eman ta zabal zazu

Universidad
del País Vasco

University of the
Basque Country

Contribution to Supervised Representation Learning: Algorithms and Applications

Author

Ahmad Khoder

Supervisors

Fadi Dornaika
Abdelmalik Moujahid

Donostia-San Sebastián
April 2021

ACKNOWLEDGEMENT

I would like to thank everyone who has contributed to my academic achievements. Firstly, I would like to express my sincere appreciation to my supervisors for their continuous support and guidance throughout the thesis duration. In addition, I would like to thank my parents and siblings who supported me with love until I reached the level I am at today. Without all of you, all of this would have been so difficult. A special thanks goes to my friends in Lebanon and Spain. Finally, I would like to thank the University of the Basque Country for trusting me and giving me the opportunity to present my dissertation. Thank you all for your unwavering support.

Ahmad

AK

ABSTRACT

In this thesis, we focus on supervised learning methods for pattern categorization. In this context, it remains a major challenge to establish efficient relationships between the discriminant properties of the extracted features and the inter-class sparsity structure.

Our first attempt to address this problem was to develop a method called "Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity" (RDA_FSIS). This method performs feature selection and extraction simultaneously. The targeted projection transformation focuses on the most discriminative original features while guaranteeing that the extracted (or transformed) features belonging to the same class share a common sparse structure, which contributes to small intra-class distances.

In a further study on this approach, some improvements have been introduced in terms of the optimization criterion and the applied optimization process. In fact, we proposed an improved version of the original RDA_FSIS called "Enhanced Discriminant Analysis with Class Sparsity using Gradient Method" (EDA_CS). The basic improvement is twofold: on the first hand, in the alternating optimization, we update the linear transformation and tune it with the gradient descent method, resulting in a more efficient and less complex solution than the closed form adopted in RDA_FSIS. On the other hand, the method could be used as a fine-tuning technique for many feature extraction methods. The main feature of this approach lies in the fact that it is a gradient descent based refinement applied to a closed form solution. This makes it suitable for combining several extraction methods and can thus improve the performance of the classification process.

In accordance with the above methods, we proposed a hybrid linear feature extraction scheme called "feature extraction using gradient descent with hybrid initialization" (FE_GD_HI). This method, based on a unified criterion, was able to take advantage of several powerful linear discriminant methods. The linear transformation is computed using a descent gradient method. The strength of this approach is that it is generic in the sense that it allows fine tuning of the hybrid solution provided by different methods.

Finally, we proposed a new efficient ensemble learning approach that aims to estimate an improved data representation. The proposed method is called "ICS Based Ensemble Learning for Image Classification" (EM_ICS). Instead of using multiple classifiers on the transformed features, we aim to estimate multiple extracted feature subsets. These were obtained by multiple learned linear embeddings. Multiple feature subsets were used to estimate the transformations, which were ranked using multiple feature selection techniques. The derived extracted feature subsets were concatenated into a single data representation vector with strong discriminative properties.

Experiments conducted on various benchmark datasets ranging from face images, handwritten digit images, object images to text datasets showed promising results that outperformed the existing state-of-the-art and competing methods.

Keywords: Machine Learning, Pattern Classification, Discriminant Embedding, Manifold learning, Linear Embedding, Image Categorization, Supervised Learning, Hybrid Embedding, Hybrid Initialization, Computer Vision, Ensemble Learning, Fine Tuning, Gradient Descent Optimization.

RESUMEN

La presente tesis está enfocado en los métodos de aprendizaje supervisado para la categorización de patrones. En este contexto, sigue siendo un gran desafío establecer relaciones eficientes entre las propiedades discriminantes de las características o atributos extraídos y la estructura de escasez entre clases .

El primer intento para abordar este problema fue desarrollar un método llamado "Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity (RDA_FSIS)”. Este método realiza la selección y extracción de características simultáneamente. La transformación de proyección perseguida se centra en identificar las características originales más discriminativas al tiempo que garantiza que las características extraídas (o transformadas) que pertenecen a la misma clase compartan una estructura dispersa común, lo que contribuye a reducir la distancia entre objetos de la misma clase. Al hilo de lo anterior, se han introducido algunas mejoras relacionadas con el criterio de optimización o función objetivo así como el proceso de optimización aplicado. En efecto, propusimos una versión mejorada del algoritmo original RDA_FSIS llamada "Enhanced Discriminant Analysis with Class Sparsity using Gradient Method (EDA_CS) ". Las mejoras destacadas son: por un lado, incorporar el método de descenso de gradiente en el proceso de adaptación y y ajuste de la transformación lineal, resultando en una solución más eficiente y menos compleja que la forma cerrada adoptada en RDA_FSIS. Por otro lado, el método propuesto podría usarse como una técnica de sintonización precisa para muchos métodos de extracción de características. El rasgo principal de este enfoque radica en el hecho de que es un refinamiento basado en el descenso de gradiente aplicado a una solución en forma cerrada. Esto lo hace adecuado para combinar varios métodos de extracción y, por lo tanto, puede mejorar el rendimiento del proceso de clasificación.

De acuerdo con los métodos anteriores, se ha propuesto un esquema lineal híbrido de extracción de características llamado "feature extraction using gradient descent with hybrid initialization (FE_GD_HI)”. Este método, basado en un criterio de optimización unificado, fue capaz de aprovechar las ventajas de varios métodos de análisis discriminante lineal. La clave radica en que se trata de un esquema genérico que permite un ajuste fino de la solución híbrida proporcionada por diferentes métodos.

Por último, se ha presentado un nuevo enfoque de aprendizaje por conjuntos eficiente que tiene como objetivo estimar una representación de datos mejorada. El método propuesto se denomina ""ICS Based Ensemble Learning for Image Classification (EM_ICS)". En lugar de aprender múltiples clasificadores usando las características transformadas, nuestro objetivo consistia en estimar varios subconjuntos de características usando múltiples variedades de aprendizaje lineal. Éstos subconjuntos de características han servido para estimar las transformaciones que posteriormente se han ordenado utilizando múltiples técnicas de selección de características. Finalmente, los distintos subconjuntos de características extraídos se han concatenado para dar lugar a un solo vector de representación de datos con fuertes propiedades discriminatorias.

Los experimentos realizados en distintos conjuntos de datos de referencia incluyendo imágenes faciales, imágenes de dígitos escritos a mano, imágenes de objetos, y conjuntos de datos de texto han mostrado resultados prometedores que han registrado mejoras significativas en comparación con los métodos existentes.

Palabras clave: Aprendizaje automático, Clasificación de patrones, Incrustación discriminante, Aprendizaje múltiple, Incrustación lineal, Categorización de imágenes, Aprendizaje supervisado, Incrustación híbrida, Inicialización híbrida, Visión por computador.

Contents

Part I : Contribution to supervised data representation learning

Part II : Selected Publications

List of figures

List of tables

Part I : Contribution to supervised data representation learning

CHAPTER 1

General Introduction

Nowadays, the evolution of modern technologies has led to an exponential increase in the amount of data generated in a variety of fields, such as medicine, manufacturing, finance, banking, public services, e-commerce, and business intelligence and strategy, to name a few.

For most of these areas, data analysis is a crucial step in enabling decision-making systems to respond efficiently to the actual current demands of the world. Efforts in this area have proven that it is not only the quantity of data that allows better evaluations, but also the quality of the data, its relevance, adequacy and reliability. Therefore, there is a genuine need to generate and process high quality data using less computational and storage resources.

For this purpose, the use of machine learning techniques (ML) has become a necessity. ML techniques aim to exploit data structures to achieve optimized data processing. In general, these methods provide better data representation by revealing hidden data patterns that help to extract relevant information. Basically, there exist three settings for machine learning approaches, namely: supervised, semi-supervised and unsupervised learning. These three settings are mainly differentiated by the availability and use of data labels in the learning process. Data labels are specific information that categorizes the data samples, in other words, the labels assign each data sample to the appropriate class or group. In supervised learning, the framework exploits the data labels in the learning process, for this type of learning all data labels should be available. Unsupervised learning frameworks do not require the data labels in the learning process, so the availability of the label information is not necessary at all. The third and last type is semi-supervised learning, which in fact can be considered as a compromise between supervised and unsupervised learning. The latter methods use all the training samples (labeled and unlabeled) to obtain the intrinsic geometric structure of the entire training data.

Depending on the context knowledge, the approaches of ML are traditionally divided into classification, regression and clustering. The first two are considered as supervised learning techniques, while the latter is considered as unsupervised learning approach. Classification aims to categorize the data according to certain criteria (e.g., image classification, objects, etc.) under different labels (classes). On the other hand, regression predicts continuous valued outputs. Clustering refers to the partitioning of the data set into multiple groups called clusters. The goal is to partition the data so that points within a cluster are very similar and points in different clusters are different. It determines the grouping among the unlabeled data.

Classification approaches are widely used in machine learning, computer vision and various other fields as they can model many real-world applications. In general, datasets are represented by two-dimensional (2-D) matrices, with columns corresponding to data samples and rows corresponding to their characterizing features. The number of features that represent the data samples is referred to as the "dimensionality" of the data. A feature can be identified as one of the following: relevant, irrelevant or redundant. Relevant features are mainly the features that contribute to a better predictive model and hence higher classification performance. These features provide useful information and are the ones that the model should extract and select among all other candidates. Irrelevant features do not contribute in any way to the improvement of the predictive model. They do not provide useful information and sometimes can even worsen the classification process, they express noise with respect to a particular relevance evaluation criterion. Redundant features are those that can be correlated, they also do not contribute to the improvement of the model. On the contrary, these features can lead to a more complex, ineffective and computationally expensive learning process.

Motivated by the desire to obtain optimized, relevant and tighter data representations, dimensionality reduction techniques are proposed and implemented. Dimensionality reduction techniques mainly aim at reducing the number of features representing the data samples in order to achieve better data interpretation. Dimensionality reduction can be performed using two approaches, namely feature selection and feature extraction. The former simply identifies the most relevant features of the data and selects the subset that contains these features without applying any core changes to the meaning of the original features. The selected features are candidates from the larger original set, so feature selection techniques subsequently produce a lower-dimensional space. On the other hand, a feature extraction technique provides a new lower-dimensional space created from a new set of features. Feature extraction can be performed using linear or nonlinear methods. Most feature extraction methods focus on estimating a linear transformation that maps the original features to another space from which latent variables can be obtained. The need to design efficient and discriminative low-dimensional embedding spaces for data representation is a key challenge that has long been pursued by researchers. Learning appropriate representations of data that allow extracting and organizing discriminative information is an important step in machine learning. It can reduce memory and computational requirements and, more importantly, improve the performance of subsequent classifiers or other machine learning techniques. This explains why representation learning is increasingly becoming a major research topic [113, 144, 180, 222, 213]. Although there has been tremendous progress in achieving some of the goals of such feature engineering, there is still much work to be done. In fact, most data representation learning methods suffer from a number of drawbacks related to the quality of the extracted data.

This work contributes to data representation learning by employing several linear projection models capable of performing feature ranking and extraction simultaneously. We focused on studying different learning representation algorithms and their applications to image categorization tasks. More specifically, we focused on "supervised learning" for image categorization. All the data representations provided by the proposed approaches have demonstrated their superiority over a wide list of powerful competing methods. Most of our proposed methods integrate the concept of implicit feature ranking along with class sparsity, which allowed these methods to gain many powerful discriminative capabilities.

Contributions

Throughout this thesis report, we have proposed several supervised linear feature extraction methods that have shown promising results outperforming many existing methods. The main findings are summarized as follows:

- We have provided a comprehensive and concise literature review on machine learning types and dimensionality reduction. We have provided several examples that allow a proper comprehension about these topics by highlighting their strengths, limitations and variants.
- We proposed a supervised feature extraction algorithm targeting image categorization applications. This method exploited multiple types of sparsity in a joint framework and delivered high-end performance. Specifically, our proposed framework integrates two types of sparsity, the first is achieved by imposing the $\ell_{2,1}$ norm constraint on the transformation matrix to ensure that our models implicitly perform feature selection. The second type is achieved by imposing the inter-class sparsity constraint on the projected samples, which helped to ensure a common sparsity structure for the samples sharing the same class. The proposed framework has retrieved the Linear Discriminant Analysis transformation considering the aforementioned types of sparsity. An orthogonal reconstruction matrix was also introduced to improve the proposed approach's robustness to noise.
- We have proposed an enhanced feature extraction approach that further improves the discriminative capabilities provided by our first contribution. The improved criterion differs in two ways: the global criterion and the optimization

process. We have used the gradient algorithm to compute and update the transformation matrix instead of the closed form solution. The proposed framework is considered as a fine-tuning tool that allows tuning the embeddings provided by existing linear approaches. In general, it is possible to improve the performance of other feature extraction methods using our proposed method.

- We have introduced a hybrid initialization scheme for the transformation matrix, which has assured very useful properties. To compute the transformation matrix (embedding) we used the steepest descent gradient algorithm. It is well known that the gradient algorithm requires a good initial guess to perform well. We set the initial guess of the sought embedding as a combination of the solutions provided by multiple linear feature extraction methods. Then, we start applying our proposed algorithm iteratively until a more discriminative transformation is obtained. Through this introduced hybrid scheme, our proposed approaches were able to inherit the powerful discriminant properties provided by the linear methods used in the hybrid construction of the transformation matrix.

- We proposed an ensemble learning based approach that exploits the utilization of multiple feature subsets to construct enhanced and more discriminative data representations. Our scheme uses multiple feature selection algorithms to construct different feature subsets. Each of these subsets is then separately fed to a learner and a prediction is obtained to form a single model. The main idea of our proposed algorithm is to combine the projections provided by multiple models in order to construct a single, more powerful data representation.

- Our proposed methods can be applied to other types of data, not just images. It is true that most of our contributions emphasized working with image datasets, however our proposed methods can be applied to different types of data. To prove that, we extended our experiments by applying our method on synthetic and text datasets. The original motivation for this extension that is presented in section 4.5 is to highlight the discriminative power of the proposed method using non-image datasets.

Thesis Organization and Research Outline

The content of this thesis is divided into two main parts. The first part provides an overview and discussion of general machine learning concepts. More specifically, Part I comprises six chapters. The current chapter presents a general introduction to my dissertation and highlights the main findings of the thesis, including the organization of the PhD dissertation and the research outline. Chapter 2 presents the background as well as the state of the art relevant to my work, which includes a general overview of machine learning algorithms, dimensionality reduction techniques, graph-based and deep learning approaches, and finally some preliminaries and tools. Chapter 3 describes the experimental setups used in the experiments conducted in this thesis. This chapter provides a detailed description of the datasets and descriptors used in various experiments. Chapter 4 provides a brief summary of the contributions made in this thesis. Each contribution is presented in detail in a separate chapter in the second part of the report. Chapter 5 presents conclusions derived from this thesis and highlights some limitations and future work. The final chapter serves to list our publications and contributions with a brief summary for each contribution.

Part II presents the main articles written while working on the thesis, including those published and those submitted and in the revision phase.

CHAPTER 2

Background and State of the art

Contents

During my PhD, I have investigated several machine learning paradigms, from basic and classical supervised and unsupervised learning methods to more general and recent semi-supervised techniques. In addition to that, I have explored various feature extraction and selection algorithms. In this chapter, I will present a general description of all these techniques and discuss some related works and preliminaries.

2.1 Machine Learning Types

The learning problems we consider can be roughly categorized into supervised, semi-supervised, and unsupervised. In supervised learning, the goal is to predict the value of an outcome measure (or label), usually quantitative or categorical, based on several input features. In contrast, outcome measure do not exist in unsupervised learning, the goal is to describe how the data is organized or clustered using only the set of input features. In semi-supervised learning we are concerned with the design of models in the presence of both labeled and unlabeled data, that is we only have outcome measures for a subset of the data.

2.1.1 Supervised Learning

In machine learning, the distinction in the nature of the output variables has led to a naming convention for the prediction tasks: Regression, when we predict quantitative outputs, and Classification, when we predict qualitative outputs.

In both tasks, the main objective is extracting the specific structures of the input data that lead to the derivation of correct output data.

Supervised learning methods require both the training data alongside with the corresponding labels in the training process. Suppose our data matrix is given by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ where D and N denote the dimensionality and the total number of samples, respectively, supervised learning approaches require the label matrix $\mathbf{F} \in \mathbb{R}^{N \times C}$ to learn and extract the targeted features. N and C represent the total number of samples and the number of classes, respectively.

It is well-known that supervised learning approaches outperform both semi-supervised and unsupervised techniques at classification or regression tasks. This is normal due to the fact that the supervised learning models take advantage of the available label information of the data samples used to train the model. Therefore, learning the data structure will be more efficient, leading to better discrimination properties, hence better classification and regression.

Common supervised learning algorithms include logistic regression, naive bayes, support vector machines, artificial neural networks, and others. Supervised learning methods are widely investigated and gained much attention in the machine learning and computer vision fields. A vast number of methods have been and are being proposed for various tasks including (image categorization, classification, medical images, feature extraction, feature selection, graph-based embedding, and many more tasks) [88, 89, 44, 188, 186, 36, 146, 106, 80, 218, 211, 106, 125].

2.1.2 Unsupervised Learning

In general, unsupervised learning is used for various tasks, such as clustering, dimensionality reduction, representation learning, and others. In all these tasks, knowledge about the inherent structure of the data is pursued without any label availability. We state some examples about famous unsupervised algorithms, the well-known principal component analysis [81], k-means clustering and some extensions [114, 160, 82], and autoencoders [26, 153]. Unsupervised learning is very useful in exploratory analysis as it can automatically identify the data structure.

Another task where unsupervised learning can be important is dimensionality reduction. Dimensionality reduction techniques, as explained earlier, aim to reduce the dimensionality of the data and thus provide an efficient representation of the original data using a smaller number of features. Usually, in representation learning, capturing the relationships between features, allow us to represent original data using the latent features that interrelate the original features. This sparse latent structure is often represented using far fewer features than the original ones. This allows for less intensive and computationally expensive data processing. In other contexts, dimensionality reduction can be used to convert data from one modality to another.

Similar to other tasks, unsupervised learning has its advantages and drawbacks. Since unsupervised learning works without the data labels, no prior knowledge about the instances is required. Moreover, this setting is simple, requires fewer computations and is faster than other learning settings. The most obvious drawback is that this setting does not have access to the data labels. Many current studies are still conducted on unsupervised feature extraction or selection techniques [78, 118, 133, 166, 122, 214].

2.1.3 Semi-Supervised Learning

In real-world applications, the data labeling process is very challenging. In other words, collecting the data labels is a very demanding and time-consuming process, in the sense that it might be unrealistic to collect the labels of all the data. Another reason that makes the labeling process infeasible is the use of applications where there is a constant stream of data (e.g., social networks). Collecting a portion of the data labels is favorable, since it is cheaper and relatively require less processing. In general, only a very small portion of the data is required to be labeled, so semi-supervised learning techniques can perform efficiently. For all these reasons, many researchers have adopted the semi-supervised settings and proposed novel algorithms targeting this type of learning. Semi-supervised learning can be regarded as a compromise between supervised and unsupervised learning. Semi-supervised models take advantage of a small amount of labeled training data along with a large amount of unlabeled training data to derive the best embedding spaces, since the former is less expensive and easier to obtain. This is how semi-supervised learning works:

Suppose that we have a data matrix $\mathbf{X} \in \mathbb{R}^{D\times N}$. In reality, semi-supervised learning sees the data matrix as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l, \mathbf{x}_{l+1}, ..., \mathbf{x}_{l+u}] \in \mathbb{R}^{D\times(l+u)}$, where D represents the dimensionality of the data (number of features) and $N = l + u$ represents the total number of data samples. Semi-supervised learning algorithms uses both labeled $\mathbf{x}_i|_{i=1}^{l}$ and unlabeled samples $\mathbf{x}_i|_{i=l+1}^{l+u}$, where l and u denote the number of labeled and unlabeled data samples, respectively. Thus, the original data matrix in the semi-supervised learning algorithms is divided into two parts,

$\mathbf{X}_{\mathcal{L}} = [\mathbf{x}_1, \mathbf{x}_2, ..., ...\mathbf{x}_l] \in \mathbb{R}^{D \times l}$ and $\mathbf{X}_{\mathcal{U}} = [\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, ..., ...\mathbf{x}_{l+u}] \in \mathbb{R}^{D \times u}$, these are the data matrix associated with labeled and unlabeled samples, respectively. In many cases, the main goal of semi-supervised methods is to derive the labels of the unlabeled data samples, the soft label matrix is usually denoted by $\mathbf{F} \in \mathbb{R}^{N \times C}$ where C denotes the total number of classes of the data. In a semi-supervised context, $\mathbf{F} = \begin{pmatrix} \mathbf{F}_{\mathcal{L}} \\ \mathbf{F}_{\mathcal{U}} \end{pmatrix}$, where $\mathbf{F}_{\mathcal{L}}$ consists of the labels of the labeled data samples. The strengths of semi-supervised approaches are many, these methods are stable, simple, efficient, and do not require a large number of labeled data samples, thus requiring less learning time, which makes them very fast. The disadvantages of this setting are that they are mostly not applicable to network level data, also they provide lower classification performance than the supervised methods. This is normal due to the lack of label information that the supervised methods use in the learning process. In general, semi-supervised methods must follow a number of assumptions about the data to justify using a small set of labeled data to make inferences about the unlabeled data points. The first is the continuity assumption, which refers to the assumption that data points that are "close" to each other are more likely to share a common label. In addition, the second assumption is that data naturally form discrete clusters and that points in the same cluster are more likely to have a common label. Semi-supervised learning methods also assume that the data lies roughly in a lower-dimensional space (or manifold) than the original space.

Some common semi-supervised methods are transductive support vector machines and graph-based methods such as label propagation, feature extraction, and feature selection [119, 169, 215, 137, 158, 42].

Table 1 contains the notation and symbols used in this section. Table 2 illustrates a brief summary on the three discussed learning types in this section.

2.2 Overview on Dimensionality Reduction

In real-world applications and with the advent of so-called Big Data, the problem of dealing with high-dimensional data always arises [61]. Normally, real data is repre-

Table 1: **Brief description of the main notations used in the machine learning types section.**

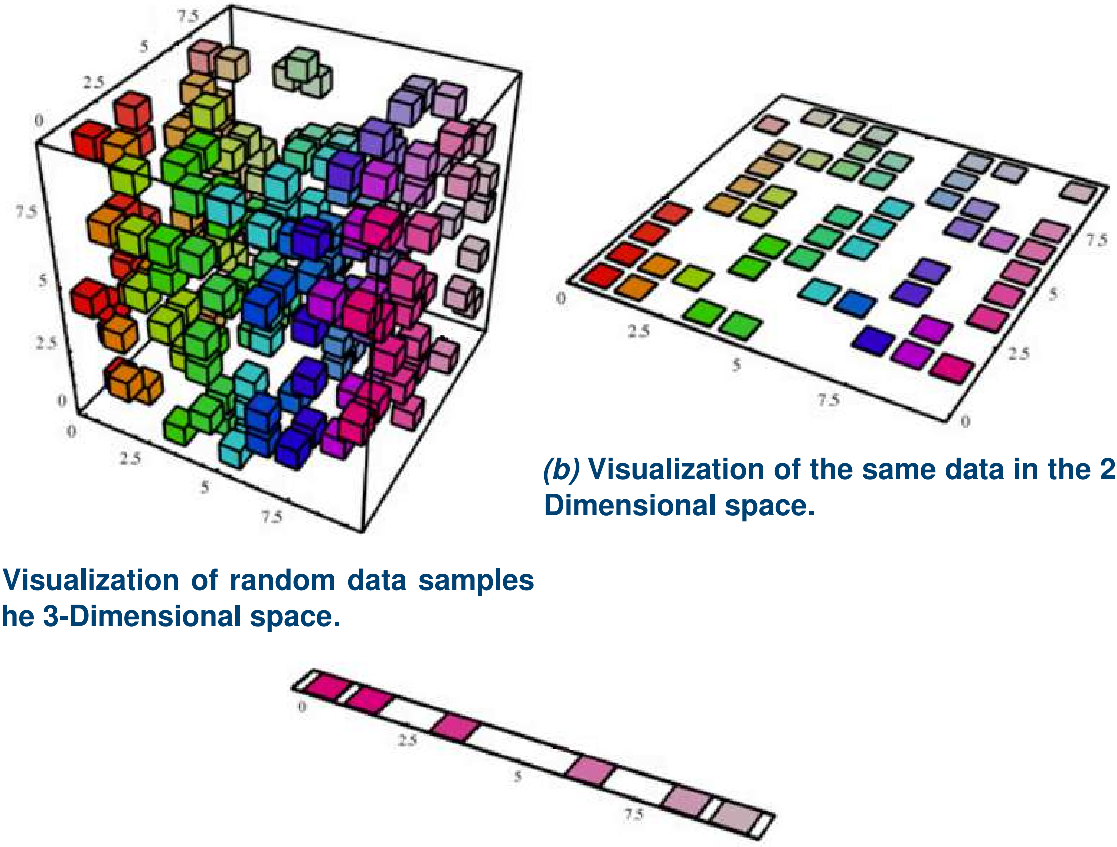
Symbol	Description
$\mathbf{X}$	Data matrix
N	Number of samples
D	Dimension of data
$\mathbf{F}$	Matrix of data labels
l	Number of labeled samples
u	Number of unlabeled samples
C	Number of classes
$\mathbf{Y}$	Projected data matrix
$\mathbf{Q}$	Transformation matrix

Table 2: **Machine learning types comparison.**

	Description	
Learning Types	Unsupervised	No available information about data labels Do not use data labels in the training process Very fast Used for exploratory purposes, dimensionality reduction, feature extraction,...
	Semi-supervised	Information about part of the data labels (More realistic in real world applications) Use of both labeled and (part) unlabeled samples in the training process Relatively fast Used for label propagation and feature extraction (data projection) Lower performance than that of supervised techniques
	Supervised	All data labels are available Use only labeled samples in the training process Normally slower than unsupervised and semi-supervised learning Used for feature extraction, feature selection, classification, and other tasks High performance

sented through a large number of features, which makes it very challenging to deal with these data. Applications used in various fields such as gaming, photography, image processing, machine learning, classification and data storage are very challenging due to the high dimensionality of the data. In most cases, processing this data requires enormous memory and computational resources. In addition, high dimensional data is prone to be affected by noise.

Figure 1 shows a graphical illustration of the curse of dimensionality concept. In this example, in the case of the 1-dimensional space shown in Figure 1c, there are only 10 possible positions, therefore 10 datum are required to create a representative sample that covers the problem space. In the case of a two-dimensional (2-D)space, there exist 10^2 possible positions, so 100 datum are required to create a representative sample for the problem space. This is illustrated in Figure 1b. In the case of a three-dimensional (3-D) space, there are 10^3 possible positions, so the number of required datum to create a representative sample covering the problem space would be 1000. The number of needed datum continues to grow exponentially.



(b) **Visualization of the same data in the 2-Dimensional space.**

(a) **Visualization of random data samples in the 3-Dimensional space.**

(c) **Visualization of the same data projected on 1-Dimensional space.**

Figure. 1: **Overview about the curse of Dimensionality [1].**

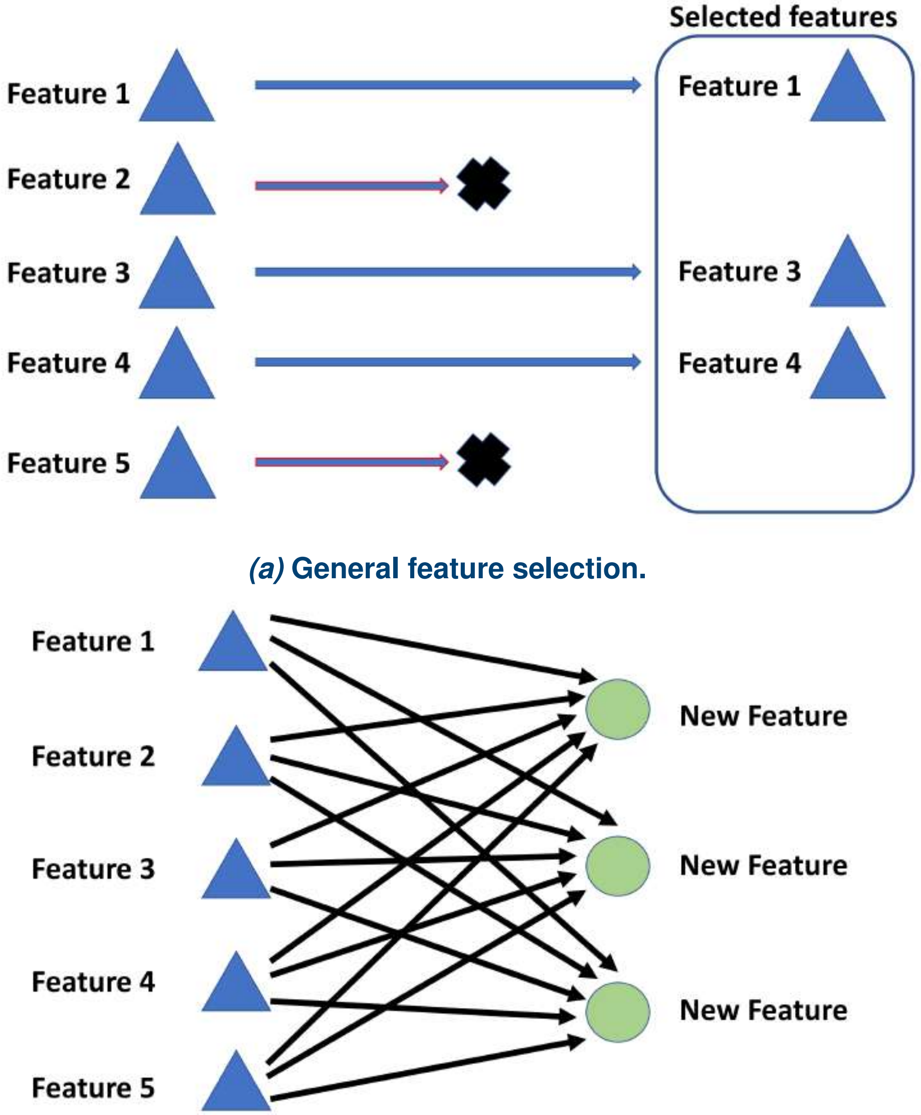
In order to address this problem, dimensionality reduction techniques have been proposed. These reduction techniques have recently come into prominence due to their high efficiency [83, 138]. Dimensionality reduction works by reducing the number of features (referred to as the dimension of the data) while preserving the intrinsic data structure. Decreasing the dimensionality of the data helps in several ways, it helps in data compression, achieving efficient learning and inference, overcoming the "curse of dimensionality", data de-noising in addition to achieving better visualization [177, 154, 199, 219].

In the pattern recognition and machine learning fields, dimensionality reduction can be achieved using two approaches, namely: (i) feature extraction or (ii) feature selection. Until our current date, Linear Discriminant Analysis (LDA) [171] along with Principal Component Analysis (PCA) [161] hold the places for two of the most popular dimensionality reduction feature extraction approaches that have demonstrated efficiency over linear structured data. However, the reality is that various real-world applications deal with non-linear structured data, where PCA and LDA may fail. This is where the importance of manifold learning methods targeting feature extraction becomes apparent.

2.2.1 Overview on Feature Selection

Usually, a feature can be classified as relevant, irrelevant, or redundant. An irrelevant feature is the one that does not contribute to the predictive model's enhancement, moreover, it may even degrade the classification performance if it is considered during the classification process. In contrast, relevant features contribute to the achievement of a discriminative prediction model, hence leading to a more efficient classification performance. These are the targeted features that the model aims to select among all others. A redundant feature does not lead to better performance of the model in the classification process. Many researches concluded that the use of the original data features does not guarantee the best performance in learning tasks [59, 196].

Feature selection refers to selecting subsets of the most relevant features that represent the data in the most efficient way. These features are selected from the original data features after being ranked by their importance according to certain mathematical operations. Many works have investigated various feature selection techniques in the field of pattern recognition [99, 113, 144, 180, 222].



(a) **General feature selection.**

(b) **General feature extraction.**

Figure. 2: **Feature selection vs feature extraction illustration. Feature selection: Selects features from the original data features and discard others. Feature extraction: Extracts a new set of features from the original data.**

Figure 2 illustrates the main difference between feature selection and feature extraction techniques.

As stated earlier, feature selection approaches aim to select the most relevant and representative features of the data according to different criteria. Several approaches have been proposed in recent years. Of these, we mention: Fisher score [45], Relief [92], Relief-F [147, 94], mutual information [11], Hilbert Schmidt Independence Criterion (HSIC) [163], Laplacian score [65], in addition to Trace Ratio criterion [132]. These mentioned methods have contributed greatly in feature selection applications and achieved very good performances.

In general, feature selection techniques work as follows. Suppose we have a set of "d" features, this set of features is denoted by R, the main objective of feature selection techniques is to select a subset S of m features with $m < d$, that maximizes the criterion F .

$$S^* = \underset{S \subseteq R}{\arg\max}\, F\left(S\right) \quad s.t.\ |S| = m \tag{2.1}$$

|S| in equation 2.1 represents the cardinality of the set S.

Some of the most popular feature selection techniques used in the pattern recognition field are Fisher score, Relief, Relief-F and many more.

2.2.1.1 Fisher Score

Fisher score works by computing the score of each data feature and then selecting each feature accordingly. Fisher algorithm computes the score of the i-th feature S_i using:

$$S_i = \frac{\sum_{j=1}^{C} n_j \left(\mu_{ij} - \mu_i\right)^2}{\sum_{j=1}^{C} n_j\, \rho_{ij}^2} \tag{2.2}$$

where ρ_{ij} and μ_{ij} represent the variance and mean of the i-th feature associated with the j-th class. n_j denotes the number of instances in the j-th class and μ_i is the mean of the i-th feature. C denotes the number of classes.

Most original feature selection methods work by computing the score features individually while ignoring the combination of features. This may lead to non-optimal results, hence we obtain incorrect feature importance estimations. For simplicity, we consider working with two features f_1 and f_2. In some cases, the scores of both features may be low, however, the score of the combination of these two features is high. In this case, these algorithms discard the two features f_1 and f_2, although they should be selected. The same can happen in the case of using redundant features, the algorithm can select both of them although neither of them should be selected.

2.2.1.2 Relief Algorithm

Most methods used to approximate feature reliability presume conditional independence of features and are therefore less suitable for problems that might involve more feature interaction. Relief-based algorithms (Relief, Relief-F, and RRelief-F) do not simply make this assumption.

These algorithms have been shown to be reliable, conscious of contextual information, and can effectively estimate the quality and relevance of features or attributes in problems with high attribute dependency. Relief algorithms are based on the concept of local margins for each feature. These margins should be large enough for relevant features. These algorithms are widely considered as feature subset selection methods used in the pre-processing phase before training the model [92]. They are still one of the most popular pre-processing algorithms to date [37]. They are actually general feature estimators that have been successfully used in a multitude of environments. Inspired by instance-based learning, the authors in [92] proposed the classical Relief algorithm. Relief is optimized for two-class problems. The basic principle of the algorithm is to consider not only the disparity of feature values and variance in classes, but also the distance between instances.

Consider the feature vector **v** and the feature vectors of the instance closest to **v** from each class. The closest instance belonging to the same class is referred to as near-hit (NH), and the closest instance with a different class is denoted as near-miss (NM).

Relief algorithm [94] iteratively computes the weight for the i-th feature by:

$$W_i = W_i - (V_i - \mathrm{NH}_i)^2 + (V_i - \mathrm{NM}_i)^2 \tag{2.3}$$

2.2.1.3 Relief-F Algorithm

Authors in [94] improved the Relief algorithm. They developed an extension of the original Relief, called Relief-F, which improves the original algorithm by estimating margins more reliably. Irrelevant attributes, either the redundant or noisy ones, can influence and affect the selection of the nearest neighbors. This makes the estimation of the margins unreliable. To address this issue, Relief-F searches for the "k" nearest (NH's) and (NM's) rather than a single (NH and NM) and averages the contribution of all k nearest (NH's) and (NM's). The nearest neighbors' selection is very important in Relief-F. The purpose is to find the nearest neighbors with respect to important attributes. In all our experiments, "k" was set to 10, which, empirically, gives satisfactory results. For some problems, significantly better results can be obtained by tuning "k" (as is typical for the majority of machine learning algorithms). Many studies have been conducted to explore the feature selection ability using Relief-F algorithm [147]. More details on Relief variants can be found in [71].

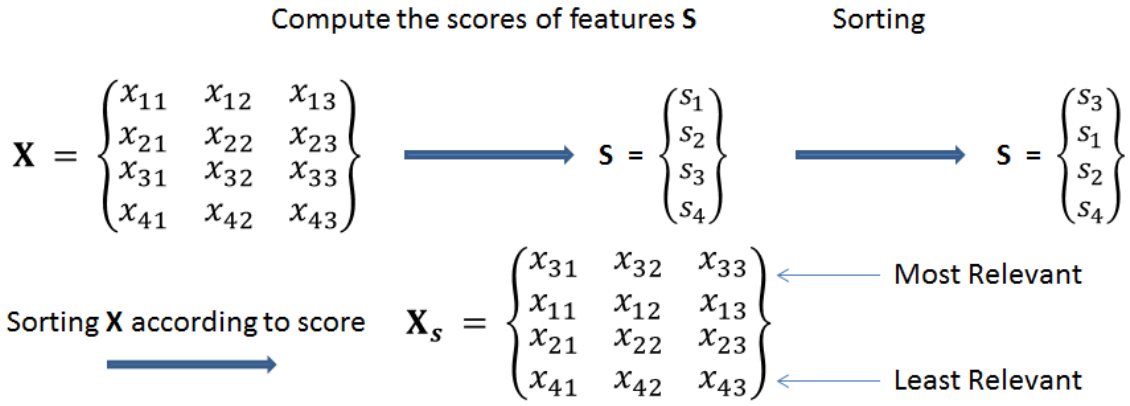

Compute the scores of features S
Sorting
$\mathbf{X} = \begin{Bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{Bmatrix}$
$\mathbf{S} = \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{Bmatrix}$
$\mathbf{S} = \begin{Bmatrix} s_3 \\ s_1 \\ s_2 \\ s_4 \end{Bmatrix}$
Sorting X according to score
$\mathbf{X}_s = \begin{Bmatrix} x_{31} & x_{32} & x_{33} \\ x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{41} & x_{42} & x_{43} \end{Bmatrix}$
Most Relevant
Least Relevant

***Figure. 3:* Feature Ranking General Methodology.**

Let us consider a simple example where we have the original data matrix $\mathbf{X} \in \mathbf{R}^{4\times3}$, **s** is a 4-vector containing the computed score associated with each feature, and

$\mathbf{X}_s$ represents the data matrix after feature ranking. Figure 3 illustrates the general feature ranking methodology.

2.2.1.4 Robust multi-label feature selection with dual-graph regularization

Feature selection remains a heavily studied topic to this day, with many recent approaches being proposed. One of the recently proposed feature selection approaches is the Robust Multi-label Feature Selection based on Dual-graph (DRMFS) [74]. The authors proposed a novel method based on dual-graph regularization. The two used graphs are namely: (i) feature graph regularization, in addition to (ii) label graph regularization. The former was adopted in order to preserve the geometric structure of the features. The latter was used to explore the correlations of the data labels. Furthermore, the authors have imposed the $\ell_{2,1}$ norm constraint on the loss function in order to ensure more robustness to their approach.

The main objective of this proposed scheme is to compute the feature weight matrix $\mathbf{W}$. The authors imposed the $\ell_{2,1}$ norm in addition to a non-negative constraint onto the feature weight matrix to enhance the row sparsity property.

The (DRMFS) algorithm minimizes the following objective function:

$$\min_{\mathbf{W}} = ||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_{2,1} + \alpha Tr(\mathbf{W}^T\mathbf{L}^X\mathbf{W}) + \beta Tr(\mathbf{W}\mathbf{L}^Y\mathbf{W}^T) + \gamma||\mathbf{W}||_{2,1} \ s.t. \mathbf{W} \geq 0 \tag{2.4}$$

where $\mathbf{W} \in \mathbb{R}^{d\times c}$, $\mathbf{X} \in \mathbb{R}^{d\times n}$ and $\mathbf{Y} \in \mathbb{R}^{n\times c}$ denote the feature weight matrix, the feature matrix and the label matrix, respectively. d, c and n represent the dimensionality, the number of classes and the number of samples, respectively. $\mathbf{L}^X$ and $\mathbf{L}^Y$ represent the feature and label graph Laplacian matrices, respectively. Detailed information on the computation of Laplacian graphs can be found in [8]. α, β and γ are three balance parameters to their corresponding terms. Once the solution for problem 2.4 is obtained, it is possible to evaluate the most important top $k-$features by computing $||\mathbf{W}_{i*}||_2$ $(1 \leq i \leq d)$ and selecting the features corresponding to the k highest scores.

2.2.2 Overview on Feature extraction

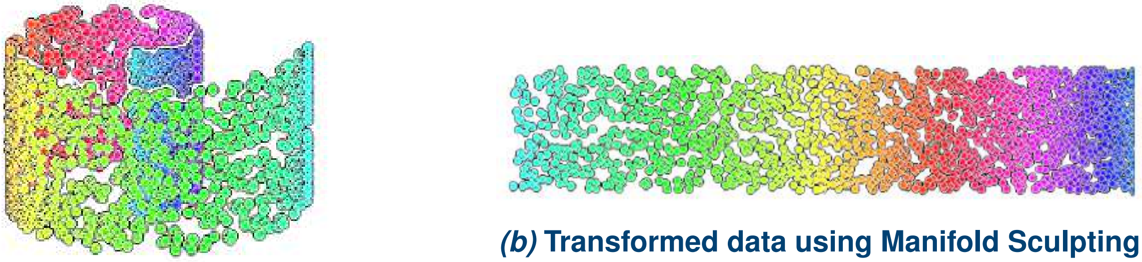
As we mentioned in section 2.2, dimensionality reduction can be achieved using two approaches, namely: feature extraction or feature selection. Feature extraction methods are those that create a set of new features based on certain transformations and/or combinations of the original features. An overview about the feature extraction methodology was illustrated in figure 2b. There are a huge number of feature extraction methods in the literature due to the high importance and large investigations and contributions in this field. Section 2.4 presents some typical feature extraction methods related to our contribution in this thesis.

Manifold learning can be classified as a type of feature extraction. In the case of dealing with non-linear structured data, manifold learning approaches are the solution to obtain efficient data representations. Generally, a manifold is a surface without a particular form. It does not necessarily have to be a plane, it can have any shape.

Manifold learning methods attempt to understand and learn the underlying data structure. These methods aim to reduce the dimensionality of the data while maintaining the high- dimensional data distribution, they allow each data sample to be described as a function of only a few underlying parameters. Manifold learning or feature extraction methods aim to uncover these parameters to derive a low-dimensional data representation. These methods assume that the data points are samples from a low-dimensional manifold (latent space) embedded in a high-dimensional dimensional space (ambient space).

Manifold learning approaches can often be regarded as a non-linear version of PCA. In PCA, the data is projected onto a low-dimensional space. This is restrictive in the sense that those surfaces are all linear. We know that PCA usually searches for a plane surface to describe the data, which may not exist. This may lead to an inappropriate data representation. Manifold learning solves this problem in a very efficient manner. The main concept of manifold learning clearly states that any pair of data samples that are close in the original space should also be close in the low-dimensional space. An example of the manifold smoothness is depicted in figure

4, which presents the famous swiss roll example. Manifold transformation from figure (4a) to (4b) was conducted using the Manifold Sculpting method [52].



(b) **Transformed data using Manifold Sculpting**

(a) **Original data**

Figure. 4: **Synthetic Swiss Roll example. The left part of this figure depicts the original data in a 3D space. The right part depicts the non-linear embedding of the same data in the low dimensional 2D space [52].**

By looking at Figure 4, we can notice that the classification process of the data in the original space is challenging (left part of the figure), while the classification of the data in the transformed space is much more efficient (right part of the figure).

Working with feature extraction techniques has many advantages. It allows working with lower-dimensional data, which is less computationally expensive to handle. Feature extraction methods also lead to obtaining more discriminant data representations, that can boost the classification performance while allowing the use of simpler classifiers. For these reasons, feature extraction techniques are nowadays intensively studied in the pattern recognition, machine learning and computer vision fields. Many recent studies have been conducted in order to obtain discriminant data representations [186, 188, 44, 89, 88].

To illustrate the concept of latent and ambient spaces, an additional example was presented through Figures 5 and 6. Both of these figures are intended to visualize the distribution of the Tetra synthetic dataset samples in both the original space and the embedded space. The Tetra dataset was defined in [175, 176], it consists of 400 data points belonging to four classes and lying in $\mathbb{R}^3$. This dataset presents the challenge associated with low inter-cluster distances. In other words, one can observe how the clusters represented by spheres are very close to each other by looking at Figure 5, which makes the classification process hard to implement. On the other hand, better class discrimination is provided by projecting the original data into the embedding spaces delivered by the feature extraction methods Robust sparse Linear Discriminant Analysis RSLDA [186], Linear Discriminant Analysis LDA [171] and Feature Extraction using Gradient Descent FE_GD [88]. The distribution of samples in the latent space (2D space) is shown in Figure 6.

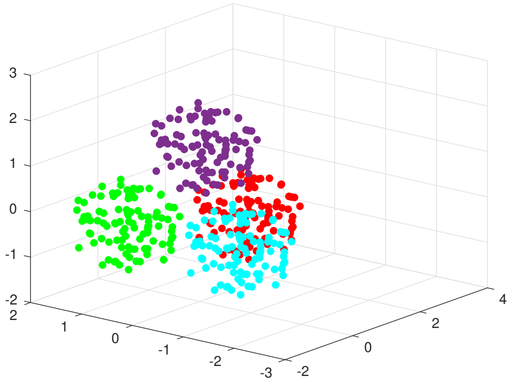


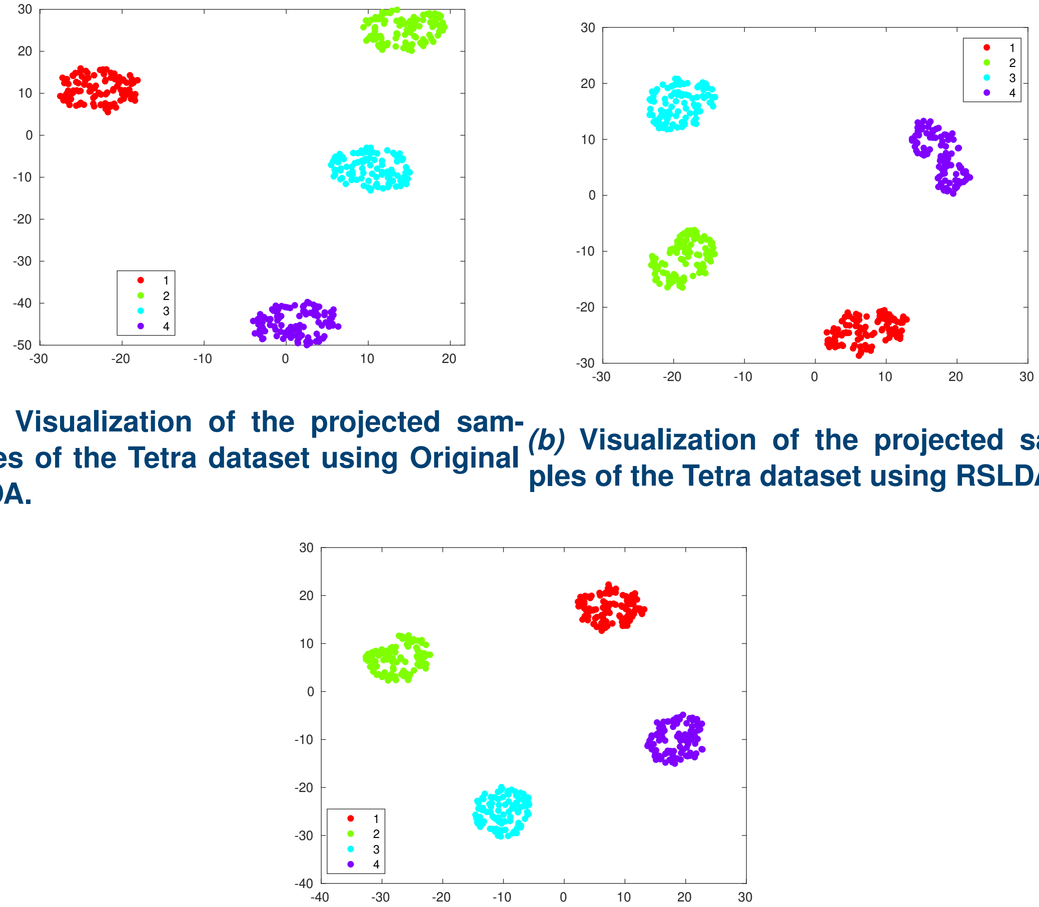
Figure. 5: **Visualization of the Tetra dataset points in the original space. These 3D points belong to four large full spheres close to each other.**

2.3 Graph-Based Learning

In recent years, graph-based learning has attracted much interest in the pattern recognition and computer vision fields. Graph theory has been introduced and merged with the manifold learning concept. This has led to promising results. Many graph-based Manifold Learning techniques have been proposed in recent years for the purpose of extracting relevant features from original data [8, 148, 170]. Some examples of graph-based manifold algorithms are the famous Locally Linear Embedding (LLE) [148], ISOMAP [170] and Laplacian Eigenmap (LE) [8]. These algorithms are based on ideas from both manifold space and graph theory.

2.3.1 Graph Construction

Generally speaking, graph structure encodes inter-dependencies among constituents and provides a convenient representation of high-dimensional data, which is the main reason that graph construction has become an important research topic in



***(a)* Visualization of the projected samples of the Tetra dataset using Original LDA.**

***(b)* Visualization of the projected samples of the Tetra dataset using RSLDA.**

***(c)* Visualization of the projected samples of the Tetra dataset using FE_GD.**

***Figure. 6:* TSNE visualization of the projected samples of the Tetra dataset using LDA, RSLDA, and the first proposed variant FE_GD.**

manifold learning field. Researchers in graph theory field mainly focus on analyzing and mining information patterns from graphs. In this section, we will briefly enumerate and discuss some classical and widely used graph construction methods.

Two of the most famous classical graph construction approaches are the K-nearest neighbors (KNN) graph in addition to the ϵ-neighborhoods graph [170]. These two methods aim to compute the edge weight matrix, also called the affinity matrix **W** based on the distance $d(x_i, x_j)$ or the similarity $sim(x_i, x_j)$ between the two points x_i and x_j.

$$W_{ij} = \begin{cases} sim(x_i, x_j) & \text{if } x_i \text{ and } x_j \text{ are nearest neighbors.} \\ \mathbf{0} & \text{if } x_i \text{ and } x_j \text{ are not nearest neighbors.} \end{cases} \quad (2.5)$$

The constructed affinity matrix **W** is a symmetrical matrix due to the fact that the similarity between the two entries x_i and x_j is equal no matter what the starting point is. In other words, the expression $sim(x_i, x_j) = sim(x_j, x_i)$ always holds true.

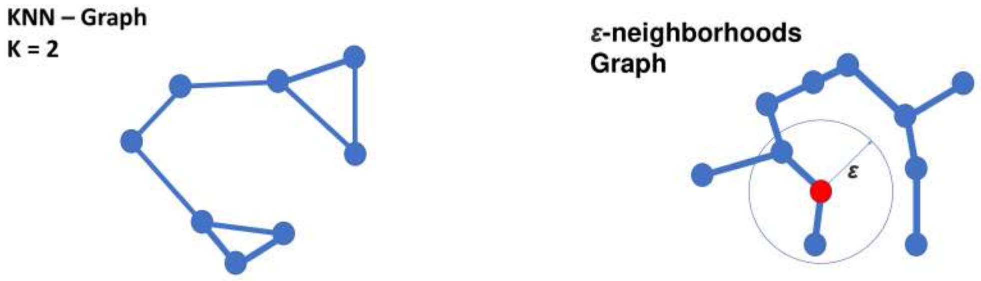
Generally speaking, the edge matrix **W** is subject to the following constraints:

- $W_{ij} = 0$ indicates the absence of an edge connecting the two nodes i and j.
- $W_{ii} = 0$, $i = 1, ..., N$ where N denotes the total number of data samples (nodes).
- All weight edges are non-negative, $W_{ij} \geq 0$.
- $W_{ij} = W_{ji}$.

In the ϵ-neighborhoods graphs, the base criteria depends on the Euclidean norm between x_i and x_j. In this method, the connection between x_i and x_j is only established in the case when $||x_i - x_j||^2 < \epsilon$, where $||.||$ denotes the Euclidean norm. One common problem that can occur using this strategy is the possibility of getting some disconnected nodes, this can happen in the cases where the value of ϵ is not carefully defined.

To overcome the above limitation, K-nearest neighbors (KNN) graphs were used. For each node, KNN method searches for the set of the nearest neighbors of that node and establishes a link between the node and the "K" nearest nodes in that set. The choice of "K" is usually important and affects the performance. KNN-based graphs were found to perform reasonably well and resulted in decent data representations. The downside of these graphs arises when dealing with large datasets and when the number of neighbors required to construct the graph is large. In this particular case, large computational resources are required to construct the graph.

Figure 7a illustrates a typical example of graph construction using the K-nearest neighbors algorithm in the case where the value assumed for K is 2, while figure 7b shows the obtained graph using the ϵ-neighborhoods algorithms.



(a) **Typical K-nearest neighbors graph with K=2.**

(b) **Typical ϵ-neighborhoods graph.**

Figure. 7: **Typical graph construction examples using KNN and ϵ-neighborhoods algorithms.**

Usually, subsequent to computing the graph, the weights are updated using the heat kernel function as follows: Let $t \in \mathbf{R}$

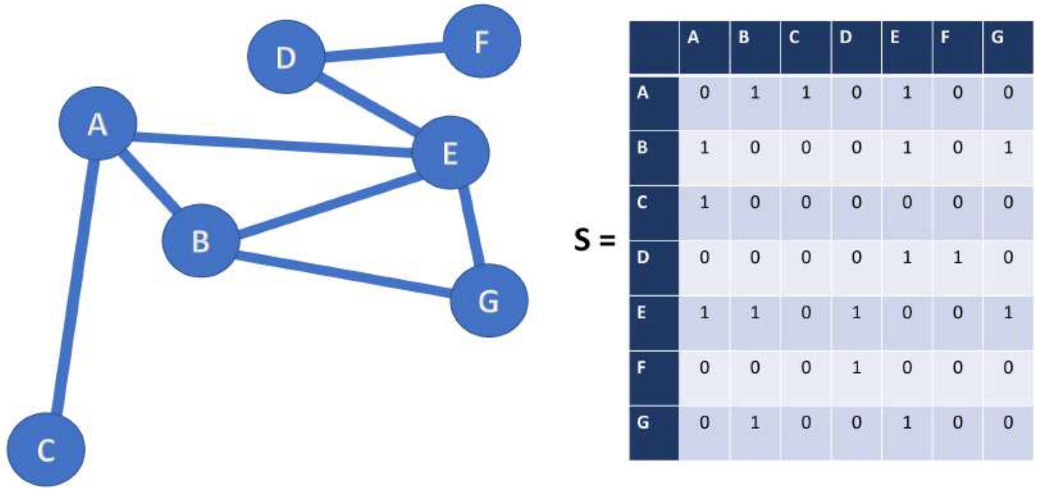
$$W_{ij} = \begin{cases} e^{-\frac{||x_i - x_j||^2}{t}} & \text{if nodes } i \text{ and } j \text{ are connected.} \\ \mathbf{0} & \text{if nodes } i \text{ and } j \text{ are not connected.} \end{cases} \tag{2.6}$$

We described above the criteria used for graph construction using both (KNN) and ϵ-neighborhoods algorithms. The general concept of graph-based algorithms is as follows. Each data sample is represented as a node. Let $\mathbf{G}(\mathbf{P}, \mathbf{E})$ be the graph where $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, ..., \mathbf{p}_N\}$ is the set of nodes, N denotes the total number of data samples, and $\mathbf{E}$ is the set of edges. W_{ij} denotes the edge weight between the two nodes $\mathbf{p}_i$ and $\mathbf{p}_j$. The value of W_{ij} may depend on several factors (e.g., labels of samples i and j or the distance separating these two samples in the original space).

In general, the real interpretation of W_{ij} is a measure of the similarity between the two nodes $\mathbf{p}_i$ and $\mathbf{p}_j$, so usually a high value of W_{ij} indicates high similarity between the two samples $\mathbf{p}_i$ and $\mathbf{p}_j$ and vice versa. A graph can be either weighted or unweighted. In a weighted directed graph, each connection between two particular nodes is given a specific weight.

In recent years, multiple researches aimed at incorporating adjacency graphs into the manifold learning dimensionality reduction frameworks. The main goal has been to derive a low-dimensional space that represents the local structure of the data [9, 67, 69, 148]. First, an adjacency graph is constructed to model the underlying geometry of the data. Then, a mapping is constructed to preserve the local or global structure of the graph in the embedding space.

An example of a classical adjacency graph containing 7 nodes is shown in Figure 8, the similarity scores in this example are set to binary weights.



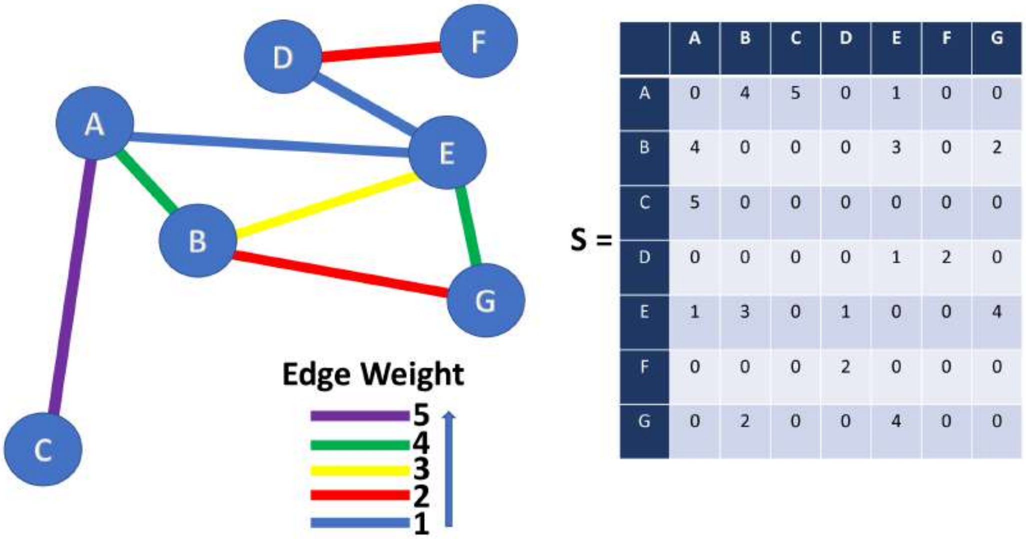
S =

	A	B	C	D	E	F	G
A	0	1	1	0	1	0	0
B	1	0	0	0	1	0	1
C	1	0	0	0	0	0	0
D	0	0	0	0	1	1	0
E	1	1	0	1	0	0	1
F	0	0	0	1	0	0	0
G	0	1	0	0	1	0	0

***Figure. 8:* Adjacency graph and its corresponding similarity matrix.**

An example of a weighted graph is illustrated in figure 9. The similarity scores in this example are computed using equation (2.7), where $EW^{(i,j)}$ represents the edge weight between nodes p_i and p_j.

$$W_{ij} = \begin{cases} EW^{(i,j)} & \text{if } p_i \text{ and } p_j \text{ are connected.} \\ 0 & \text{if } p_i \text{ and } p_j \text{ are not connected.} \end{cases} \tag{2.7}$$



$$S = \begin{array}{c|ccccccc} & A & B & C & D & E & F & G \\ \hline A & 0 & 4 & 5 & 0 & 1 & 0 & 0 \\ B & 4 & 0 & 0 & 0 & 3 & 0 & 2 \\ C & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ E & 1 & 3 & 0 & 1 & 0 & 0 & 4 \\ F & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ G & 0 & 2 & 0 & 0 & 4 & 0 & 0 \end{array}$$

Figure. 9: **Weighted graph alongside with the corresponding similarity matrix.**

2.3.2 Graph-Based embedding

Graph-based learning demonstrated remarkable superiority in the pattern recognition and machine learning areas, which is why it has gained so much importance and is focused on by many researchers in these fields [54, 66, 73, 76, 79, 100]. Numerous methods serving different purposes have made use of graph theory. Some of them have merged the idea of manifold space with the graph theory to develop and produce powerful discrimination methods. Graphs have proven to be powerful tools for data analysis applications. Moreover, graphs can represent data in a simple yet effective manner. For these reasons, graph-based algorithms are nowadays studied in various domains such as: semi-supervised learning for label propagation and regression [93, 181, 43], feature selection [226, 111, 223, 179], graph-based embedding [199, 200, 149], community discovery, spectral clustering [149, 184] and many more.

Various classical graph-based manifold algorithms ushered a new era of graph-based learning for the Pattern Recognition field. Locally Linear Embedding (LLE) [148], ISOMAP [170], Laplacian Eigenmap (LE) [8], Linear Neighborhood Propagation (LNP) [181], Locality Preserving Projections (LPP), and Graph-optimized Locality

Preserving Projections (GoLPP) [217]. These algorithms are based on ideas from both manifold space and graph theory.

Lets consider the mathematical model of the graph as defined in section 2.3.1, where the graph is represented by **G**(**P**,**E**) with nodes **P** and edges **E**. In some cases, a graph can also be represented with three tuples as **G**(**P**,**E**,**W**), where W_{ij} represents the edge weight between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. N denotes the total number of data samples.

2.3.2.1 Locally Linear Embedding

Locally Linear Embedding (LLE) is a classical unsupervised manifold learning approach. In other words, it does not require data labels to operate. LLE determines the non-linear global data structure by exploiting the local linear reconstructions. It formulates its learning problem as a neighborhood-preserving embedding. The main goal of the method is minimizing the reconstruction error of all local neighborhoods in the entire dataset.

First, the adjacency matrix used in LLE is computed through either the K-nearest neighbors (KNN) or the ϵ- neighborhoods method. After that, all non-zero entries of the weight matrix W are computed by the reconstruction of the sample $\mathbf{x}_i$ from its K neighboring points.

LLE computes the weighted matrix **W** of **G**(**P**,**E**) by the following formula:

$$\phi\left(\mathbf{W}\right) = \sum_{i=1}^{N} || \, \mathbf{x}_i - \sum_{j=1}^{N} W_{ij}\mathbf{x}_j \, ||^2 \qquad s.t. \sum_{j=1}^{N} W_{ij} = 1. \tag{2.8}$$

After obtaining **W** the embedding matrix $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, ..., \mathbf{z}_N) \in \mathbf{R}^{m \times N}$ can be obtained by minimizing the following criterion: (The following problem can be solved by eigen decomposition)

$$\phi\left(\mathbf{Z}\right) = \sum_{i=1}^{N} || \, \mathbf{z}_i - \sum_{j=1}^{N} W_{ij}\mathbf{z}_j \, ||^2 \quad s.t. \frac{1}{N}\mathbf{Z}^T\mathbf{Z} = \mathbf{I}, \ \sum_{i} \mathbf{z}_i = 0. \tag{2.9}$$

Authors in [84] extended the work on graph-based embedding related to LLE by proposing a manifold-based similarity adaptation for label propagation technique. In their proposed method, the authors used a Gaussian Kernel function with a different parameter for each dimension of the data to define the weight matrix. Their method performed well and was able to enhance the performance of the classical LLE, but the convergence of the solution to a global minimum was not guaranteed because the proposed objective function is not convex.

2.3.2.2 A Global Geometric Framework for Nonlinear Dimensionality Reduction ISOMAP

Authors in [170] proposed a nonlinear manifold learning method that was able to recover the underlying structure of data under certain assumptions, namely: (i) isometry and (ii) convexity.

Suppose we have the original parameter space Θ, the mapping Ψ. $\mathbf{x}_i$ and $\mathbf{x}_j$ denote two points on the manifold, and $d_G(\mathbf{x}_i, \mathbf{x}_j)$ denotes the shortest distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ travelled along the manifold. The vector θ denotes the control parameters underlying a measuring device and the manifold as the enumeration $\mathbf{x} = \Psi(\theta)$ of all possible measurements as the parameters vary.

The two assumptions are the following:

- Isometry: The mapping Ψ preserves the geodesic distances.

  $d_G(\mathbf{x}_i, \mathbf{x}_j) = |\theta - \theta'| \quad \forall \mathbf{x}_i \leftrightarrow \theta, \mathbf{x}_j \leftrightarrow \theta'$

  where $|.|$ denotes Euclidean distance.

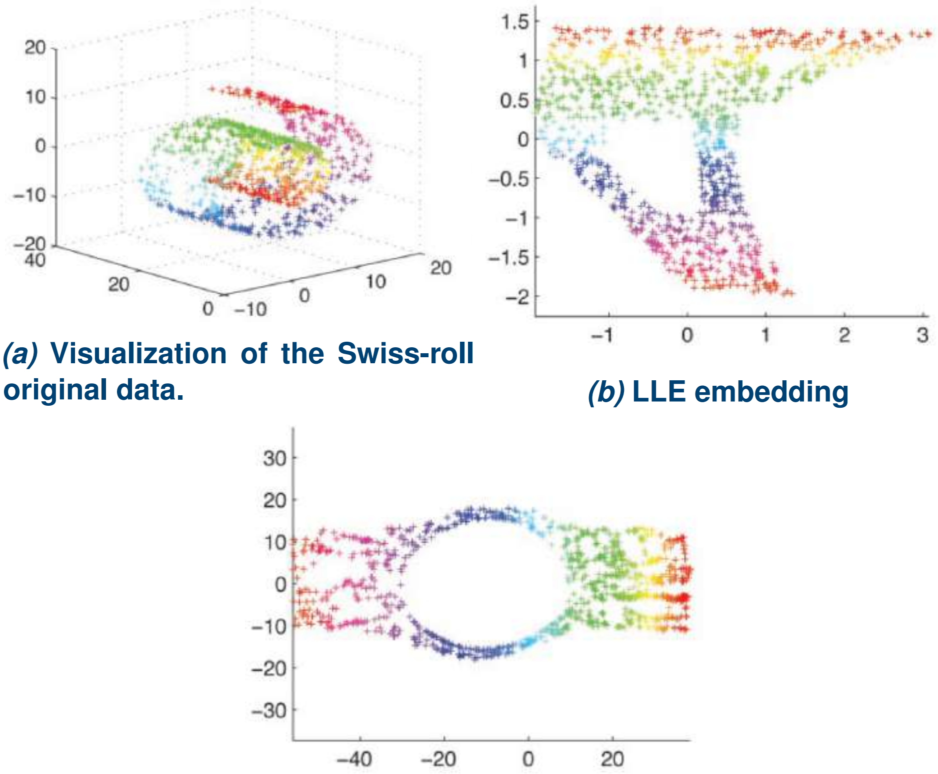
- Convexity: The parameter space Θ is a convex subset of $\mathbf{R}^d$. If (θ,θ') is a pair of parameters points in Θ, then $\{(1-t)\,\theta + t\,\theta' : t \in (0,1)\}$ should lie in Θ.

Under these conditions, ISOMAP was able to recover Θ up to rigid motion. ISOMAP can perform manifold feature learning as follows. In the first step, ISOMAP uses the famous k-nearest neighbors graph and ϵ-neighborhoods, and sets the edge lengths equal to $d(\mathbf{x}_i, \mathbf{x}_j)$. Assuming that the graph is denoted by **G**(**P**,**E**), ISOPMAP then defines the shortest path between $\mathbf{x}_i$ and $\mathbf{x}_j$ over **G** as $d_G(\mathbf{x}_i, \mathbf{x}_j)$.

Finally, the low-dimensional embedding **Z** can be computed by minimizing the following problem:

$$\Phi\left(\mathbf{Z}\right) = \sum_{i,j} \left(d\left(\mathbf{z}_i, \mathbf{z}_j\right) - d_G\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)^2 \tag{2.10}$$

This solution for this minimization problem can be obtained using the multidimensional scaling algorithm [170].



(a) **Visualization of the Swiss-roll original data.**

(b) **LLE embedding**

(c) **ISOMAP embedding**

Figure. 10: **Visualization of (a) Original Swiss-roll data. (b) LLE embedding with K = 12. (c) ISOMAP with K = 7. Detailed information about the Swiss-roll dataset and this illustration can be found in [39].**

Figure 10 illustrates a visualization of a Swiss-roll data [39], LLE embedding with K = 12, and ISOMAP embedding with K = 7. Detailed information about this illustration can be found in [39].

2.3.2.3 Laplacian Eigenmap

Laplacian Eigenmap (LE) [8] is another graph-based method aimed at tracing high-dimensional data. This method has been used in several applications, it can be used either to reduce the high dimensional data or to derive a powerful data representation [9]. It is a joint method built on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold and the connections to the heat equation. The main strong-point of LE is that it works to keep the mapping of nodes i and j which have large weight value W_{ij} as close as possible. LE also uses the k-nearest neighbors and the ϵ-neighborhoods methods to set the edges between the nodes, and then utilizes either simple or heat kernel methods to estimate the edge weights. Assume that **L** denotes the Laplacian matrix, LE minimizes the following:

$$\begin{aligned}\Phi\left(\mathbf{Z}\right) =& \frac{1}{2}\sum_{i,j=1}^{N}\left|\left|\mathbf{z}_i-\mathbf{z}_j\right|\right|^2 W_{ij}\\ =& trace\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)\ s.t. \mathbf{Z}^T\mathbf{D}\mathbf{Z}=\mathbf{I}\end{aligned} \tag{2.11}$$

Solution can be obtained using eigen decomposition.

2.3.2.4 Linear Neighborhood Propagation

Authors in [181] proposed the Linear Neighborhood Propagation (LNP) method, this scheme explored neighborhood properties. The method relies on the assumption that there is a possibility to linearly reconstruct each data sample based on its neighborhood. The graph computed by this method is used in label propagation. The adjacency matrix is constructed using the K nearest neighbors of each sample and the weights are computed by minimizing the following problem:

$$\mathbf{W}\left(i,:\right) = \min || \, \mathbf{x}_i - \sum_{j=1}^{N} W_{ij}\mathbf{x}_j \, ||^2 \qquad s.t. \sum_{j=1}^{N} W_{ij}=1, \quad W_{ij}\geq 0. \tag{2.12}$$

Where $\mathbf{W}\left(i,:\right)$ denotes the $i-$th row of the matrix **W**.

2.3.2.5 Locality Preserving Projections and Extensions

Most of the embedding methods mentioned above are non-linear, these methods provide good data representations, however, these methods cannot be applied when dealing with out of samples data. For this purpose, the need for the linearization of these embedding approaches arises. Many studies targeting linear embedding frameworks were conducted. We mention among these, Locality Preserving Projection (LPP) [68, 198, 217] and Neighborhood Preserving Embedding (NPE) [66]. These two are respectively considered as the linear versions of Laplacian Eigenmap (LE) and Locally Linear Embedding (LLE). LPP is an unsupervised method with basic graph construction properties. Let us adopt the notations shown in Table 3

Table 3: **Some notations.**

$\mathbf{X}$	Data matrix
W_{ij}	Coefficient noting the similarity between the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$
$\mathbf{y}_i$ and $\mathbf{y}_j$	1-dimensional projection of $\mathbf{x}_i$ and $\mathbf{x}_j$ in the new space.
$\mathbf{D}$	Diagonal matrix
$\mathbf{L}$	Laplacian matrix
$\mathbf{a}$	Projection vector
$\mathbf{A}$	Projection matrix

Let the diagonal entries of $D_{ii} = \sum_j W_{ij}$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$. For 1D projection case (data samples are projected on a single axis), LPP can be written as:

$$\min \sum_{i,j} (y_i - y_j)^2 W_{ij} \tag{2.13}$$

Since a linear embedding is targeted, the mapping can be applied on all data samples using the derived projection matrix (this a vector in the case of 1D projection). We have $y_i = \mathbf{a}^T \mathbf{x}_i$. Equation 2.13 can be transformed to:

$$\begin{aligned} \min \frac{1}{2} \sum_{i,j} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j \right)^2 W_{i,j} &= \min \;\; \mathbf{a}^T \mathbf{X} \left(\mathbf{D} - \mathbf{W} \right) \mathbf{X}^T \mathbf{a} \\ &= \min \;\; \mathbf{a}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{a} \end{aligned} \tag{2.14}$$

Although LPP provides good performance, it has some drawbacks that need to be addressed. The most obvious of these drawbacks is the need for a separate process in order to compute the graph. To address this problem, authors proposed the Graph-optimized Locality Preserving Projections (GoLPP) [217] which minimizes the classical LPP objective function over the linear transformation and the affinity matrix jointly in a single criterion. The optimized graph criterion showed more discriminative power than the original criterion. GoLPP minimizes the following:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{W}} \frac{trace\left(\mathbf{A}^T\mathbf{X}\hat{\mathbf{L}}\mathbf{X}^T\mathbf{A}\right)}{trace\left(\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}\right)} + \mu \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}\ln\left(W_{ij}\right) \\ s.t. W_{ij} \geq 0, \sum_{j=1}^{N} W_{ij} = 1 \end{aligned} \tag{2.15}$$

Assuming that $\hat{\mathbf{W}} = \mathbf{W} + \mathbf{W}^T$, $\hat{\mathbf{L}}$ denotes the Laplacian matrix defined as $\hat{\mathbf{L}} = \hat{\mathbf{D}} - \hat{\mathbf{W}}$. $\hat{\mathbf{D}}$ is actually a diagonal matrix whose diagonal elements are the sum of rows sums of $\hat{\mathbf{W}}$.

Many recent studies have been conducted to either achieve more discriminative embeddings, or for dimensionality reduction goals [216, 227]. Researchers are now focusing on joint methods that exploit both the transformation matrix and graphs in a single criterion. Authors in [209] proposed the "Joint graph optimization and projection learning for dimensionality reduction" (JGOPL). The authors in [209] adopted the $\ell_{2,1}$ norm to measure the distance for the loss function, which provides a more robust method against outliers. Moreover, the same approach demonstrated very good local structure preserving properties. A locality constraint is introduced into the (JGOPL) criterion to prevent a sample from joining the distant samples during graph optimization. Other recent approaches have been proposed to extend graph-based embedding to the semi-supervised setting, where a small fraction of the data labels is required and can lead to better learning [227, 131]. An example of one work addressing the semi-supervised setting is the work proposed in [131], where the structured graph achieves the ideal neighbors assignment, based on which an optimal low-dimensional subspace can be learned.

2.3.2.6 Exponential Local discriminant embedding

Local Discriminant Embedding (LDE) [22] is a powerful discriminant analysis method. It was originally proposed to overcome a few of the classical Linear Discriminant Analysis limitations. LDE extends the main concept of LDA with the aim of performing local discrimination. The main goal of LDE is to estimate a linear mapping that simultaneously maximizes the local margin between heterogeneous samples and brings the homogeneous samples closer to each other.

Assuming that the data matrix is denoted by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$, where N represents the total number of data samples, $W_{w,ij}$ denotes the similarity between two homogeneous samples $\mathbf{x}_i$ and $\mathbf{x}_j$, $W_{b,ij}$ denotes the similarity between two heterogeneous ones.

$\mathbf{D}_w$ and $\mathbf{D}_b$ denote two diagonal matrices whose entries are the column sums of $\mathbf{W}_w$ and $\mathbf{W}_b$, respectively. $\mathbf{L}_w = \mathbf{D}_w - \mathbf{W}_w$ and $\mathbf{L}_b = \mathbf{D}_b - \mathbf{W}_b$ show the corresponding Laplacian matrices.

LDE computes the sought linear transformation matrix $\mathbf{A}$ by maximizing the following criterion:

$$\frac{trace\left(\mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}\right)}{trace\left(\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A}\right)} = \frac{trace\left(\mathbf{A}^T\hat{\mathbf{S}}_b\mathbf{A}\right)}{trace\left(\mathbf{A}^T\hat{\mathbf{S}}_w\mathbf{A}\right)} \tag{2.16}$$

where the symmetric matrices $\hat{\mathbf{S}_w} = \mathbf{X}\mathbf{L}_w\mathbf{X}$ and $\hat{\mathbf{S}_b} = \mathbf{X}\mathbf{L}_b\mathbf{X}$ denote the locality-preserving within-class and the locality-preserving between-class scatter matrix, respectively.

Although LDE usually provides a good representation, this algorithm is affected by the small simple size problem. The Small Simple Size (SSS) problem occurs when the number of images used in the training set is significantly smaller than the number of pixels (or features) in each instance. The same problem also occurs in the cases where the rank of $\mathbf{L}_w \leq N-1$. LDE can be affected by the SSS problem in the same way as LDA and many other linear embedding techniques. Many algorithms try to solve this issue, one of them is Exponential local discriminant embedding (ELDE) [41]. The main idea of ELDE was to replace the scatter matrices $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$ with their exponential versions, in this way, if the (SSS) problem occurs, in other words in the case when $\hat{\mathbf{S}}_w$ is singular (has zero eigenvalues and thus cannot be inverted), SSS would not occur. In this way, instead of solving equation 2.16, ELDE solves the following:

$$\max_{\mathbf{A}} \frac{trace\left(\mathbf{A}^T exp\left(\hat{\mathbf{S}}_b\right)\mathbf{A}\right)}{trace\left(\mathbf{A}^T exp\left(\hat{\mathbf{S}}_w\right)\mathbf{A}\right)} \tag{2.17}$$

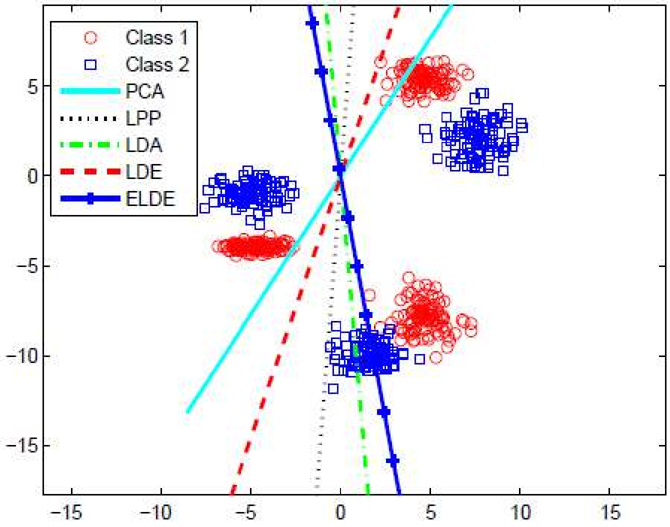


Figure. 11: **Projection direction of ELDE together with that of four linear methods (LDE, LDA, LPP and PCA) [41].**

Figure 11 illustrates an example of the multi-modal datasets representing two classes of 2D samples. Each class is distributed as three separated Gaussians having different parameters. The projection direction of ELDE together with that of four linear methods are plotted. We can see that every method has provided a different direction according to its objective function.

Many recent research is still being conducted based on local embedding whether for image classification or dimensionality reduction purposes [77, 136, 203, 64].

2.4 Typical Linear Feature Extraction Methods

The contributions provided through this thesis report have been influenced by several works and investigations. In this section, we will give a brief description of some typical linear feature extraction methods related to the realized contributions of this thesis.

2.4.1 Principal Component Analysis

We discussed dimensionality reduction and its importance in section 2.2, Principal Component Analysis (PCA) is one of the most well-known unsupervised approaches used for dimensionality reduction purposes. In general, PCA is most useful when the data lies on or close to a linear sub-space of the data set. For this type of data, PCA finds a basis for the linear subspace and allows to disregard the irrelevant features. Let us briefly describe how (PCA) works.

Given a dataset where each data sample has a dimensionality D, i.e., each point consists of D features, the main goal of PCA is to compute a set of D-dimensional vectors aligned with the directions of maximum variance of the data. The number of computed vectors is D, they are referred to as Principal Components and denoted as (PCs). The computed Principal Components have the following properties:

- The computed PCs are uncorrelated. This is due to the fact that these principal components form an orthonormal basis, they are described by being not only perpendicular to each other, but also having unit lengths.
- Principal components are associated with data variance. In particular, the first component (PC) is aligned with the direction of maximum variance, the second with the direction representing the second highest variance, the third with the next direction, etc...

These PCs have several uses, the most important of which are: (i) projecting the original data onto these PC's, and (ii) using these PCs to synthesize new points. The former can be implemented by applying the dot product of an input data point with the principal component, which returns a scalar value that is the projection of that point onto this PC. In principle, D-dimensional input data can be projected onto its D Principal Components, however, it is usually only interesting to select the PCs that represent a high data variance to project onto, this can be chosen manually or based on a set threshold. For example, by selecting the first m PCs describing the highest data variance where $m << D$ to project onto these, this is where dimensionality reduction is achieved.

We give some details about the computational process of the orthonormal transformation matrix **P** $\in \mathbb{R}^{D\times D}$, which consists of the PCs: **P** is computed according to the following constraints:

$\mathbf{Y} = \mathbf{P}^T\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{N\times D}$ denotes the original data matrix consisting of N samples of dimensionality D, and the columns of **Y** contain the projection onto the principal components $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. Moreover, $\mathbf{Y}\mathbf{Y}^T = \mathbf{U}$, where **U** denotes the covariance (diagonal) matrix of the projected points **Y** which are uncorrelated.

Mathematically, the covariance matrix $\mathbf{U} = \mathbf{Y}\mathbf{Y}^T$ can be expressed by:

$$\mathbf{Y}\mathbf{Y}^T = (\mathbf{P}^T\mathbf{X})(\mathbf{P}^T\mathbf{X})^T = \mathbf{P}^T(\mathbf{X}\mathbf{X}^T)\mathbf{P}.$$

We want the obtained quantity in the above equation to be a diagonal matrix **U** in which:

$$\mathbf{P}^T(\mathbf{X}\mathbf{X}^T)\mathbf{P} = \mathbf{U}.$$

If we multiply the left side of the equation by **P** and the right side by $\mathbf{P}^T$ we obtain:

$$\mathbf{X}\mathbf{X}^T = \mathbf{P}\mathbf{U}\mathbf{P}^T$$

We know that $\mathbf{P}\mathbf{P}^T = \mathbf{I}$, and the Singular Value Decomposition of the quantity $\mathbf{X}\mathbf{X}^T$ give us the following:

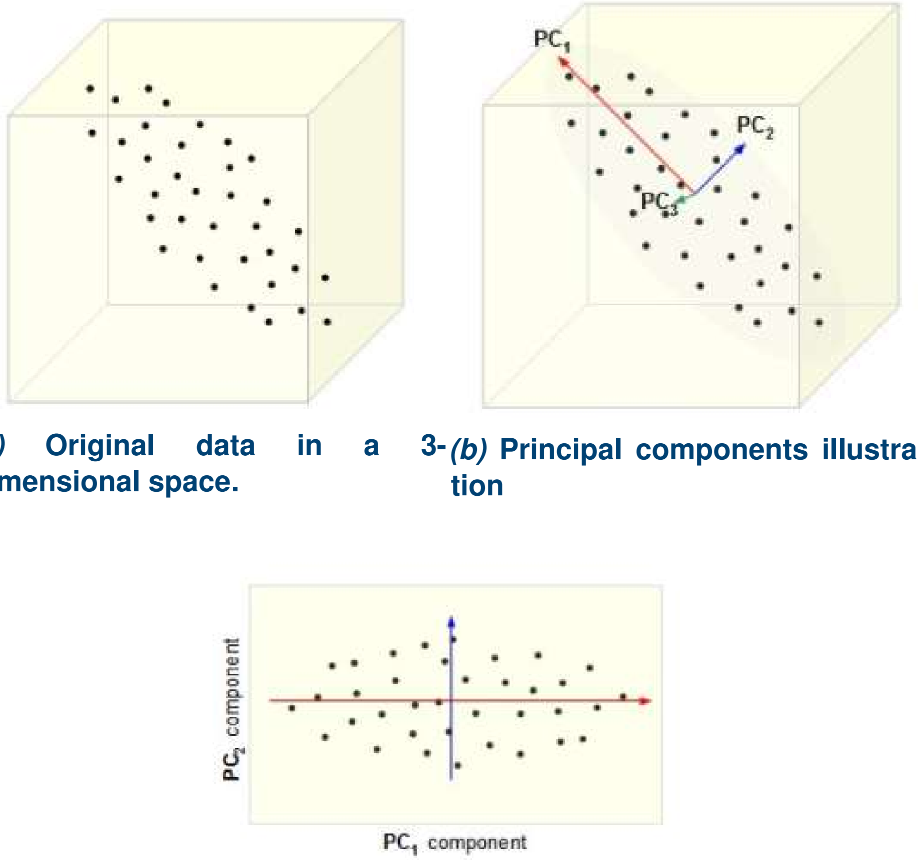
$$\mathbf{X}\mathbf{X}^T = \mathbf{V}\mathbf{S}\mathbf{W}^T$$

where $\mathbf{V}$ and $\mathbf{W}$ contain the left and right eigenvectors of the quantity $\mathbf{X}\mathbf{X}^T$ and $\mathbf{S}$ is a diagonal matrix containing the corresponding eigenvalues. By combining the last two equations obtained above, we derive that $\mathbf{P}\mathbf{U}\mathbf{P}^T = \mathbf{V}\mathbf{S}\mathbf{W}^T$ and since $\mathbf{X}\mathbf{X}^T$ is constructed as a symmetric matrix, the left and right eigenvectors $\mathbf{W}$ and $\mathbf{V}$ will be equal. This leads to $\mathbf{P}\mathbf{U}\mathbf{P}^T = \mathbf{V}\mathbf{S}\mathbf{V}^T$. We know that $\mathbf{P}$ and $\mathbf{V}$ are orthonormal, this concludes that $\mathbf{P} = \mathbf{V}$ and $\mathbf{U} = \mathbf{S}$. Thus, $\mathbf{U}$ is a diagonal matrix and the projected data $\mathbf{Y}$ are uncorrelated. This also proves that the PCs corresponding to the data matrix $\mathbf{X}$ are given by the eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ of the original (centered) data.

In short, PCA can be used as a feature extraction or dimensionality reduction approach, it uses the eigenvectors of the data's covariance to perform dimensionality reduction. PCA focuses on finding mutually orthogonal basis functions to obtain the directions of maximum variance in the data. It will preserve pairwise Euclidean distances. Figure 12 illustrates the dimensionality reduction process using the principal component analysis method, sub-figure (12a) presents a random original data lying in a 3-dimensional space, sub-figure (12b) shows the computed principal components of the data driven by the direction of the maximum variance of the data, and finally sub-figure (12c) shows the projection of the original data onto the first and second PCs, while ignoring the third one.

2.4.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is one of the most commonly used feature extraction methods in supervised learning. Till date, LDA [171] is still considered as a favored tool for supervised classification tasks due to its simplicity and robustness [60]. Similar to any other method, Linear Discriminant Analysis has its own advantages and limitations. One strength is that LDA performs efficiently in low-dimensional



(a) **Original data in a 3-dimensional space.**

(b) **Principal components illustration**

(c) **Data projection onto the first two Principal components**

Figure. 12: **Principal Component Analysis illustration [2].**

environments. However, LDA fails in the case where the number of predictor variables is very large compared to the number of observations. In this particular case, the within-class matrix will be singular, hence it will not be possible to apply LDA. Another scenario where LDA also fails is when the linear boundaries do not provide good separation of classes in the data. Many methods have been proposed to overcome the limitations of classical Linear Discriminant Analysis and have proved to be very efficient in the image classification field. This has resulted in classical LDA being one of the most successful bases for novel algorithms. In other words, LDA-based approaches have shown outstanding performances in the image classification field.

LDA [171] requires the labeling information of the training data in order to compute the best projection subspace in which the test samples will be then projected onto in order to be classified. Let C denote the number of classes in the data and n denote the number of samples in class c. LDA estimates a transformation matrix where the desired space maximizes the between-class variance and minimizes the within-class variance. In other words, LDA aims to find a linear projection that increases the distance between samples belonging to different classes and, conversely, decreases the distance between samples belonging to the same class.

Suppose μ, μ_i are the mean of all data samples and the mean of samples belonging to the i-th class, respectively. These means can be calculated as $\mu = \frac{1}{n}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\mathbf{x}_j{}^i$ and $\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_j{}^i$.

First, LDA computes the between-class scatter matrix $\mathbf{S}_b$ using the following formula:

$$\mathbf{S}_b = \frac{1}{n}\sum_{i=1}^{C} n_i \left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T \tag{2.18}$$

then the within-class scatter matrix $\mathbf{S}_w$ is calculated as follows

$$\mathbf{S}_w = \frac{1}{n}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\left(\mathbf{x}_j{}^i - \mu_i\right)\left(\mathbf{x}_j{}^i - \mu_i\right)^T \tag{2.19}$$

LDA aims to estimate a projection space that maximizes the between-class variance and minimizes the within-class variance. In the case where only one projection axis is needed, the projection axis $\mathbf{p}$ can be obtained by solving the following Fisher criterion: [46]

$$\mathbf{p} = \arg\max_{\mathbf{p}} \frac{\mathbf{p}^T\,\mathbf{S}_b\,\mathbf{p}}{\mathbf{p}^T\,\mathbf{S}_w\,\mathbf{p}} \tag{2.20}$$

The above problem (2.20) can be transformed to a difference form that is given by [207, 98]:

$$\mathbf{p} = \arg\min_{\mathbf{p}^T\mathbf{p}=1} \mathbf{p}^T\left(\mathbf{S}_w - \mu\,\mathbf{S}_b\right)\mathbf{p} \tag{2.21}$$

where μ is a small positive constant. By solving Eq. (2.21), we can observe that the optimal projection vector $\mathbf{p}$ is the eigenvector associated with the smallest eigenvalue of $\mathbf{S}_w - \lambda\mathbf{S}_b$. Finally, for more than one projection axis, the projection matrix $\mathbf{P} \in \mathbb{R}^{d\times k}$ consists of the k eigenvectors associated with the k smallest eigenvalues of $\mathbf{S}_w - \lambda\mathbf{S}_b$.

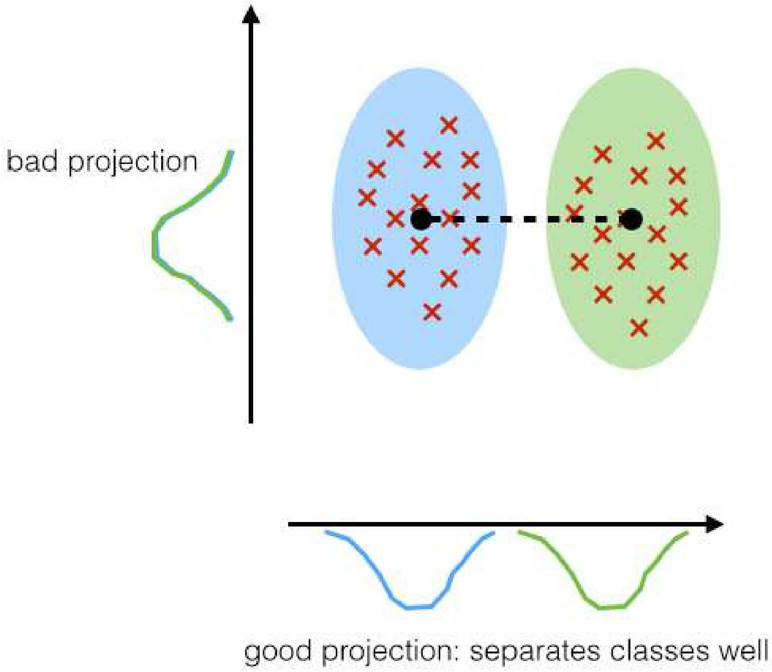


Figure. 13: **Illustration of the LDA projection axis [3].**

Figure 13 illustrates random data points where the data consists of two classes. Usually the Linear Discriminant Analysis projection is of dimension $(C-1)$, where C denotes the total number of classes. Figure 13 presents the LDA projection axis that provides good class separation (horizontal axis), therefore, better discrimination.

2.4.3 Robust Sparse Linear Discriminant Analysis

The original LDA method suffers from several problems. First, LDA may suffer from the small sample size (SSS) problem, which makes the LDA algorithm infeasible in certain cases. Another problem of LDA is that it is very sensitive to noise. The third problem is that classical LDA is also sensitive to the number of projection directions. Many LDA based techniques have been proposed to overcome some of the classical LDA problems and provide better performance and efficiency, namely: orthogonal LDA

(OLDA) [208], uncorrelated LDA (ULDA) [206] and many others. The main motivation of OLDA and ULDA was addressing the small sample size problem that may occur in classical LDA. Also, two-dimensional linear discriminant analysis (2DLDA) [202] has been proposed for the same purpose, 2DLDA can be directly applied to the image matrix, which can use the structural information of the image for feature extraction. Another issue is that LDA fails to represent non-Gaussian distributed data. For this reason, authors in [225] proposed Manifold Partition Discriminant Analysis (MPDA) to solve the latter problem. MPDA jointly uses the neighbour information in addition to the label information in order to construct a discriminative embedding space. Sparse LDA (SLDA) [143] was proposed to overcome the issue of the presence of redundant features in the data. SLDA imposed the sparse constraint and was able to learn a sparse discriminant space. It is true that SLDA performs well on most classification tasks, but it still lacks the ability to implicitly perform feature selection. Recently, with the advent of deep learning methods, authors in [40] extended the original LDA criterion into a deep neural network framework and called it deep linear discriminant analysis (DeepLDA). The main goal of DeepLDA is to learn a model that can concentrate as much discriminative power as possible on the $C-1$ directions, with C denoting the class number. Similar to other deep architectures, DeepLDA provided efficient performance on large-scale image datasets. However, it requires a large amount of training samples to train the feature extraction network. Moreover, it is too difficult to interpret the model with the complex network structures.

All the above methods have contributed significantly to image and object classification, however, these approaches still have many shortcomings. These methods are still not able to provide the best features assessment, they are not robust to noise, and these methods cannot preserve discriminant information according to the selected number of projection directions and dimensions.

In order to address these issues and seek for an embedding space that provides better discrimination properties, authors in [186] proposed the Robust Sparse Discriminant Analysis (RSLDA) method, where the authors imposed the $\ell_{2,1}$ norm over the targeted transformation matrix to ensure that their method performs feature selection and extraction simultaneously. In addition, the authors introduced a sparse error term to fit noise during the learning process. They adopted the ℓ_1 norm for the error term to give the model the ability to handle the sparse noise. RSLDA is designed to be considered as a joint framework that integrates PCA and LDA into a single model. Moreover, the authors introduced an orthogonal matrix to connect the data in the original and transformed space and keep the main energy of the original data in the discriminant subspace.

RSLDA is a supervised LDA-based method used for feature extraction. Aiming to overcome some drawbacks of LDA technique and extract the features while keeping the main energy of the data and improving the robustness to noise, RSLDA solves the following optimization problem:

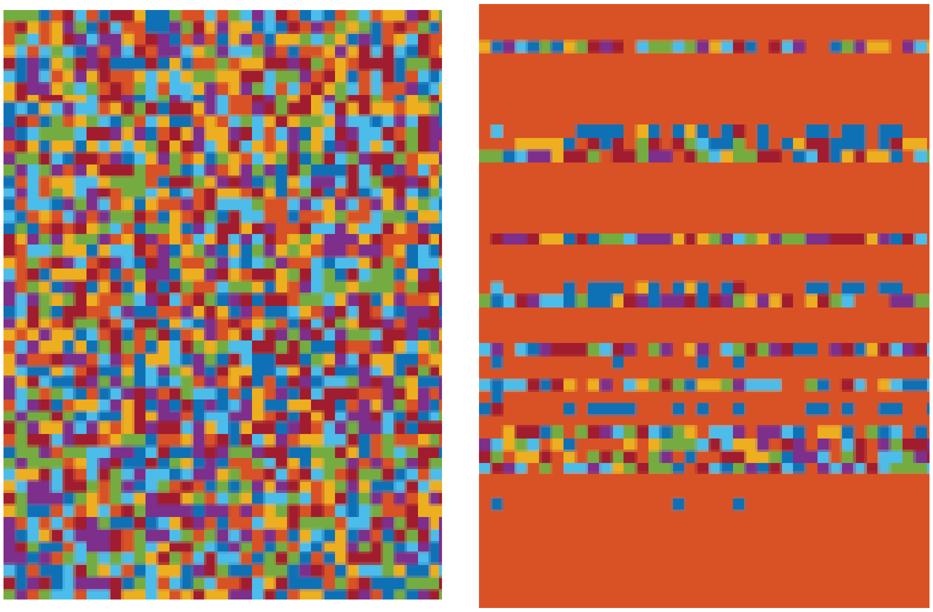
$$\min_{\mathbf{P},\mathbf{Q},\mathbf{E}} Tr\left(\mathbf{Q}^T\left(\mathbf{S}_w - \mu\,\mathbf{S}_b\right)\mathbf{Q}\right) + \lambda_1\,\|\mathbf{Q}\|_{2,1} + \lambda_2\,\|\mathbf{E}\|_1 \quad s.t. \quad \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} + \mathbf{E}, \mathbf{P}^T\,\mathbf{P} = \mathbf{I} \tag{2.22}$$

where $\mathbf{Q} \in \mathbb{R}^{d\times m}$ and $\mathbf{P} \in \mathbb{R}^{d\times m}$ denote the projection matrix and orthogonal reconstruction matrix, respectively, with $(m < d)$. λ_1 and λ_2 are trade-off parameters used to determine the importance of the different terms. $\mathbf{S}_w$ and $\mathbf{S}_b$ are the within-class and between-class scatter matrices respectively. **E** is the error matrix and μ is a constant used to balance the two scatter matrices.

The $\ell_{2,1}$ norm of the transformation matrix **Q** used in the optimization problem (2.22) can be calculated using equation (2.23).

$$\|\mathbf{Q}\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{d} q_{ij}^2} \tag{2.23}$$

According to [186], RSLDA learns a discriminant subspace and has less information loss than other LDA-based algorithms. Besides, RSLDA addresses the issue of model sensibility to reduced dimensions and can therefore provide very good performance even in cases where the projected space has very few dimensions. Figure 14 illustrates the projection matrices derived from both the original LDA in addition to the RSLDA algorithms. By looking at sub-figure 14b, it is obvious that RSLDA can identify and select the most discriminative features from the original data. One can see that many rows in the projection matrix associated with RSLDA have zero values, which represent irrelevant features that the model can discard.



(a) **Transformation matrix obtained by LDA.**

(b) **Transformation matrix obtained by RSLDA.**

Figure. 14: **Comparison of the transformation matrices provided by the original LDA and RSLDA on the Extended Yale B face database using 15 training samples from each class randomly selected. Note: only the first 50 rows of the projection matrices are visualized for comparison [186].**

More information on Robust Sparse Linear Discriminant Analysis can be found in [186].

2.4.4 Inter-Class Sparsity based Discriminative Least Square Regression

Least squares regression (LSR) has proven its effectiveness in classification tasks in the pattern classification and computer vision fields. LSR provided promising results especially in face recognition [197], microarray gene classification [112], cancer classification [58] and image retrieval tasks [50]. The main objective of LSR is to learn an embedding space that links the source and target data with minimal regression errors.

Several LSR-based methods were proposed and contributed to enhance the original (LSR) framework [6, 152, 151, 51, 20, 25], each of these methods provided significant improvement over the original (LSR). LSR-based methods have always been found to be very efficient as they possess several problem solving properties. In particular, these methods are known to overcome the small sample size (SSS) problem that LDA may also face. At the same time they greatly improve the computational performance compared to other type-based methods [49, 173]. In addition to being robust to the SSS problem, LSR-based approaches have proven to be more flexible to the introduction of novel regularization parameters than other conventional methods. This fact enabled these methods to achieve better data interpretability, resulting in superior performance. The supervised approach Linear Regression (LR) is arguably one of the most popular LSR-based methods. It has proven to be particularly powerful in classification applications. Under certain conditions, LR can be considered as an equivalent for LDA. This was discussed in the paper entitled Least Squared Linear Discriminant Analysis [205].

Generally, LR operates as follows: First LR approach defines a label matrix linked to the class labels. Next, LR seeks for a transformation matrix that can perfectly transform the samples into their corresponding labels. However, many issues still exist in the above LR based methods. The first of them is that the target matrix is too strict and inappropriate for classification [182, 110]. Strict binary label matrices usually lead to constant regression response distances for different class samples, which leads to a disturbance in the learning process. This is contradicting the reality which states that samples belonging to different classes should be as far apart as possible after transformation. This targeted labeled matrix needs to be relaxed in order to achieve superior performance. Another major problem with the LR based methods is that these methods can lead to overfitting the system. This can happen because sometimes these methods ignore the relationships among samples, in other words, they only focus on fitting the samples to the corresponding labels [194, 17].

Suppose $\mathbf{X} \in \mathbb{R}^{D \times N}$ is a training fata matrix and $\mathbf{Y} \in \mathbb{R}^{C \times N}$ denotes the corresponding label matrix, where D, N and C denote the feature dimension, the number of training samples and the number of classes, respectively. The standard LR (StLR) aims to learn a projection that transforms the given training samples into their respective class labels by minimizing:

$$\min_{\mathbf{Q}} = ||\mathbf{Y} - \mathbf{QX}||_F^2 + \lambda||\mathbf{Q}||_F^2 \tag{2.24}$$

where $\mathbf{Q} \in \mathbb{R}^{C \times D}$ is the projection matrix and λ is a regularization parameter to set the importance of the regularization term. $||.||_F$ is the Frobenius norm.

Several methods have been proposed to address the problems associated with LR based methods [183, 221]. In order to address these major issues, authors in [193] proposed the discriminative LSR (DLSR) that provided several innovations. First, the proposed method introduced a relaxed label matrix instead of the strict binary one. Moreover, DLSR presented the ϵ-dragging approach which aims to enlarge the distances of regression targets of different classes. DLSR demonstrated promising classification potentials, however, the adopted approach to relax the label matrix resulted in enlarging the regression responses distances between same class data samples. This disrupts the learning process.

To address this problem and further achieve better discriminative properties, authors in [188] proposed a new relaxed label regression method called Inter-Class Sparsity based Discriminative Least Square Regression (ICS_DLSR). The proposed approach generates a linear embedding of the unknown label space, where the space dimension is equal to the number of classes. This approach was able to produce a model in which the margins of data samples of the common class are significantly reduced, as opposed to those of samples of different classes, which were amplified. This was achieved by inserting a class-based row sparsity on the projected features. Unlike the previously mentioned LSR-based approaches, ICS_DLSR aims to preserve the row-sparsity consistency property of samples belonging to the same class. ICS_DLSR ensures that the regression responses distance between shared class samples will be significantly reduced, which can result in better performance. ICS_DLSR was able to achieve its goals by imposing a novel inter-class sparsity regularization term on the transformation. In addition, a sparsity error term was introduced into the LSR model to relax the strict label matrix for regression.

Overall, the main goal of ICS_DLSR is to provide an embedding space in which the same class transformed samples share a common sparse structure. In this sense, an inter-class sparsity constraint was introduced into the original least square regression model, such that the margins of samples belonging to the same class are greatly reduced, while the margins of samples belonging to different classes are enlarged.

Specifically, the authors of ICS_DLSR introduced two additional terms into the StLR framework. First, a novel inter-class sparsity constraint was introduced to ensure that the transformed samples share a common class structure. Second, a sparsity error term was also introduced to relax the strict label matrix **Y**. After this introduction, the proposed global criterion becomes as follows:

$$\min_{\mathbf{Q}} = ||\mathbf{Y} + \mathbf{E} - \mathbf{QX}||_F^2 + \frac{\lambda_1}{2}||\mathbf{Q}||_F^2 + \lambda_2 \sum_{i=1}^{C} ||\mathbf{QX}_i||_{2,1} + \lambda_3 ||\mathbf{E}||_{2,1} \tag{2.25}$$

where **Q** is the projection matrix, λ_1, λ_2 and λ_3 are three regularization parameters to determine the importance of the corresponding terms. $||.||_F$ is the Frobenius norm. C denotes the total number of classes. The authors of ICS_DLSR used the alternating direction method (ADM) [115, 116, 201] to derive the solution for the unknowns. ICS_DLSR was able to achieve outstanding results on classification tasks.

2.5 Overview on Deep Learning

In recent years, Deep Learning [101] has gained much attention in various fields. These methods provided outstanding performance breakthroughs in several areas, namely: speech recognition, natural language processing and computer vision [96, 103]. With the principle objective being implicitly capturing intricate structures of big data, deep learning techniques allowed multiple processing layers models to represent data with multiple levels of abstraction. Deep learning consists of several methods, the best known of which are neural networks, hierarchical probabilistic models, in addition to a large number of supervised and unsupervised feature learning approaches. The main impetus came from the desire to construct a model that mimics how the brain perceives and understands multi-model information. This has inspired much research in recent years. The initial development of a neural network was proposed by authors in [126], where the main objective of the authors was to understand how the human brain can develop complex patterns using interconnected cells called neurons. These authors proposed the McCulloch-Pitts (MCP) model, which is a basic neuron model that was a pioneering contribution to the field. Many methods were then proposed leading to the current "deep learning era". One of the most important contributions that led to the " deep learning era" is the work done in [72]. Authors in [72] introduced the multi layers (Deep Belief Network). Based on the Boltzmann Machines, the Deep Belief Network trains one layer at a time, guiding the training of intermediate levels of representation using unsupervised learning performed locally at each level. Deep Belief Networks based methods have demonstrated very efficient properties and are still being developed until our current date. These networks have been used by researchers for many tasks (e.g., medical image analysis [85, 87], cancer classification [5], hyperspectral image classification [224], electroencephalography [128]).

Some of the deep learning approaches were found to be significantly superior to the regular state-of-the-art techniques in various tasks (e.g., visual, audio, medical, social, and sensory). These techniques were able to process complex data in a more efficient manner. As time progressed, several factors contributed to the wide adoption of Deep Learning, the first of which would be the emergence of high-quality, labeled, and large datasets. Another reason that encourages more researchers to explore and investigate Deep Learning-based approaches is the rise of powerful parallel GPU computing, which has led to a significant acceleration in deep models' training. The wide availability of open source toolboxes specifically designed for Deep Learning has also helped.

2.5.1 Deep Convolutional Neural Networks

In general, the concept of convolutional neural networks is composed of two elements known as "Artificial Neural Networks" in addition to a set of operations known as "convolutions". If we recall the concept of neural networks, it is a system composed of artificial neurons that simulates the biological neurons for a specific task. Figure 15 illustrates a simple view over an artificial neuron, where $f(.)$ corresponds to the activation function. The inputs represented in the input set $X = (x_1, x_2, ..., x_N)$ are connected to f via the set of weights $\Omega = (\omega_1, \omega_2, ..., \omega_N)$ and the bias b, the output is finally represented by Z. The convolution operation mainly consists of applying some filters to an input signal.

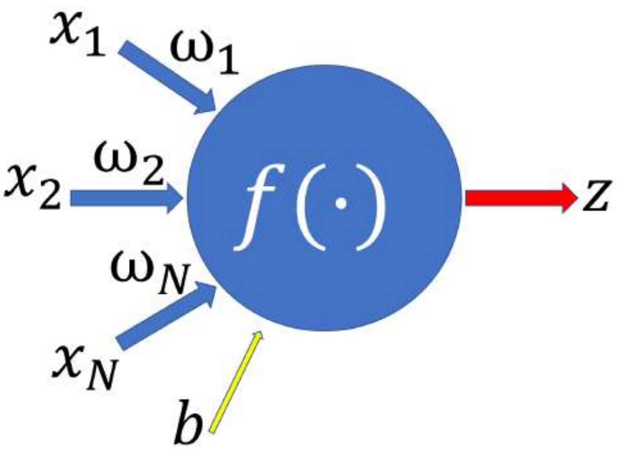


Figure. 15: **Simple artificial neuron illustration.**

Convolution Neural Networks (CNNs) are considered the most representative supervised deep learning models. CNNs proved to be very competitive and powerful in computer vision and image processing tasks (e.g., Image Classification, Image Segmentation, Object Detection, Video Processing, in addition to Natural Language Processing.) [27, 117]. A typical CNN architecture generally consists of alternating layers of convolution and pooling. These are succeeded by single or multiple fully connected layers. In some works, a fully connected layer is replaced by a global average pooling layer. An example of a simple CNN architecture is illustrated in Figure 16.

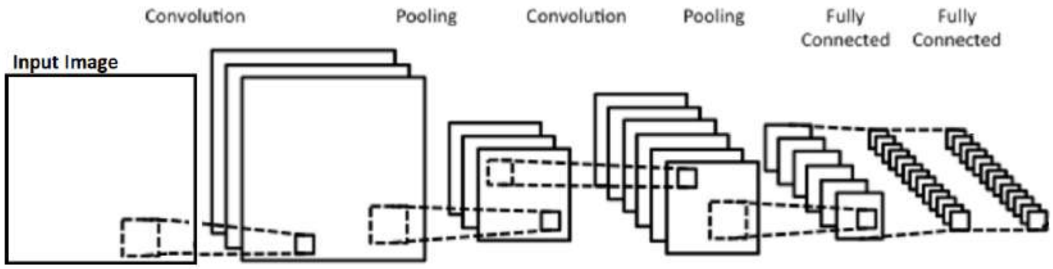


Figure. 16: **Simple Artificial Neural Network architecture.**

Convolutional Neural Networks acquire their efficient learning strength due to the multiple feature extraction stages they employ. In recent years, CNNs based approaches attracted much interest in the image processing and computer vision fields. Many methods with different architectures have been proposed and showed very optimistic performance, where the significant improvement achieved from one method to another is mostly related to novel architectural innovations (e.g., depth of the network, width, etc...).

Historically, several contributions targeting (CNNs) have been proposed. It all started in 1989 when a CNN was first proposed by the authors of [102]. Authors of [102] utilized the backpropagation method in the training process, the proposed framework sets the weights according to the target. Authors in [104] presented the Convolutional Neural Network as a feedforward multilayered hierarchical network, where each layer performs several transformations. The output of the convolutional kernels is then assigned to the nonlinear processing unit (activation function). The proposal of AlexNet by the authors in [96] was a breakthrough in the field, where the authors achieved a remarkable classification performance using the ImageNet dataset. Knowing that deep CNN methods require significant computational power, Alexnet has used parallel computing in the training process to overcome the shortcomings of the hardware. The network depth of AlexNet was extended from 5 (LeNet) to 8 layers to ensure that the network is applicable to multiple image categories. Since the authors of [96] knew that increasing the network depth generally leads to overfitting, they ensure that their algorithm skips some transformational units during the training process. This idea was inspired by the work in [29, 164], which presents a simple way to prevent neural networks from overfitting. Figure 17 illustrates the architecture of Alexnet. The success of AlexNet have pushed a lot of other researchers to adopt CNNs, resulting in many innovations. Much work was then implemented by altering the structure of the networks and designing new blocks.

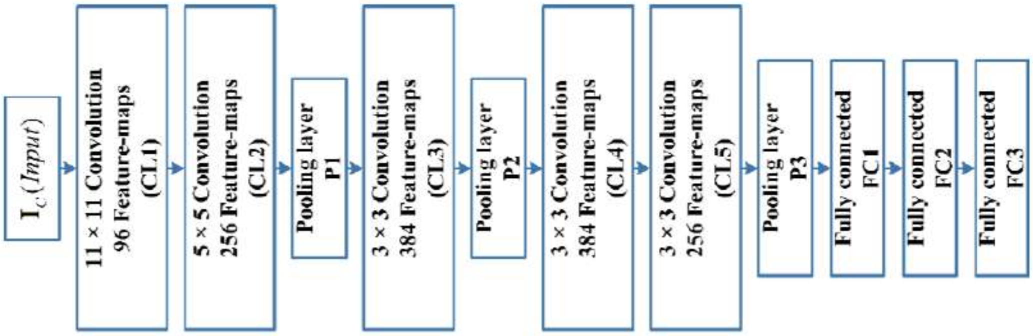

I_C (Input)
11 × 11 Convolution 96 Feature-maps (CL1)
5 × 5 Convolution 256 Feature-maps (CL2)
Pooling layer P1
3 × 3 Convolution 384 Feature-maps (CL3)
Pooling layer P2
3 × 3 Convolution 384 Feature-maps (CL4)
3 × 3 Convolution 256 Feature-maps (CL5)
Pooling layer P3
Fully connected FC1
Fully connected FC2
Fully connected FC3

Figure. 17: **AlexNet architecture showing its 8 layers [86].**

Another major work that made a huge impression was the authors' work in [159]. They proposed the very deep convolutional neural network (VGG), which demonstrated efficient performance in large-scale image classification and localization problems. VGG is characterized by its simplicity, homogeneous topology and increased depth. VGG is regarded as an innovative object recognition model that supports up to 19 layers, it has the ability of outperforming baselines on many tasks and datasets outside of ImageNet. The main drawback of VGG is that it is computationally intensive and requires high computational resources due to the use of 138 million parameters. Authors in [168] introduced the (GoogleNet) network, where the proposed network architecture allows achieving high performance with reduced computational cost. GoogleNet is even deeper than the above mentioned networks, it consists of a total of 22 layers. The architecture of GoogleNet is where the inventive idea of split, transform and merge with the corresponding block known as inception block was initially introduced. The proposed Inception Block introduced the concept of branching within a layer, which allowed the abstraction of features at different spatial scales. GoogleNet applied parameter tuning, which resulted in a huge reduction in the number of parameters, which significantly reduced the computational requirements. Another strength of GoogleNet is its fast convergence rate, which was achieved by introducing the auxiliary learners concept. GoogleNet showed very good characteristics both computationally and in terms of performance, however, its heterogeneous topology that needs to be customized from one module to another is its main limitation. Moreover, the use of GoogleNet may lead to the loss of relevant information at some points. This is due to the representation bottleneck that drastically reduces the feature space in the next layer.

Another widely used and well-known deep network is ResNet [62]. ResNet has greatly influenced the deep neural networks architectural innovations by introducing the concept of residual learning. Despite of having an architecture 8 and 20 times deeper than VGG and AlexNet, respectively, ResNet showed lower computational complexity than the two aforementioned networks. The authors in [62] empirically showed that ResNet, which consists of 50, 101, and 152 layers, leads to higher performance in image classification tasks than a simple network with 34 layers. Figure 18 illustrates the architecture of the residual block used by ResNet.

The concept of residual learning then inspired subsequent networks, such as Inception-ResNet, Wide ResNet, ResNeXt and others [167, 195, 212].

Similar to other tasks, Deep Learning using CNNs has both advantages and disadvantages. It is proven that the hierarchical structure of deep CNNs provides the ability to extract low, med and high-level features. Deep architectures usually have an advantage over the conventional architectures when it comes to complex learning problems. CNN based methods have shown performance enhancement over the conventional methods [134, 174]. However, deep learning approaches require enormous amounts of computational resources compared to conventional feature extraction and manifold learning approaches. Deep learning methods require expensive GPUs and hundreds of machines, which is very costly. Moreover, deep architectures are significantly slower than the conventional machine learning feature

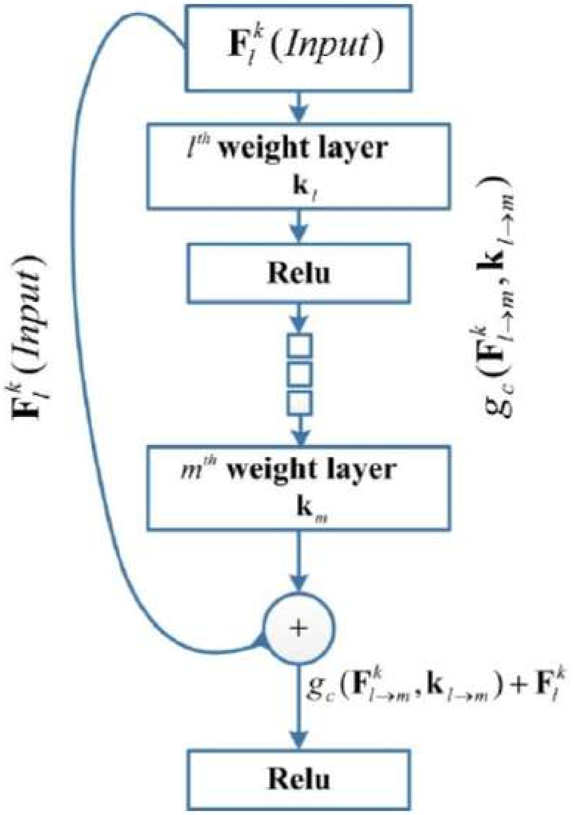

$\mathbf{F}_l^k\,(Input)$
l^{th} weight layer
$\mathbf{k}_l$
Relu
m^{th} weight layer
$\mathbf{k}_m$
$+$
$g_c(\mathbf{F}_{l\to m}^k, \mathbf{k}_{l\to m}) + \mathbf{F}_l^k$
Relu
$\mathbf{F}_l^k\,(Input)$
$g_c(\mathbf{F}_{l\to m}^k, \mathbf{k}_{l\to m})$

Figure. 18: **ResNet architecture of the residual block [86].**

extraction approaches. Another drawback is the fact that Deep Learning methods require very large amounts of data to demonstrate superiority over other feature extraction techniques.

These above factors have kept the focus on conventional machine learning algorithms. In particular, conventional supervised learning is very promising due to its high performance and learning ability. Additionally, conventional supervised learning approaches are noticeably simpler to implement and require far less computational resources and processing time. At the same time, most of the recent supervised learning approaches have demonstrated very efficient performance in various domains, especially in computer vision, image and object classification, and many others. Compared to Deep Learning based methods, conventional methods are better suited for tasks with small databases. For these purposes, we decided to propose several novel supervised learning algorithms for image categorization. Our proposed methods were able to demonstrate their efficiency and ensure high discrimination capabilities.

2.5.2 Graph-based Deep Learning

Deep Learning has experienced a breakthrough in recent years. Throughout history, the majority of Deep Learning studies have focused on different dimensional Euclidean- structured data (e.g., acoustic signals, images, and videos).

In general, 2008 was the year that the most important work exploiting Deep Learning using manifolds and graphs was introduced. Authors in [155] proposed the "graph neural network model", which became a breakthrough in the field.

Various techniques such as convolutional neural networks (CNN), recurrent neural networks and others, led to a breakthrough in the machine learning community. These techniques led to a surge in performance that could not have been imagined just a few years ago. The major success of learning techniques, particularly those related to convolutional neural networks, has merely came by working in the euclidean domain. However, in a multitude of other fields, we need to deal with various types of data that are best represented by manifolds and graphs (e.g., social networks, regulatory networks, 3D shapes). Much research and experimentation is being conducted for the purpose of generalizing deep learning frameworks to non-Euclidean structured data such as graphs and manifolds [15, 34, 70, 91].

As we have mentioned earlier, many researchers were fueled by the need to investigate how deep learning can be applied to non-euclidean data. Therefore, the first thing they thought of was to generalize convolution through spectral approaches, in this case the main idea was basically to generalize the Fourier convolution theorem to graph and manifold structured data. In other words, applying convolution in the spectral domain rather than the spatial one. To accomplish this, researchers began by considering the eigenfunctions of the Laplacian graph as a generalized version of the typical Fourier basis. One can obtain the graph convolution as follows: First, one projects the original signal over the Laplacian eigenfunctions, thus realizing a graph Fourier transform. Second, one multiplies the obtained spectrum by a set of spectral coefficients and third, one projects everything back to the original domain. An illustration of the steps used in spectral approaches is shown in Figure 19. The Laplacian has an eigenvalue decomposition $\Delta = \Phi\Lambda\Phi^T$, where $\Phi = (\phi_1, ..., \phi_n)$ are the orthonormal eigenvectors and $\Lambda = \text{diag}(\lambda_1, ..., \lambda_n)$ is the diagonal matrix of the corresponding eigenvalues. Given a signal $f = (f_1, ..., f_n)^T$ on the vertices of the graph G, its graph Fourier transform is given by $\hat{f} = \Phi^T f$.

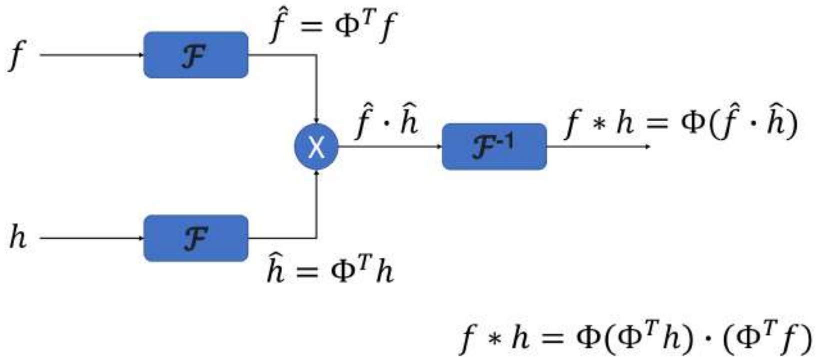


***Figure. 19:* Illustration of spectral approaches steps.**

The authors in [70] worked on Convolutional Neural Networks (CNN)s on graphs in spectral domain. Another important contribution is the work of authors in [34], authors implemented the CNNs on spectral graphs with the goal of designing fast localized convolutional filters on graphs. Authors in [91] proposed the famous Graph Convolutional Networks (GCN) algorithm aiming at a semi-supervised approach for learning the soft label in a transductive setting with graph-structured data. Another approach with semi-supervised setting is the method proposed by the authors in [108]. After that, many other algorithms related to the originally proposed one (GCN) have been proposed. The most famous of them are Graph Attention Networks [178], Graph Spatial-temporal networks [157], Graph Auto -encoders [18] , and Graph Generative Networks [210]. Other methods for improving spectral graph convolutional networks were later proposed [105, 109, 191, 204, 123, 124, 23, 24].

Despite the efficient performance provided by the spectral methods, there is a problem, namely that the eigenfunctions of the Laplacian are inconsistent across different domains. In other words, when using the same input signal and coefficients, the result is different from one case to another. This comes from the fact that graph Laplacian eigenfunctions exhibit different behavior across different domains. To solve this problem and to extend the convolution in a consistent way across different domains, the researchers proposed a second family of approaches, namely spatial approaches [127]. The main idea of spatial approaches is to apply a template to a neighborhood representation obtained by mapping the neighbors to a finite fixed structure.

The authors in [127] extended the Fourier operation to the non-Euclidean domain and generalized CNNs to graph- and manifold-structured data as well.

Here is a brief overview of how Deep Learning has been mathematically deployed on graphs. First, let's start with some notations. The assumed notations used in this section are summarized in the Table 4

***Table 4:* Notations used in the current section.**

Symbol	Description	Constraint/Formula
W	Adjacency Matrix	$w_{ij} = w_{ji}$ $w_{ij} = 0$ if$(i,j) \notin \varepsilon$ $w_{ij} > 0$ if$(i,j) \in \varepsilon$
$\mathcal{G}$	Undirected weighted graph	$\mathcal{G} = (\{1, ..., n\}, \varepsilon, \mathbf{W})$
Δ	Unnormalized Graph Laplacian	$\Delta = \mathbf{D} - \mathbf{W}$ $\mathbf{D} = \text{Diag}(\sum_j w_{ij}, i = 1, ..., n)$
$\Delta = \Phi\Lambda\Phi^T$	Laplacian eigendecomposition	–
Φ	Orthonormal eigenvectors	$\Phi = (\Phi_1, ..., \Phi_n)$
Λ	Diagonal matrix of corresponding eigenvalues	$\Lambda = \text{Diag}(\lambda_1, ..., \lambda_n)$

In harmonic analysis, the eigenvectors play the role of Fourier atoms, thus the eigenvalues can be interpreted as frequencies. First, given a signal $f = (f_1, ..., f_n)^T$ on the vertices of the graph $\mathcal{G}$, its graph Fourier transform is given by $\hat{f} = \Phi^T f$. The spectral convolution in the Euclidean case of two signals f and g can be defined as the element-wise product of their Fourier transforms as follows:

$$f \star g = \Phi\,(\Phi^T f) \circ (\Phi^T g) = \Phi\,\mathrm{diag}(\hat{g_1}, ..., \hat{g_n})\hat{f} \tag{2.26}$$

Authors in [16] used the spectral convolution presented in equation 2.26 to generalize CNNs on graphs as follows:

$$f_l^{out} = \xi\left(\sum_{l'=1}^{p} \Phi_k \hat{G}_{l,l'} \Phi_k^T f_{l'}^{in}\right) \tag{2.27}$$

where $F^{in} = (f_1^{in}, ..., f_p^{in})$ and $F^{out} = (f_1^{out}, ..., f_q^{out})$ denote the p and q-dimensional input and output signals on the vertices of the graph, respectively. $F^{in} \in \mathbf{R}^{n\times p}$ and $F^{out} \in \mathbf{R}^{n\times q}$. $\Phi = (\Phi_1, ..., \Phi_k)$ is an $n \times k$ eigenvectors-matrix and finally $\hat{G}_{l,l'} \in \mathbf{R}^{k\times k}$ is a diagonal matrix of spectral multipliers corresponding to a particular filter in the frequency domain, and ξ is a nonlinearity applied to the vertex-wise function values. This method proved very good contributions, however it has several drawbacks. One of these drawbacks is the high computational cost required for the process.

In order to address this issue and alleviate (reduce) the computational cost, authors in [34] used the Chebyshev polynomial basis. In that case, the spectral filters are represented as follows:

$$g_\alpha(\Delta) = \sum_{j=0}^{r-1} \alpha_j T_j(\hat{\Delta}) = \sum_{j=0}^{r-1} \alpha_j \Phi T_j(\hat{\Lambda}) \Phi^T \tag{2.28}$$

where $T_j(\lambda) = 2\lambda T_{j-1}(\lambda) - T_{j-2}(\lambda)$ represents the Chebyshev polynomial of degree j, when $T_1(\lambda) = \lambda$ and $T_0(\lambda) = 1$.

$\hat{\Delta} = 2\lambda_n^{-1}\Delta - I$ is a rescaled Laplacian such that its eigenvalues $\hat{\Lambda} = 2\lambda_n^{-1}\Lambda - I$ lie in the interval [-1, 1], and α denotes an r-dimensional polynomial coefficients vector for parameterizing the filter. By using this approach, the authors were able to address several drawbacks that arise in classical spectral convolution. Moreover, the computational complexity was alleviated from $\mathcal{O}(n^2)$ to $\mathcal{O}(rn)$, where r indicates how many times the Laplacian was applied.

Authors in [91] proposed the Graph Convolutional Network algorithm (GCN), which also contributed in a decent way in the graph-based networks area. The proposed algorithm is related to the work in [34] with additional assumptions. Assuming that $r = 2$, $\alpha = \alpha_0 = -\alpha_1$ and λ_n is approximately equal to 2 ($\lambda_n \approx 2$), the filter is expressed as:

$$g_\alpha(f) = \alpha(I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) f \tag{2.29}$$

By analyzing the constructed filter, we can realize the fact that the maximum eigenvalue of $I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ can be 2, therefore the filter is numerically unstable. In order to solve this problem, the authors renormalized the filter from equation 2.29, which is then expressed as follows:

$$g_\alpha(f) = \alpha \hat{D}^{-\frac{1}{2}} \hat{W} \hat{D}^{-\frac{1}{2}} f \tag{2.30}$$

where $\hat{W} = W + I$ and $\hat{D} =$ diag$(\sum_{j \neq i} \hat{w}_{ij})$.

2.5.3 Deep Metric Learning

Lately, Convolutional Neural Networks has achieved remarkable success in the fields of pattern recognition and computer vision. Metric learning is directly based on a distance metric that aims to assemble the similarity between different images. Recently, many researchers in the computer vision community are exploring deep metric learning approaches. These methods combine the idea of deep neural networks with the main objective of manifold learning. Deep metric learning uses neural networks to automatically learn discriminative features from images by optimizing a given objective function. These methods can show their superiority over conventional methods in many cases (e.g., faces of the same person when presented in different poses, expressions, illuminations). While metric learning has limited ability to capture nonlinearity in the data, deep metric learning helps in capturing the non-linear feature structure by learning a nonlinear transformation of the feature space. Building efficient classification models is strongly related to the design of appropriate loss functions that enable optimal class discrimination. In recent years, deep metric learning has been shown to deliver satisfactory results for various tasks such as face recognition, image classification, pattern recognition, anomaly detection, etc. Several methods have exploited deep metric learning, designed several loss functions and provided very good discrimination capabilities [185, 142, 190, 162, 90] .

2.6 Other Tools

Throughout the presented contributions in this thesis, we have used several well-known schemes in order to achieve our objectives. We have used these general schemes as tools that have helped us to accomplish our work. Whether they are considered mathematical tools or general ideas related to machine learning, these methods have contributed in enhancing the performance of our proposed approaches and achieving the desired optimal results. We have used several approaches, the most important of which are: (i) gradient descent algorithm and (ii) ensemble learning. In this section, we will briefly introduce these schemes, and describe how we used them to achieve our goals.

2.6.1 Gradient Descent

Gradient descent (GD) is an iterative optimization scheme used to minimize the function by moving toward the steepest descent direction in each iteration. The way the gradient method is applied differs through various fields. In machine learning and classification, gradient is used to iteratively update the parameter values of the desired model.

In general, the solution of optimization problems can be found using two approaches, the first is the "closed-form solution" and the second is the "gradient descent" method.

Throughout our multiple contributions, we have used both approaches to solve our proposed optimization problems, we have a adopted the closed-form solution for some of our suggested schemes and the gradient method for the others. The gradient algorithm has demonstrated excellent characteristics in solving unconstrained optimization problems. Besides its ability to provide accurate solutions, it is also characterized by its simplicity and low computational complexity.

Knowing that the general idea of gradient descent is mathematically related to the derivative of a function, one should first know the exact role of the derivative in the procedure. Understanding the real interpretation of the derivative is as important as being able to calculate it. Andrew Trask has given a very nice explanation about derivatives and how they work in his book "Grokking Deep Learning".

Gradient descent has always been used in machine learning and optimization to find the minimum of a convex function. This algorithm relies on properties of the first derivative to figure out in which direction and with what magnitude coefficients of the function should be modified. One of the most important constraints to work with the gradient descent algorithm is that the cost function should be differentiable, otherwise it is not possible to apply the gradient descent algorithm. One of the most commonly used cost functions for regression models is undoubtedly the mean squared error (MSE) function. Given that i, m, **y** and **ŷ** denote the index of samples, the number of samples, the expected and predicted values, respectively, the MSE can be calculated as follows:

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2.31}$$

It is usually very common for the cost function to be represented by the letter J. The number of derivatives that need to be calculated are related to the number of parameters in the desired function.

Figure 20 illustrates the general methodology of the gradient descent algorithm, showing the direction of gradient descent and the cost function value according to the weight.

If we want to interpret the gradient geometrically, it is possible to think of the derivative as the slope of the tangent line to the graph at the given point. Synonymous with the word "gradient" is the word "slope". The values of the model parameters are

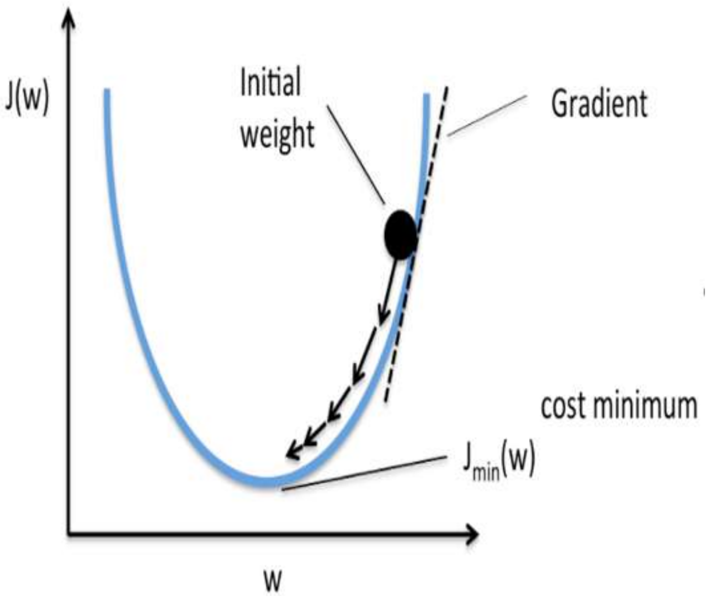


Figure. 20: **Gradient descent algorithm general methodology [4].**

.

usually randomized at the beginning. Their values determine the location of the point on the error curve (for a model with one parameter), the error surface (for a model with two parameters), or the error function (for more than two parameters). The goal of the Gradient Descent algorithm is to identify the parameter values for which the error is minimal. Figure 21 illustrates an error surface for a random function, where the minimum error value is represented by a white dot on the figure (Image Source: https://towardsdatascience.com/improving-vanilla-gradient-descent-f9d91031ab1d).

Let us suppose that w, w', b, b' and α denote the current weight value, the new weight value, the current bias value, the new bias value, and the learning rate, respectively. In order to achieve convergence, the parameters are generally updated iteratively as follows:

$$w' = w_0 - \alpha * \frac{\partial J(w_0, b)}{\partial w_0} \tag{2.32}$$

$$b' = b_0 - \alpha * \frac{\partial J(w_0, b)}{\partial b_0} \tag{2.33}$$

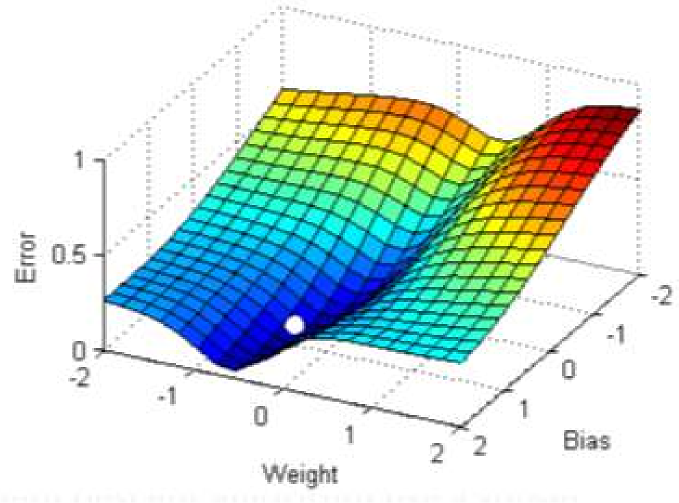


Figure. 21: **Error surface illustration.**

.

where $\frac{\partial J(w_0,b)}{\partial w_0}$, and $\frac{\partial J(w_0,b)}{\partial b_0}$ denote the derivative of the cost function J with respect to w_0 and b, respectively.

The parameters should be updated until the value of the cost function stops decreasing, and the whole process can be terminated if the current model state is already satisfactory.

In the machine learning field, the gradient algorithms have been used in a vast number of applications to solve the optimization problems associated with the learning models. (GDs) have demonstrated excellent properties in solving unconstrained optimization problems. They are characterized by their simplicity and low complexity. Many variations of the gradient approach have been proposed, tested and have shown their effectiveness. Some similar ones are "Adaptive Gradient Techniques" (AGT) which are very effective when working with sparse data. Adaptive gradients boost data's robustness [33]. AGT algorithms have been found to have certain limitations. It has been proved that the utilization of AGT eliminates the need to manually tune the learning rate. However, they reach a stage where they are unable to acquire new information due to the accumulation of square gradients in the denominator. As each additional term is positive, the cumulative sum tends to increase during training. This in essence, decreases the learning rate and ultimately makes it very poor. Many novel methods implement the gradient descent techniques in neural networks, these methods achieved very promising performance [38, 165]. Authors in [48] proposed a novel fast gradient approach for image classification using neural networks, which showed very promising performance.

2.6.2 Ensemble Learning

In the machine learning field, especially when talking about the methods where the main target is to provide a discriminative embedding space, a single model is usually sought. A model is usually constructed by specific mathematical operations guided by the global criterion of the corresponding algorithm. Once the model is obtained, it is used for the desired task (e.g., classification or some other performance evaluation protocol). Working with single models provided by powerful algorithms has always been an efficient approach in classification tasks. However, one may ask some questions such as: "Is it necessary that the performance obtained using a single model is the optimal performance that a given algorithm can provide?", and "Does working with a single model always reflect the full potential and discriminative properties of the algorithm?".

In reality, it is not necessary that learning with a single model always leads to the optimal performance provided by a proposed method. To address this problem and investigate how to improve the performance of different methods, some research investigated ensemble learning methods. An ensemble learning combines the predictions from multiple machine learning models into a single model that can reduce the generalization error. They offer increased flexibility and can scale in proportion to the amount of training data available. A few widely used ensemble approaches are bagging [14] and boosting [120].

The main idea of ensemble learning is to blend and combine the predictions from multiple models. These models are usually very good models and each of them provides a good discrimination property on its own. By combining these models, one obtains a single model that is characterized by improved discrimination ability. This leads to better classification. So the hypothesis is that in the case where the models are correctly combined, this can lead to more accurate and/or robust models. Ensemble learning consists of several methodologies (e.g. stacking, boosting, bagging, etc...). Figure 22 presents an overview of one general structure of ensemble learning methodology, where multiple subsets of the training data are used to create multiple models. The obtained models are then fed to a model combiner, resulting in a final model. The obtained final model can then be used for the desired tasks (e.g., classification).

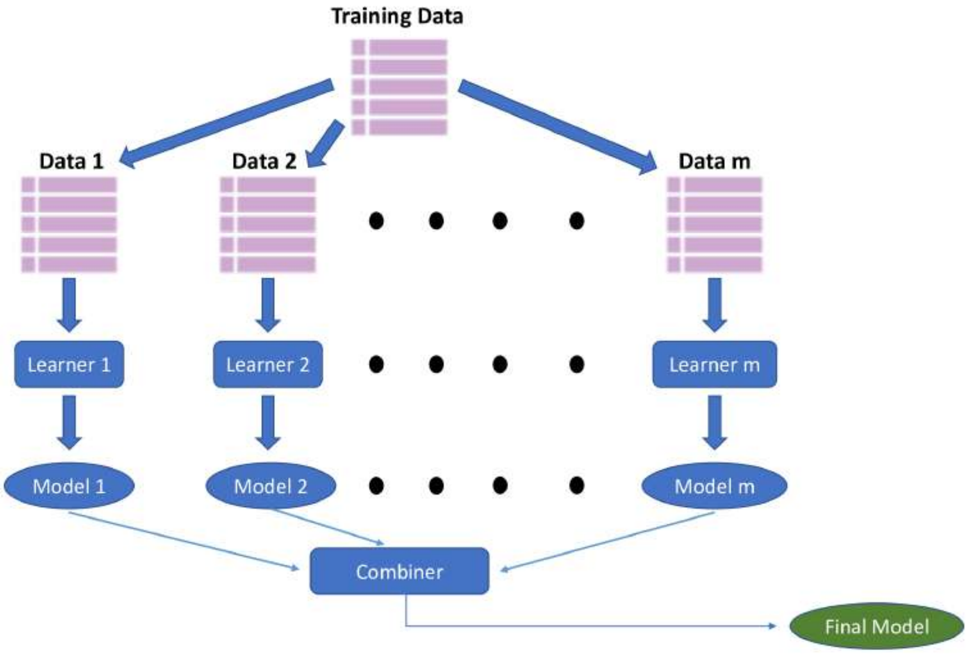


Figure. 22: **Ensemble Learning Overview. Note: the presented overview is a general structure, many components can be realized in several different ways.**

A variety of ensemble learning methods have been applied to classification tasks, mostly using deep convolutional neural networks (CNNs) for image classification. The reason is that ensemble learning has shown promising and excellent contribution to improve the performance of neural networks [35].

The performance of a single model is usually measured by its ability to determine the best predictor for the data. This can only be inferred after the classification process is complete. There is no way to realize this information beforehand by exploiting only the treated data and the optimization problem [97]. This was addressed in [141, 97].

This research focused on using a cross-validation strategy to evaluate the performance of each model. This strategy is referred to as the "discrete Super Learner selector".

Another view to ensure improved performance may be to estimate the optimal combination of models that leads to the best predictor. This has been well studied in the literature. Brieman addressed several related works regarding the theoretical properties of ensemble learning in [14] where he summarized the works of [12, 47, 55, 145, 150]. Another well-known strategy used in ensemble learning is called "stacking" [189], it involves combining the predictions of multiple models on the same data set. Many researchers have proposed linear combination approaches that introduce stacking into the ensemble of models [189, 14].

To derive the most efficient combination of models, the work described in [14] examined stacked regression using cross-validation. The cross-validation based work was extended with the aim of finding the best combination of predictors by proposing the "Super Learner" approach [97]. This framework showed superiority and very good contributions in several domains, namely: online learning [10], medicine [121, 192], spatial prediction applications [32] in addition to mortality prediction [19, 140].

CHAPTER 3

Experimental setup and Datasets

Contents

To carry out the various experiments reported in this thesis, we used several datasets of different types and scales. Image datasets depicting faces (with lighting, pose, and expression variations), objects, scenes, handwritten digits, and others were used to measure the performance of our proposed methods. In addition, a synthetic non-image dataset and an artificial pattern dataset were used to achieve more reliability. In this thesis, we focus on image classification tasks, so we used a wide range of image descriptors in our experiments. In this chapter, we give a brief description of the datasets and image descriptors used in this thesis, while explaining the experimental setups and the pre-processing techniques used (if applied).

3.1 Experimental Setup

To ensure a fair comparison between the proposed and competing approaches, the different experiments were conducted using the same experimental setup (datasets, percentage of training/test samples, dimensionality reduction techniques, etc.).

There are a variety of competing methods that we selected for comparison with our methods, these methods were selected based on the convenience of the experiments in each paper. In other words, the competing approaches sometimes differ from one paper to another. However, most of our works share these following methods as basic competing methods: K-nearest neighbors (KNN) [95], Support Vector Machines (SVM) [21], Linear Discriminant Analysis (LDA) [171], Local Discriminant Embedding (LDE) [22], Principal Coefficients Embedding (PCE) [139], Inter-class sparsity based least square regression (ICS_DLSR) [188] and Robust sparse LDA (RSLDA) [186]. Some additional methods including Linear Regression Based Classification (LRC) [129], Low-rank Linear Regression (LRLR) [17], Low-rank Ridge Regression (LRRR) [17], Sparse Low-rank Regression (SLRR) [17], Low-rank Preserving Projection via Graph Regularized Reconstruction (LRPP_ GRR) [187], Manifold Partition Discriminant Analysis (MPDA) [225], Sparse Uncorrelated Linear Discriminant Analysis SULDA [220], and Exponential Local Discriminant Embedding ELDE [41] are added to enrich the comparison for the Extended Yale B and the large PubFig83 dataset.

In addition to the previous ones, some deep learning methods were also included in our evaluations to measure the performance of our proposed methods against deep approaches. The derived results are presented in the corresponding tables in the contribution chapters.

For most of our experimental findings, the classifications were performed with 10 randomly selected splits for each dataset. In other words, the classification rates presented in the tables of our experiments are reported as the average classification accuracy over the 10 splits, unless otherwise stated in the results section for each experiment. We note that the SVM used in the experiments is the Linear SVM. It was implemented using LIBSVM library[1].

In our experiments, different training/testing percentages are used for each dataset. For each method, an embedding is first computed using the training portion of the data. The training and test data are then projected using the estimated embedding. Classification of the test data is then performed using either the Nearest Neighbor classifier (NN) [28] or the Support Vector Machines (SVM) classifier [57].

Most experiments invoked a dimensionality reduction of the raw features before feeding them to the learning models and classifiers. In most of our experiments, the data was pre-processed by using Principal Component Analysis (PCA), which was used as a dimensionality reduction technique while preserving 100% of the data's energy. We note that in some experiments performed, PCA was not used at all to illustrate the ability of the method in selecting the most relevant original features.

The reported classification rates were selected from the combination of the best parameter configurations and correspond to the average over 10 randomly selected splits, as mentioned earlier. In case a specific method required some kind of tuning or has a specific parameter, information can be found in the relevant chapters in Part II of this report.

[1] *https : //www.csie.ntu.edu.tw/ cjlin/libsvm/*

3.2 Datasets

Several real and synthetic sample datasets are used in our work. These datasets include face images, object images, handwritten digit datasets, and scene datasets. Despite the fact that most of our work focuses on image classification, a synthetic non-image dataset as well as a text dataset have also been used for broader evaluation.

3.2.1 Face Datasets

- **Extended Yale B Face Dataset**[2]**:** This dataset [53] is constructed from images of faces taken at different illuminations and facial expressions for each subject. The used dataset is a cropped version which contains between (58 and 64) images for each of the 38 individuals. It contains a total number of 2414 images, each of which is rescaled to 32× 32 pixels and represented through gray scale representation. Raw brightness images of dimension 1024 are used in the experiments. The reported results were obtained after we used 10, 15, 20 and 25 samples from each class as training samples and the remaining as test samples.

- **LFW-a Dataset** [3]**:** The Labeled Faces in the Wild-a (LFW-a) [75] dataset. While maintaining the structure of the original LFW dataset, LFW-a contains the images from the LFW dataset after alignment with commercial face alignment software. The used dataset contains images from 141 different classes with a total number of 3,408 gray-scale images, each rescaled to 32×32 pixels. Raw brightness images of dimension 1024 are used in the experiments. The reported results were obtained after we used 5, 6, 7, and 8 image samples from each class as training samples and the rest as test samples.

- **Georgia Face dataset** [4]**:** The Georgia face dataset contains a total of 750 images representing 50 individuals. Each individual is represented by 15 images showing frontal and tilted faces with different facial expressions, lighting

[2] *http : //vision.ucsd.edu/ leekc/ExtY aleDatabase/ExtY aleB.html*
[3] *https : //talhassner.github.io/home/projects/lfwa/index.html*
[4] *http : //www.anefian.com/research/face_reco.htm*

conditions, and scales. The images used are cropped and resized to 32×32 pixels for each image. Raw-brightness images (dimension 1024) are used in the experiments. The reported results were obtained after we used 3, 5, 7 and 9 image samples from each class as training samples and the remaining as test samples.

- **Honda dataset** [5]**:** The Honda face dataset contains a total number of 2,277 face images. It consists of 22 classes with approximately 97 images per class. The images represent faces subjected to different conditions. Raw brightness images are used in the experiments. The reported results were obtained after we used 10, 20, 30 and 50 image samples from each class as training samples and the rest as test samples.

- **FEI dataset** [6]**:** The FEI face dataset contains images of students and staff from FEI. It is a face dataset that contains a set of colored face images taken against a white background. The images are in an upright frontal position with a profile rotation of up to approximately 180 degrees. This dataset contains a total of 700 images, 14 images for each of the 50 subjects. The images are resized to 32 × 32 pixels. Raw brightness images of dimension 1024 are used. The reported results were obtained after we used 5, 6, 7, and 8 image samples from each class as training samples and the rest as test samples.

- **PubFig83 dataset** [7]**:** The PubFig83 dataset is a large scaled and challenging dataset that contains 13,002 images representing faces, collected in different situations (e.g., facial expressions, illuminations, backgrounds, and different poses). The images in this dataset represent 83 different individuals, each of which has between 46 and 231 images. We used 8720 images for training and the remaining 4282 for testing. HOG , LBP and Gabor wavelet features are extracted and concatenated from the aligned face images, then finally reduced

[5] *http : //vision.ucsd.edu/ leekc/HondaUCSDVideoDatabase/HondaUCSD.html*

[6] *https : //fei.edu.br/ cet/facedatabase.html*

[7] *http : //www.briancbecker.com/blog/research/pubfig83 − lfw − dataset/*

to 2048 dimensions using PCA. The methods are compared with respect to the experimental settings presented in [7].

3.2.2 Objects Datasets

- **COIL20 Object Dataset [8]:** The Columbia Object Image Library (COIL20) [130] dataset is constructed from images of different objects, with each object rotated around a vertical axis. The dataset used in our works contains images of 20 objects, each with 72 images, resulting in a total number of 1,440 images. The image descriptor used is the Local Binary Patterns (LBP) [107]. The uniform LBP histogram (59 values) was used. Three LBP descriptors are constructed from the image using 8 points and three values for the radius (R=1, 2 and 3 pixels). Thus, the final concatenated descriptor has 177 values. The results are obtained after we use 20, 25, 30 and 35 image samples from each class as training samples and the remaining as test samples.

- **Caltech101 Dataset [9]:** The used Caltech101 dataset contains images of objects belonging to 101 classes. The full Caltech dataset, consisting of 256 classes, can be found at [56]. It is a well-known, challenging set that contains a set of images with complicated backgrounds. We used a cropped version of the original Caltech dataset, which consists of 3,030 images, 30 images for each of the 101 classes. The reported results were obtained after we used 5 image samples from each class as training samples and the rest as test samples.

  The image descriptor used is the block-based LBP [107] representation. We have used 100 blocks. For each block, we extract the uniform LBP histogram (59 values). Thus, the length of the image descriptor is 5900.

  Moreover, we use the deep features provided by the ResNet-50 [63] convolutional neural network. This is a 50 layer convolutional neural network that is trained on the ImageNet database. By using this network, we are able to extract

[8] $http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php$

[9] $http://www.vision.caltech.edu/Image_Datasets/Caltech101/$

the image representation in the Average Pooling layer. The latter is considered as an image descriptor with a 2048-dimensional vector.

3.2.3 Handwritten digits

- **USPS Digits Dataset [10]:** The US Postal Service or abbreviated (USPS) [156] is a handwritten digits dataset used for digit recognition. This dataset contains 110 images for each digit from 0 to 9, thus, it consists of 10 classes, each of which contains 110 images, so a total of 1100 images are used in this dataset, the dimension of the images is 16×16. Raw-brightness images are used. The reported results were obtained after we used 30, 40, 55 and 65 image samples from each class as training samples and the remaining as test samples.

- **MNIST dataset [11]:** The Modified National Institute Of Standards and Technology dataset, abbreviated as (MNIST), is a challenging and large dataset that contains images of handwritten digits. The dataset used in the experiments contains a total number of 60,000 images representing 10 classes. The image descriptor used for the MNIST dataset has a length of 2048 and is obtained from the ResNet-50 convolutional neural network. The results are obtained after we use 1000 image samples from each class as training samples and the rest as test samples.

3.2.4 Scene Datasets

- **Outdoor Scene dataset [12]:** This scene dataset contains 2,688 images belonging to 8 groups. The descriptor used consists of 256 HOG features.

3.2.5 Text Datasets

- **20 News text dataset [13]:** Originally, the data in this dataset is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related (e.g. comp.sys.ibm.pc.hardware /

[10] *https : //www.kaggle.com/bistaumanga/usps − dataset*
[11] *http : //yann.lecun.com/exdb/mnist/*
[12] *https : //github.com/sudalvxin/SMSC/tree/master/data*
[13] *http : //qwone.com/ jason/20Newsgroups/*

comp.sys.mac.hardware), while others are highly unrelated (e.g. misc.forsale / soc.religion.christian). For our analyzes, we adopted a cropped version of the 20 newsgroups dataset with binary occurrence data for 100 words over 16,242 postings. The selected dataset contains a total of 2000 samples belonging to 4 classes.

3.2.6 Synhetic Datasets

- **Tetra synthetic dataset:** The Tetra dataset was defined in [175, 176]. This dataset consists of 400 data points belonging to four classes. The data points are in $\mathbb{R}^3$, this dataset presents the challenge associated with small inter-cluster distances.

Figure 23 presents some typical images associated with some of the datasets used in our evaluations.

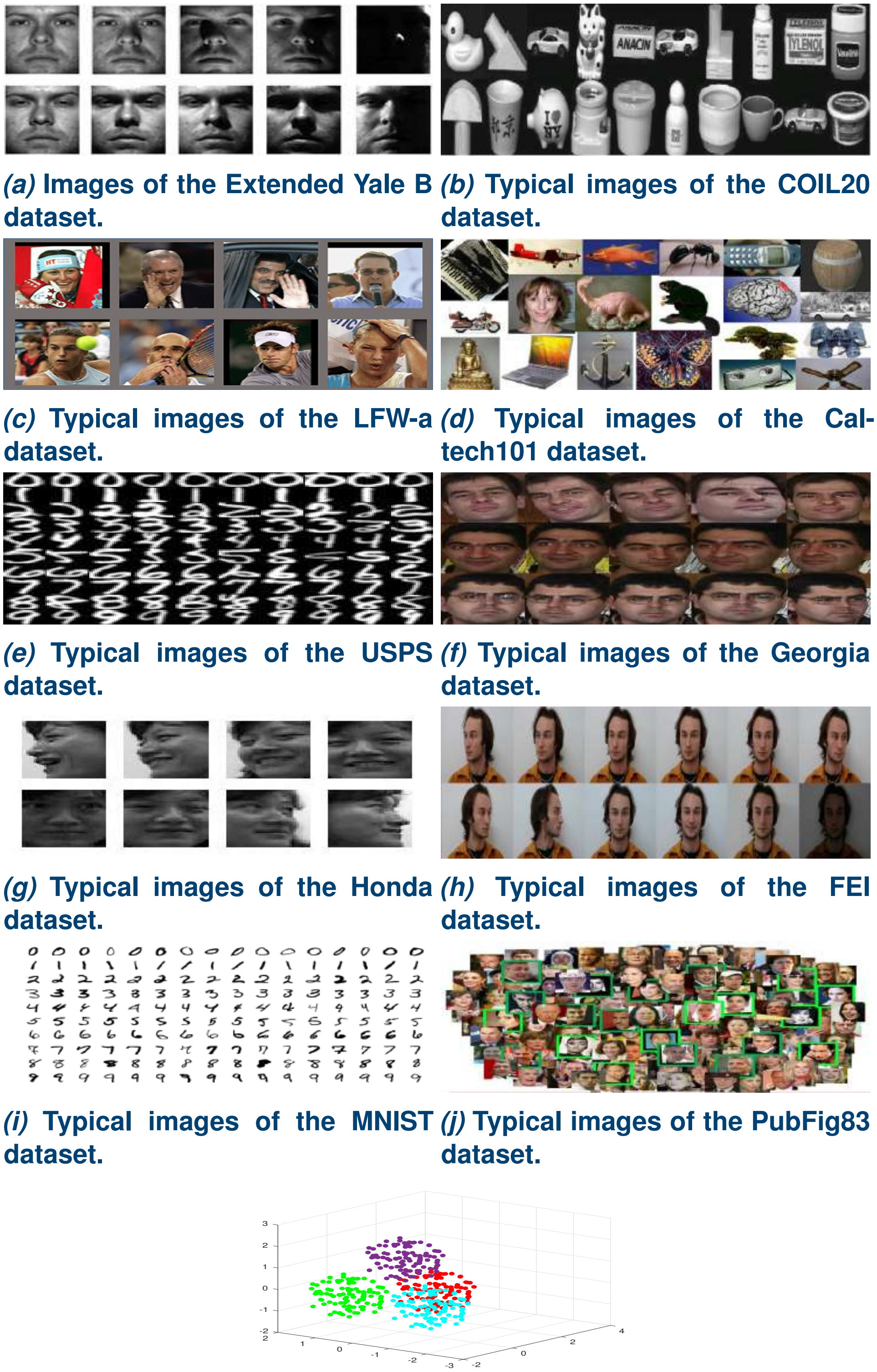
Table 5 illustrates a brief description over the used datasets in this report.

Table 5: **Datasets brief description.**

Dataset	Type	# Samples	# Features	# Classes	Descriptor
Extended Yale B	Face (images)	2414	1024	38	RAW-brightness images
LFW-a	Face (images)	3,408	1024	141	RAW-brightness images
Georgia	Face (images)	750	1024	50	RAW-brightness images
Honda	Face (images)	2277	1024	22	RAW-brightness images
FEI	Face (images)	700	1024	50	RAW-brightness images
PubFig83	Face (images)	13,002	2048	83	Concatenation of HOG,LBP and Gabor wavelet
COIL20	Object (images)	1440	177	20	3 concatenated Local Binary Pattern histograms
Caltech101	Object (images)	3,030	5900	101	3 Block-based LBP (100 blocks * 59)
			2048		Deep features (ResNet-50)
USPS	Digits (images)	1100	256	10	RAW-brightness images
MNIST	Digits (images)	60,000	2048	10	ResNet-50
20 News	Text	2,000	100	4	Term Frequency times Inverse Document Frequency
Outdoor Scene	Scene (images)	2,688	256	8	HOG features
Tetra	Synthetic	400	3	4	Coordinates

3.3 Descriptors

In the computer vision field, image descriptors are descriptions of the content in images. In general, descriptors provide elementary characteristics of images (e.g., the shape, the color, the texture, etc.). These descriptors have a good knowledge of the objects represented in the images and allow efficient interpretation of the image



(a) Images of the Extended Yale B dataset.

(b) Typical images of the COIL20 dataset.

(c) Typical images of the LFW-a dataset.

(d) Typical images of the Caltech101 dataset.

(e) Typical images of the USPS dataset.

(f) Typical images of the Georgia dataset.

(g) Typical images of the Honda dataset.

(h) Typical images of the FEI dataset.

(i) Typical images of the MNIST dataset.

(j) Typical images of the PubFig83 dataset.

(k) Visualization of the Tetra dataset.

Figure. 23: Typical images of different datasets.

contents. There are various types of descriptors, those that represent the raw images and others that can be extracted and learned through a special process for better data representation. In this thesis, we have worked with several descriptors and used all the latter in classification tasks. The descriptors used can be found in the last column of Table 5.

CHAPTER 4

Contributions

Contents

Our dissertation is mainly concerned with the development of novel supervised learning feature extraction techniques intended for image categorization applications. During the PhD study, we were able to provide several algorithms that ensured the delivery of discriminative and efficient embedding spaces for the data. We provided powerful and efficient data representations that we used in classification tasks. In addition to learning methods, we also exploited ensemble learning approaches and proposed a novel supervised scheme based on the ensemble learning concept. The proposed scheme was able to provide significantly superior performance compared to the single model. Moreover, based on the idea that the data usually consists of class-shared and class-specific information, we developed a criterion that captures the discriminative capabilities that can be provided by exploiting the class-specific information of the data. We made this possible by extracting the useful information from each class of the data separately. This last contribution is still under investigation and up to this point only preliminary results have been obtained, thus is presented as future work in the perspectives section 5.2.

Indeed, there exist some connections between most of our contributions. Multiple contributions in this thesis share the same overall modeling, however, these contributions differ on many levels. The main differences between the first three contributions are: the chosen optimization approach, the initialization of the embedding matrix, and the exploitation of the hybrid initialization scheme for the sough linear embedding. The fourth contribution is the only one that is not linked to the others in this report. It is an ensemble learning based approach that exploits the use of multiple feature subsets and multiple feature selection methods to provide more discriminant data representations. The remaining papers extend the experimental results.

In this chapter, we will present a brief summary of the contributions presented in this thesis. Detailed information and the complete methodology for each proposed method can be found in the corresponding chapter presented in the second part (Part II) of this report.

4.1 Linear embedding by joint Robust Discriminant Analysis and Inter-class Sparsity

The classical Linear Discriminant Analysis (LDA) and its variants are one of the best known and most widely used supervised feature extraction approaches. These methods have been used for various classification tasks. However, they have some limitations that need to be overcome. The main limitation is that the projection obtained by LDA does not provide good interpretability of the features. In addition, most LDA-based approaches do not provide feature ranking, hence they lack the ability of selecting the most relevant data features. In order to enhance the discrimination ability and provide better feature extraction, we propose a novel supervised method for multi-class classification that performs feature selection and extraction simultaneously. The targeted transformation focuses on the most discriminative original features while ensuring that the transformed features (extracted features) belonging to each class share a common sparse structure. Our proposed method is entitled Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity (RDA_FSIS). The corresponding model integrates two types of sparsity. The first type is achieved by imposing the $\ell_{2,1}$norm constraint on the projection matrix to ensure that the suggested scheme implicitly performs feature selection. The second type of sparsity is achieved by imposing the inter-class sparsity constraint on the projected samples to ensure a common sparsity structure in each class. An orthogonal matrix is also introduced in our model to guarantee that the extracted features can retain the main variance of the original data, thus improving the robustness to noise. The proposed method retrieves the LDA transformation by considering the two introduced types of sparsity. We solved the proposed criterion as a non-convex optimization problem using the alternating direction method of multipliers [13]. Through our optimization, we used the closed form solution in order to compute the sought transformation matrix in each iteration.

Various experiments are conducted on multiple image datasets with different types and scales. The projected features are used for multi-class classification. The obtained results show that the proposed method outperforms other competing methods by learning a more compact and discriminative transformation.

Figure 24 illustrates the principle of the proposed model, exploiting original features and inter-class sparsity. Yellow dots, red triangles, and blue squares represent samples from the first, second, and C-th class, respectively. The left part of the figure illustrates the input data (as a cloud of points and as a data matrix). The right part illustrates the expected projection of the cloud and the data matrix. Given [**X**= **X**$_1$,**X**$_2$, ..., **X**$_C$] denote the samples from the first class to the C-th class. [**Q**^{T} **X**$_1$,**Q**^{T} **X**$_2$,...,**Q**^{T} **X**$_C$] are the projected samples. **Q** is the sought transformation matrix.

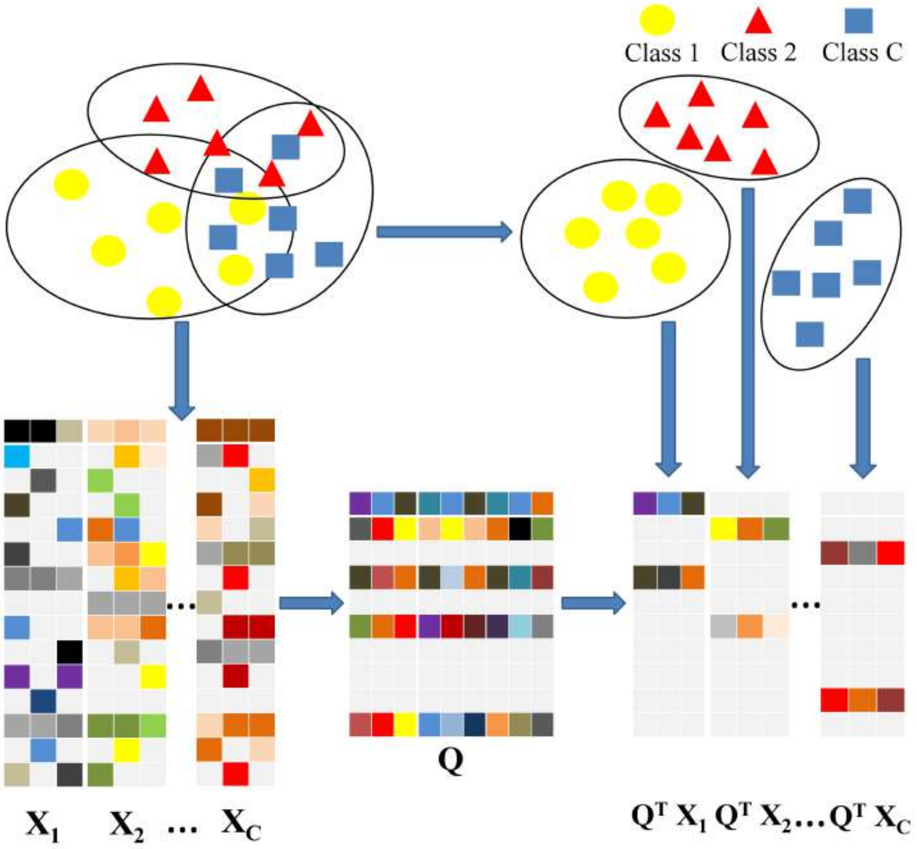


Figure. 24: **Illustration of the RDA_FSIS method where the original features and inter-class sparsity are exploited.**

Table 6: **Mean classification performance (%) of the RDA_FSIS method using the Extended Yale B dataset.**

No	KNN	SVM	LDA	LDE	ELDE	PCE	SULDA	MPDA	ICS_DLSR	RSLDA	RDA_FSIS
10	69.8	73.85	82.32	79.92	85.85	86.39	84.61	83.67	86.56	86.79	**88.27**
15	75.2	80.02	86.76	83.77	89.30	89.23	88.72	86.82	89.53	89.93	**91.73**
20	80.24	85.79	90.7	88.44	93.07	92.19	91.66	90.38	93.14	93.59	**95.11**
25	82.24	89.03	92.17	90.43	94.09	93.35	92.14	91.79	94.50	94.92	**96.23**

Tables 6-7 illustrate the mean classification rates of the RDA_FSIS among other

Table 7: **Mean classification performance (%) of the RDA_FSIS method on the tested datasets.**

Dataset\Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**RDA_FSIS**
COIL20	20	94.58	97.65	96.19	95.00	94.87	**98.04**	96.73	97.85
	25	95.79	98.22	97.07	96.12	95.99	98.22	97.74	**98.60**
	30	96.65	98.70	97.81	97.01	97.49	98.75	98.26	**99.10**
	35	97.14	98.81	98.15	97.42	98.11	99.12	98.68	**99.36**
Georgia	3	52.57	56.22	48.18	52.77	46.43	59.73	62.32	**62.67**
	5	61.28	66.98	59.20	62.14	56.18	71.12	73.48	**74.28**
	7	66.73	72.83	67.83	67.10	62.15	78.38	78.82	**79.98**
	9	71.40	77.53	72.57	72.13	66.37	82.57	82.77	**83.30**
Honda	10	64.12	71.32	65.95	65.74	61.86	70.79	69.90	**72.48**
	20	77.69	83.60	79.39	79.25	75.33	82.95	83.03	**84.19**
	30	84.78	89.09	85.84	86.24	82.55	88.20	89.04	**89.44**
	50	91.36	94.15	92.28	92.34	90.03	93.53	94.13	**94.54**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	93.19	**94.01**
	6	90.35	92.93	94.18	92.15	88.73	93.65	94.25	**94.63**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.66	**96.09**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.43	**96.67**
USPS	30	87.01	88.21	84.91	83.54	72.01	88.46	89.45	**90.05**
	40	88.56	90.40	86.19	85.3	72.30	90.16	91.11	**91.27**
	55	90.51	92.09	88.64	87.16	73.32	91.25	**92.65**	92.56
	65	91.76	93.16	89.29	88.58	74.11	91.53	92.89	**93.33**
LFWA-a	5	9.90	12.72	20.51	9.98	9.44	22.56	24.70	**28.07**
	6	10.57	13.61	25.28	10.49	10.26	25.72	28.42	**30.98**
	7	11.06	14.70	28.62	11.24	10.98	29.04	31.50	**33.28**
	8	11.35	15.72	32.42	11.71	11.73	31.92	32.48	**35.80**

Table 8: **Mean classification accuracies (%) of different methods on the Caltech101 dataset using LBP and deep features.**

Caltech101	**5 training samples**	
Method	LBP features	Deep features
ICS_DLSR	17.20	84.86
RSLDA	16.00	85.34
RDA_FSIS	**17.81**	**85.69**

competing methods using a part of the tested datasets. The classifier used for classification is the Nearest Neighbors (NN) classifier with the number of neighbors set to one (1-NN classifier). The depicted rates are the average over 10 random splits and each corresponds to a different number of training samples.

Table 8 presents the classification performance of the RDA_FSIS method along with other competing methods using the Caltech101 dataset in the cases where the descriptor varies between LBP and deep features. For the case of deep features, we did not use the PCA preprocessing. The bold numbers denote the best results obtained in each experiment. Table 9 illustrates the classification performance of the

Table 9: **Mean classification performance (%) of the RDA_FSIS method on the PubFig83 dataset.**

Method	Classification accuracy
KNN	63.35
SVM	82.60
LDA	77.95
LDE	62.89
ELDE	65.88
PCE	50.40
SULDA	81.26
MPDA	67.89
ICS_DLSR	85.19
RSLDA	84.78
DeepLDA	44.35
Alexnet	64.00
Resnet50	**90.40**
RDA_FSIS	84.84

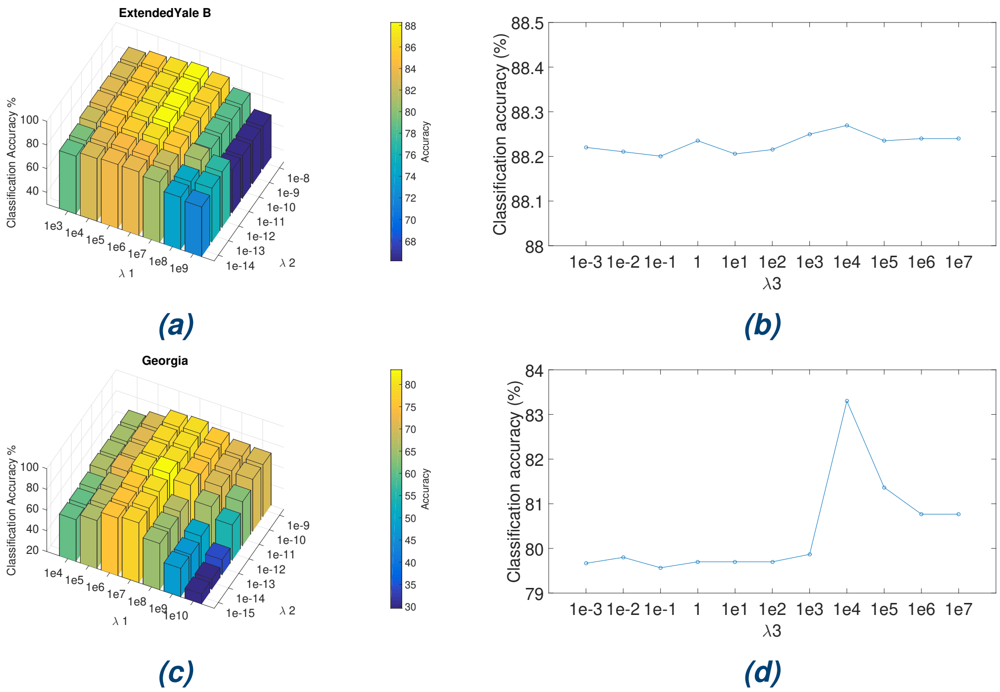
RDA_FSIS method using a single split for the large-scale PubFig83 dataset. The classifier used to obtain these results is the Nearest Neighbor (NN) classifier.

Figure 25 illustrates the parameter sensitivity of the proposed RDA_FSIS approach. In this figure, we explored how the classification performance varies depending on the use of different parameter combinations for the proposed approach.

Detailed information about this contribution are presented in Part II, Chapter 1.

4.2 An enhanced approach to the robust discriminant analysis and class sparsity based embedding

The main goal of this approach is to improve linear feature extraction used for supervised multi-class classification problems. Inspired by our proposed RDA_FSIS framework, we propose a unifying criterion that is able to retain the advantages of our powerful linear discriminant method by exploiting several types of sparsity. The proposed approach differs from the first contribution in two ways, namely: (i) The global criterion and (ii) the optimization process. In this proposed method, we have adopted the gradient descent approach to estimate the sought linear transformation instead of



(a)

(b)

(c)

(d)

Figure. 25: **Classification Performance (%) of the RDA_FSIS method according to the parameters combinations using the Extended Yale B and Georgia datasets in which 10 and 9 samples from each class are used for training, respectively. In subfigures (a) and (c), λ_3 is fixed, while in subfigures (b) and (d), λ_1 and λ_2 are fixed.**

using the closed form solution. Considering that the projection matrix requires a good initial estimate (since it is estimated by a steepest gradient descent scheme), we have used two initialization procedures leading to two variants of the proposed algorithm. The first variant is entitled Robust Discriminant Analysis using Gradient Descent (RDA_GD). In this variant, the initial estimate of the linear transformation matrix is set to the solution of the RSLDA method, which makes the transformation inherit the feature selection capability provided by RSLDA. The second variant, referred to as Enhanced Discriminant Analysis with Class Sparsity EDA_CS, sets the initial guess to the solution provided by our previously proposed RDA_FSIS. This allowed the second proposed variant to inherit the feature ranking along with the inter-class sparsity advantages exploited by the RDA_FSIS method. Although the main goal of the current work is to refine the solution provided by the "Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity" (RDA_FSIS) method, the proposed learning model can be used to refine the solution of many other linear methods.

The proposed framework can be considered as a fine-tuning technique that can be applied to various linear feature extraction methods.

Experiments have been conducted on several public image datasets of different types, including objects, faces and digits. The proposed framework compared favorably with several competing methods.

The derived findings are summarized in Table 10. This table depicts the classification rates as well as the standard deviations of the two proposed variants and the competing methods using multiple datasets. The results are obtained using different training and testing percentages of the data and are the average rate obtained over 10 random splits.

The last row in Table 10 illustrates the classification accuracy using the large scale MNIST dataset (60,000 images). The results for the MNIST dataset were obtained using a **single split** adopting 1000 samples from each class for training.

For detailed information about this contribution, please refer to Part II, Chapter 2.

4.3 A hybrid discriminant embedding with feature selection: application to image categorization

In this contribution, we have presented a unified and hybrid discriminant embedding method that minimizes the loss of discriminative information. This method is the first work that introduces the hybrid initialization process in the field, which allows the proposed approach to inherit the discriminative capabilities provided by various schemes simoultaneously. The proposed method differs from the existing related methods at many levels in terms of criterion design, optimization technique and initialization process. As for the criterion design, the proposed method integrates LDA and a variant of PCA into a joint learning framework. It inherits the excellent discrimination capability of LDA while enabling the reconstruction of the original data with minimal

***Table 10:* Mean classification performance (%) of the two variants of the enhanced discriminant approach using gradient descent technique on the tested datasets.**

Dataset\Method	Train. / class	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	RDA_FSIS	**RDA_GD**	**EDA_CS**
USPS	30	87.01±1.5	88.21±1.2	84.91±1.7	83.54±1.3	72.01±1.1	88.46±0.8	89.45±1.2	90.05±0.8	89.50±1.2	**90.40**±0.8
	40	88.56±1.6	90.40±0.9	86.19±0.9	85.3±1.2	72.30±1.7	90.16±0.7	91.11±1.0	91.27±0.9	**91.81**±1.1	91.76±0.5
	55	90.51±1.4	92.09±0.8	88.64±1.0	87.16±1.7	73.32±2.2	91.25±1.2	92.65±1.1	92.56±1.2	93.07±1.0	**93.40**±1.0
	65	91.76±1.3	93.16±0.9	89.29±1.5	88.58±1.1	74.11±1.9	91.53±1.3	92.89±1.0	93.33±1.0	93.71±0.9	**93.73**±0.6
Honda	10	64.12±2.1	71.32±2.1	65.95±2.2	65.74±2.2	61.86±2.2	70.79±2.5	69.90±2.1	72.48±2.0	70.16±1.9	**72.73**±2.0
	20	77.69±1.2	83.60±1.0	79.39±1.4	79.25±1.7	75.33±1.4	82.95±1.2	83.03±1.3	84.19±1.4	83.60±1.2	**84.40**±1.4
	30	84.78±1.3	89.09±1.0	85.84±1.1	86.24±1.1	82.55±1.8	88.20±1.0	89.04±1.2	89.44±1.0	89.41±1.1	**89.66**±1.1
	50	91.36±0.9	94.15±1.2	92.28±1.1	92.34±0.8	90.03±0.7	93.53±0.6	94.13±0.8	**94.54**±1.0	94.53±0.8	94.45±0.9
FEI	5	88.98±2.5	91.18±2.3	92.60±3.6	90.67±2.6	86.04±3.2	92.16±2.7	93.19±2.5	94.01±2.3	93.81±2.6	**94.24**±2.7
	6	90.35±2.7	92.93±2.8	94.18±3.9	92.15±2.7	88.73±3.7	93.65±2.7	94.25±2.3	94.63±2.3	94.75±2.5	**94.80**±1.9
	7	92.60±3.6	94.31±2.5	95.60±3.5	94.26±3.0	91.09±4.2	95.20±2.2	95.66±1.5	96.09±1.5	96.20±1.5	**96.26**±1.8
	8	94.27±2.9	95.23±2.2	96.03±3.5	95.57±2.4	93.20±4.4	96.17±1.9	96.43±1.6	96.67±1.7	**96.97**±1.7	96.87±2.0
COIL20	20	94.58±0.9	97.65±1.3	96.19±0.8	95.00±0.7	94.87±1.6	98.04±0.5	96.73±0.6	97.85±0.6	96.89±0.6	**98.05**±0.6
	25	95.79±0.8	98.22±0.7	97.07±0.8	96.12±0.7	95.99±1.3	98.22±0.6	97.74±0.7	98.60±0.5	97.89±0.5	**98.74**±0.5
	30	96.65±0.6	98.70±0.8	97.81±0.5	97.01±0.6	97.49±0.7	98.75±0.1	98.26±0.7	99.10±0.4	98.52±0.6	**99.15**±0.5
	35	97.14±0.7	98.81±0.8	98.15±0.3	97.42±0.6	98.11±0.6	99.12±0.4	98.68±0.6	99.36±0.4	98.80±0.6	**99.55**±0.2
Georgia	3	52.57±1.4	56.22±2.3	48.18±2.8	52.77±2.3	46.43±2.3	59.73±2.1	62.32±2.2	62.67±2.0	62.35±2.2	**63.05**±1.6
	5	61.28±1.5	66.98±1.9	59.20±1.9	62.14±1.6	56.18±1.9	71.12±1.3	73.48±1.6	74.28±1.1	73.54±1.5	**74.68**±1.2
	7	66.73±1.5	72.83±1.2	67.83±2.4	67.10±2.0	62.15±1.8	78.38±1.4	78.82±1.1	79.98±1.7	79.42±1.7	**80.30**±1.3
	9	71.40±1.0	77.53±2.0	72.57±3.0	72.13±2.3	66.37±2.9	82.57±2.1	82.77±2.2	83.30±2.1	82.80±2.2	**83.33**±2.1
Extended Yale B	10	69.80±4.5	73.85±5.6	82.32±5.1	79.92±4.3	86.39±3.1	86.56±4.5	86.79±4.8	88.27±4.5	87.10±4.4	**88.59**±4.1
	15	75.20±4.5	80.02±4.6	86.76±4.7	83.77±4.9	89.23±3.4	89.53±3.8	89.93±3.8	91.73±3.6	90.04±3.8	**91.89**±3.6
	20	80.24±2.5	85.79±2.8	90.70±2.4	88.44±2.2	92.19±1.4	93.14±2.2	93.59±2.5	95.11±1.8	93.75±2.5	**95.22**±1.8
	25	82.24±3.3	89.03±1.5	92.17±1.3	90.43±2.1	93.35±1.0	94.50±1.1	94.92±1.2	96.23±0.8	95.02±1.2	**96.33**±0.7
MNIST	1000	91.75	97.58	85.74	93.22	93.77	98.02	97.95	98.25	98.21	**98.30**

information loss. The proposed method integrates the inter-class sparsity constraint into an LDA framework which pursued the transformed samples belonging to the same classes to have the same row-sparsity structure. The proposed method offers many advantages due to its hybrid initialization capability. Our framework is generic in the sense that it allows the combination and tuning of other linear discriminant embedding methods, thus the method automatically inherits the advantages of these methods. We used the gradient descent algorithm to find the solution to our proposed criterion, rather than the closed-form solution used in ICS_DLSR and RSLDA, for example. The gradient algorithm provides faster, less complex and more accurate solutions than the closed form solutions. Moreover, the proposed linear transformation is generic and can be used for many types of objects (signals, images and texts) and many types of descriptors (including both regular and stable image features). In our work, we have used and tested different types of image descriptors. Image raw brightness, Local Binary patterns and deep features (provided by Deep Convolutional Neural Networks) were used as image descriptors for the tested datasets.

We proposed two initialization procedures for the linear transformation, resulting in two variants of the proposed algorithm. The first procedure refines the RSLDA solution using the proposed model's objective function, this variant is denoted as Feature Extraction Using Gradient Descent FE_GD. The second procedure sets the initial transformation matrix to a hybrid combination of transformation matrices obtained from two methods: Inter-class sparsity based discriminative least square regression, denoted as ICS_DLSR [188] and RSLDA [186]. The second variant is referred to as Feature Extraction Using Gradient Descent With Hybrid initialization FE_GD_HI. The suggested approach inherits the advantages of two powerful discriminant methods at two levels: (1) the hybrid transformation initialization and (2) the refinement via the proposed single new criterion.

The proposed method is also capable of obtaining a well-constructed projection space that ensures high classification performance; it can additionally be used in tuning an already obtained projection matrix. The proposed method can be generic in the sense that any hybrid initial transformation matrix can be fed into our algorithm and then a more discriminative solution for the transformation matrix is obtained, resulting in higher classification performance.

The conducted experiments proved the efficiency of the proposed method in classification tasks using multiple scaled datasets.

Tables 11 and 12 illustrates the achieved classification performance of the two proposed variants. The results presented in Table 11 are the average classification rate obtained over 10 random training splits, where in each split a random portion of the data is used for training. On the other hand, the results illustrated in Table 12 correspond to the recognition rate obtained with a single split.

Figures 26 and 27 present the classification performance behaviour according to the chosen dimension for the FE_GD and FE_GD_HI methods using the Extended

Table 11: **Mean classification performance (%) of FE_GD and FE_GD_HI methods on the tested datasets.**

Dataset\Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**FE_GD**	**FE_GD_HI**
USPS	30	87.01	88.21	84.91	83.54	72.01	88.46	89.45	89.50	**90.29**
	40	88.56	90.40	86.19	85.3	72.30	90.16	91.11	**91.81**	91.46
	55	90.51	92.09	88.64	87.16	73.32	91.25	92.65	**93.07**	92.87
	65	91.76	93.16	89.29	88.58	74.11	91.53	92.89	**93.71**	93.49
Honda	10	64.12	71.32	65.95	65.74	61.86	70.79	69.90	70.16	**72.14**
	20	77.69	83.60	79.39	79.25	75.33	82.95	83.03	83.60	**84.64**
	30	84.78	89.09	85.84	86.24	82.55	88.20	89.04	89.41	**90.12**
	50	91.36	94.15	92.28	92.34	90.03	93.53	94.13	94.53	**95.10**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	93.19	93.81	**94.58**
	6	90.35	92.93	94.18	92.15	88.73	93.65	94.25	94.75	**95.08**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.66	96.20	**96.29**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.43	**96.97**	96.40
COIL20	20	94.58	97.65	96.19	95.00	94.87	**98.04**	96.73	96.89	97.66
	25	95.79	98.22	97.07	96.12	95.99	98.22	97.74	97.89	**98.59**
	30	96.65	98.70	97.81	97.01	97.49	98.75	98.26	98.52	**99.08**
	35	97.14	98.81	98.15	97.42	98.11	99.12	98.68	98.80	**99.39**

Table 12: **Mean classification performance (%) of FE_GD and FE_GD_HI methods on the MNIST dataset.**

Dataset\Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**FE_GD**	**FE_GD_HI**
MNIST	1000	91.75	97.58	85.74	93.22	93.77	98.02	97.95	98.21	**98.33**

Yale B and Honda datasets, respectively, were 10 samples per class are used for training. By analyzing the latter figures, we can observe that our proposed approach provides a very stable performance on lower dimensions, from which we can deduce that the information loss is minimized.

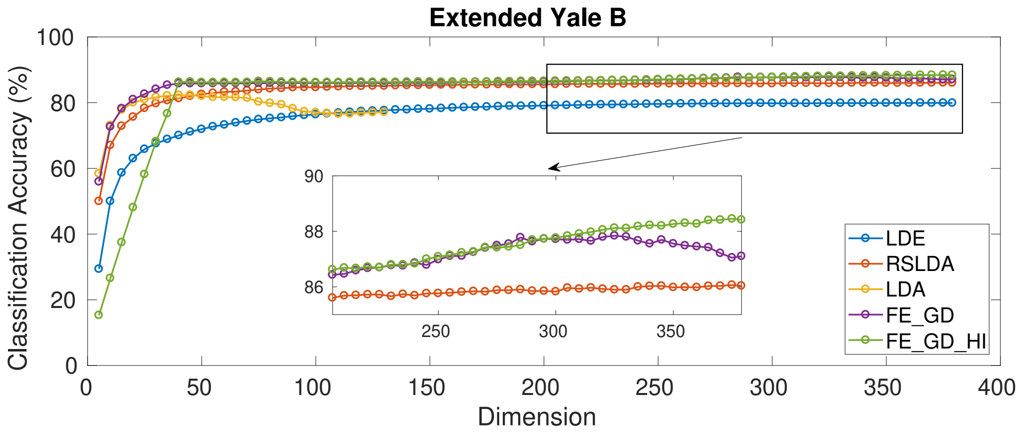


Figure. 26: **Classification performance (%) vs. dimension for the FE_GD and FE_GD_HI methods using the Extended Yale B dataset.**

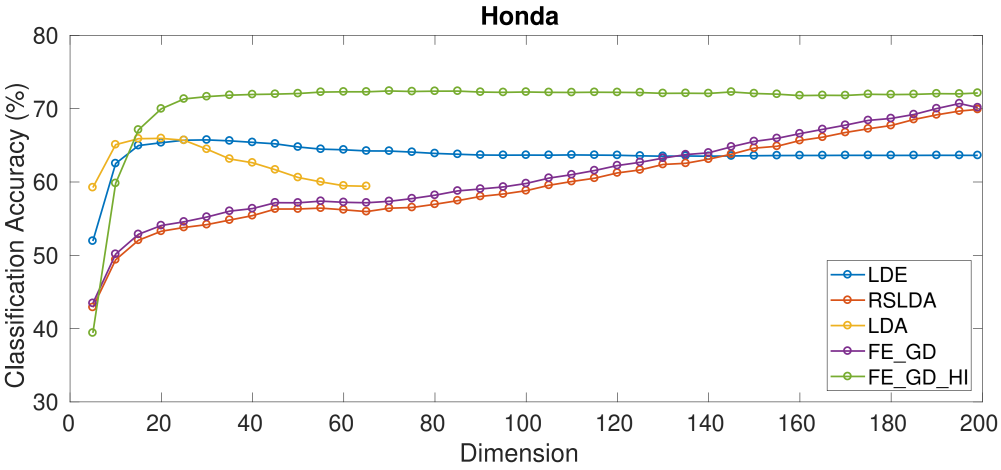
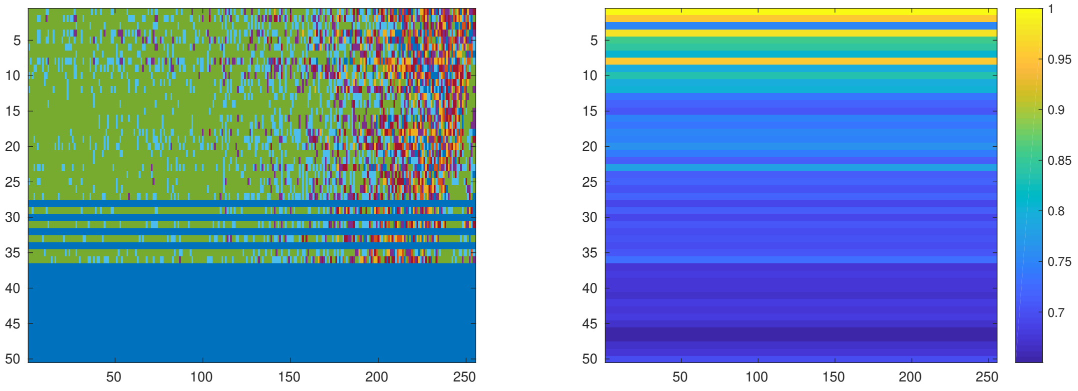


Figure. 27: **Classification performance (%) vs. dimension for the FE_GD and FE_GD_HI methods using the Honda dataset.**

Figure 28 visualizes the first 50 rows of the transformation matrix computed by the first variant of our proposed method. The plotted transformation matrix corresponds to the USPS digits dataset where 30 samples from each class were used for training.



(a) **Transformation matrix computed by FE_GD.**

(b) **Row norms of the transformation matrix derived from FE_GD.**

Figure. 28: **Visualization of the first 50 rows of the transformation matrix computed by FE_GD using the USPS dataset.**

Detailed information about this contribution can be found in the corresponding paper presented in Part II, Chapter 3.

4.4 Ensemble Learning via Feature Selection and Multiple Transformed Subsets: Application to Image Classification

Constructing the model via a limited set of features or via a single transformation can sometimes limit classification performance and lead to non-optimal results that some algorithms are capable of delivering. For this purpose, ensemble learning methods have been investigated. The main goal of these methods is to learn a set of models that provide features or predictions whose joint use could lead to better performance than that obtained by the single model.

In this contribution, we propose a new efficient ensemble learning approach that is able to enhance the classification performance of linear discriminant embedding methods. The main idea of our proposed algorithm is to blend and combine the projected data from multiple models. These models are usually very good models and each of them, considered individually, provides a good discriminant characteristic. By combining these models, we have derived a single model described by its improved discriminant ability. This leads to better classification. So the hypothesis is that in the case where the models are correctly combined, this can lead to more accurate and/or robust models. As a case study, we consider the efficient "Inter-class sparsity discriminative least square regression" [188] method in our work. Our proposed approach has succeeded in estimating an improved data representation. Instead of deploying multiple classifiers on the transformed features, we aim to estimate multiple extracted feature subsets obtained by multiple learned linear embeddings. These are associated with subsets of ranked original features. Multiple feature subsets were used to estimate the transformations. The derived extracted feature subsets were concatenated into a single data representation vector used in the classification process. Our scheme exploited multiple feature selection algorithms, namely: (i) Fisher score, (ii) Relief-F [94], in addition to (iii) Robust multi-label feature selection with dual-graph regularization" (DRMFS) algorithm [74]. We proposed three variants for our ensemble learning approach, where each variant differs from the others in the adopted feature selection techniques used for feature ranking. Multiple ranked feature subsets were used in the training process with the ICS_DLSR algorithm and their corresponding outputs were used to construct multiple models. The output of these models are concatenated to form a single data representation used in the classification process. The targeted models were constructed using different subsets of the original data. The design of the proposed approach ensures that each created model contains the most relevant features that efficiently describe the data. Relevant features are considered each time such that even if less relevant features are found, they do not harm the classification performance. The original data features were ranked using different and combined feature selection techniques.

The delivered outcomes have proven that the proposed approach may offer significant enhancement and is able to outperform competing methods. Our proposed approach was benchmarked on various datasets of different sizes and types and achieved competitive results.

Figure 29 depicts a graphical illustration of the main steps of our contribution. For the sake of simplicity, in the example of the latter figure, the case of **three models** creation was assumed. The presented figure shown demonstrates the complete process, which includes the following: Ranking of the original features of the data, construction of subgroups, model creation, concatenation and classification.

Tables 13 - 16 present some of the classification results obtained by applying our proposed ensemble scheme (three variants) over several datasets of different scale and complexity.

The paper containing detailed information about this ensemble of models based method along with the complete experiments is presented at Part II, Chapter 4.

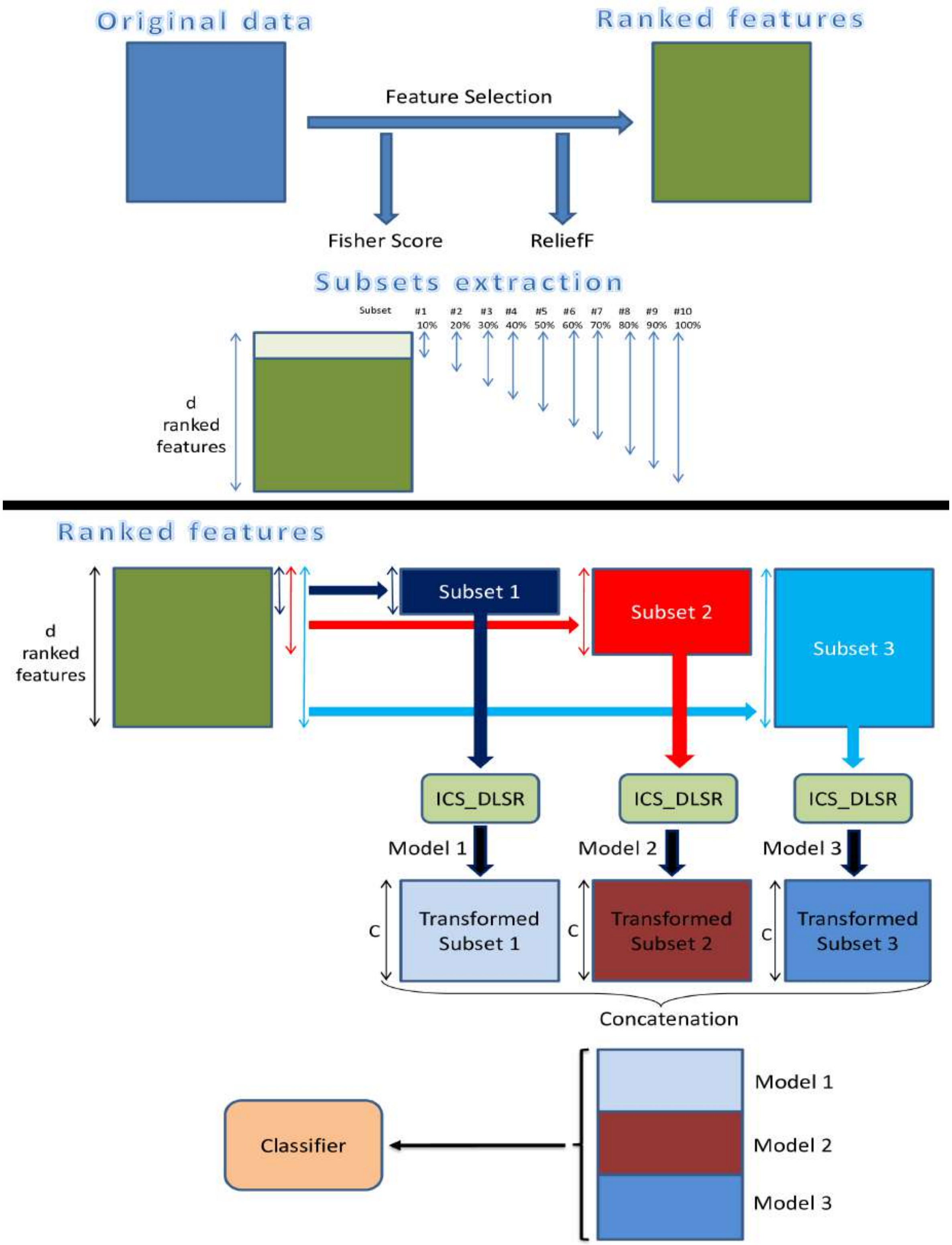


Figure. 29: **Proposed Ensemble Learning Methodology.**

Table 13: **Mean classification performance (%) of the first proposed ensemble learning approach EM_ICS_FS on the Extended Yale B dataset. The performance of some competing methods is also depicted.**

					Ext. Yale B				
Training Samples	Method	KNN	SVM	LDA	LDE	PCE	SULDA	RSLDA	RDA_GD
10		69.80	73.85	82.32	79.92	86.39	84.61	86.79	87.10
15		75.20	80.02	86.76	83.77	89.23	88.72	89.93	90.04
20		80.24	85.79	90.7	88.44	92.19	91.66	93.59	93.75
25		82.24	89.03	92.17	90.43	93.35	92.14	94.92	95.02
	Method	LRC	LRLR	LRRR	SLRR	LRPP_GRR	MPDA	ICS_DLSR	**EM_ICS_FS**
10		81.65	84.63	87.76	87.95	84.82	83.67	86.56	**88.46**
15		88.92	86.31	91.09	89.75	89.07	86.82	89.53	**91.43**
20		91.74	88.93	93.19	92.58	91.42	90.38	93.14	**94.49**
25		93.78	90.98	95.51	94.24	92.25	91.79	94.50	**95.88**

Table 14: **Mean classification performance (%) of the first proposed ensemble learning approach EM_ICS_FS on the LFW-a dataset. The performance of some competing methods is also depicted.**

Dataset\Method	Training Samples	KNN	SVM	LDA	LDE	PCE	RSLDA	RDA_GD	ICS_DLSR	**EM_ICS_FS**
LFW-a	5	9.90	12.72	20.51	9.98	9.44	24.70	25.11	22.56	**27.38**
	6	10.57	13.61	25.28	10.49	10.26	28.42	28.61	25.72	**31.75**
	7	11.06	14.70	28.62	11.24	10.98	31.50	31.82	29.04	**36.07**
	8	11.35	15.72	32.42	11.71	11.73	32.48	32.69	31.92	**39.71**

Table 15: **Mean classification performance (%) of the second proposed ensemble learning approach EM_ICS_HS on the FEI dataset. The performance of the ICS_DLSR method is also depicted.**

Dataset\Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_HS**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	92.20	**92.56**
	6	90.35	92.93	94.18	92.15	88.73	93.65	93.88	**94.20**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.14	**95.43**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.00	**96.27**

Table 16: **Comparison of the mean classification performance of our three proposed variants (Ensemble of models) using the Outdoor Scene dataset. The performance of some competing methods is also depicted.**

		Outdoor Scene		
Training Samples		**Methods**		
	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_HS**	**EM_ICS_DRMFS**
50	68.19	68.75	**68.84**	68.80
70	69.41	**70.51**	70.15	70.11
90	69.64	**70.60**	70.41	70.45
110	70.21	71.03	**71.05**	70.78

4.5 A Supervised Discriminant Data Representation: Application to Pattern Classification

This contribution is mainly an extension of the work presented in section 4.3. In this work, we proposed a discriminant feature extraction method that inherits the advantages of two recent powerful discriminant methods. The obtained transformation encapsulates two different types of discrimination, namely inter-class sparsity in addition to robust LDA. Similar to the work presented in section 4.3, we exploited the hybrid initialization process for the transformation matrix in order to obtain a more powerful discrimination. We used the gradient descent algorithm instead of the closed form approach to derive a solution for the proposed criterion. In this work, we have extended the experiments to include more datasets. The initial motivation for this extension was highlighting our proposed variants discrimination power using non-image datasets. Knowing that the original experiments only included studies on image datasets, in this contribution we conducted a study on a synthetic dataset as well as on a document dataset.

One of the added experiments was conducted on the synthetic Tetra dataset [172]. This dataset consists of 400 data points belonging to four classes. The original data points of this dataset are in $\mathbb{R}^3$, but in our experiments the dimension was augmented to 100 so that each data sample is represented by 100 features. The 3-dimensional dataset is transformed into a high-dimensional dataset $\in \mathbb{R}^{100}$ using a random projection matrix.

This dataset was chosen because it presents the challenge associated with small inter-cluster distances. The distance between clusters is minimal. Tetra's data points are visualized in Figure 30. One can see that the clusters are almost touching.

Figure 31 illustrates the T-SNE visualization of the projected samples of the Tetra dataset using the original linear discriminant analysis LDA, RSLDA as well as the first variant of our proposed method SDA_G_1. By observing this figure, it is noticeable that our method provides very good class separation properties and leads to the best intra-class compactness among the competing methods. The proposed method ensures superior performance when applied to datasets with small inter-cluster distances.

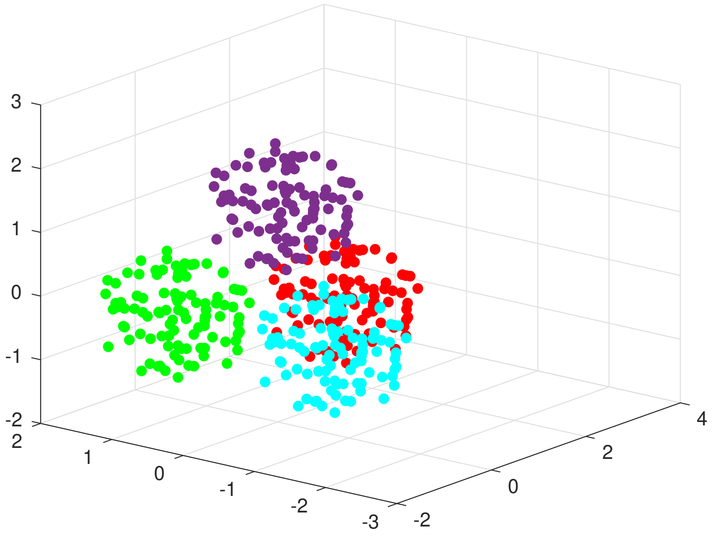
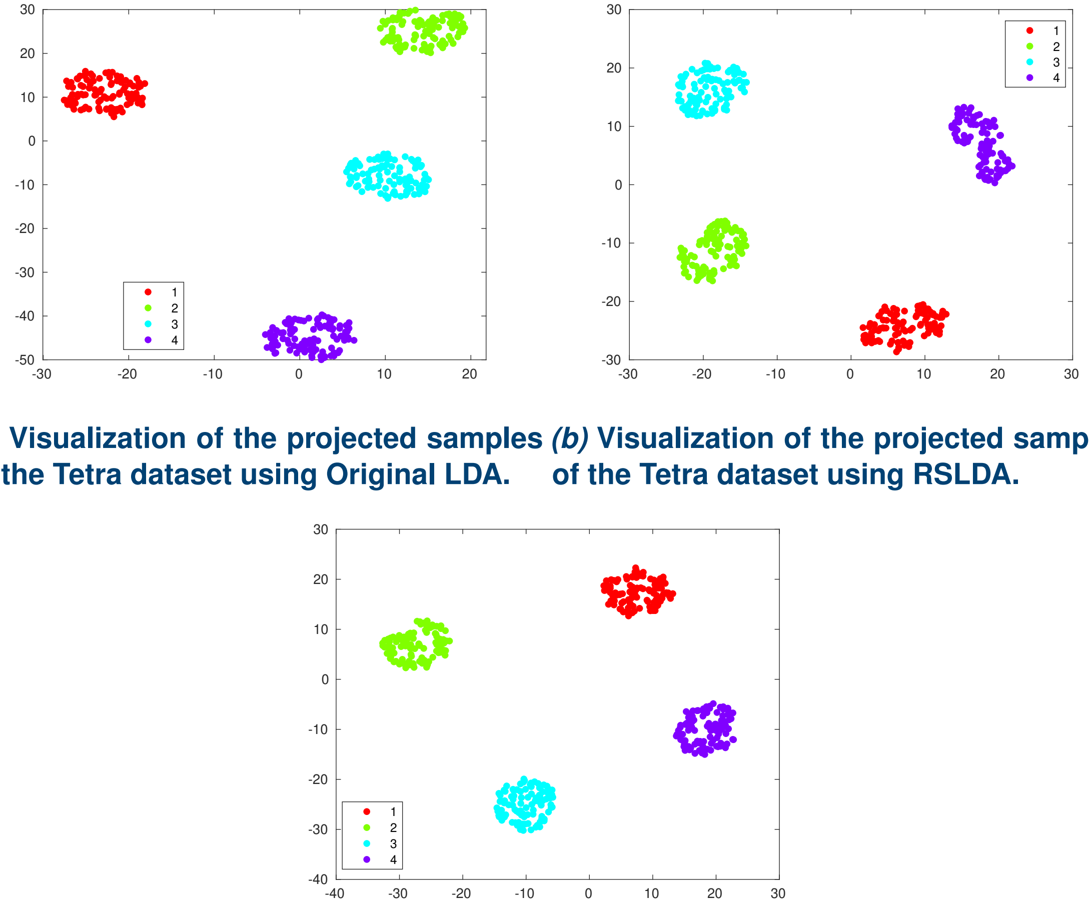


Figure. 30: **Visualization of the Tetra dataset points in the original space. These 3D points belong to four large full spheres close to each other.**

In addition, we investigated the classification performance of our proposed approach over the News20 text dataset. Our findings were obtained using 10 splits, where 20% and 30% of the data samples from each class were used for training and the remaining samples were used for testing. Table 17 depicts the average classification performance obtained using the News20 text dataset.

Table 17: **Classification performance (%) on the News20 text dataset.**

News20			
Training Percentage			
20%		30%	
Method	Classification accuracy	Method	Classification accuracy
LDA	68.04	LDA	68.70
RSLDA	68.11	RSLDA	68.88
SDA_G_1	68.38	**SDA_G_1**	69.10
SDA_G_2	**68.87**	**SDA_G_2**	**69.58**



(a) **Visualization of the projected samples of the Tetra dataset using Original LDA.**

(b) **Visualization of the projected samples of the Tetra dataset using RSLDA.**

(c) **Visualization of the projected samples of the Tetra dataset using SDA_G_1.**

Figure. 31: **TSNE visualization of the projected samples of the Tetra dataset using LDA, RSLDA, and the first proposed variant SDA_G_1.**

CHAPTER 5

Conclusions and Perspectives

Contents

5.1 Conclusions

The main objective of this thesis is to propose and develop multiple linear supervised learning methods for pattern recognition, specifically for image categorization. The proposed methods have contributed significantly in the computer vision and classification field. The proposed approaches are classified as supervised learning methods since our methods require the data labels in the learning process. Extensive experiments have been carried out to test the efficiency of the proposed algorithms on various benchmark datasets of different types and scales. Satisfactory results were obtained, all the proposed approaches were able to outperform the competing methods, whether they are recent or state-of-the-art methods. This dissertation is divided into two parts, the former presents a general introduction in addition to the background of our work, and the latter presents our contributions, each in its own chapter. In part II chapter 1, we proposed a novel discriminant supervised method that aims to learn an informative and discriminative projection space for the data. The first proposed approach is referred to as (RDA_FSIS), it involves two types of sparsity in a unifying criterion, the former comes from imposing the $\ell_{2,1}$ norm over the transformation matrix, which allows the proposed method to implicitly perform feature ranking. The latter comes from imposing the inter-class sparsity constraint over the transformed features, which allows the transformed features in each class to share a common sparse structure. This resulted in better discriminative properties and thus more efficient classification. Second, in part II chapter 2, we also proposed an enhanced version of the (RDA_FSIS) method. The proposed method differs from the original method in both the optimization process and the global criterion. We have used the gradient descent approach to derive the solution of the sought linear transformation instead of the closed form solution. The second proposed method consists of two variants called (EDA_CS) and (RDA_GD). These were used to improve the efficiency of (RSLDA) in addition to our proposed (RDA_FSIS) method. Third, we introduced the idea of hybrid initialization for the embedding space in part II chapter 3. We proposed a hybrid discriminant embedding method that ensures feature selection. The two proposed variants of this method mainly differ in the initialization process, with the second proposed variant being the most efficient. This method can inherit the discrimination power from other linear methods. It can be considered as a fine-tuning technique that can improve the embedding of existing linear methods. Fourth, in part II chapter 4, we also aimed at estimating a more discriminative data representation that increases classification efficiency, which we could achieve with an ensemble of models learning technique. In this proposed method, the embedding was computed using different subsets of the data through different scenarios where the original data features were classified using multiple and hybrid feature selection techniques. Three variants of this procedure were proposed and tested and yielded satisfactory results (EM_ICS_FS), (EM_ICS_HS) and (EM_ICS_DRMFS). The difference between the first three variants of the proposed approach is mainly that the data are ordered using different feature selection techniques.

5.2 Limitations and future work

In the course of working on this thesis, we have found that many extensions and improvements can be made at various points. This would lead to better versions of the proposed approaches. In this section, we will suggest several future avenues. Future work may envisage the following six tracks:

- All the approaches proposed in this dissertation are supervised learning methods. Therefore, these methods require the data labels to work. Since collecting the data labels is a significant challenge, it can be computationally intensive and time consuming at the same time in real applications. The proposed frameworks will be extended to the semi-supervised setting, where the training process uses both labeled and unlabeled samples.

- As a second track, it is possible to improve some of the solutions of our methods by using other mathematical approaches for the optimization process. For example, for the methods where we computed the reconstruction matrix using a closed form solution, we can propose an alternative that uses other numerical approaches (e.g., gradient descent and others).

  Moreover, since we know that gradient descent based approaches require a good initial estimate for the solution, it is possible to further increase the efficiency of our methods by using novel initialization schemes for the linear transformation matrices we are looking for.

- In our work, we have explored and studied the importance of the selected features in the learning process. Usually, better data representations lead to better classification performance, so the choice of data features to work with is very important. As future work, we will explore more about the feature selection techniques used in our ensemble learning based proposed approach. We will use more powerful and diverse feature selection techniques that will allow better data importance analysis and allow our proposed approach to derive more discriminative data representations.

- We have proposed several methods that can derive linear models for supervised learning environments. These models have excellent properties, but they are still shallow. Therefore, it would be very interesting to convert these shallow models into deep neural network models that provide better data representations. Thus, better performance can be achieved.

- It is known as a fact that image data can be represented by multiple views. Data can be collected from different sources or represented by different types of descriptors (e.g. HOG [30], Gabor [31], LBP [134], GIST [135], deep features, etc...). These descriptors can capture different aspects of the data and complement each other. Most of the work in this thesis deals with single image data, where each data pattern is represented by a single descriptor type. The only exception is the case of the PubFig83 dataset, where the descriptor used was formed by concatenating HOG, LBP and Gabor. Therefore, it would be very interesting to extend the developed methods to the case of multi-view embedding.

  In this case, an objective function would be designed for estimating either multiple individual embeddings or a consensus embedding from the data matrices. In this way, more optimal solutions can be expected.

- Another perspective is to exploit the idea that data usually consists of three properties, namely: (i) cross-class information, (ii) class-specific Information, in addition to some (iii) sparse perturbations. This is in contrast to traditional projection learning methods that work with the assumption that the discriminative data features share a common subspace. The criterion we are looking for aims to decompose the original high-dimensional data into class-common and class-specific subspaces using multiple learned projection matrices. By designing such a feature extraction algorithm, it is possible to address the problem of multi-class image classification with training data of small sample size and provide a criterion that significantly minimizes the information loss. It is expected that this idea will provide promising classification performance.

CHAPTER 6

Publications

Contents

Our research and work has led to many outcomes, we have been able to publish or submit several articles to international journals and conferences during the time of completion of our thesis. Part II will contain the most important articles, each of them as a separate chapter. This chapter will be a concise summary of our main articles.

6.1 Publication included in this thesis

Part II - Chapter 1: Linear embedding by joint Robust Discriminant Analysis and Inter-class Sparsity

- In this journal paper, we introduced a novel supervised method that aims to derive a competitive embedding space. Our method integrates two types of sparsity in a single criterion. The proposed approach is characterised by implicitly performing feature selection and extraction simultaneously. The first type of sparsity was achieved by imposing the $\ell_{2,1}$ norm over the transformation matrix in order to ensure the ability to perform feature selection. The second type of sparsity was achieved by imposing an inter-class sparsity constraint on the transformed features to ensure that samples within the same class share a common sparse structure. Our method was compared with several competing methods as well as with a number of state-of-the-art methods using different types of datasets. The results show that the proposed approach was able to outperform the competing methods.

Part II - Chapter 2: An enhanced approach to the robust discriminant analysis and class sparsity based embedding

- This journal paper presents a work aimed at improving the discriminative power of the proposed RDA_FSIS method. The developed approach differs from the latter in many ways, where both the criterion and the optimization process are different. While in the original RDA_FSIS method, the targeted transformation matrix was updated via a closed form solution, in the method proposed in this paper, we used the gradient descent method to find a solution for the linear transformation. This guarantees a better and enhanced solution than the closed-form solution used by most of competing methods. In this work, two variants of our method have been proposed, which differ mainly in that they use the output transformation matrix derived from a linear method as input to our algorithm. The use of the proposed approach was able to improve the performance of several linear projection methods and in particular the original RDA_FSIS algorithm. This was demonstrated on several datasets of different types and sizes.

Part II - Chapter 3: A hybrid discriminant embedding with feature selection: application to image categorization

- In this journal paper, we have proposed a unifying criterion that can retain the advantages of linear discriminant embedding and inter-class sparsity. The proposed framework can be considered as a fine-tuning method and a fusing method for linear discriminant embedding methods. The initial solution of the transformation matrix sought by our method was set to a hybrid combination of the solutions obtained by two embedding methods. Then, the sought transformation matrix was effectively updated via a gradient descent method. The introduction of the hybrid combination construction scheme for embedding allows our proposed approach to be generic in the sense that it can be used to improve the performance of different linear methods that have not been tested. The experiments conducted have shown that the method proposed in this article was able to outperform competing methods.

Part II - Chapter 4: Ensemble Learning via Feature Selection and Multiple Transformed Subsets: Application to Image Classification

- In this journal paper, we have proposed an ensemble learning method based on class sparsity-based regression. The proposed supervised method is used for multi-class classification tasks. The proposed approach enabled the estimation of an extended and improved data representation. Multi-ranked feature subsets were used to estimate the predictors. With the derived predictors or projections, we used the ensemble learning method to form a single data representation vector to be used in the classification process. The proposed approach was able to significantly improve the classification efficiency compared to learning with a single model, which we demonstrated in our experiments with different datasets.

Part II - Chapter 5: A Supervised Discriminant Data Representation: Application to Pattern Classification

- In this journal paper, we proposed a novel supervised approach that can derive a discriminative and efficient data representation that leads to excellent classification performance. The proposed approach computes the projection matrix via the gradient approach and simultaneously incorporates a PCA variant reconstruction matrix, which is used to preserve the energy of the original data.

The proposed approach takes advantages of the suggested initialization process in a way that it inherits the advantages of multiple linear methods at once.

6.2 Publications not included in this thesis

Feature Extraction by Joint Robust Discriminant Analysis and Inter-class Sparsity

- This conference paper is a summary of the work in our paper "Linear embedding by joint Robust Discriminant Analysis and Inter-class Sparsity". It contains part of the experiments performed in the original paper and was accepted and published at the conference "25th International Conference on Pattern Recognition" ICPR2020 Milan - Italy.

Hybrid Feature Extraction Using Robust LDA and Inter-class Sparsity for Image Categorization

- This conference paper consists of a summarized version of our work "A hybrid discriminant embedding with feature selection: application to image categorization". It contains part of the experiments conducted in the original paper and was published in the "Electronic Imaging 2021, Image Processing: Algorithms and Systems XIX." conference.

Feature Extraction and Selection via Robust Discriminant Analysis and Class Sparsity

- In this conference paper, we summarized our work "A hybrid discriminant embedding with feature selection: application to image categorization" presented in part II - chapter 3. We took part of the experiments conducted and wrote a conference paper that was published at the ICPR2021 conference in Milan - Italy.

6.3 List of publications

In this section, we will present the complete list of our publications during the thesis time. Entries presented in bold represent the publications included in this thesis report.

International Journals:

- **Dornaika, F., and A. Khoder. "Linear embedding by joint Robust Discriminant Analysis and Inter-class Sparsity." Neural Networks 127 (2020): 141-159.**
- **Khoder, A., and F. Dornaika. "An enhanced approach to the robust discriminant analysis and class sparsity based embedding." Neural Networks 136 (2021): 11-16.**
- **Khoder, A., and F. Dornaika. "A hybrid discriminant embedding with feature selection: Application to image categorization." Applied Intelligence (2020): 1-17.**
- **Khoder, A., and F. Dornaika. "Ensemble Learning via Feature Selection and Multiple Transformed Subsets: Application to Image Classification". Currently submitted to Applied Soft Computing.**
- **Khoder, A., F. Dornaika, and Moujahid, A. "A Supervised Discriminant Data Representation: Application to Pattern Classification." Revised version submitted to the International Journal of Machine Learning and Cybernetics.**

International Conferences:

- Dornaika. F., and A. Khoder. " Feature extraction by joint Robust Discriminant Analysis and Inter-class Sparsity.", IEEE International Conference on Pattern Recognition 2021.
- Khoder. A., and F. Dornaika. " Feature extraction and Selection via Robust

Discriminant Analysis and Class Sparsity.”, IEEE International Conference on Pattern Recognition 2021.

- Khoder. A., and F. Dornaika. ” Hybrid feature extraction using robust LDA and inter-class sparsity for image categorization.”, SPIE Electronic Imaging - Image Processing: Algorithms and Systems XIX, 2021.

References

[1] https://bigsnarf.wordpress.com/category/thoughts/page/4/.

[2] https://cnx.org/contents/9cMfjngH@6.3:Av9d0v4w@10/Dimensionality-Reduction-Methods-for-Molecular-Motion.

[3] https://sebastianraschka.com/Articles/2014_python_lda.html.

[4] http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/.

[5] A. M. Abdel-Zaher and A. M. Eldeib. Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46:139–144, 2016.

[6] H. Abdi. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):97–106, 2010.

[7] B. Becker and E. Ortiz. Evaluating open-universe face identification on the web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 904–911, 2013.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, volume 14, pages 585–591, 2001.

[9] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[10] D. Benkeser, C. Ju, S. Lendle, and M. van der Laan. Online cross-validation-based ensemble learning. *Statistics in Medicine*, 37(2):249–260, 2018.

[11] M. Bennasar, Y. Hicks, and R. Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[12] J. O. Berger and M. Bock. Combining independent normal mean estimation

problems with unknown variances. *The Annals of Statistics*, pages 642–648, 1976.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[15] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[16] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[17] X. Cai, C. Ding, F. Nie, and H. Huang. On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1124–1132, 2013.

[18] S. Cao, W. Lu, and Q. Xu. Deep neural networks for learning graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[19] A. Chambaz, W. Zheng, and M. Van Der Laan. Data-adaptive inference of the optimal treatment rule and its mean reward. the masked bandit. 2016.

[20] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

[21] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[22] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 846–853. IEEE, 2005.

[23] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020.

[24] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[25] V. Cherkassky and Y. Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural Networks*, 17(1):113–126, 2004.

[26] H. Choi, M. Kim, G. Lee, and W. Kim. Unsupervised learning approach for network intrusion detection system using autoencoders. *The Journal of Supercomputing*, 75(9):5597–5621, 2019.

[27] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in Neural Information Processing Systems*, 25:2843–2851, 2012.

[28] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34(8):1–17, 2007.

[29] G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613. IEEE, 2013.

[30] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[31] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[32] M. M. Davies and M. J. Van Der Laan. Optimal spatial prediction using ensemble

machine learning. *The International Journal of Biostatistics*, 12(1):179–201, 2016.

[33] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, et al. Large scale distributed deep networks. 2012.

[34] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

[35] L. Deng and J. C. Platt. Ensemble deep learning for speech recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[36] S. Dhivya, J. Sangeetha, and B. Sudhakar. Copy-move forgery detection using surf feature extraction and svm supervised learning technique. *Soft Computing*, pages 1–12, 2020.

[37] T. G. Dietterich. Machine-learning research. *AI Magazine*, 18(4):97–97, 1997.

[38] E. Dogo, O. Afolabi, N. Nwulu, B. Twala, and C. Aigbavboa. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pages 92–99. IEEE, 2018.

[39] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[40] M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. *arXiv preprint arXiv:1511.04707*, 2015.

[41] F. Dornaika and A. Bosaghzadeh. Exponential local discriminant embedding

and its application to face recognition. *IEEE Transactions on Cybernetics*, 43(3):921–934, 2013.

[42] F. Dornaika and Y. El Traboulsi. Learning flexible graph-based semi-supervised embedding. *IEEE Transactions on Cybernetics*, 46(1):206–218, 2015.

[43] F. Dornaika and Y. El Traboulsi. Joint sparse graph and flexible embedding for graph-based semi-supervised learning. *Neural Networks*, 114:91–95, 2019.

[44] F. Dornaika and A. Khoder. Linear embedding by joint robust discriminant analysis and inter-class sparsity. *Neural Networks*, 127:141–159, 2020.

[45] R. O. Duda and P. E. Hart. Dg stork pattern classification. *John Wiely and Sons*, 2001.

[46] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[47] B. Efron and C. Morris. Combining possibly related estimation problems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 35(3):379–402, 1973.

[48] A. El Mouatasim. Fast gradient descent algorithm for image classification with neural networks. *Signal, Image and Video Processing*, 14:1565–1572, 2020.

[49] X. Fang, S. Teng, Z. Lai, Z. He, S. Xie, and W. K. Wong. Robust latent subspace learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2502–2515, 2017.

[50] Q. Feng, Y. Zhou, and R. Lan. Pairwise linear regression classification for image set retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4865–4872, 2016.

[51] J. Gao, D. Shi, and X. Liu. Significant vector learning to construct sparse kernel regression models. *Neural Networks*, 20(7):791–798, 2007.

[52] M. Gashler, D. Ventura, and T. R. Martinez. Iterative non-linear dimensionality reduction with manifold sculpting. In *NIPS*, volume 8, pages 513–520, 2007.

[53] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):643–660, 2001.

[54] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang. Deformed graph laplacian for semisupervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 26(10):2261–2274, 2015.

[55] E. J. Green and W. E. Strawderman. A james-stein type estimator for combining unbiased and possibly biased estimators. *Journal of the American Statistical Association*, 86(416):1001–1006, 1991.

[56] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[57] S. R. Gunn et al. Support vector machines for classification and regression. *ISIS Technical Report*, 14(1):5–16, 1998.

[58] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

[59] N. Han, J. Wu, Y. Liang, X. Fang, W. K. Wong, and S. Teng. Low-rank and sparse embedding for dimensionality reduction. *Neural Networks*, 108:202–216, 2018.

[60] D. J. Hand. Classifier technology and the illusion of progress. *Statistical Science*, pages 1–14, 2006.

[61] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan. The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[62] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image

recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[63] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[64] L. He, H. Yang, and L. Zhao. Tensor subspace learning and classification: Tensor local discriminant embedding for hyperspectral image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[65] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *Advances in Neural Information Processing Systems*, 18:507–514, 2005.

[66] X. He, D. Cai, S. Yan, and H.-J. Zhang. Neighborhood preserving embedding. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1208–1213. IEEE, 2005.

[67] X. He and P. Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16(16):153–160, 2004.

[68] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[69] X. He, S. Yan, Y. Hu, and H.-J. Zhang. Learning a locality preserving subspace for visual recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 385–392. IEEE, 2003.

[70] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[71] S. Hijazi. *Semi-supervised Margin-based Feature Selection for Classification*. PhD thesis, Université du Littoral Côte d'Opale; Université Libanaise, école doctorale . . . , 2019.

[72] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[73] E. Hu, S. Chen, D. Zhang, and X. Yin. Semisupervised kernel matrix learning by kernel propagation. *IEEE Transactions on Neural Networks*, 21(11):1831–1841, 2010.

[74] J. Hu, Y. Li, W. Gao, and P. Zhang. Robust multi-label feature selection with dual-graph regularization. *Knowledge-Based Systems*, 203:106126, 2020.

[75] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[76] H. Huang, J. Liu, and Y. Pan. Semi-supervised marginal fisher analysis for hyperspectral image classification. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:377–382, 2012.

[77] X. Huang, H. Qiao, B. Zhang, and X. Nie. Supervised polarimetric sar image classification using tensor local discriminant embedding. *IEEE Transactions on Image Processing*, 27(6):2966–2979, 2018.

[78] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang. Superpca: A superpixelwise pca approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4581–4593, 2018.

[79] L. O. Jimenez-Rodriguez, E. Arzuaga-Cruz, and M. Vélez-Reyes. Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data. *IEEE Transactions on Geoscience and Remote Sensing*, 45(2):469–483, 2007.

[80] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural

networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[81] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[82] H. Kamper, K. Livescu, and S. Goldwater. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 719–726. IEEE, 2017.

[83] M. Kan, S. Shan, Y. Su, D. Xu, and X. Chen. Adaptive discriminant learning for face recognition. *Pattern Recognition*, 46(9):2497–2509, 2013.

[84] M. Karasuyama and H. Mamitsuka. Manifold-based similarity adaptation for label propagation. *Advances in Neural Information Processing Systems*, 26:1547–1555, 2013.

[85] M. Kaur and D. Singh. Fusion of medical images using deep belief networks. *Cluster Computing*, pages 1–15, 2019.

[86] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, 2020.

[87] A. Khatami, A. Khosravi, T. Nguyen, C. P. Lim, and S. Nahavandi. Medical image analysis using wavelet transform and deep belief networks. *Expert Systems with Applications*, 86:190–198, 2017.

[88] A. Khoder and F. Dornaika. A hybrid discriminant embedding with feature selection: application to image categorization. *Applied Intelligence*, pages 1–17, 2020.

[89] A. Khoder and F. Dornaika. An enhanced approach to the robust discriminant

analysis and class sparsity based embedding. *Neural Networks*, 136:11–16, 2021.

[90] D. H. Kim and B. C. Song. Virtual sample-based deep metric learning using discriminant analysis. *Pattern Recognition*, 110:107643, 2021.

[91] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[92] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pages 249–256. Elsevier, 1992.

[93] E. Kokiopoulou and P. Frossard. Graph-based classification of multiple observation sets. *Pattern Recognition*, 43(12):3988–3997, 2010.

[94] I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.

[95] L. Kozma. k nearest neighbors algorithm (knn). *Helsinki University of Technology*, 2008.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[97] M. J. Laan. van der, eric c. polley, and alan e. hubbard. “super learner.”. *Statistical Applications in Genetics and Molecular Biology*, 6, 2007.

[98] Z. Lai, Y. Xu, Z. Jin, and D. Zhang. Human gait recognition via sparse discriminant projection learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1651–1662, 2014.

[99] P. Langley. others,“selection of relevant features in machine learning”. *Defense Technical Information Center*, 1994.

[100] J. Lardies, H. Ma, and M. Berthillier. Source localization using a sparse representation of sensor measurements. In *Acoustics 2012*, 2012.

[101] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[102] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[103] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[104] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256. IEEE, 2010.

[105] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[106] C. Li, H. Chen, L. Zhang, N. Xu, D. Xue, Z. Hu, H. Ma, and H. Sun. Cervical histopathology image classification using multilayer hidden conditional random fields and weakly supervised learning. *IEEE Access*, 7:90378–90397, 2019.

[107] L. Li, P. W. Fieguth, and G. Kuang. Generalized local binary patterns for texture classification. In *BMVC*, volume 123, pages 1–11, 2011.

[108] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[109] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[110] X. Li, M. Chen, F. Nie, and Q. Wang. Locality adaptive discriminant analysis. In *IJCAI*, pages 2201–2207, 2017.

[111] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie. Generalized uncorrelated regres-

sion with adaptive graph for unsupervised feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5):1587–1595, 2018.

[112] Y. Li and A. Ngom. Nonnegative least-squares methods for the classification of high-dimensional biological data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2):447–456, 2013.

[113] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu. Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2138–2150, 2013.

[114] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003.

[115] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[116] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *arXiv preprint arXiv:1109.0367*, 2011.

[117] X. Liu, Z. Deng, and Y. Yang. Recent progress in semantic image segmentation. *Artificial Intelligence Review*, 52(2):1089–1106, 2019.

[118] Y. Liu, F. Nie, Q. Gao, X. Gao, J. Han, and L. Shao. Flexible unsupervised feature extraction for image classification. *Neural Networks*, 115:65–71, 2019.

[119] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng. Structured optimal graph based sparse feature extraction for semi-supervised learning. *Signal Processing*, 170:107456, 2020.

[120] H. Lu and R. Mazumder. Randomized gradient boosting machine. *SIAM Journal on Optimization*, 30(4):2780–2808, 2020.

[121] A. R. Luedtke and M. J. van der Laan. Super-learning of an optimal dynamic treatment rule. *The International Journal of Biostatistics*, 12(1):305–332, 2016.

[122] X. Ma, Y. Lin, Z. Nie, and H. Ma. Structural damage identification based on unsupervised feature-extraction via variational auto-encoder. *Measurement*, 160:107811, 2020.

[123] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang. Graph convolutional networks with eigenpooling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 723–731, 2019.

[124] F. Manessi, A. Rozza, and M. Manzo. Dynamic graph convolutional networks. *Pattern Recognition*, 97:107000, 2020.

[125] A. Martinazzo, M. Espadoto, and N. S. Hirata. Self-supervised learning for astronomical image classification. *arXiv preprint arXiv:2004.11336*, 2020.

[126] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133*, 1944.

[127] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[128] F. Movahedi, J. L. Coyle, and E. Sejdić. Deep belief networks for electroencephalography: A review of recent contributions and future outlooks. *IEEE Journal of Biomedical and Health Informatics*, 22(3):642–652, 2017.

[129] I. Naseem, R. Togneri, and M. Bennamoun. Linear regression for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2106–2112, 2010.

[130] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20). 1996.

[131] F. Nie, X. Dong, and X. Li. Unsupervised and semisupervised projection with graph optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[132] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676, 2008.

[133] S. Nurmaini, R. Umi Partan, W. Caesarendra, T. Dewi, M. Naufal Rahmatullah, A. Darmawahyuni, V. Bhayyu, and F. Firdaus. An automated ecg beat classification system using deep neural networks with an unsupervised feature extraction technique. *Applied Sciences*, 9(14):2921, 2019.

[134] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[135] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[136] D. Ou, K. Tan, Q. Du, J. Zhu, X. Wang, and Y. Chen. A novel tri-training technique for the semi-supervised classification of hyperspectral images based on regularized local discriminant embedding feature extraction. *Remote Sensing*, 11(6):654, 2019.

[137] Q.-Q. Pang and L. Zhang. Semi-supervised neighborhood discrimination index for feature selection. *Knowledge-Based Systems*, 204:106224, 2020.

[138] Y. H. Pang, J. T. A. Beng, and F. S. Abas. Regularized locality preserving discriminant embedding for face recognition. *Neurocomputing*, 77(1):156–166, 2012.

[139] X. Peng, J. Lu, Z. Yi, and R. Yan. Automatic subspace learning via principal coefficients embedding. *IEEE Transactions on Cybernetics*, 47(11):3583–3596, 2016.

[140] R. Pirracchio, M. L. Petersen, M. Carone, M. R. Rigon, S. Chevret, and M. J. van der Laan. Mortality prediction in intensive care units with the super icu

learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

[141] E. C. Polley and M. J. Van der Laan. Super learner in prediction. 2010.

[142] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019.

[143] Z. Qiao, L. Zhou, and J. Z. Huang. Sparse linear discriminant analysis with applications to high dimensional low sample size data. *International Journal of Applied Mathematics*, 39(1), 2009.

[144] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[145] J. Rao and K. Subrahmaniam. Combining independent estimators and estimation in linear regression with unequal variances. *Biometrics*, pages 971–990, 1971.

[146] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81, 2019.

[147] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1):23–69, 2003.

[148] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[149] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 65–72, 2019.

[150] D. B. Rubin and S. Weisberg. The variance of a linear combination of independent estimators using estimated weights. *Biometrika*, 62(3):708–709, 1975.

[151] D. Ruppert, S. J. Sheather, and M. P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.

[152] D. Ruppert and M. P. Wand. Multivariate locally weighted least squares regression. *The Annals of Statistics*, pages 1346–1370, 1994.

[153] M. Sahasrabudhe, Z. Shu, E. Bartrum, R. Alp Guler, D. Samaras, and I. Kokkinos. Lifting autoencoders: Unsupervised learning of a fully-disentangled 3d morphable model using deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[154] L. K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee. Spectral methods for dimensionality reduction. *Semi-supervised Learning*, 3, 2006.

[155] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[156] A. K. Seewald. Digits-a dataset for handwritten digit recognition. *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*, 2005.

[157] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

[158] C. Shi, C. Duan, Z. Gu, Q. Tian, G. An, and R. Zhao. Semi-supervised feature selection analysis with structured multi-view sparse regularization. *Neurocomputing*, 330:412–424, 2019.

[159] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[160] K. P. Sinaga and M.-S. Yang. Unsupervised k-means clustering algorithm. *IEEE Access*, 8:80716–80727, 2020.

[161] L. I. Smith. A tutorial on principal components analysis. 2002.

[162] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[163] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 823–830, 2007.

[164] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[165] F. Sultana, A. Sufian, and P. Dutta. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 122–129. IEEE, 2018.

[166] Q. Sun and S. Bourennane. Hyperspectral image classification with unsupervised feature extraction. *Remote Sensing Letters*, 11(5):475–484, 2020.

[167] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[168] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[169] B. Tang and L. Zhang. Local preserving logistic i-relief for semi-supervised feature selection. *Neurocomputing*, 399:48–64, 2020.

[170] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[171] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2):169–190, 2017.

[172] M. C. Thrun and A. Ultsch. Clustering benchmark datasets exploiting the fundamental clustering problems. *Data in Brief*, 30:105501, 2020.

[173] R. Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[174] C. Tomasi. Histograms of oriented gradients. *Computer Vision Sampler*, pages 1–6, 2012.

[175] A. Ultsch. Kohonen's self organizing feature maps for exploratory data analysis. *Proc. INNC90*, pages 305–308, 1990.

[176] A. Ultsch. Self-organizing neural networks for visualisation and classification. In *Information and Classification*, pages 307–313. Springer, 1993.

[177] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

[178] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[179] Y. Wan, X. Chen, and J. Zhang. Global and intrinsic geometric structure embedding for unsupervised feature selection. *Expert Systems with Applications*, 93:134–142, 2018.

[180] D. Wang, F. Nie, and H. Huang. Feature selection via global redundancy minimization. *IEEE Transactions on Knowledge and Data Engineering*, 27(10):2743–2755, 2015.

[181] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2007.

[182] L. Wang and C. Pan. Groupwise retargeted least-squares regression. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4):1352–1358, 2017.

[183] L. Wang, X.-Y. Zhang, and C. Pan. Msdlsr: Margin scalable discriminative least squares regression for multicategory classification. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12):2711–2717, 2015.

[184] R. Wang, F. Nie, and W. Yu. Fast spectral clustering with anchor graph for large hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2003–2007, 2017.

[185] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[186] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen, and Y. Xu. Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):390–403, 2018.

[187] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan. Low-rank preserving projection via graph regularized reconstruction. *IEEE Transactions on Cybernetics*, 49(4):1279–1291, Apr. 2019.

[188] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu. Inter-class sparsity based discriminative least square regression. *Neural Networks*, 102:36–47, 2018.

[189] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[190] B. Wu, Z. Chen, J. Wang, and H. Wu. Exponential discriminative metric embedding in deep learning. *Neurocomputing*, 290:108–120, 2018.

[191] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph

convolutional networks. In *International Conference on Machine Learning*, pages 6861–6871. PMLR, 2019.

[192] R. Wyss, S. Schneeweiss, M. van der Laan, S. D. Lendle, C. Ju, and J. M. Franklin. Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*, 29(1):96–106, 2018.

[193] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang. Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11):1738–1754, 2012.

[194] S. Xiang, Y. Zhu, X. Shen, and J. Ye. Optimal exact least squares rank minimization. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 480–488, 2012.

[195] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[196] J. Xu, B. Tang, H. He, and H. Man. Semisupervised feature selection based on relevance and redundancy criteria. *IEEE Transactions on Neural Networks and Learning Systems*, 28(9):1974–1984, 2016.

[197] Y. Xu, X. Fang, Q. Zhu, Y. Chen, J. You, and H. Liu. Modified minimum squared error algorithm for robust classification and face recognition experiments. *Neurocomputing*, 135:253–261, 2014.

[198] Y. Xu, A. Zhong, J. Yang, and D. Zhang. Lpp solution schemes for use with face recognition. *Pattern Recognition*, 43(12):4165–4176, 2010.

[199] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2006.

[200] C. Yang, Y. Feng, P. Li, Y. Shi, and J. Han. Meta-graph based hin spectral

embedding: Methods, analyses, and insights. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 657–666. IEEE, 2018.

[201] J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013.

[202] J. Yang, D. Zhang, X. Yong, and J.-y. Yang. Two-dimensional discriminant transform for face recognition. *Pattern Recognition*, 38(7):1125–1129, 2005.

[203] X. Yang, G. Liu, Q. Yu, and R. Wang. Stable and orthogonal local discriminant embedding using trace ratio criterion for dimensionality reduction. *Multimedia Tools and Applications*, 77(3):3071–3081, 2018.

[204] L. Yao, C. Mao, and Y. Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

[205] J. Ye. Least squares linear discriminant analysis. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1087–1093, 2007.

[206] J. Ye, R. Janardan, Q. Li, and H. Park. Feature reduction via generalized uncorrelated linear discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1312–1322, 2006.

[207] J. Ye and T. Xiong. Null space versus orthogonal linear discriminant analysis. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1073–1080. ACM, 2006.

[208] J. Ye and T. Xiong. Null space versus orthogonal linear discriminant analysis. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1073–1080, 2006.

[209] Y. Yi, J. Wang, W. Zhou, Y. Fang, J. Kong, and Y. Lu. Joint graph optimization and projection learning for dimensionality reduction. *Pattern Recognition*, 92:258–273, 2019.

[210] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pages 5708–5717. PMLR, 2018.

[211] X. Yuan, Y. Gu, Y. Wang, C. Yang, and W. Gui. A deep supervised learning framework for data-driven soft sensor modeling of industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4737–4746, 2019.

[212] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[213] S. Zang, Y. Cheng, X. Wang, and J. Ma. Semi-supervised flexible joint distribution adaptation. In *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, pages 19–27, 2019.

[214] H. Zhang, R. Zhang, F. Nie, and X. Li. An efficient framework for unsupervised feature selection. *Neurocomputing*, 366:194–207, 2019.

[215] J. Zhang, P. Zhang, B. Li, L. Jing, and T. Lv. Semisupervised feature extraction based on collaborative label propagation for hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 17(11):1958–1962, 2019.

[216] L. Zhang and L. Qiao. A graph optimization method for dimensionality reduction with pairwise constraints. *International Journal of Machine Learning and Cybernetics*, 8(1):275–281, 2017.

[217] L. Zhang, L. Qiao, and S. Chen. Graph-optimized locality preserving projections. *Pattern Recognition*, 43(6):1993–2002, 2010.

[218] P. Zhang, H. He, and L. Gao. A nonlinear and explicit framework of supervised manifold-feature extraction for hyperspectral image classification. *Neurocomputing*, 337:315–324, 2019.

[219] T. Zhang, D. Tao, X. Li, and J. Yang. Patch alignment for dimensionality

reduction. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1299–1313, 2008.

[220] X. Zhang, D. Chu, and R. C. Tan. Sparse uncorrelated linear discriminant analysis for undersampled problems. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7):1469–1485, 2015.

[221] X.-Y. Zhang, L. Wang, S. Xiang, and C.-L. Liu. Retargeted least squares regression algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2206–2213, 2014.

[222] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang. Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):689–700, 2015.

[223] W. Zheng, X. Zhu, Y. Zhu, R. Hu, and C. Lei. Dynamic graph learning for spectral feature selection. *Multimedia Tools and Applications*, 77(22):29739–29755, 2018.

[224] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb. Learning to diversify deep belief networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3516–3530, 2017.

[225] Y. Zhou and S. Sun. Manifold partition discriminant analysis. *IEEE Transactions on Cybernetics*, 47(4):830–840, 2016.

[226] R. Zhu, F. Dornaika, and Y. Ruichek. Learning a discriminant graph-based embedding with feature selection for image categorization. *Neural Networks*, 111:35–46, 2019.

[227] R. Zhu, F. Dornaika, and Y. Ruichek. Semi-supervised elastic manifold embedding with deep learning architecture. *Pattern Recognition*, 107:107425, 2020.

Part II : Selected Publications

CHAPTER 1

Linear Embedding by Joint Robust Discriminant Analysis and Inter-Class Sparsity

Neural Networks 127 (2020) 141–159

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Linear embedding by joint Robust Discriminant Analysis and Inter-class Sparsity

F. Dornaika [a,b,*], A. Khoder [a]

[a] University of the Basque Country UPV/EHU, San Sebastian, Spain
[b] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

ARTICLE INFO

Article history:
Received 31 August 2019
Revised and accepted 16 April 2020
Available online 22 April 2020

Keywords:
Linear discriminant analysis
Feature extraction
Feature selection
Inter-class sparsity
Image classification

ABSTRACT

Linear Discriminant Analysis (LDA) and its variants are widely used as feature extraction methods. They have been used for different classification tasks. However, these methods have some limitations that need to be overcome. The main limitation is that the projection obtained by LDA does not provide a good interpretability for the features. In this paper, we propose a novel supervised method used for multi-class classification that simultaneously performs feature selection and extraction. The targeted projection transformation focuses on the most discriminant original features, and at the same time, makes sure that the transformed features (extracted features) belonging to each class have common sparsity. Our proposed method is called Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity (RDA_FSIS). The corresponding model integrates two types of sparsity. The first type is obtained by imposing the $\ell_{2,1}$ constraint on the projection matrix in order to perform feature selection. The second type of sparsity is obtained by imposing the inter-class sparsity constraint used for ensuring a common sparsity structure in each class. An orthogonal matrix is also introduced in our model in order to guarantee that the extracted features can retain the main variance of the original data and thus improve the robustness to noise. The proposed method retrieves the LDA transformation by taking into account the two types of sparsity. Various experiments are conducted on several image datasets including faces, objects and digits. The projected features are used for multi-class classification. Obtained results show that the proposed method outperforms other competing methods by learning a more compact and discriminative transformation.

1. Introduction

In reality, most of the data are represented through a large number of features. Various types of data including high quality images, videos and many others have most of the time a large dimensionality which makes these data hard and challenging to be handled. Several applications in many fields, e.g. gaming, photography, image processing, machine learning, classification and data storage, are very demanding due to the high dimensionality of the data that needs to be handled and thus require a large amount of memory for storage as well as a lot of processing power. In general, few relevant features can represent the original data in a more efficient way than other features. Besides, original high dimensional data contain redundant features or noises which can lead to the disturbance of the learning process that exploits these data. Using these high dimensional data will generally lead to an increase in the processing complexity and time which is a problem that needs to be addressed. It is well known that the use of the original data will not guarantee the best performance in learning tasks as many researches concluded (Han et al., 2018; Xu, Tang, He, & Man, 2016). Thus, the best way to solve this problem is to select and extract the most representative features from the data. Data can be then handled via these extracted features. Many researchers focused on tackling the problem of high dimensionality by proposing two main approaches: (i) feature selection, and (ii) feature extraction. Nowadays, these two approaches are highly investigated and play an important role in learning systems (Kwak & Choi, 2002). Many methods have proved to be effective in selecting and extracting the most discriminative features to represent original data. One of these methods is the well-known Principal Component Analysis (PCA) (Smith, 2002) which is mainly used as a preprocessing technique for the data since it is able to learn a low-dimensional projection while preserving the energy of the original data. Another well-known feature extraction method is the Linear Discriminant Analysis (LDA) (Martínez & Kak, 2001). LDA aims to learn a projection that minimizes the distance among samples belonging to the same class and increases the distance

[*] Correspondence to: University of the Basque Country UPV/EHU. Manuel Lardizabal, 1, 20018 San Sebastian, Spain.
E-mail address: fdornaika@gmail.com (F. Dornaika).

https://doi.org/10.1016/j.neunet.2020.04.018
0893-6080/© 2020 Elsevier Ltd. All rights reserved.

among samples belonging to different classes. LDA is a supervised method which uses the label information of the samples to learn the linear transformation. LDA showed very good performance in classification tasks where the datasets are linearly separable. In recent times, several methods were proposed for the purpose of obtaining a linear projection. Most of these methods have shown a superior performance in real-world applications. However, these methods are not able to perform feature selection of the original data while computing the projection.

In Tao, Hou, Nie, Jiao, and Yi (2015), the authors have proposed a method capable of extracting and selecting the most discriminative features of the original data. This was done by applying an $\ell_{2,1}$ norm row-sparsity constraint on the projection matrix associated with the linear discriminant analysis.

In this paper, we introduce a new supervised method that simultaneously performs feature selection and extraction. Thus, the proposed method provides a data representation scheme in which the provided features are relevant for classification tasks.

The main contributions are as follows. First, the paper introduces a novel method for linear data projection in which the transformed features have a common structure in each class and satisfy the Linear Discriminant Analysis criterion. In addition, the proposed method is able to perform feature selection and extraction simultaneously. It also includes a simple auto-encoder model in order to get a robust linear transform. Second, the paper presents extensive and various experiments showing that the proposed linear method outperforms other competing linear methods in almost all of the tested cases using similar setups for fair comparison and using several image datasets.

The proposed method has the following features:

- It performs a sparse and robust LDA. The imposed row sparsity of the linear transformation implicitly performs a weighting of the original features. The robustness is obtained by making sure that the data can be well recovered from the low dimensional representation.
- The linear projection of the data provides a common structure for the features of each class by imposing the transformed features to have a common sparsity in each class. This constraint on the projected data can enhance the class separation.
- Since the explicit ranking of the original features is a by-product of the proposed method, it can efficiently provide feature selection of the original features without running any expensive computation.

The remainder of the paper is structured as follows. Section 2 describes some related works and presents the main notations used in this paper. Section 3 describes the problem formulation and detailed solution to the proposed method. Section 4 gives the experimental setup and presents the obtained results. Section 5 concludes the paper.

2. Preliminaries and related work

In this section, we will briefly describe some methods related to our work. Then we review the Linear Discriminant Analysis (LDA) and focus on how it can be used as a method for ranking the original features (Fan, Xu, & Zhang, 2011; Martínez & Kak, 2001). In addition, we will review the Robust and Sparse Linear Discriminant Analysis (RSLDA) (Wen et al., 2018) method and we will show how introducing the $l_{2,1}$ norm constraint can be used for feature selection (Tao et al., 2015).

Table 1
Main notations used in the paper.

Notation	Description
d	Dimensionality of original data
N	Number of data samples
C	Number of classes
n_c	Number of samples in the cth class
$\mathbf{x}_i$	The ith data sample $\in \mathbb{R}^d$
$\mathbf{X}$	Training data samples $\in \mathbb{R}^{d\times N}$
$\mathbf{Q}$	Projection matrix $\in \mathbb{R}^{d\times d}$
$\mathbf{D}$, $\mathbf{U}$	Diagonal matrix
$\mathbf{I}$	Identity matrix

2.1. Notations

This subsection will be dedicated to the introduction of some notations that will be used in our paper. Matrices are represented by bold capital letters and vectors are represented by small bold letters.

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ be the training set with N training samples from C classes, and d the dimension of each sample; each sample $\mathbf{x}_i$ is a column vector with d features $\in \mathbb{R}^d$.

The $l_{2,1}$ norm of a matrix $\mathbf{A} \in \mathbb{R}^{d\times N}$ is calculated by $\|\mathbf{A}\|_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{N} a_{ij}^2}$, the l_1 norm of a matrix $\mathbf{A}$ is calculated as follows $\|\mathbf{A}\|_1 = \sum_{j=1}^{N} \sum_{i=1}^{d} |a_{ij}|$, the l_2 norm of a vector $\mathbf{b} = [b_1, b_2, \ldots, b_d]$ is calculated as $\|\mathbf{b}\|_2 = \sqrt{\sum_{i=1}^{d} b_i^2}$.

Table 1 shows the main notations used in our paper.

2.2. Related work

In recent times, various classification methods in the machine learning field have been proposed. Many techniques and methods have been proposed and implemented. They are constantly evolving with a goal of having the best performance (prediction and classification tasks) on various datasets. Very often, data lie in a high-dimensional space, i.e. they are represented by a large number of features. Some features represent the data better than others and the data contain either redundant features or noises. The key to solving this problem is applying dimensionality reduction that can filter out some noise and redundant information by reducing the original high-dimensional space to the low-dimensional intrinsic space. Hence comes the importance of feature selection and extraction.

Feature selection aims to select and extract the most relevant features from the data to efficiently represent them prior to the classification (Stanczyk, Zieloско, & Jain, 2018; Xue, Zhang, Wang, Zhang, & Li, 2018; Yang & Ong, 2012). On the other hand, feature extraction methods are generally based on feature transformation, essentially high-dimensional data projection into low-dimensional subspace (Dornaika & El Traboulsi, 2016; Zhu, Dornaika, & Ruichek, 2019a, 2019b). This type of dimensionality reduction methods can provide a data representation on which a learning task can have a high performance. This latter can be a classification, a clustering, or a regression.

A learning method is supervised whenever label information is available for all training data. It is semi-supervised if part of the data is labeled and the remaining data do not have label information. For the unsupervised methods there is no label information at all. The remaining of the section will describe some related work about data projection (i.e., feature extraction).

One of the most known unsupervised methods is Principal Component Analysis (PCA) (Smith, 2002) which learns a projection for the data while preserving its main energy. PCA is normally used as a pre-processing technique prior to the many learning algorithms (Yang, Chu, Zhang, Xu, & Yang, 2013; Zhang, Xu, Shao, & Yang, 2017). Although PCA helps in feature extraction purposes, most of the time the extracted features are not discriminant.

In Peng, Lu, Yi, and Yan (2016), the authors proposed an unsupervised projection method called Principal Component Embedding (PCE) that can automatically determine the feature dimension in addition to being robust to non-Gaussian noise.

LDA (Tharwat, Gaber, Ibrahim, & Hassanien, 2017) remains as a favored tool for data projection and for supervised classification in many applications because of its simplicity and robustness (Hand, 2006). However, despite that LDA performs quite well in simple, low-dimensional settings, it is known to fail in some cases, e.g. when the number of predictor variables is very large compared to that of observations. In this case, LDA would not be directly applicable because the within-class matrix would be singular. LDA may also fail when the linear boundaries cannot ensure good separation between classes. Many extensions of the original LDA have been proposed to overcome its limitations and enhance its performance. The work described in Huang, Liu, Lu, and Ma (2002) deals with the small sample size (SSS) problem in LDA and proposes a method to solve it making use of the null space of within class scatter matrix. The Manifold Partition Discriminant Analysis (MPDA) method (Zhou & Sun, 2016) uses both neighbor and label information to learn the projection. This method can overcome the limitation of original LDA that failed to work with data of non-Gaussian distribution.

Local discriminant embedding (LDE) (Chen, Chang, & Liu, 2005) has also been used in classification applications as an improvement of LDA. The embedding of LDE ensures that data points belonging to the same class maintain their intrinsic neighbor relations, and neighboring points belonging to different classes no longer stick to one another. However, LDE also suffers from the small sample size (SSS) problem. To overcome the limitations of the LDE method, the work described in Dornaika and Bosaghzadeh (2013) introduced the Exponential LDE (ELDE). This method can solve the sample size problem of LDE and enhance the discrimination of the obtained projection. It is based on replacing the within-in and between class scatter matrices by their exponential ones.

In Nie, Wang, Wang, and Li (2019), the authors proposed a variant of LDA in which the intra-class KNN graph is estimated. The obtained embedding space can preserve the local neighborhood structure by constructing a k-nearest neighbors (kNNs) graph on data points. The embedding space and similarity matrix are simultaneously estimated whereas the selection of neighbors is automatically done in the projection subspace rather than in the original space. In Zhang and Gao (2018), the authors introduced a non-linear approach named supervised data-dependent kernel sparsity preserving projection (SDKSPP) for dimensionality reduction. This is a non-linear variant of the sparsity preserving projection method. This deploys a data-dependent kernel instead of standard kernels to achieve performance improvements. In Gou et al. (2018), the authors proposed a discriminative dimensionality reduction technique entitled sparsity and geometry preserving graph embedding (SGPGE). It captures the sparse reconstructive relationships among training samples and discovers the intrinsic geometry and latent discrimination in high-dimensional data. The authors show that the graphs built with discriminant and geometrical information are more informative in graph embedding. In Wen, Xu, Li, Ma and Xu (2018), the authors introduced a supervised embedding method with inter-class sparsity constraint. This method, called inter-class sparsity based discriminative least square regression (ICS_DLSR), can greatly reduce the margin of intra-class and simultaneously enlarge the margin of inter-class so that a better performance is guaranteed. The transformed samples have common sparsity structure in each class.

Almost all proposed methods for feature extraction do not have the ability to select the most discriminative and important features from the original data. Indeed, the main purpose is to get new features by recombining the original ones.

In order to take into account the relevance of the original features, a sparse constraint has been added in Sparse discriminant analysis (SDA) method (Clemmensen, Hastie, Witten, & Ersboll, 2011) which is a sparse version of LDA. SDA uses the ℓ_1 (Tibshirani, 1996) or the lasso penalty to achieve sparsity in the regression framework (Efron, Hastie, Johnstone, Tibshirani, et al., 2004; Tibshirani, 1996; Zou & Hastie, 2005; Zou, Hastie, & Tibshirani, 2006). Sparse uncorrelated LDA (SULDA) (Zhang, Chu, & Tan, 2015) and sparse LDA (SLDA) (Qiao, Zhou, & Huang, 2009) have also been proposed in order to obtain a sparse subspace for feature extraction.

For the linear projection, several methods used the $\ell_{2,1}$ norm as regularization term in order to ensure row-sparsity of the linear transform. A typical method is described in Tao et al. (2015) where the $\ell_{2,1}$ norm is applied on the transformation of original linear discriminant analysis.

Another method built on the $\ell_{2,1}$ norm constraint is the Robust Sparse Linear Discriminant analysis (RSLDA) (Wen, Fang et al., 2018). Using the $\ell_{2,1}$ norm, RSLDA adaptively performs feature weighting. It also includes a robust PCA term that ensures low dimensional information to be accurately recovered. It has been shown that the RSLDA method is robust to noisy data.

2.3. Review of linear discriminant analysis (LDA) and robust sparse LDA (RSLDA)

LDA:

Linear discriminant analysis (LDA) (Tharwat et al., 2017) is a well-known algorithm used for supervised classification tasks. It requires the label information of the training data in order to estimate the best projection subspace in which test samples can be easily classified. Let C denote the number of classes in the data and n_i denote the number of samples in the ith class. LDA aims to find a linear projection which increases the distance between samples belonging to different classes, and in contrary decreases the distance between samples belonging to the same class.

Let μ, μ_i be the mean of all data samples and the mean of samples of the ith class respectively. These means can be calculated as follows $\mu = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\mathbf{x}_j{}^i$ and $\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_j{}^i$.

First, LDA calculates the between-class scatter matrix $\mathbf{S}_b$ by the following formula:

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C} n_i\,(\mu_i - \mu)(\mu_i - \mu)^T \tag{1}$$

then the within-class scatter matrix $\mathbf{S}_w$ is calculated as follows

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\mathbf{x}_j{}^i - \mu_i)(\mathbf{x}_j{}^i - \mu_i)^T \tag{2}$$

LDA aims to estimate a projection space which maximizes the between-class variance and minimizes the within-class variance. In the case where only one projected axis is needed, the projection axis $\mathbf{p}$ can be obtained by solving the following Fisher criterion: (Duda, Hart, & Stork, 2012)

$$\mathbf{p} = \arg\max_{\mathbf{p}} \frac{\mathbf{p}^T\,\mathbf{S}_b\,\mathbf{p}}{\mathbf{p}^T\,\mathbf{S}_w\,\mathbf{p}} \tag{3}$$

The above problem (3) can be transformed to a difference form that is given by (Lai, Xu, Jin, & Zhang, 2014; Ye & Xiong, 2006):

$$\mathbf{p} = \arg\min_{\mathbf{p}^T\mathbf{p}=1} \mathbf{p}^T\,(\mathbf{S}_w - \mu\,\mathbf{S}_b)\,\mathbf{p} \tag{4}$$

where μ is a small positive constant. By solving Eq. (4), we can observe that the optimal projection vector $\mathbf{p}$ is nothing but the eigenvector associated with the smallest eigenvalue of $\mathbf{S}_w - \lambda\,\mathbf{S}_b$. Finally, for more than one projection axis, the projection matrix $\mathbf{P} \in \mathbb{R}^{d\times k}$ will consist of the k eigenvectors associated with the k smallest eigenvalues of $\mathbf{S}_w - \lambda\,\mathbf{S}_b$.

Introducing the $\ell_{2,1}$ norm constraint:

In real world applications, the dimension of data could often be very high. This makes the classification and learning tasks computationally expensive.

Hence comes the importance of feature selection (FS). Indeed, we know that the data contain a large number of features that could be either redundant or in some cases irrelevant. Eliminating these features or reducing their effect can lead to some improvements and advantages like fast processing and a better classification accuracy. More importantly, FS can help in the alleviation of the curse of dimensionality in the data. Let $\mathbf{Q} \in \mathbb{R}^{d\times d}$ denote a projection matrix that operates on data samples of dimension d. The projection of a sample $\mathbf{x}$ is given by $\mathbf{Q}^T\mathbf{x}$. The $\ell_{2,1}$ norm of $\mathbf{Q}$ is given by

$$\|\mathbf{Q}\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{d} q_{ij}^2} \tag{5}$$

This norm is equal to the sum of ℓ_2 norms of all rows of the matrix.

A good feature selection/weighting can be obtained by minimizing the $\ell_{2,1}$ norm of the projection matrix as it was described in Xiang, Nie, Meng, Pan, and Zhang (2012). In this work, the authors utilized this constraint in their framework as an FS tool for classification. Whenever the rows of the matrix $\mathbf{Q}$ are equal to zero (or their ℓ_2 norms are very small), the features corresponding to these rows are irrelevant and could be removed.

Robust Sparse LDA (RSLDA):

The work described in Wen, Fang et al. (2018) introduced an LDA-based method for feature extraction. The proposed method is entitled Robust Sparse Linear Discriminant Analysis (RSLDA). It deploys the minimization of the $\ell_{2,1}$ norm of the linear transform. It also incorporates the ability to recover the original data from the low dimensional projected data.

Aiming to overcome some drawbacks of the LDA (Tharwat et al., 2017) technique, and to extract the features while holding the main energy of the data and enhancing the robustness to noise, RSLDA solves the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{P},\mathbf{Q},\mathbf{E}} Tr\ \left(\mathbf{Q}^T\left(\mathbf{S}_w - \mu\,\mathbf{S}_b\right)\mathbf{Q}\right) + \lambda_1\,\|\mathbf{Q}\|_{2,1} + \lambda_2\,\|\mathbf{E}\|_1 \\ &\quad s.t.\quad \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} + \mathbf{E},\ \mathbf{P}^T\,\mathbf{P} = \mathbf{I} \end{aligned} \tag{6}$$

where $\mathbf{Q} \in \mathbb{R}^{d\times m}$ is the projection matrix in which $(m < d)$, λ_1 and λ_2 are trade-off parameters used to determine the importance of the different terms. $\mathbf{S}_w$ and $\mathbf{S}_b$ are the within-class and between-class scatter matrices respectively. $\mathbf{E}$ is the error matrix and μ is a constant used to balance the two scatter matrices.

The $\ell_{2,1}$ norm of the transformation matrix $\mathbf{Q}$ used in the optimization problem (6) can be calculated using Eq. (5). According to Wen, Fang et al. (2018), RSLDA learns a discriminative subspace and has reduced information loss than other LDA-based algorithms. Besides, RSLDA addresses the issue of model sensibility to reduced dimensions, and can thus provide a very good performance even in cases where the projected space has very few dimensions. More information on Robust Sparse Linear Discriminant Analysis can be found in Wen, Fang et al. (2018).

3. Proposed method

In this section, we will present the motivation of our method. Then we will introduce the proposed learning model and the approach used for finding the solution to the proposed learning method.

In Section 2.3, we have briefly described how the Linear Discriminant Analysis (LDA) algorithm works. However, LDA has many limitations that need to be overcome. First, LDA (Tharwat et al., 2017) can be very sensitive to the selection of the reduced dimensions, which affects the classification rate when very few dimensions are used. This is mainly due to the fact that the associated projection matrix is obtained by solving an Eigen decomposition problem that uses global scatter matrices. In addition, LDA lacks the ability of selecting and ranking the most discriminative features from the original data. This can be seen from the estimated projection matrix that does not allow to have a good interpretability for feature relevance.

We will introduce an approach that aims to fix these drawbacks. The proposed method inherits the advantages of the methods stated in the related works section. Indeed, the proposed method aims at learning a better transformation matrix that leads to better classification performance via introducing two types of sparsity. The first type is imposed via the minimization of the $\ell_{2,1}$ norm of the projection matrix. This explicitly provides a ranking for the original data features. The second type is given by the inter-class sparsity of the projected data in which each class is forced to have common sparsity structure in the projected space. Furthermore, our introduced criterion includes a robust LDA in order to be robust in presence of noisy observation.

3.1. Problem formulation and learning model

Motivated by overcoming some of the LDA drawbacks, and inspired by the RSLDA model, we propose a novel method that can lead to a more discriminant transformation. Unlike the RSLDA model that imposes the row sparsity of the transformation matrix, our model will integrate the inter-class sparsity too.

Our proposed learning model is to learn two matrices by minimizing the following functional:

$$\begin{aligned} f(\mathbf{Q},\mathbf{P}) = Tr\ \left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1\ \|\mathbf{Q}\|_{2,1} + \lambda_2\ \|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\|_1 \\ + \lambda_3 \sum_{i=1}^{C} \|\mathbf{Q}^T\,\mathbf{X}_i\|_{2,1} \end{aligned} \tag{7}$$

where the unknown matrices are: the sought projection matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$, and the PCA orthogonal matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$. λ_1, λ_2, and λ_3 are the three trade-off parameters that determine the importance of their corresponding terms. $\mathbf{S} = \mathbf{S}_w - \mu\,\mathbf{S}_b$ is the LDA scatter matrix, with $\mathbf{S}_w$ and $\mathbf{S}_b$ being the within-class and between-class scatters matrices, respectively. $\mathbf{S}_w$ and $\mathbf{S}_b$ can be calculated using Eqs. (1) and (2) respectively, while μ is a constant used to balance the two scatter matrices. $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$ is the data matrix associated with the ith class.

The proposed learning model estimated by the minimization of the objective (7) has the following theoretical justification. Minimizing the first term tends to provide a projection matrix associated with the classic Linear Discriminant Analysis. Minimizing the second term (sum of the ℓ_2 norms of the rows of the projection matrix) reduces the over-fitting and produces an implicit ranking of the original features. The third term is a variant of the PCA constraint (Fang et al., 2017) $\mathbf{X} = \mathbf{P}\ \mathbf{Q}^T\,\mathbf{X}$. It has been introduced in our model using the ℓ_1 norm of the error matrix in order to retain the energy preserving property of PCA (Smith, 2002). It guarantees that the original data will be well recovered (Zou et al., 2006). Real data are normally corrupted by many kinds of noises. Therefore, the use of the ℓ_1 norm of the error matrix, $\|\mathbf{E}\|_1$, can compensate the random and sparse noise. Thus, the third term can be seen as a simple auto-encoder model in which the encoder is given by the matrix $\mathbf{Q}^T$ and the decoder is given by the matrix $\mathbf{P}$. The fourth term is a sum of row sparsity over the projection of each class. By minimizing this term, it is expected that each class in the projection space will have the same common sparse structure (see Fig. 1).

By introducing the variables $\mathbf{F} = \mathbf{Q}^T\,\mathbf{X}$, $\mathbf{E} = \mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}$, and $\mathbf{F}_i = \mathbf{Q}^T\,\mathbf{X}_i$ $(i = 1, \ldots, C)$, problem (7) can be written as:

$$f(\mathbf{Q}, \mathbf{E}, \mathbf{P}, \mathbf{F}) = Tr\,\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1\,\|\mathbf{Q}\|_{2,1} + \lambda_2\,\|\mathbf{E}\|_1 + \lambda_3 \sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1} \tag{8}$$

$$\min_{\mathbf{Q},\mathbf{E},\mathbf{P},\mathbf{F}} f\,(\mathbf{Q}, \mathbf{E}, \mathbf{P}, \mathbf{F}) \quad s.t.\ \mathbf{F} = \mathbf{Q}^T\,\mathbf{X}, \quad \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} + \mathbf{E}, \text{ and } \mathbf{P}^T\,\mathbf{P} = \mathbf{I}$$

Minimizing $\sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1}$ aims to ensure the common sparsity of the transformed features of samples belonging to the same classes. By joining these constraints with the $\ell_{2,1}$ norm constraint on the transformation matrix $\mathbf{Q}$, it is expected that the transformation obtained by solving problem (8) will simultaneously select the most important features, provide a robust discrimination, and generate an inter-class sparsity.

Fig. 1 illustrates the principle of the proposed model in which original features and inter-class sparsity are exploited. Yellow dots, red triangles and blue squares represent samples from the first, second and Cth class, respectively. The left part of the figure illustrates the input data (as a cloud of points and as a data matrix). The right part illustrates the expected projection of the cloud and of the data matrix.

It is well known that the $\ell_{2,1}$ norm of the matrix $\mathbf{Q}$ can be written in the form of a trace. Thus, we have:

$$\|\mathbf{Q}\|_{2,1} = Tr\,\left(\mathbf{Q}^T\,\mathbf{U}\,\mathbf{Q}\right) \tag{9}$$

where $\mathbf{U}$ is a diagonal matrix given by:

$$\mathbf{U} = \begin{pmatrix} \frac{1}{\|\mathbf{q}_1\|_2} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\|\mathbf{q}_d\|_2} \end{pmatrix} \tag{10}$$

$\mathbf{q}_i$ denotes the ith row vector of the matrix $\mathbf{Q}$.

3.2. Optimization of proposed method

Our optimization problem (8) does not have an analytical solution. Thus, we adopted an iterative scheme to obtain the solution, in which matrices are first, initialized then updated in an alternating process by fixing some unknowns and computing others.

Using the alternating direction method of multipliers (ADMM) (Boyd, Parikh, Chu, Peleato, Eckstein, et al., 2011), we have solved our optimization problem as follows. We first reformulate our problem (8) into the following augmented Lagrangian function (Courcoubetis & Weber, 2003):

$$\begin{aligned}\mathcal{L}\,(\mathbf{Q}, \mathbf{P}, \mathbf{E}, \mathbf{F}, \mathbf{Y}_1, \mathbf{Y}_2) &= Tr\,\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 Tr\,\left(\mathbf{Q}^T\,\mathbf{U}\,\mathbf{Q}\right) + \lambda_2\,\|\mathbf{E}\|_1 \\ &+ \lambda_3 \sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1} + \frac{\beta}{2}\left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}_1}{\beta}\right\|_2^2 \\ &+ \frac{\beta'}{2}\left\|\mathbf{F} - \mathbf{Q}^T\,\mathbf{X} + \frac{\mathbf{Y}_2}{\beta'}\right\|_2^2\end{aligned} \tag{11}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are two Lagrange matrices, β and β' are two small positive numbers associated with the ADMM. If we fix all variables except one, we can alternately solve each variable by minimizing the Lagrangian at a time. We proceed as follows:

- **Update Q**:
  $\mathbf{Q}$ can be obtained by fixing the variables $\mathbf{P}$, $\mathbf{E}$ and $\mathbf{F}$, and minimizing the resulting problem:

$$\begin{aligned}\mathcal{L}(\mathbf{Q}) &= Tr\,\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 Tr\,\left(\mathbf{Q}^T\,\mathbf{U}\,\mathbf{Q}\right) \\ &+ \frac{\beta}{2}\left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}_1}{\beta}\right\|_2^2 \\ &+ \frac{\beta'}{2}\left\|\mathbf{F} - \mathbf{Q}^T\,\mathbf{X} + \frac{\mathbf{Y}_2}{\beta'}\right\|_2^2\end{aligned} \tag{12}$$

  The optimal $\mathbf{Q}$ can be obtained by vanishing the derivative of the Lagrangian with respect to $\mathbf{Q}$. Since the matrix $\mathbf{U}$ is given by Eq. (10), the derivative of $Tr\,\left(\mathbf{Q}^T\mathbf{U}\mathbf{Q}\right) = \|\mathbf{Q}\|_{2,1}$ is $\mathbf{U}\,\mathbf{Q}$. From $\frac{\partial\mathcal{L}(\mathbf{Q})}{\partial\mathbf{Q}} = \mathbf{0}$, we can obtain:

$$\mathbf{Q} = \left(2\,\mathbf{S} + \lambda_1\,\mathbf{U} + \beta\,\mathbf{X}\,\mathbf{X}^T + \beta'\,\mathbf{X}\,\mathbf{X}^T\right)^{-1}\left(\beta\,\mathbf{X}\,\mathbf{M}^T\,\mathbf{P} + \beta'\,\mathbf{X}\,\mathbf{M}'^T\right) \tag{13}$$

  where $\mathbf{M} = \mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}_1}{\beta}$, $\quad \mathbf{M}' = \mathbf{F} + \frac{\mathbf{Y}_2}{\beta'}$. It is worthy noting that the matrix $\mathbf{U}$ depends on the elements of the matrix $\mathbf{Q}$. In order to get a tractable solution for $\mathbf{Q}$, the diagonal matrix $\mathbf{U}$ is supposed to have an initial guess that is fixed at the first iteration. This trick is also used in many iterative algorithms dealing with the minimization of the $\ell_{2,1}$ norm of a matrix. Once $\mathbf{Q}$ is updated, we update the associated diagonal matrix $\mathbf{U}$ using Eq. (10).
- **Update P:**
  The orthogonal matrix $\mathbf{P}$ can be obtained by fixing the variables $\mathbf{Q}$, $\mathbf{E}$ and $\mathbf{F}$ and minimizing the resulting problem:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}}{\beta}\right\|_2^2 \tag{14}$$

  Suppose $\mathbf{M} = \mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}_1}{\beta}$, problem (14) becomes:

$$\begin{aligned}&\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{M} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \\ &= \min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} Tr\,(\mathbf{M}^T\,\mathbf{M} - 2\mathbf{M}^T\,\mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}) \\ &= \max_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} Tr\,(\mathbf{P}^T\,\mathbf{M}\,\mathbf{X}^T\,\mathbf{Q})\end{aligned} \tag{15}$$

  Problem (15) can be solved by performing a singular value decomposition of the matrix $\mathbf{M}\,\mathbf{X}^T\,\mathbf{Q}$. Let the SVD decomposition be given by $SVD\,(\mathbf{M}\,\mathbf{X}^T\,\mathbf{Q}) = \mathbf{B}\,\Sigma\,\mathbf{V}^T$. We can have a solution of $\mathbf{P}$ by (Zou et al., 2006):

$$\mathbf{P} = \mathbf{B}\,\mathbf{V}^T \tag{16}$$

- **Update E**:
  $\mathbf{E}$ can be obtained by fixing $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{F}$ and minimizing:

$$\min_{\mathbf{E}} \lambda_2\,\|\mathbf{E}\|_1 + \frac{\beta}{2}\left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} - \mathbf{E} + \frac{\mathbf{Y}_1}{\beta}\right\|_2^2 \tag{17}$$

  We can have the solution to Problem (17) as follows:

$$\mathbf{E} = shrink_e(\mathbf{E}_0) \tag{18}$$

  where $e = \frac{\lambda_2}{\beta}$, $\mathbf{E}_0 = \mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} + \frac{\mathbf{Y}_1}{\beta}$, and $shrink_e(.)$ is the element-wise shrinkage operator with parameter e (Candès, Li, Ma, & Wright, 2011). This is given by $shrink_e(x) = sign(x)\,max(|x| - e, 0)$.

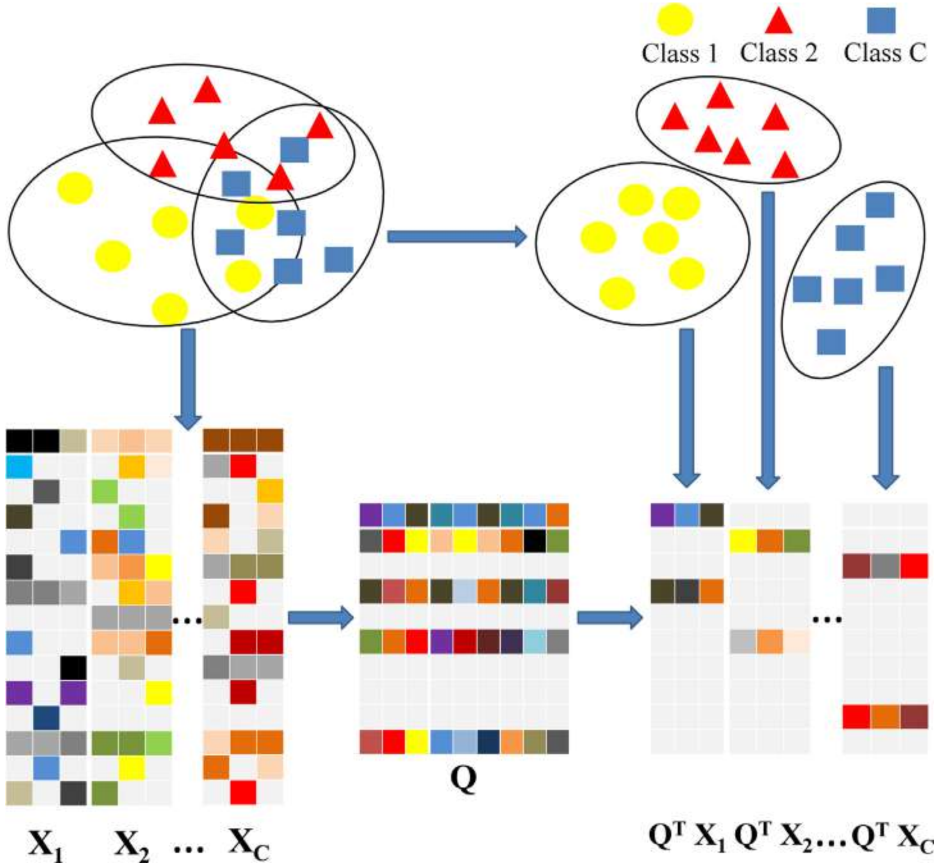

Class 1
Class 2
Class C
Q
$\mathbf{X_1}$ $\mathbf{X_2}$ ... $\mathbf{X_C}$
$\mathbf{Q^T X_1}$ $\mathbf{Q^T X_2}$...$\mathbf{Q^T X_C}$

Fig. 1. Illustration of the proposed method. [$\mathbf{X}_1$, $\mathbf{X}_2$, ..., $\mathbf{X}_C$] denote the samples from the first class to the Cth class. [$\mathbf{Q}^T\,\mathbf{X}_1$, $\mathbf{Q}^T\,\mathbf{X}_2$, ..., $\mathbf{Q}^T\,\mathbf{X}_C$] are the projected samples. $\mathbf{Q}$ is the sought transformation matrix. Yellow dots, red triangles and blue squares represent samples from the first, second and Cth class, respectively. The left part of the figure illustrates the input data (as a cloud of points and as a data matrix. The right part illustrates the expected projection of the cloud and of the data matrix.

- **Update F**:

  F can be obtained by fixing variables **Q**, **P**, **E** and minimizing:

$$\min_{\mathbf{F}} \lambda_3 \sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2} \left\| \mathbf{F} - \mathbf{Q}^T\,\mathbf{X} + \frac{\mathbf{Y}_2}{\beta'} \right\|_2^2 \tag{19}$$

Let

$$\mathbf{H} = \mathbf{Q}^T\,\mathbf{X} - \frac{\mathbf{Y}_2}{\beta'} \tag{20}$$

Here, we will refer to the fact that minimizing $\sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1}$ is the same as minimizing $\|\mathbf{F}_i\|_{2,1}$, separately with $(i = 1, \ldots, C)$ and C is the number of classes. **F** is the horizontal concatenation of the $\mathbf{F}_i$ matrices. We have $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_C]$. Similarly, $\mathbf{Q}^T\,\mathbf{X}$ and **H** are the horizontal concatenation of C matrices, i.e., $\mathbf{Q}^T\mathbf{X} = [\mathbf{Q}^T\,\mathbf{X}_1, \mathbf{Q}^T\,\mathbf{X}_2, \ldots, \mathbf{Q}^T\,\mathbf{X}_C]$, and $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_C]$.

By plugging Eq. (20) into Eq. (19), the latter becomes:

$$\begin{aligned} &\min_{\mathbf{F}} \lambda_3 \sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2} \|\mathbf{F} - \mathbf{H}\|_2^2 \\ &= \min_{\mathbf{F}} \sum_{i=1}^{C} \left(\lambda_3\, \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2}\, \|\mathbf{F}_i - \mathbf{H}_i\|_2^2 \right) \end{aligned} \tag{21}$$

Referring to the above stated fact which states that solving the summation of **F** is the same as solving for each subset $\mathbf{F}_i$ separately we obtain the following:

$$\begin{aligned} &\min_{\mathbf{F}} \sum_{i=1}^{C} \left(\lambda_3 \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2}\, \|\mathbf{F}_i - \mathbf{H}_i\|_2^2 \right) \\ &\Leftrightarrow \sum_{i=1}^{C} \min_{\mathbf{F}_i} \left(\lambda_3\, \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2}\, \|\mathbf{F}_i - \mathbf{H}_i\|_2^2 \right) \end{aligned} \tag{22}$$

Thus, every matrix $\mathbf{F}_i$ is solved by:

$$\min_{\mathbf{F}_i} \left(\lambda_3\, \|\mathbf{F}_i\|_{2,1} + \frac{\beta'}{2}\, \|\mathbf{F}_i - \mathbf{H}_i\|_2^2 \right) \tag{23}$$

According to Liu, Ji, and Ye (2009), $\mathbf{F}_i$ is given by:

$$[\mathbf{F}_i]_j = \begin{cases} \dfrac{\left\|[\mathbf{H}_i]_j\right\|_2 - \lambda_3/\beta'}{\left\|[\mathbf{H}_i]_j\right\|_2} \cdot [\mathbf{H}_i]_j & \text{if } \left\|[\mathbf{H}_i]_j\right\|_2 > \lambda_3/\beta' \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{24}$$

where $[\mathbf{H}_i]_j$ and $[\mathbf{F}_i]_j$ are the jth row vectors of $\mathbf{H}_i$ and $\mathbf{F}_i$, respectively.

- **Update $\mathbf{Y}_1$, $\mathbf{Y}_2$, β and β'**:

  The Lagrange multipliers $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and the penalty terms β and β' are updated as follows:

$$\mathbf{Y}_1 = \mathbf{Y}_1 + \beta\,(\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X} - \mathbf{E}) \tag{25}$$

$$\mathbf{Y}_2 = \mathbf{Y}_2 + \beta'\,(\mathbf{F} - \mathbf{Q}^T\,\mathbf{X}) \tag{26}$$

$$\beta = \min(\rho\,\beta, \beta_{max}) \tag{27}$$

$$\beta' = \min(\rho\, \beta', \beta'_{max}) \tag{28}$$

Note that β_{max} and β'_{max} are constant.

Algorithm 1 summarizes our proposed method and describes the main steps for solving the problem (8).

Algorithm. 1. Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity (RDA_FSIS)

```
Input:           Data samples $\mathbf{X} \in \mathbb{R}^{d \times N}$, Parameters $\lambda_1, \lambda_2, \lambda_3$
Output:          $\mathbf{P}, \mathbf{Q}$ and $\mathbf{E}$
Initialization:  $\mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{0}$
                 $\mathbf{Q} = \mathbf{0}$ or Random matrix.
                 $\mathbf{F} = \mathbf{Q}^T \mathbf{X}$
                 $\mathbf{E} = \mathbf{0}$
                 $\beta$ and $\beta' = 10^{-8}$
                 $\mu = 10^{-4}$
                 $\beta_{max}$ and $\beta'_{max} = 10^{8}$
                 $\rho = 10$
                 $\mathbf{P} = \arg\min_{\mathbf{P}} Tr(\mathbf{P}^T \mathbf{S} \mathbf{P})$
Process:         ADMM (alternating direction method of multipliers):
                 Repeat
                     Fix all, Update $\mathbf{Q}$ using (13), and update $\mathbf{U}$ using (10).
                     Fix all, Update $\mathbf{P}$ using (16).
                     Fix all, Update $\mathbf{E}$ using (18).
                     Fix all, Update $\mathbf{F}$ using (24).
                     Update $\mathbf{Y}_1, \mathbf{Y}_2, \beta$ and $\beta'$ using (25), (26), (27), and (28), respectively.
                 Until convergence
```

The projection of the training and test samples is carried out using the estimated projection matrix **Q**. This is given by $\mathbf{Z} = \mathbf{Q}^T \mathbf{X}$ and $\mathbf{z} = \mathbf{Q}^T \mathbf{t}$ where **X** is the training data and **t** is a test sample.

3.3. Computational complexity

In this section, we analyze the computational complexity of the proposed algorithm (see Algorithm 1). This algorithm consists of five main steps that we have described above for calculating the unknown matrices **Q**, **P**, **E**, **F**, and finally updating the multipliers $\mathbf{Y}_1$ and $\mathbf{Y}_2$. Regarding the steps of the algorithm, the last step has clearly the least computational cost since it consists of simple matrix additions and multiplications. Other steps also have no obvious effect on the computational cost of the algorithm like steps three and four which came from Eqs. (18) and (24). These steps only consist of simple matrix operations and thus their computational cost can be ignored. The main computational complexity of the proposed algorithm takes place in the first two steps. The first step requires a matrix inversion (or equivalently solving a linear system whose square matrix size is $d \times d$) with a complexity of $\mathcal{O}(d^3)$ for a $d \times d$ matrix. The second step is the singular value decomposition of a $d \times d$ matrix $(\mathbf{M}\,\mathbf{X}^T\mathbf{Q})$. Thus, the computational cost of the second step is $\mathcal{O}(d^3)$. Let τ represent the number of iterations of the proposed algorithm. The overall computational complexity of the proposed method will be $\mathcal{O}(\tau\,(d^3 + d^3)) = \mathcal{O}(\tau\,(2\,d^3))$. The computational complexities of different learning methods are presented in Table 2. The performance of these methods will be quantified in the next section. In the conducted evaluation, we will show that the best learning methods in terms of classification accuracy are ICS_DLSR, RSLDA and the proposed RDA_FSIS. From Table 2, we can see clearly that the computational complexity of the proposed method is comparable to that of ICS_DLSR and RSLDA methods.

Table 2
Complexity of different learning methods.

Method	Complexity for training
NN	–
SVM	$\mathcal{O}(N^2 d + N^3)$
LDA	$\mathcal{O}(d^3)$
LDE	$\mathcal{O}(N^2 + d^3)$
PCE	$\mathcal{O}(N\,d + N^2\,d)$
ICS_DLSR	$\mathcal{O}(\tau\,d^3)$
RSLDA	$\mathcal{O}(\tau\,(d^2 N + 4d^3))$
RDA_FSIS	$\mathcal{O}(\tau\,(2\,d^3))$

3.4. Convergence analysis

Since the overall model in (8) is nonconvex, it is difficult to guarantee its convergence to a local minimum. However, empirical evidence suggests that the proposed algorithm has a good convergence behavior (see Figs. 15 and 16). Appendix presents a proof of weak convergence of the proposed algorithm showing that under mild conditions, any limit point of the iteration sequence generated by Algorithm 1 is a stationary point that satisfies the Karush–Kuhn–Tucker (KKT) conditions.

4. Performance study

In this section, we will present the experimental results obtained by the proposed method. The experiments are conducted on different face and object datasets in addition to handwritten digits databases.

In a given conducted experiment, the number of training images per class is fixed, making sure that balanced classes are considered during training. Training images are randomly selected from the datasets and used for training while the rest are used for testing. For a better comparison, we adopted several amounts of training in order to study the performance of the methods when supervision information (the number of training images) is increased. The number of training samples per class can vary between 1 and $(N_c - 1)$ where N_c represents the number of images per each class. Inspired by many published works, we adopted several (training percentages) numbers of training samples per class in our experiments.

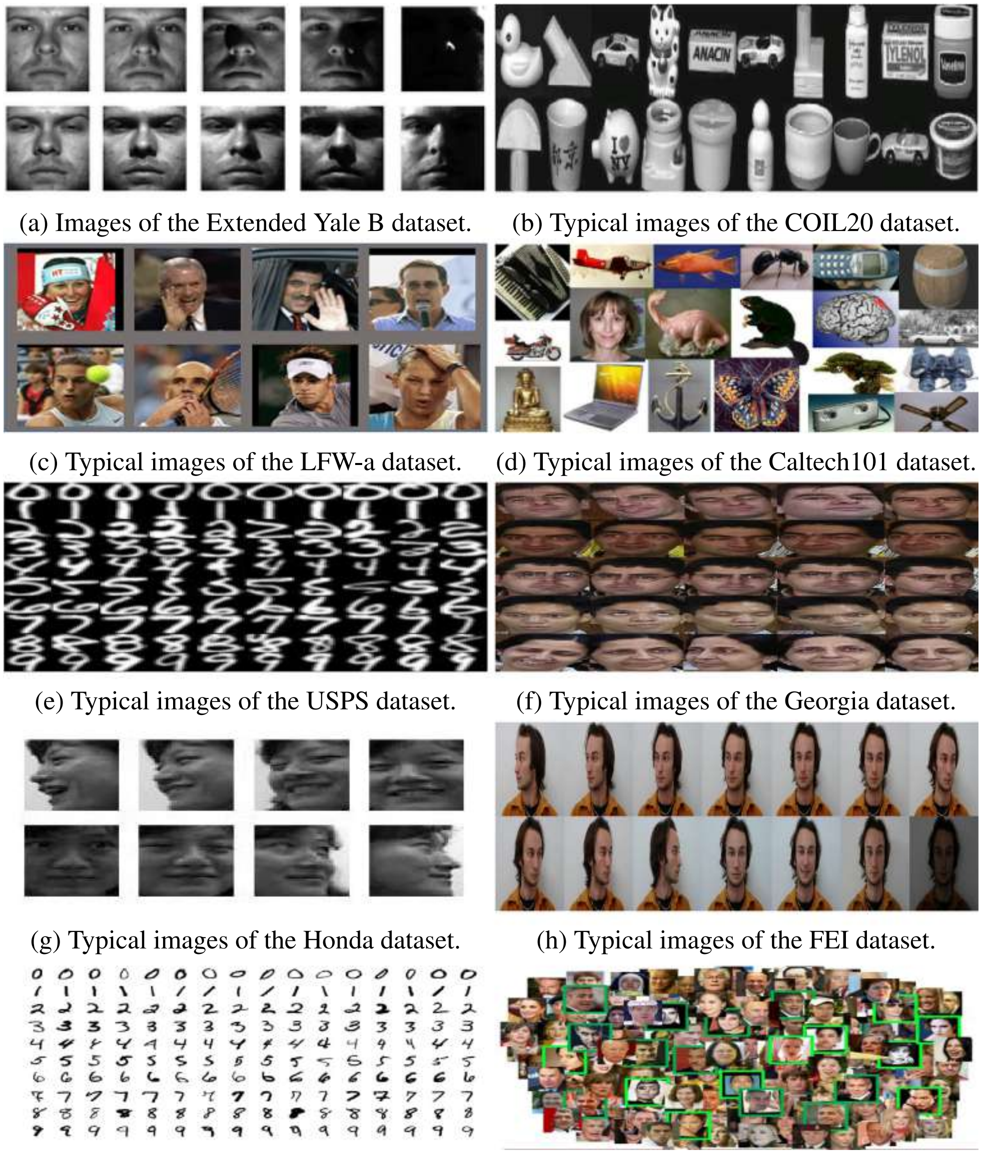
The method is evaluated according to its classification performance on ten different datasets that are described and summarized in the following section.

4.1. Datasets

This section is dedicated to state and give detailed information about image datasets used in this paper, different types and different sizes of datasets are introduced including two large-scale ones (see Fig. 2).

- **Extended Yale B Face Dataset**[1]: This dataset (Georghiades, Belhumeur, & Kriegman, 2001) is constructed from images of faces taken in different illuminations and facial expressions for each subject. The dataset used in this paper in the cropped version which contains between (58 and 64) images for each one of the 38 individuals. It contains a total number of 2414 images each is rescaled to 32 × 32 pixels and represented through gray scale representation. Raw brightness images of dimension 1024 are used in the

[1] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.



(a) Images of the Extended Yale B dataset. (b) Typical images of the COIL20 dataset.

(c) Typical images of the LFW-a dataset. (d) Typical images of the Caltech101 dataset.

(e) Typical images of the USPS dataset. (f) Typical images of the Georgia dataset.

(g) Typical images of the Honda dataset. (h) Typical images of the FEI dataset.

(i) Typical images of the MNIST dataset. (j) Typical images of the PubFig83 dataset.

Fig. 2. Typical images in different datasets.

experiments. The reported results are obtained after we used 10, 15, 20, and 25 samples from each class as training samples and the remaining are used as test samples.

- **COIL20 Object Dataset**[2]: The Columbia Object Image Library (COIL20) (Nene, Nayar, Murase, et al., 1996) dataset is constructed from images of different objects, in which each object is rotated around a vertical axis. The dataset used in this paper contains images of 20 objects with 72 images for each, thus leading to a total number of 1440 images. The image descriptor used is the Local Binary Patterns (LBP) (Li, Fieguth, & Kuang, 2011). We used the uniform LBP histogram (59 values). Three LBP descriptors are constructed from the image using 8 points and three values for the radius ($R = 1$, 2, and 3 pixels). Thus, the final concatenated descriptor has 177 values. The results are obtained after we used 20, 25, 30, and 35 image samples from each class as training samples and the remaining are used as test samples.
- **LFW-a Dataset**[3]: The Labeled Faces in the Wild-a (LFW-a) (Huang, Mattar, Berg, & Learned-Miller, 2008) dataset. While maintaining the same structure as in the original LFW dataset, LFW-a contains the images of the LFW dataset after alignment using a commercial face alignment software. The dataset used in this paper contains images from 141 different classes with a total number of 3408 gray-scale images each rescaled to 32 × 32 pixels. Raw brightness images of dimension 1024 are used in the experiments. The reported results are obtained after we used 5,6,7, and 8 image samples from each class as training samples and the remaining are used as test samples.
- **Caltech101 Dataset**[4]: The Caltech101 dataset used in this paper is the one that contains images of objects belonging to 101 classes. The full Caltech dataset which consists of 256 classes can be found at Griffin, Holub, and Perona (2007). It is a well-known challenging set which contains a set of

[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[3] https://talhassner.github.io/home/projects/lfwa/index.html.

[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101/.

images of complicated backgrounds. We used a cropped version of the original Caltech dataset which consists of 3030 images, 30 images for each one of the 101 classes. The reported results are obtained after we used 5 image samples from each class as training samples and the remaining are used as test samples.

The image descriptor used is the bock-based LBP (Li et al., 2011) representation. We used 100 blocks. For each block, we extract the uniform LBP histogram (59 values). Thus, the length of the image descriptor is 5900.

Moreover, we adopt the deep features provided by the ResNet-50 (He, Zhang, Ren, & Sun, 2016) convolutional neural network. This is a 50 layer convolutional neural network that is trained on the ImageNet database. By using this network, we are able to extract the image representation in the Average Pooling layer. The latter is considered as the image descriptor that has 2048 dimensional vector.

- **USPS Digits Dataset**[5]: The US Postal Service or abbreviated as (USPS) (Seewald, 2005) is a handwritten digits dataset used for digits recognition, this dataset contains 110 images for each digit from 0 to 9, thus it consists of 10 classes, each one contains 110 images, so a total number of 1100 images is used in this dataset, the dimension of images is 16 × 16. Raw-brightness images are used. The reported results are obtained after we used 30, 40, 55, and 65 image samples from each class as training samples and the remaining are used as test samples.
- **Georgia Face dataset**[6]: The Georgia face dataset contains a total number of 750 images that represent 50 individuals. Each individual is represented by 15 images which show frontal and tilted faces with different facial expressions, lighting conditions and scale. The images used are cropped and resized to 32 × 32 pixel for each image. Raw-brightness images (dimension 1024) are used in the experiments. The reported results are obtained after we used 3, 5, 7, and 9 image samples from each class as training samples and the remaining are used as test samples.
- **Honda dataset**[7]: The Honda face dataset contains a total number of 2277 face images. It consists of 22 classes with approximately 97 images per class. The images represent faces submitted to different conditions. Raw brightness images are used in the experiments. The reported results are obtained after we used 10, 20, 30, and 50 image samples from each class as training samples and the remaining are used as test samples.
- **FEI dataset**[8]: The FEI face dataset contains pictures of students and staff at FEI. It is a face dataset that contains a set of colorful face images taken against a white background. The images are in an upright frontal position with profile rotation of up to about 180 degrees. This dataset contains a total number of 700 images, 14 images for each one of the 50 people. Images are resized to 32 × 32 pixels. Raw brightness images of dimension 1024 are used. The reported results are obtained after we used 5, 6, 7, and 8 image samples from each class as training samples and the remaining are used as test samples.
- **MNIST dataset**[9]: The Modified National Institute Of Standards and Technology abbreviated as (MNIST) dataset is a challenging big dataset containing images of handwritten digits. The dataset used in the experiments contains a total number of 60,000 images representing 10 classes. The image descriptor used for the MNIST dataset is of length 2048 and is obtained from the ResNet-50 convolutional neural network. The results are obtained after we have used 1000 images samples from each class as training samples and the remaining are used as test samples.
- **PubFig83 dataset**[10]: The PubFig83 dataset is a large scale and challenging dataset that contains 13,002 images representing faces, collected with different situations (e.g. face expressions, illuminations, background and different poses). The images in this dataset represent 83 different persons where each has from 46 to 231 images. We have used 8720 images for training and the remaining 4282 were used for testing. HOG, LBP, and Gabor wavelet features are extracted from the aligned face images and concatenated, then finally reduced to 2048 dimensions with PCA. The methods are compared with respect to the experimental settings presented in Becker and Ortiz (2013).

4.2. Experimental setup

For fair comparison, experiments are performed using the same experimental setup (datasets, percentage of training/test samples, dimensionality reduction techniques, etc.) The proposed method is compared with the following methods: K-nearest neighbors (KNN) (Kozma, 2008), Support Vector Machines (SVM) (Chang & Lin, 2011), Linear Discriminant Analysis (LDA) (Tharwat et al., 2017), Local Discriminant Embedding (LDE) (Chen et al., 2005), PCE (Peng et al., 2016), ICS_DLSR (Wen, Xu et al., 2018) and Robust sparse LDA (RSLDA) (Wen, Fang et al., 2018). Some additional methods including SULDA (Zhang et al., 2015), MPDA (Zhou & Sun, 2016) and ELDE (Dornaika & Bosaghzadeh, 2013) are added to enrich the comparison for the Extended Yale B and the PubFig83 large dataset. For the PubFig83 large scale dataset, some deep learning methods are also tested. The results are shown in the corresponding tables. All results are obtained on 10 randomly selected splits for each dataset, unless specified otherwise in the result figure caption. We report the average classification accuracy over the 10 splits. We note that the SVM used in the experiments is the Linear SVM. It was implemented using LIBSVM library.[11]

In our experiments, different training and test percentages are used for each dataset as mentioned in Section 4.1. For each dataset and for each method, an embedding is first computed using the training part of the data. The training and test data are then projected using the estimated embedding. Classification of the test data is then performed using either the Nearest Neighbor classifier (NN) (Cunningham & Delany, 2007) or the Support Vector Machines (SVM) classifier (Gunn et al., 1998). Most of the experiments invoked a dimensionality reduction of the raw features before feeding them to the learning models and classifiers. In our experiments, PCA is used as a dimensionality reduction technique and used to preserve 100% energy. We note that, in some conducted experiments, PCA was not used at all in order to illustrate the ability of the method in selecting the most relevant original features.

Moreover, we adopted a simple normalization for the projected data. We proceeded as follows. Let $\mathbf{X}$ be the training data matrix and $\mathbf{Q}$ be the learned projection matrix. The projected data matrix is $\mathbf{Z} = \mathbf{Q}^T\mathbf{X}$. Each element of the matrix $\mathbf{Z}$ is normalized using the following formula:

$$Z_{ij} \longleftarrow \frac{Z_{ij} - min(\mathbf{Z})}{max(\mathbf{Z}) - min(\mathbf{Z})}$$

[5] https://www.kaggle.com/bistaumanga/usps-dataset.

[6] http://www.anefian.com/research/face_reco.htm.

[7] http://vision.ucsd.edu/~leekc/HondaUCSDVideoDatabase/HondaUCSD.html.

[8] https://fei.edu.br/~cet/facedatabase.html.

[9] http://yann.lecun.com/exdb/mnist/.

[10] http://www.briancbecker.com/blog/research/pubfig83-lfw-dataset/.

[11] https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Table 3
Mean classification accuracies (%) of different methods on the Extended Yale B dataset.

No	KNN	SVM	LDA	LDE	ELDE	PCE	SULDA	MPDA	ICS_DLSR	RSLDA	RDA_FSIS
10	69.8	73.85	82.32	79.92	85.85	86.39	84.61	83.67	86.56	86.79	**88.27**
15	75.2	80.02	86.76	83.77	89.30	89.23	88.72	86.82	89.53	89.93	**91.73**
20	80.24	85.79	90.7	88.44	93.07	92.19	91.66	90.38	93.14	93.59	**95.11**
25	82.24	89.03	92.17	90.43	94.09	93.35	92.14	91.79	94.50	94.92	**96.23**

Table 4
Mean classification accuracies (%) of different methods on the tested datasets.

Dataset	Method							
	Training samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**RDA_FSIS**
COIL20	20	94.58	97.65	96.19	95.00	94.87	**98.04**	96.73	97.85
	25	95.79	98.22	97.07	96.12	95.99	98.22	97.74	**98.60**
	30	96.65	98.70	97.81	97.01	97.49	98.75	98.26	**99.10**
	35	97.14	98.81	98.15	97.42	98.11	99.12	98.68	**99.36**
Georgia	3	52.57	56.22	48.18	52.77	46.43	59.73	62.32	**62.67**
	5	61.28	66.98	59.20	62.14	56.18	71.12	73.48	**74.28**
	7	66.73	72.83	67.83	67.10	62.15	78.38	78.82	**79.98**
	9	71.40	77.53	72.57	72.13	66.37	82.57	82.77	**83.30**
Honda	10	64.12	71.32	65.95	65.74	61.86	70.79	69.90	**72.48**
	20	77.69	83.60	79.39	79.25	75.33	82.95	83.03	**84.19**
	30	84.78	89.09	85.84	86.24	82.55	88.20	89.04	**89.44**
	50	91.36	94.15	92.28	92.34	90.03	93.53	94.13	**94.54**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	93.19	**94.01**
	6	90.35	92.93	94.18	92.15	88.73	93.65	94.25	**94.63**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.66	**96.09**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.43	**96.67**
USPS	30	87.01	88.21	84.91	83.54	72.01	88.46	89.45	**90.05**
	40	88.56	90.40	86.19	85.3	72.30	90.16	91.11	**91.27**
	55	90.51	92.09	88.64	87.16	73.32	91.25	**92.65**	92.56
	65	91.76	93.16	89.29	88.58	74.11	91.53	92.89	**93.33**
LFWA-a	5	9.90	12.72	20.51	9.98	9.44	22.56	24.70	**28.07**
	6	10.57	13.61	25.28	10.49	10.26	25.72	28.42	**30.98**
	7	11.06	14.70	28.62	11.24	10.98	29.04	31.50	**33.28**
	8	11.35	15.72	32.42	11.71	11.73	31.92	32.48	**35.80**

Table 5
Mean classification accuracies (%) of different methods on the Caltech101 dataset using original and Deep features.

Caltech101	5 training samples	
Method	LBP features	Deep features
ICS_DLSR	17.20	84.86
RSLDA	16.00	85.34
RDA_FSIS	**17.81**	**85.69**

Table 6
Mean classification accuracies (%) of different methods on the MNIST dataset.

KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	RDA_FSIS
91.75	97.58	85.74	93.22	93.77	98.02	97.95	**98.25**

Table 7
Mean classification accuracies (%) of different methods on the PubFig83 dataset.

Method	Classification accuracy
KNN	63.35
SVM	82.60
LDA	77.95
LDE	62.89
ELDE	65.88
PCE	50.40
SULDA	81.26
MPDA	67.89
ICS_DLSR	85.19
RSLDA	84.78
DeepLDA	44.35
Alexnet	64.00
Resnet50	**90.40**
RDA_FSIS	84.84

where $min(\mathbf{Z})$ and $max(\mathbf{Z})$ denote the minimum and maximum values in the $\mathbf{Z}$ matrix respectively. These two values are stored in order to perform the same normalization (shifting and rescaling) on the projected test data.

The reported classification rates of the methods are chosen from the best parameter configurations and correspond to the average over 10 randomly selected splits as mentioned before.

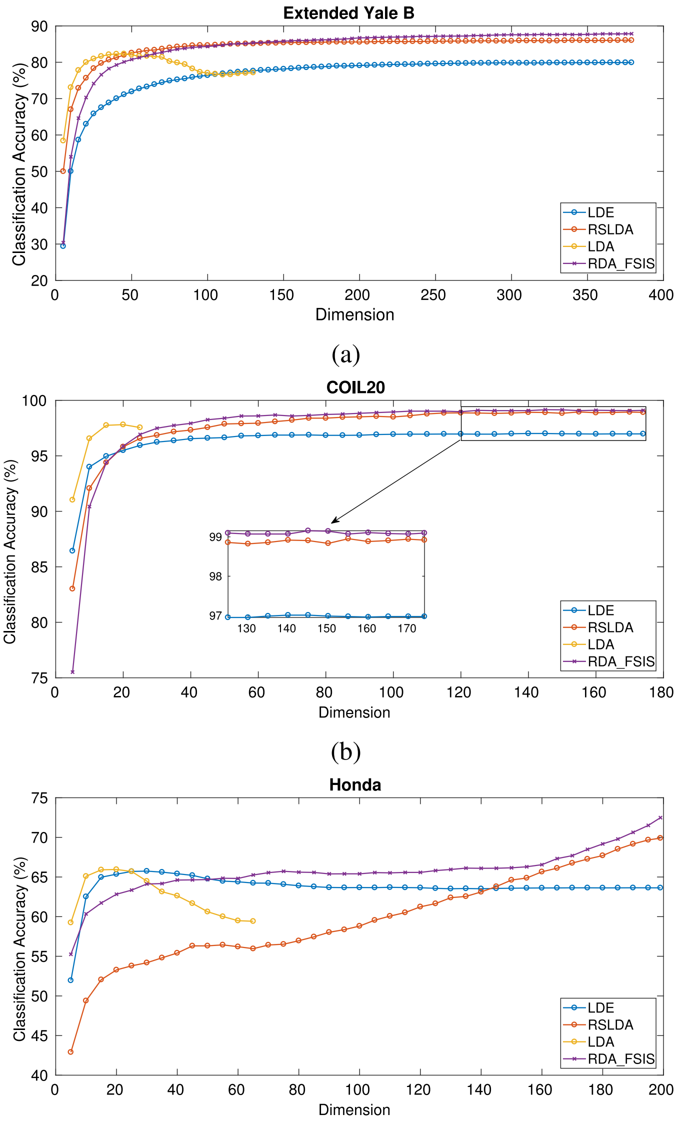
4.3. Experimental results

In this section, we will present the results obtained in our experiments. We will compare our proposed method with the other methods mentioned in Section 4.2.

Tables 3–4 present the mean classification rates of the proposed and competing methods on the Extended Yale B, COIL20, Georgia, Honda, FEI, USPS and LFW-a, respectively. The classifier used in the projection space was the NN classifier. The depicted rates are the average over 10 random splits, and correspond to different numbers of training samples in each case. The left column in every table depicts the number of training images per class.

Table 5 presents the mean classification rate of the proposed and competing methods on the Caltech101 dataset in both cases using the LBP features and the deep features. We emphasize that, for the classification results using the deep features, we did not perform any preprocessing using PCA. Bold numbers denote the best results obtained in each experiment.

Experimental results using large-scale datasets: Tables 6 and 7 present the mean classification accuracy of our proposed method alongside with competing methods for a single split and



(a)

(b)

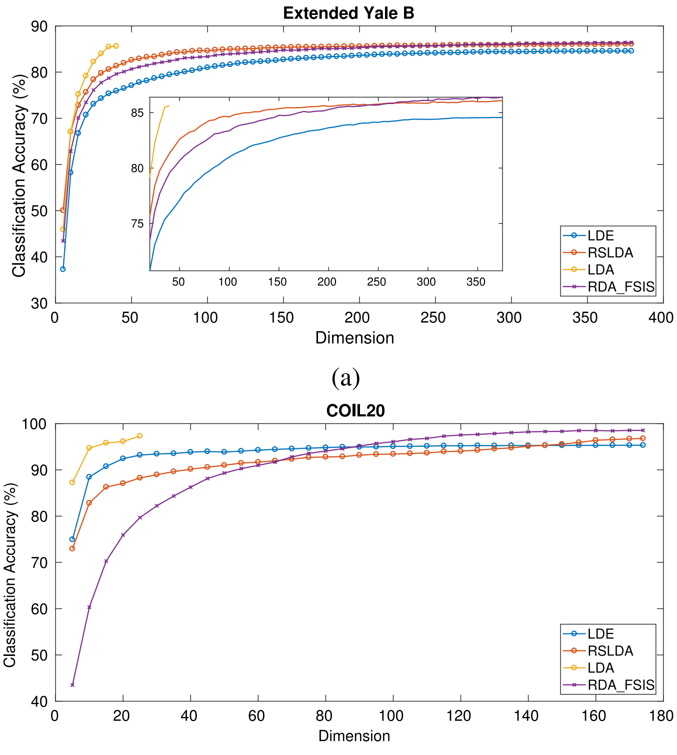
(c)

Fig. 3. Classification rates (%) vs. dimension on (a) Extended Yale B, (b) COIL20 and (c) Honda datasets, in which 10,30 and 10 samples from each class are used for training respectively and the rest for testing using Nearest neighbor classifier (**NN**).

using the experimental settings stated in Section 4.1 for the MNIST and PubFig83 datasets, respectively. The classifier used to obtain these results is the Nearest Neighbor (NN) classifier.

Besides the compared methods mentioned above, three typical deep learning methods, i.e., DeepLDA (Dorfer, Kelz, & Widmer, 2015), Alexnet (Krizhevsky, Sutskever, & Hinton, 2012), and Resnet50 (He et al., 2016) were also evaluated on the PubFig83 database. For DeepLDA and Alexnet, the 8720 training images of PubFig83 are used for training without any pre-trained models. For Resnet50, we use a pre-trained net that is fine-tuned on the 8720 training images of PubFig83. The experimental results are shown in Table 7. As can be seen, the obtained results compete with deep learning paradigms. Although deep learning paradigms may provide more discriminant features, they require a good pre-trained network as well as a large training dataset.

Fig. 3 illustrates the obtained classification performance (%) of the LDE (Chen et al., 2005), LDA (Tharwat et al., 2017), RSLDA (Wen, Fang et al., 2018) and our proposed method vs. the dimension of the projected features for the (a) Extended Yale B (10 training samples used), (b) COIL20 (30 training samples used) and (c) HONDA (10 training samples used) datasets respectively. The results were obtained using the **Nearest Neighbor** (NN) Classifier.



(a)

(b)

Fig. 4. Classification rates (%) of the proposed and competing methods on the Extended Yale B dataset (a) and on the COIL20 dataset (b). 10 and 30 training samples from each class were used respectively. The classifier used was the SVM Classifier.

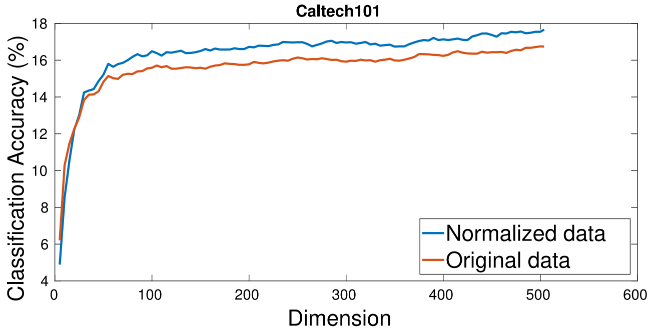
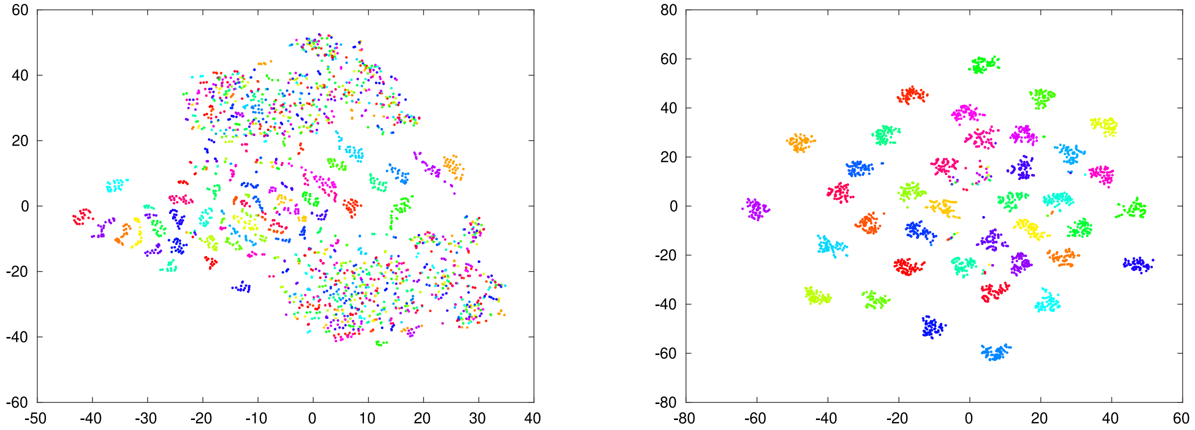


Fig. 5. Classification rates (%) vs. the number of dimensions of our proposed method on the Caltech101 database in which 5 samples from each class are used for training and using the **KNN** Classifier.

Fig. 4 illustrates the average performance on 10 splits of the proposed method and competing methods vs. the number of dimensions for the Extended Yale B and COIL20 datasets in which 10 and 30 samples from each class are used as training and the rest as test samples. The results were obtained using the SVM Classifier.

Effect of projected data normalization. Fig. 5 illustrates the performance enhancement obtained by our proposed method when the projected features are normalized before the classification process. This figure shows the results obtained with the Caltech101



(a) Original features.

(b) Projected features.

Fig. 6. t-SNE visualization of (a) the original feature space and (b) the features obtained after a projection by our method, on the Extended Yale B dataset with 25 training samples.

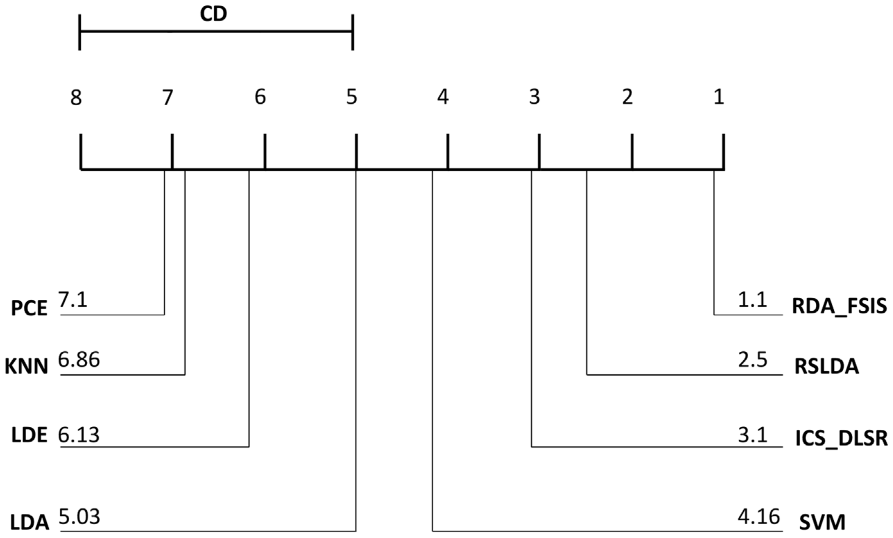


Fig. 7. CD diagram of different methods.

dataset using 5 samples from each class as training samples and the rest as test samples. The results correspond to three randomly selected splits. The red curve shows the performance obtained when the projected data are not normalized. The blue curve depicts the performance when the projected data are normalized.

t-SNE visualization. Fig. 6 shows the distribution of the 2414 images of the Extended Yale B dataset (training and test samples) using the t-SNE (Maaten & Hinton, 2008) technique. In this case, 25 images from each class are used for training (i.e., learning the projection). Fig. 6a shows the distribution of the images of the dataset when the t-SNE uses the original features, while Fig. 6b shows the distribution of the same images when t-SNE uses the projected features obtained by our proposed method.

4.4. Statistical analysis

Statistical analysis of the results can be obtained by performing the Friedman test (Demšar, 2006). This test is used to compare the average ranks of different algorithms. The null hypothesis states that all the algorithms are equivalent, and thus, their ranks should be equal. If the null hypothesis is rejected, one can perform a post-hoc test (the Nemenyi test) to find out which algorithms significantly differ. The Friedman test (run on the average rank of the 8 methods) stated that the performance of all 8 methods is not the same. The Critical Distance CD is computed (Demšar, 2006). In our case, we have a total of 8 methods with 30 evaluations.

Fig. 7 shows the CD diagram for the 8 methods, where the average rank of each is marked along the axis. The CD diagram allows to have groups of methods that are significantly different.

Experimental results have shown that it is more meaningful to apply the Friedman test on the proposed method in addition to the two most competing methods RSLDA (Wen, Fang et al., 2018) and ICS_DLSR (Wen, Xu et al., 2018). The diagram resulting from the test is shown in Fig. 8.

4.5. Parameter sensitivity analysis

In this section, we will study the effect of the algorithm parameters on the final performance. We will also study the effect of removing the term $\sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1}$ in the objective criterion of our method (8).

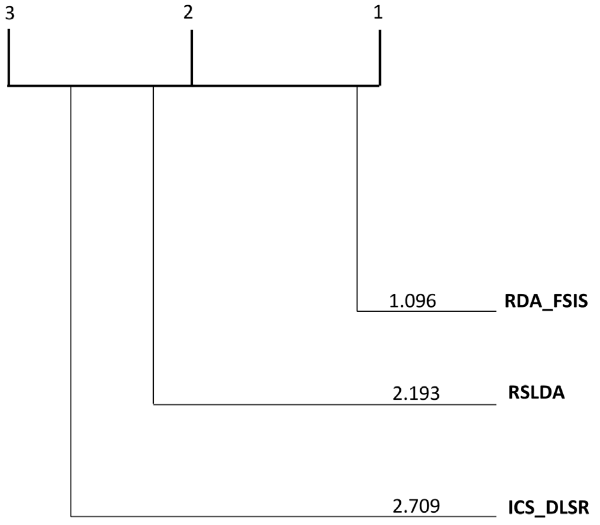


Fig. 8. Statistical Analysis diagram of the proposed method and two most competing methods.

Our proposed method has mainly three balance parameters λ_1, λ_2, and λ_3. λ_1 controls the row sparsity of the sought linear transform $\mathbf{Q}$, λ_2 enforces the robust PCA and λ_3 imposes the inter-class sparsity of the projected data.

We will quantify the classification rates over the test data when these parameters vary. Again, we will adopt ten random splits in order to compute these rates.

Figs. 9–11 study the influence of the parameters of our proposed method in terms of recognition rate (%) using the Extended Yale B, Georgia and USPS datasets, respectively. The number of training images taken from each class was fixed to 10, 9 and 40 for the Extended Yale B, Georgia and USPS datasets, respectively. Sub-figures (a) of the three figures show the recognition rates when the parameters λ_1 and λ_2 vary while λ_3 is fixed. Subfigures (b) show the recognition rates when the parameter λ_3 varies while λ_1 and λ_2 are kept fixed for the best combination.

From the above results that depict the performance using a grid search, we can have a rough idea about the optimal domains for each parameter and for each dataset.

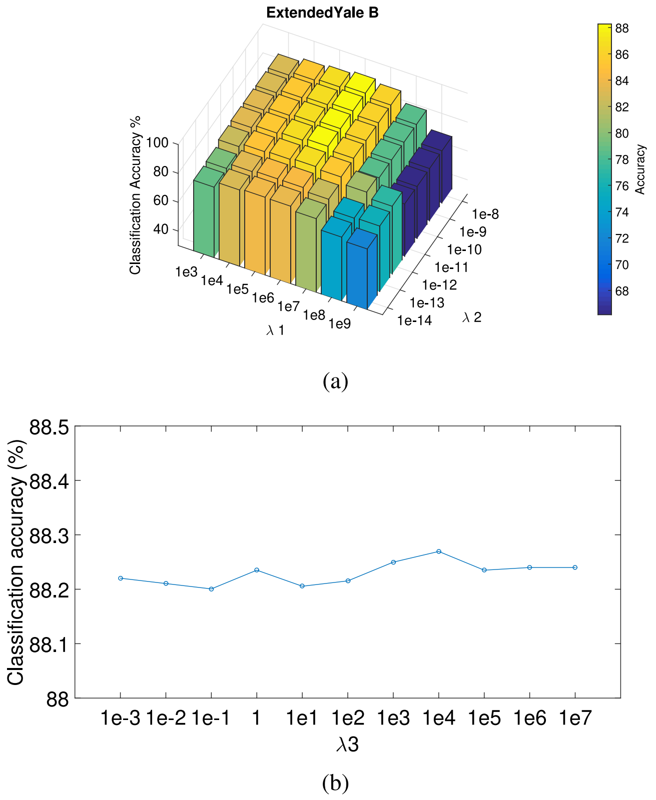
Therefore, Fig. 9a shows that a satisfactory performance can be obtained on the Extended Yale dataset when λ_1 and λ_2 are chosen in the ranges $[10^5, 10^7]$ and $[10^{-12}, 10^{-10}]$, respectively. On the other hand, according to Fig. 9b, any value for λ_3 will almost result in the same performance for this dataset.

Fig. 10a (Georgia dataset) shows that satisfactory performance can be obtained for λ_1 in $[10^6, 10^8]$ and λ_2 in $[10^{-13}, 10^{-11}]$. In addition, Fig. 10b shows that values of λ_3 have a noticeable effect on the classification and should be less than 10^5 for the Georgia dataset.

Fig. 11a (USPS dataset) shows that satisfactory performance can be obtained for λ_1 in $[10^6, 10^8]$ and for λ_2 in $[10^{-9}, 10^{-7}]$. Fig. 11b shows that the chosen values of λ_3 have no effect on the recognition in the case of the USPS dataset.

By combining the quantitative results of the sensitivity of the three parameters, we can deduce that λ_1 should be large (e.g., 10^6), λ_2 should be very small (e.g., 10^{-7}) and λ_3 should be greater than one.

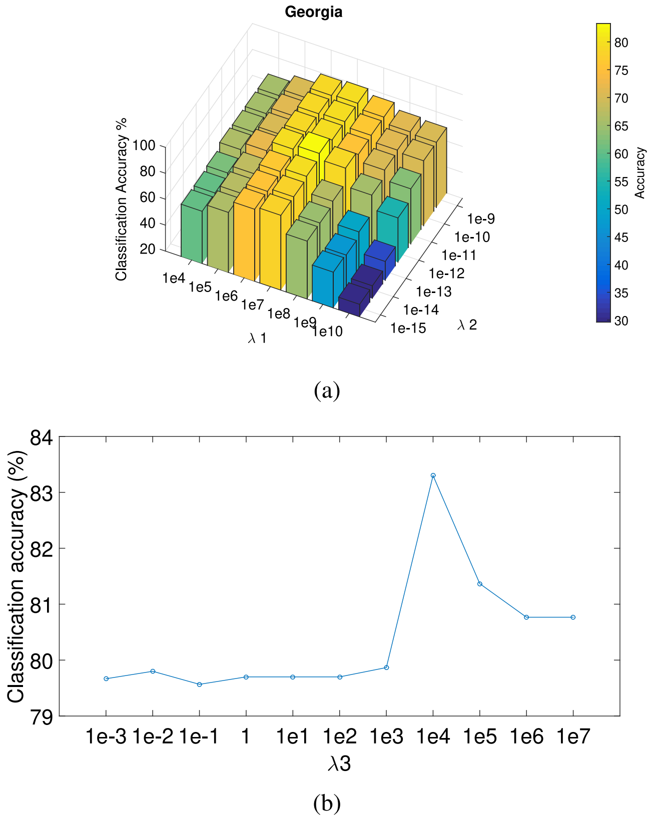
In a second group of experiments, we study the effect of inter-class sparsity on the final performance. To this end, we remove the term $\sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1}$ in the objective criterion of our method (8). This implies the removal of the constraints that ensure the common sparsity of the transformed features of samples in each class from the global criterion.



(a)

(b)

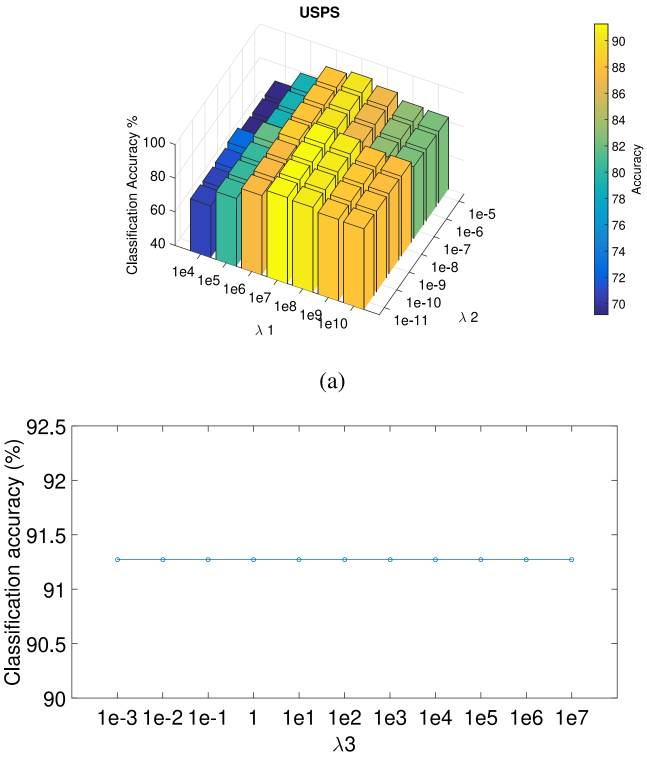
Fig. 9. Classification accuracy (%) according to parameters combinations using the Extended Yale B dataset in which 10 samples from each class are used as training. (a) λ_3 is fixed, (b) λ_1 and λ_2 are fixed.



(a)

(b)

Fig. 10. Classification accuracy (%) according to parameters combinations using the Georgia dataset in which 9 samples from each class are used as training. (a) λ_3 is fixed, (b) λ_1 and λ_2 are fixed.



(a)

(b)

Fig. 11. Classification accuracy (%) according to parameters combinations using the USPS digits dataset in which 40 samples from each class are used as training. (a) λ_3 is fixed, (b) λ_1 and λ_2 are fixed.

Table 8
Mean classification accuracy (%) of the proposed method on some datasets with and without the presence of the inter-class sparsity constraint **F**.

Dataset	Training samples	Without inter-class sparsity	With inter-class sparsity
Extended Yale	10	86.36	88.27
COIL20	20	97.01	97.85
Georgia	9	82.94	83.30
Honda	20	83.63	84.19

Table 8 shows the recognition rate (%) obtained by our method with and without the presence of the inter-class constraint in the global criterion. We can clearly see that the introduction of these constraints yields a better classification in general and is important in the formulation of our algorithm.

4.6. Explicit feature selection and their effect

This section investigates the effect of feature ranking and selection in the original data matrix **X** and the projection data **Q** obtained by our method. We recall that our learning model provides a ranking of the original features. Indeed, the score of the *i*th original feature is given by the ℓ_2 norm of the *i*th row of the linear transform **Q**.

The purpose is to provide another linear projection space that exploits the explicit ranking of the original features. Let $\mathbf{X}_s$ and $\mathbf{Q}_s$ denote the selected data matrix and projection matrix using the computed scores $\|\mathbf{Q}_{i*}\|_{i=1}^{d}$ where $\mathbf{Q}_{i*}$ is the *i*th row of the matrix **Q**. To compute $\mathbf{X}_s$ and $\mathbf{Q}_s$, we proceed as follows (see Fig. 12).

Process: After applying our proposed method, we obtain the projection matrix **Q**. First, we calculate the score corresponding to each row of the matrix **Q** by calculating its ℓ_2 norm. Thus, we have $score(i) = \|\mathbf{Q}_{i*}\|_2$ where $i = 1, \ldots, d$.

Second, we sort the obtained scores in a descending order so that the most important features are in the top. Third, we rank the original *d* features of the data **X** and the projection matrix **Q** according to the sorted scores.

Let *s* be the number of selected features in the original space ($s \leq d$). The obtained ranked **X** and **Q** are then cropped (only the *s* top rows in these two matrices are kept. The resulting matrices are denoted by $\mathbf{X}_s \in \mathbb{R}^{s\times N}$ and $\mathbf{Q}_s \in \mathbb{R}^{s\times d}$, respectively. Finally, we obtain the projected data $\mathbf{Z} = {\mathbf{Q}_s}^T\mathbf{X}_s$. We emphasize that the dimensionality of the projection space is the same if we use either $\mathbf{Z} = \mathbf{Q}^T\mathbf{X}$ or ${\mathbf{Q}_s}^T\mathbf{X}_s$. However, these two projections are different. Furthermore, by varying the number *s* of selected features, we can obtain several projection spaces.

More importantly, we notice that these new projection spaces do not need to solve the objective function (8). Indeed, the computation of matrices $\mathbf{X}_s$ and $\mathbf{Q}_s$ is very efficient since it only requires norms computation followed by a ranking and selection of their rows.

Fig. 13 illustrates the recognition rate of the proposed method as a function of the original features, *s*, selected from the data matrix **X** and the projection matrix **Q**.

Fig. 13a shows the recognition accuracy of the proposed method vs. the number of original features for the Extended Yale B dataset in which 5 samples per class are used for training and the rest for testing. The experiment is conducted on a single split, and the studied features are the original ones (not processed by PCA).

Fig. 13b shows the recognition accuracy of the proposed method for the USPS dataset in which 30 samples per class are used for training and working on 10 splits without PCA.

Fig. 13c denotes the recognition accuracy of the proposed method for the Caltech101 dataset in which 5 samples per class are used for training and the rest for testing. Here, the deep features (2048 features) of Caltech101 dataset are used and the results shown in Fig. 13c correspond to the average of 3 splits. In this case no PCA was applied on the data.

In Fig. 13, the **blue** plot depicts the recognition accuracy of the test data after computing a projection space based on a subset of the original features in the raw data matrix **X** and in the **Q without** any ranking. The **red** plot shows the recognition accuracy vs. the dimension of ranked $\mathbf{X}_s$ and $\mathbf{Q}_s$ as explained above.

Fig. 13 (see the red curves) shows that the use of selected original features yields better performance for the same number of original features (see the blue curves).

The same figure (see the red curves) shows that the proposed method can achieve and guarantees superiority over other methods, when selecting 73% of the original features (750 features out of 1024) for the Extended Yale B dataset and 66% of the original features (170 features out of 256) for the USPS dataset. For the Caltech101 dataset selecting around 1400 features from a total of 2048 (69%) is enough to ensure that the proposed method achieves superiority over the compared methods in this paper.

Fig. 14 illustrates the most relevant pixels in the USPS images. Pixels in yellow color denote the best 66% (170 features) of the original pixels. These former are found by the estimated projection matrix as being the most important features used in classification process. Pixels in dark blue correspond to the least important features. In general, we can see that the pixels on borders of the image are not useful. In contrary, the pixels belonging to the center of the image are the most important in classification.

In general, *s* can have any value above 75% of the original features in order to guarantee our method's superiority over competing methods. It is worthy noticing that our proposed method is superior to other competing methods in this paper even when all original features are used.

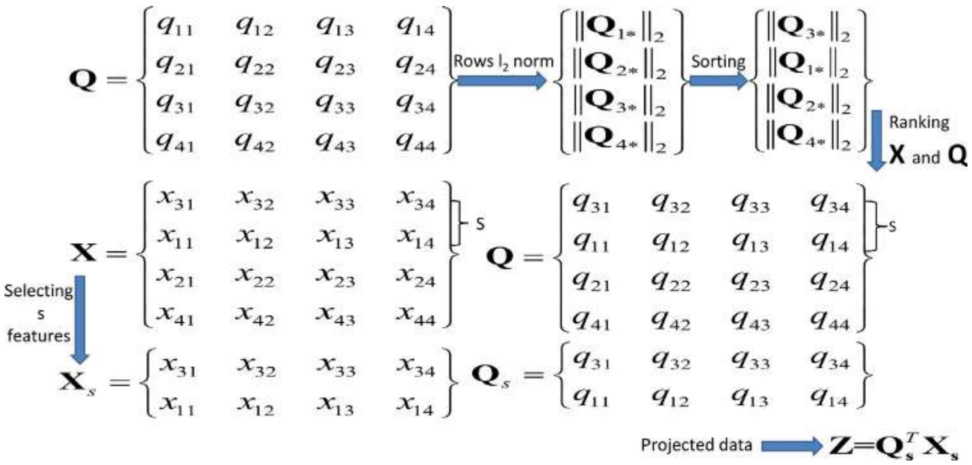
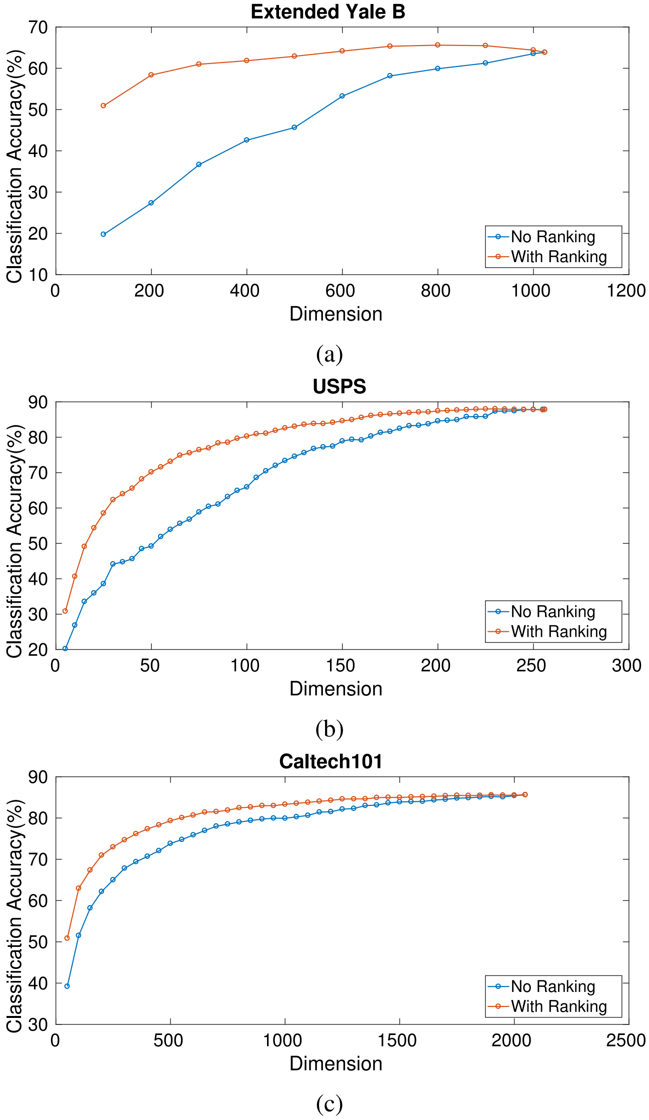


Fig. 12. Illustration of features selection and ranking, the example given and described shows the ranked features (3,2,1,4), the scores are sorted in a descending order which indicates in this example that row number 3 has the highest score and row number 4 has the lowest one.



(a)

(b)

(c)

Fig. 13. Recognition rates (%) vs. the number of selected features from **X** and **Q** for (a) the Extended Yale B database, (b) the USPS digits dataset, and (c) the Caltech101 dataset, in which 5 and 30 and 5 samples from each class respectively are used for training and the remaining samples for testing, (**NN**) classifier is used.

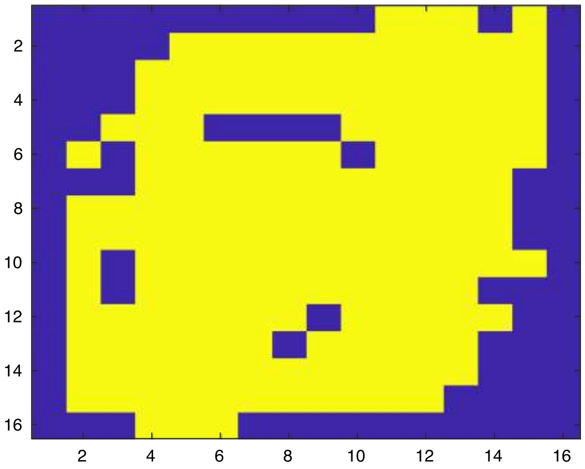


Fig. 14. Map of selected pixels (66%) associated with USPS images.

4.7. Convergence analysis

We have solved the proposed method as an optimization non-convex problem (8) using the alternating direction method of multipliers (Boyd et al., 2011). We plotted the objective function of our problem with respect to the number of iterations.

The objective function is calculated through:

$$f = Tr\,(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}) + \lambda_1\,\|\mathbf{Q}\|_{2,1} + \lambda_2\,\|\mathbf{E}\|_1 + \lambda_3 \sum_{i=1}^{C} \|\mathbf{F}_i\|_{2,1}$$

Figs. 15 and 16 show the objective function of the proposed method for the Extended Yale and COIL20 datasets, respectively. These figures illustrate the recognition rates as a function of the number of iterations. As can be seen, the objective function decreases with the increase of the number of iterations and gets close to a stable value within a limited number of iterations, which demonstrates a good convergence of the proposed method. These two figures illustrate also the recognition rates as a function of the number of iterations.

4.8. Analysis of the results and method comparison

The experimental results presented in the figures and tables of this paper demonstrate the superiority of the proposed

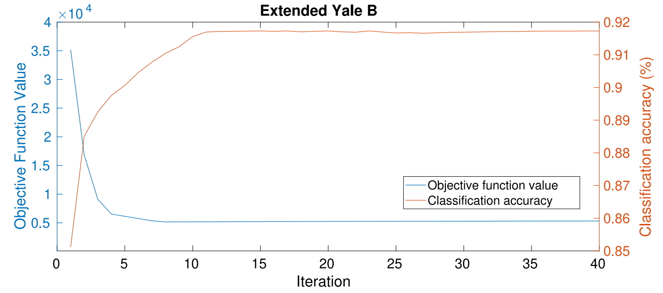


Fig. 15. Objective function value and classification accuracy of the proposed method with respect to the number of iterations. The objective function corresponds to the Extended Yale B dataset in which 10 samples from each class are randomly selected and used as training samples.

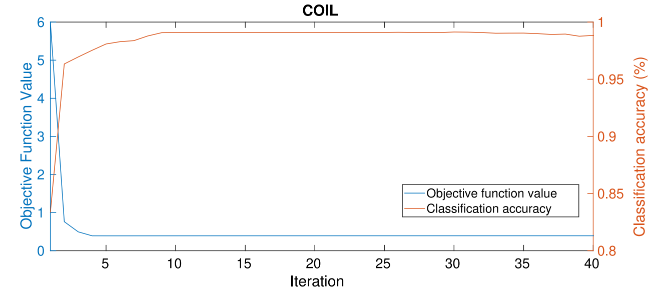


Fig. 16. Objective function value and the classification accuracy of the proposed method with respect to the number of iterations. The objective function corresponds to the COIL20 dataset in which 30 samples from each class are randomly selected and used as training samples.

method compared to other competing methods. However, many observations can be made.

In most of cases, the proposed method provided superior recognition accuracy (%) than other competing methods in the tested cases depicted in Tables 3–7. Nevertheless, the proposed method was outperformed by the RSLDA method while using the USPS digits dataset in only the case in where 55 training samples from each class were used for training. However, for the same dataset (USPS), in the remaining cases related to other training percentages, the proposed method led to better performance and again outperformed its competing methods.

It is important to emphasize that the proposed method provided a superior performance over the competing methods when the SVM classifier was used (Gunn et al., 1998) as depicted in Fig. 4. Both KNN and SVM classifiers can be used to achieve superior classification using the proposed projection method.

In addition, the proposed method achieved high classification accuracy without using a lot of features. Fig. 3 demonstrates that the proposed method performs well on low dimensions with few features in the projected space.

The t-SNE visualization presented in Fig. 6 to illustrate the distribution of images for the Extended Yale dataset demonstrates that the projected data are better discriminated. Indeed, the images belonging to the same class are grouped close to each other while the ones belonging to different classes are pushed away, leading to a good classification property.

Also, it is worthy to note that the normalization of the data after performing the projection using the proposed method and before the classification process could lead to an enhancement in the classification process for some datasets.

We have noticed that the recognition rates of all methods for the images of some datasets like LFW-a and Caltech10 are noticeably low if classic hand-crafted image descriptors are used. These datasets are challenging with complicated backgrounds and highly variable appearances. However, with the use of deep features as image descriptors, the recognition rates of the proposed method as well as other competing methods increase in a noticeable way (see the Caltech101 dataset results).

We should notice that the objective of this paper is the comparison of projection methods in the same context and not choosing the best image descriptor.

From the above results, we can conclude that the proposed method was, in general, superior to its closest competitors (i.e., the RSLDA and ICS_DLSR methods). This observed superiority is due to the following reasons. First, the RSLDA method does not impose a common sparse structure of the projected data of each class; whereas our proposed method imposes such property. Second, the ICS_DLSR method is only concerned with mapping the original space to the label space and imposes common sparse structure for the projected data of each class. It neglects the discrimination information that can be gained by integrating the robust LDA criterion with feature ranking. In a nutshell, the proposed method simultaneously integrated all desired properties into one single objective function.

5. Conclusion

In this paper, we have presented a novel discriminant supervised method which aims to learn an informative and discriminative projection space for the data. The proposed model was solved in an iterative way and showed good convergence property. It can simultaneously select and use the most discriminative features from the data by minimizing the $\ell_{2,1}$ norm of the projection matrix. More discriminant and representative features were obtained by combining classical Linear Discriminant Analysis alongside with the inter-class sparsity constraint that ensures common sparsity of the transformed features in each class.

Through this combination, the proposed method achieved higher classification rates than many competing methods.

Future work may address many updates and could be divided into four tracks. First, the proposed framework can be extended to the semi-supervised setting in which the training data have labeled and unlabeled samples. Second, since the proposed criterion is nonlinear, we may propose a refinement of the obtained solution of the projection matrix using some gradient-based approaches. The third track is to transform the current shallow model into a deep model that exploits several linear transformations which can lead to better data representation. Since the proposed learning model is built upon the Linear Discriminant Analysis criterion (first term in the proposed objective function), the fourth track can be the trial of other discriminant criteria that can be merged with it or replace it.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Proposition. *Let* $\theta \triangleq (\mathbf{Q}, \mathbf{P}, \mathbf{F}, \mathbf{E}, \mathbf{Y}_1, \mathbf{Y}_2)$ *and* $\left\{\theta^t\right\}_{t=1}^{\infty}$ *be the sequence generated by Algorithm 1 and suppose that* $\{\theta\}_{t=1}^{\infty}$ *is bounded at* $\lim_{t\to\infty}\left[\theta^{t+1}-\theta^t\right] = \mathbf{0}$. *Then every limit point of* $\left\{\theta^t\right\}_{t=1}^{\infty}$ *satisfies the KKT conditions. Thus, whenever* $\{\theta\}_{t=1}^{\infty}$ *converges, it converges to a KKT point.*

Proof. Let us assume that the proposed algorithm reaches a stationary point. The Karush–Kuhn–Tucker (KKT) conditions are derived as follows (we note that the procedure of solving $\mathbf{P}$ does not involve in the Lagrange multipliers, and thus, we do not prove the KKT condition for it):

1. $\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\mathbf{X} = \mathbf{E}$ (29)
2. $\mathbf{F} = \mathbf{Q}^T\mathbf{X}$ (30)
3. $\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = 2\ \mathbf{SQ} + \lambda_1\mathbf{UQ} + \beta\,[\mathbf{XX}^T\mathbf{Q} - \mathbf{XM}^T\mathbf{P}] + \beta'[\mathbf{XX}^T\mathbf{Q} - \mathbf{XM}'^T] = \mathbf{0}$ (31)
4. $\mathbf{E} = shrink_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta})$ (32)
5. $\frac{\partial \mathcal{L}}{\partial\,|\mathbf{F}_i|_j} = [\mathbf{Y}_2(i)]_j + \lambda_3\frac{\partial\left\|[\mathbf{F}_i]_j\right\|_2}{\partial[\mathbf{F}_i]_j} = \mathbf{0}$ $\forall i = 1,\ldots,C,\ j = 1,\ldots,d$ (33)

where $\mathbf{F}_i$ and $\mathbf{Y}_2(i)$ are the ith submatrices of $\mathbf{F}$ and $\mathbf{Y}_2$ corresponding to the samples of the ith class, respectively. $[\mathbf{F}_i]_j$ and $[\mathbf{Y}_2(i)]_j$ denote the jth row vectors of $\mathbf{F}_i$ and $\mathbf{Y}_2(i)$, respectively.

- Let $(\mathbf{Q}^+, \mathbf{P}^+, \mathbf{F}^+, \mathbf{E}^+, \mathbf{Y}_1{}^+, \mathbf{Y}_2{}^+)$ be the solution at the next iteration (i.e. at iteration $t+1$). The Lagrange multipliers $\mathbf{Y}_1$ and $\mathbf{Y}_2$ from Algorithm 1 are given by:

$$\mathbf{Y}_1{}^+ = \mathbf{Y}_1 + \beta\,(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} - \mathbf{E}) \tag{34}$$

$$\mathbf{Y}_2{}^+ = \mathbf{Y}_2 + \beta'(\mathbf{F} - \mathbf{Q}^T\mathbf{X}) \tag{35}$$

If the sequences of variables $\left\{\mathbf{Y}_1{}^t\right\}$ and $\left\{\mathbf{Y}_2{}^t\right\}$ converge to a stationary point as $(\mathbf{Y}_1^+ - \mathbf{Y}_1) \to \mathbf{0}$ and $(\mathbf{Y}_2^+ - \mathbf{Y}_2) \to \mathbf{0}$, then we have:

$$\mathbf{X} - \mathbf{PQ}^T\mathbf{X} - \mathbf{E}\ \to \mathbf{0} \tag{36}$$

$$\mathbf{F} - \mathbf{Q}^T\mathbf{X}\ \to \mathbf{0} \tag{37}$$

Thus, the first two KKT conditions are satisfied.

- According to Algorithm 1, the expression of $\mathbf{Q}^+$ is given by:

$$\mathbf{Q}^+ = \mathbf{A}^{-1}(\mathbf{BXM}^T\mathbf{P} + \beta'\mathbf{XM}'^T) \tag{38}$$

where

$$\mathbf{A} = (2\ \mathbf{S} + \lambda_1\mathbf{U} + \beta\mathbf{XX}^T + \beta'\mathbf{XX}^T) \tag{39}$$

$$\begin{aligned}\mathbf{Q}^+ - \mathbf{Q} \to \mathbf{0}\ &\Rightarrow \mathbf{A}^{-1}(\mathbf{BXM}^T\mathbf{P} + \beta'\mathbf{XM}'^T) - \mathbf{Q} = \mathbf{0}\\ &\Rightarrow 2\mathbf{SQ} + \lambda_1\mathbf{UQ} + \beta[\mathbf{XX}^T\mathbf{Q} - \mathbf{XM}^T\mathbf{P}] + \beta'\mathbf{XM}'^T\\ &= \mathbf{0}\end{aligned} \tag{40}$$

Thus, the third KKT condition is satisfied.

- We have:

$$\mathbf{Y}_1 = \lambda_2\frac{\partial}{\partial\mathbf{E}}\|\mathbf{E}\|_1 \tag{41}$$

This yields:

$$\begin{aligned}\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta} &= \mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\mathbf{X} + \frac{\lambda_2}{\beta}\frac{\partial}{\partial\mathbf{E}}\|\mathbf{X} - \mathbf{PQ}^T\mathbf{X}\|_1\\ &= \Gamma_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X})\end{aligned} \tag{42}$$

where $\Gamma_{\frac{\lambda_2}{\beta}}(x)$ denotes element-wise scalar function that is given by $\Gamma_{\frac{\lambda_2}{\beta}}(x) = x + \frac{\lambda_2}{\beta}\partial|x|$.

By using the inverse function of $\Gamma_{\frac{\lambda_2}{\beta}}(x)$ in the above equation, we get the expression of $\mathbf{E}$ as:

$$\mathbf{E}^+ = \Gamma^{-1}_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta}) \tag{43}$$

From Shen, Wen, and Zhang (2014), the inverse of the function $\Gamma_{\frac{\lambda_2}{\beta}}(x)$ is approximated by the element-wise shrinkage operator with parameter equal to $\frac{\lambda_2}{\beta}$ (defined in Eq. (18)):

$$\mathbf{E}^+ \cong shrink_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta}) \tag{44}$$

$$\mathbf{E}^+ - \mathbf{E} = shrink_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta}) - \mathbf{E} \tag{45}$$

When $\mathbf{E}^+ - \mathbf{E} \to \mathbf{0}$, then $\mathbf{E} = shrink_{\frac{\lambda_2}{\beta}}(\mathbf{X} - \mathbf{PQ}^T\mathbf{X} + \frac{\mathbf{Y}_1}{\beta})$. Thus, the fourth KKT condition is satisfied.

- Since $\mathbf{F} = \mathbf{Q}^T\mathbf{X}$ (second KKT condition) and $\mathbf{H} = \mathbf{Q}^T\,\mathbf{X} - \frac{\mathbf{Y}_2}{\beta'}$ (definition of the matrix $\mathbf{H}$ in Eq. (20)), we have:

$$\frac{\mathbf{Y}_2}{\beta'} = \mathbf{F} - \mathbf{H} \tag{46}$$

From Eq. (24), when $\{\mathbf{F}\}_{t=1}^{\infty}$ converges, we have:

$$[\mathbf{F}_i]_j^+ - [\mathbf{F}_i]_j = \begin{cases} [\mathbf{H}_i]_j - \dfrac{\lambda_3\,[\mathbf{H}_i]_j}{\beta'\left|\left|[\mathbf{H}_i]_j\right|\right|_2} - [\mathbf{F}_i]_j = \mathbf{0} & \text{if } \left\|[\mathbf{H}_i]_j\right\|_2 > \dfrac{\lambda_3}{\beta'}\\ -\,[\mathbf{F}_i]_j = \mathbf{0}, & \text{if } \|\,[\mathbf{H}_i]_j\,\|_2 \le \dfrac{\lambda_3}{\beta'}\end{cases} \tag{47}$$

where $\mathbf{H}_i$ is the ith submatrix of $\mathbf{H}$ corresponding to the samples of the ith class and $[\mathbf{H}_i]_j$ denotes the jth row vector of $\mathbf{H}_i$.

From Liu et al. (2009), we have:

$$\frac{\partial\left(\left|\left|[\mathbf{F}_i]_j\right|\right|_2\right)}{\partial\,[\mathbf{F}_i]_j} = \begin{cases}\dfrac{[\mathbf{F}_i]_j}{\left|\left|[\mathbf{F}_i]_j\right|\right|_2}, & \text{if } [\mathbf{F}_i]_j \neq \mathbf{0}\quad (\|\,[\mathbf{H}_i]_j\,\|_2 > \dfrac{\lambda_3}{\beta'})\\ \dfrac{\beta'}{\lambda_3}\,[\mathbf{H}_i]_j\,, & \text{if } [\mathbf{F}_i]_j = \mathbf{0}\quad (\|\,[\mathbf{H}_i]_j\,\|_2 \le \dfrac{\lambda_3}{\beta'})\end{cases} \tag{48}$$

– If $\|\,[\mathbf{H}_i]_j\,\|_2 > \frac{\lambda_3}{\beta'}$, then from Eq. (47), we have:

$$[\mathbf{H}_i]_j - \frac{\lambda_3}{\beta'}\frac{[\mathbf{H}_i]_j}{\left|\left|[\mathbf{H}_i]_j\right|\right|_2} - |\mathbf{F}_i|_j = \mathbf{0} \tag{49}$$

Since $[\mathbf{H}_i]_j - [\mathbf{F}_i]_j = -\frac{[\mathbf{Y}_2(i)]_j}{\beta'}$ (from Eq. (46)) and $\frac{[\mathbf{H}_i]_j}{\left|\left|[\mathbf{H}_i]_j\right|\right|_2} = \frac{[\mathbf{F}_i]_j}{\left|\left|[\mathbf{F}_i]_j\right|\right|_2}$ ($[\mathbf{H}_i]_j$ and $[\mathbf{F}_i]_j$ are collinear vectors having the same unit vector), we have:

$$\frac{-\,[\mathbf{Y}_2(i)]_j}{\beta'} - \frac{\lambda_3}{\beta'}\frac{[\mathbf{F}_i]_j}{\left|\left|[\mathbf{F}_i]_j\right|\right|_2} = \mathbf{0}$$

By using Eq. (48), the above equation can yield the last KKT condition:

$$[\mathbf{Y}_2(i)]_j + \lambda_3\frac{\partial\left\|[\mathbf{F}_i]_j\right\|_2}{\partial[\mathbf{F}_i]_j} = \mathbf{0}\qquad \forall i = 1,\ldots,C, j = 1,\ldots,d \tag{50}$$

– If $\left|\left|[\mathbf{H}_i]_j\right|\right|_2 < \frac{\lambda_3}{\beta'}$, then the equation $[\mathbf{Y}_2(i)]_j + \lambda_3 \frac{\partial \left|\left|[\mathbf{F}_i]_j\right|\right|_2}{\partial [\mathbf{F}_i]_j} = \mathbf{0}$ still holds as shown below.
From Eq. (46), we have $[\mathbf{F}_i]_j = [\mathbf{H}_i]_j + \frac{[\mathbf{Y}_2(i)]_j}{\beta'}$. In the case when $\| [\mathbf{H}_i]_j \|_2 < \frac{\lambda_3}{\beta'}$, from Eqs. (47) and (48), we have respectively $[\mathbf{F}_i]_j = \mathbf{0}$ and $\frac{\beta'}{\lambda_3} [\mathbf{H}_i]_j = \frac{\partial \left|\left|[\mathbf{F}_i]_j\right|\right|_2}{\partial [\mathbf{F}_i]_j}$. By substituting these expressions in $[\mathbf{F}_i]_j = [\mathbf{H}_i]_j + \frac{[\mathbf{Y}_2(i)]_j}{\beta'}$, the latter becomes:

$$\begin{aligned} \mathbf{0} &= \frac{[\mathbf{Y}_2(i)]_j}{\beta'} + \frac{\lambda_3}{\beta'} \frac{\partial \left|\left|[\mathbf{F}_i]_j\right|\right|_2}{\partial \, [\mathbf{F}_i]_j} \\ &\Rightarrow [\mathbf{Y}_2(i)]_j + \lambda_3 \frac{\partial \left|\left|[\mathbf{F}_i]_j\right|\right|_2}{\partial \, [\mathbf{F}_i]_j} = \mathbf{0} \quad \forall i = 1, \ldots, C, j = 1, \ldots, d \end{aligned} \tag{51}$$

Thus the final KKT condition is also proved.

Thus, the value of the sequence $\left\{\theta^t\right\}_{t=1}^{\infty}$ asymptotically satisfies the KKT condition for the objective function (7). □

References

Becker, B., & Ortiz, E. (2013). Evaluating open-universe face identification on the web. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 904–911).

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, *3*(1), 1–122.

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, *58*(3), 11.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Chen, H.-T., Chang, H.-W., & Liu, T.-L. (2005). Local discriminant embedding and its variants. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 2* (pp. 846–853). IEEE.

Clemmensen, L., Hastie, T., Witten, D., & Ersboll, B. (2011). Sparse discriminant analysis. *Technometrics*, *53*(4), 406–413.

Courcoubetis, C., & Weber, R. (2003). Lagrangian methods for constrained optimization. In *Wiley-interscience series in systems and optimization*, (p. 333). Wiley Online Library.

Cunningham, P., & Delany, S. J. (2007). K-nearest neighbour classifiers. *Multiple Classifier Systems*, *34*(8), 1–17.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, *7*, 1–30.

Dorfer, M., Kelz, R., & Widmer, G. (2015). Deep linear discriminant analysis. In *International conference on learning representations* (pp. 1–13).

Dornaika, F., & Bosaghzadeh, A. (2013). Exponential local discriminant embedding and its application to face recognition. *IEEE Transactions on Cybernetics*, *43*(3), 921–934.

Dornaika, F., & El Traboulsi, Y. (2016). Learning flexible graph-based semi-supervised embedding. *IEEE Transactions on Cybernetics*, *46*(1), 206–218.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of Statistics*, *32*(2), 407–499.

Fan, Z., Xu, Y., & Zhang, D. (2011). Local linear discriminant analysis framework using sample neighbors. *IEEE Transactions on Neural Networks*, *22*(7), 1119–1132.

Fang, X., Teng, S., Lai, Z., He, Z., Xie, S., & Wong, W. K. (2017). Robust latent subspace learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2502–2515.

Georghiades, A. S., Belhumeur, P. N., & Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 643–660.

Gou, J., Yi, Z., Zhang, D., Zhan, Y., Shen, X., & Du, L. (2018). Sparsity and geometry preserving graph embedding for dimensionality reduction. *IEEE Access*, *6*, 75748–75766.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset*. California Institute of Technology.

Gunn, S. R., et al. (1998). Support vector machines for classification and regression. *ISIS Technical Report*, *14*(1), 5–16.

Han, N., Wu, J., Liang, Y., Fang, X., Wong, W. K., & Teng, S. (2018). Low-rank and sparse embedding for dimensionality reduction. *Neural Networks*, *108*, 202–216.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical Science*, 1–14.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (cvpr)* (pp. 770–778).

Huang, R., Liu, Q., Lu, H., & Ma, S. (2002). Solving the small sample size problem of lda. In *Null* (p. 30029). Citeseer.

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'real-life'images: detection, alignment, and recognition*.

Kozma, L. (2008). *K nearest neighbors algorithm (KNN)*. Helsinki University of Technology.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kwak, N., & Choi, C.-H. (2002). Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, *13*(1), 143–159.

Lai, Z., Xu, Y., Jin, Z., & Zhang, D. (2014). Human gait recognition via sparse discriminant projection learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(10), 1651–1662.

Li, L., Fieguth, P. W., & Kuang, G. (2011). Generalized local binary patterns for texture classification. In *BMVC, Vol. 123* (pp. 1–11).

Liu, J., Ji, S., & Ye, J. (2009). Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (pp. 339–348). AUAI Press.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, *9*(Nov), 2579–2605.

Martínez, A. M., & Kak, A. C. (2001). Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(2), 228–233.

Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). *Columbia object image library (coil-20): Technical report CUCS-005-96*.

Nie, F., Wang, Z., Wang, R., & Li, X. (2019). Submanifold-preserving discriminant analysis with an auto-optimized graph. *IEEE Transactions on Cybernetics*.

Peng, X., Lu, J., Yi, Z., & Yan, R. (2016). Automatic subspace learning via principal coefficients embedding. *IEEE Transactions on Cybernetics*, *47*(11), 3583–3596.

Qiao, Z., Zhou, L., & Huang, J. Z. (2009). Sparse linear discriminant analysis with applications to high dimensional low sample size data. *International Journal of Applied Mathematics*, *39*(1).

Seewald, A. K. (2005). *Digits-a dataset for handwritten digit recognition*. Vienna (Austria): Austrian Research Institut for Artificial Intelligence Technical Report.

Shen, Y., Wen, Z., & Zhang, Y. (2014). Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods & Software*, *29*(2), 239–263.

Smith, L. I. (2002). A tutorial on principal components analysis. *Technical report*.

Stanczyk, U., Zielosko, B., & Jain, L. (2018). *Advances in feature selection for data and pattern recognition*. Springer.

Tao, H., Hou, C., Nie, F., Jiao, Y., & Yi, D. (2015). Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(4), 796–808.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *58*(1), 267–288.

Wen, J., Fang, X., Cui, J., Fei, L., Yan, K., Chen, Y., et al. (2018). Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(2), 390–403.

Wen, J., Xu, Y., Li, Z., Ma, Z., & Xu, Y. (2018). Inter-class sparsity based discriminative least square regression. *Neural Networks*, *102*, 36–47.

Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(11), 1738–1754.

Xu, J., Tang, B., He, H., & Man, H. (2016). Semisupervised feature selection based on relevance and redundancy criteria. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(9), 1974–1984.

Xue, Y., Zhang, L., Wang, B., Zhang, Z., & Li, F. (2018). Nonlinear feature selection using Gaussian kernel SVM-RFE for fault diagnosis. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*(10), 3306—3331.

Yang, J., Chu, D., Zhang, L., Xu, Y., & Yang, J. (2013). Sparse representation classifier steered discriminative projection with applications to face recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(7), 1023–1035.

Yang, J.-B., & Ong, C.-J. (2012). An effective feature selection method via mutual information estimation. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(6), 1550–1559.

Ye, J., & Xiong, T. (2006). Null space versus orthogonal linear discriminant analysis. In *Proceedings of the 23rd international conference on machine learning* (pp. 1073–1080). ACM.

Zhang, X., Chu, D., & Tan, R. C. (2015). Sparse uncorrelated linear discriminant analysis for undersampled problems. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(7), 1469–1485.

Zhang, A., & Gao, X. (2018). Supervised data-dependent kernel sparsity preserving projection for image recognition. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*(12), 4923–4936.

Zhang, Z., Xu, Y., Shao, L., & Yang, J. (2017). Discriminative block-diagonal representation learning for image recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(7), 3111–3125.

Zhou, Y., & Sun, S. (2016). Manifold partition discriminant analysis. *IEEE Transactions on Cybernetics*, *47*(4), 830–840.

Zhu, R., Dornaika, F., & Ruichek, Y. (2019a). Joint graph based embedding and feature weighting for image classification. *Pattern Recognition*.

Zhu, R., Dornaika, F., & Ruichek, Y. (2019b). Learning a discriminant graph-based embedding with feature selection for image categorization. *Neural Networks*, *111*, 35–46.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*. *Series B. Statistical Methodology*, *67*(2), 301–320.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *15*(2), 265–286.

CHAPTER 2

An Enhanced Approach to the Robust Discriminant Analysis and Class Sparsity Based Embedding

Neural Networks 136 (2021) 11–16

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Neural Networks Letter

An enhanced approach to the robust discriminant analysis and class sparsity based embedding

A. Khoder [b], F. Dornaika [a,b,c,*]

[a] Henan University, Kaifeng, China
[b] University of the Basque Country UPV/EHU, San Sebastian, Spain
[c] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

ARTICLE INFO

Article history:
Received 30 August 2020
Received in revised form 10 December 2020
Accepted 23 December 2020
Available online 30 December 2020

Keywords:
Supervised learning
Discriminant model
Feature extraction
Linear embedding
Inter-class sparsity
Image categorization

ABSTRACT

In recent times, feature extraction attracted much attention in machine learning and pattern recognition fields. This paper extends and improves a scheme for linear feature extraction that can be used in supervised multi-class classification problems. Inspired by recent frameworks for robust sparse LDA and Inter-class sparsity, we propose a unifying criterion able to retain the advantages of these two powerful linear discriminant methods. We introduce an iterative alternating minimization scheme in order to estimate the linear transformation and the orthogonal matrix. The linear transformation is efficiently updated via the steepest descent gradient technique. The proposed framework is generic in the sense that it allows the combination and tuning of other linear discriminant embedding methods. We used our proposed method to fine tune the linear solutions delivered by two recent linear methods: RSLDA and RDA_FSIS. Experiments have been conducted on public image datasets of different types including objects, faces, and digits. The proposed framework compared favorably with several competing methods.

1. Introduction

Achieving a good representation of high dimensional data was the focus of many researches. This can be carried out using different strategies. The most known ones are feature selection and feature extraction. A good data representation should also lead to better classification performance. Thus, Representation Learning (which includes feature extraction and selection) becomes a hot research topic (e.g., Langley (1994), Li, Liu, Yang, Zhou, and Lu (2013), Quinlan (2014), Raileanu and Stoffel (2004), Wang, Nie, and Huang (2015), Zang, Cheng, Wang, and Ma (2019), Zhao et al. (2015)). Feature extraction can be obtained by linear or nonlinear methods. Some of these methods have the ability to extract directly the targeted projection vectors from 2D image matrices while taking into consideration the inter-class and margin separability alongside with the intra-class compactness simultaneously. For instance, the two-dimensional maximum embedding difference (2D MED) (Wan, Li, Yang, Gai, & Jin, 2014) method proved to be efficient in feature extraction. Image data may be affected by many sorts of variations; namely: illumination conditions, poses, in addition to dealing with different facial expressions and others. In Wan et al. (2017), the authors addressed this problem. They proposed the "Local graph embedding based on maximum margin criterion via Fuzzy Set". This method has the ability of addressing and dealing with the above-mentioned problems. It proved to be very efficient when the data are affected by different types of variations.

Most of the methods focus on the estimation of a linear transformation that maps the original features to another space where latent variables can be obtained. For these methods, feature ranking or selection can be imposed by adding an $\ell_{2,1}$ norm constraint on the transformation matrix in the global criterion (e.g., Dornaika and Khoder (2020), Wen, Fang et al. (2018), Zhu, Dornaika, and Ruichek (2019)).

In this paper, we present a discriminant embedding method that retains the strengths of two recent discriminant methods, namely: (i) RSLDA (Wen, Fang et al., 2018) and (ii) ICS_DLSR (Wen, Xu, Li, Ma & Xu, 2018). The former promotes Linear Discriminant Analysis that implicitly performs feature weighting. The latter promotes inter-class sparsity which means that projected features of each class will share a common sparse structure. While the current work's goal is similar to that of our previous work (Dornaika & Khoder, 2020), the current proposed criterion as well as the deployed optimization are different.

The paper has the following main contributions. First, inspired by Dornaika and Khoder (2020), we provide a new simplified

[*] Corresponding author at: University of the Basque Country UPV/EHU, San Sebastian, Spain.
E-mail address: fadi.dornaika@ehu.eus (F. Dornaika).

https://doi.org/10.1016/j.neunet.2020.12.025
0893-6080/© 2020 Published by Elsevier Ltd.

objective function that allows the estimation of the linear transformation. It promotes class sparsity structure and robust discriminant analysis. Second, we provide an optimization algorithm in which the linear transformation is estimated by the gradient descent method. This has the advantage of providing more accurate solution than the closed-form one as will be demonstrated by the experiments. Although the main goal of the current work is to refine the solution provided by our recent "Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity" (RDA_FSIS) method, our proposed learning model can be used for refining the solution of many linear methods.

The main characteristics of the proposed model are as follows:

- The sought transformation encapsulates two different types of discrimination, namely: inter-class sparsity and robust LDA.
- The method could be adopted as a fine-tuning technique that can be used by many feature extraction methods.
- We used the gradient descent method to find a solution for the proposed criterion which guaranteed a better and less complex solution than the closed form one.

The paper is organized as follows. Section 2 presents some related researches and presents the main notations. Section 3 presents the proposed criterion as well as the associated optimization procedure. The experiments are described and presented in Section 4. Section 5 concludes the paper.

2. Related work and notations

2.1. Notations

We will refer for the training set by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$. d is the dimension of the data samples, N denotes the number of training samples, n_i denotes the number of samples corresponding to the ith class, and C is the number of classes. Every data sample $\mathbf{x}_i$ is represented by a column vector $\in \mathbb{R}^d$. $\mathbf{P}$ and $\mathbf{Q}$ denote the orthogonal and the desired projection matrix, respectively. $\mathbf{S}_b$ and $\mathbf{S}_w$ represent the between-class and within-class scatter matrices, given by $\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C} n_i\,(\mu_i - \mu)\,(\mu_i - \mu)^T$, and $\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\mathbf{x}_j^i - \mu_i)(\mathbf{x}_j^i - \mu_i)^T$ where μ, μ_i are the mean of all data samples and the mean of samples of the ith class, respectively.

The $\ell_{2,1}$ norm of a matrix $\mathbf{Z} \in \mathbb{R}^{d\times N}$ is given by $\|\mathbf{Z}\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{N} Z_{ij}^2}$. Its ℓ_1 norm is given by $\|\mathbf{Z}\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{N}|Z_{ij}|$. The ℓ_2 norm of a vector $\mathbf{z} = [z_1, z_2, \ldots, z_d]$ is obtained as follows $\|\mathbf{z}\|_2 = \sqrt{\sum_{i=1}^{d} z_i^2}$.

2.2. Related work

In recent times, researchers proposed many linear projection approaches. Some of these methods have integrated constraints that implement feature weighting/selection. Feature selection can efficiently discover the most relevant features of the data that describe the data in the best way and enhance discrimination (Stańczyk, Zielosko, & Jain, 2018; Xue, Zhang, Wang, Zhang, & Li, 2018; Yang & Ong, 2012).

The Linear Discriminant Analysis (LDA) method (Duda, Hart, & Stork, 2012) and its associated variants (e.g., Clemmensen, Hastie, Witten, and Ersbøll (2011), Zou and Hastie (2005), Zou, Hastie, and Tibshirani (2006)) are ones of the most used algorithms in the machine learning field. LDA estimates a transformation matrix in which the desired space minimizes the within-class variance and maximizes the between-class variance. Tao, Hou, Nie, Jiao, and Yi (2015) utilizes the $\ell_{2,1}$ norm of the LDA transformation.

Robust Sparse Linear Discriminant Analysis (RSLDA):

RSLDA (Wen, Fang et al., 2018) was proposed to tackle many limitations of the classical LDA (Tharwat, Gaber, Ibrahim, & Hassanien, 2017), RSLDA mainly adds the $\ell_{2,1}$ regularization of the projection matrix. This regularization ensures that the method performs feature ranking and weighting.

Inter Class Sparsity Least Square Regression:

In Wen, Xu et al. (2018), the authors propose the Inter-class sparsity based discriminative least square regression ICS_DLSR (Wen, Xu et al., 2018). The latter provides a linear mapping to the space of soft labels. It constructs a subspace in which the features obtained for each class have a common sparse structure.

Robust Discriminant Analysis with Feature Selection and Inter-class Sparsity (RDA_FSIS):

In Dornaika and Khoder (2020), we proposed a method that imposes two kinds of sparsity: the row sparsity of the linear transformation matrix, and the inter-class sparsity. The $\ell_{2,1}$ norm constraint was imposed on the corresponding matrices. The method also employed an orthogonal matrix whose role is to ensure that the projected features can preserve the main variance of the original data. Thus, it can improve the robustness to possible data noise. RDA_FSIS minimizes the following criterion:

$$f(\mathbf{Q}, \mathbf{E}, \mathbf{P}) = Tr\ \left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1\ \|\mathbf{Q}\|_{2,1} + \lambda_2\ \|\mathbf{E}\|_1 + \lambda_3 \sum_{i=1}^{C} \|\mathbf{Q}^T\mathbf{X}_i\|_{2,1}\ \ s.t.\ \ \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T\mathbf{X} + \mathbf{E} \qquad (1)$$

where $\mathbf{S} = \mathbf{S}_w - \mu\,\mathbf{S}_b$ denotes the LDA scatter matrices difference, $\mathbf{S}_w$ and $\mathbf{S}_b$ are the within-class and between-class matrices, respectively.$\mathbf{E}$ is an error matrix given by $\mathbf{E} = \mathbf{X} - \mathbf{P}\mathbf{Q}^T\mathbf{X}$. $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is the projection matrix and $\mathbf{P} \in \mathbb{R}^{d\times d}$ is the orthogonal matrix. μ is a constant used to balance the two scatter matrices. $\mathbf{X}_i$ represents the data matrix corresponding to the ith class. The optimization of (1) was carried out using the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). In each step, a closed-form solution was adopted for the linear transformation.

3. Proposed method

In this section, we will introduce our problem formulation and show the steps applied to find a solution to our problem. Our method is mainly considered as a linear projection method used for feature extraction and targeting a more discriminative transformation matrix. It is intended to enhance our previous method (RDA_FSIS) (Dornaika & Khoder, 2020). While the goal is similar to that of Dornaika and Khoder (2020), the current proposed criterion as well as the deployed optimization are different. In addition to the criterion, the current work has two main differences. First, the alternating method deployed in Dornaika and Khoder (2020) heavily utilizes closed-form solutions in its iterations for recovering the unknown linear transformation. However, in the current work, the linear transformation is updated using a gradient descent step. Second, the initialization used by the new proposed criterion utilizes an initial guess that is provided by either RSLDA or RDA_FSIS methods. Thus, two variants of the method are proposed.

Our proposed method has inherited feature ranking by exploiting the solution of RSLDA and RDA_FSIS methods as an initial guess for the sought transformation. This initial guess is then refined by a gradient descent method that is inserted in the ADMM. The latter aims to minimize the proposed criterion.

3.1. Formulation

We propose a method for the joint estimation of the projection matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$ and the orthogonal matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$. Since our goal is to perform a fine tuning of an available solution, we consider minimizing a simplified form of Eq. (1). Our proposed scheme minimizes the following criterion:

$$f(\mathbf{Q}, \mathbf{P}) = Tr\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} \|\mathbf{Q}^T\,\mathbf{X}_i\|_{2,1} + \lambda_2\,\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\|_2^2 \;\; s.t.\;\; \mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{2}$$

where $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$ is the data matrix associated with the ith class, n_i is the number of training samples belonging to the ith class, and C is the number of classes.

The first term in Eq. (2) is the LDA criterion where $\mathbf{S}$ represents the LDA scatter matrices difference. Thus, $\mathbf{S} = \mathbf{S}_w - \mu\,\mathbf{S}_b$ in which $\mathbf{S}_w$ is the within-class matrix and $\mathbf{S}_b$ is the between-class matrix. In our experiments, μ is set to 10^{-4}. The second term of the criterion is imposed to ensure that transformed features of the same class, in the projected space, obtain common sparse structure. In addition, the third term introduces a variant of Principal Component Analysis (PCA) constraint which ensures that original data would be recovered as well as possible. This is equivalent to the use of a reconstruction error term used in auto-encoders. This reconstruction term leads to a more relevant linear transformation, and hence a higher performance can be obtained. Our empirical results showed that the model obtained with this term was superior than the one obtained without it. λ_1 and λ_2 are two regularization parameters that control the significance of the different terms. The $\ell_{2,1}$ norm of a matrix $\mathbf{Z}$ can be given by:

$$\|\mathbf{Z}\|_{2,1} = Tr\left(\mathbf{Z}^T\,\mathbf{D}\,\mathbf{Z}\right) \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix that is given by:

$$\mathbf{D} = \begin{pmatrix} \frac{1}{\|\mathbf{z}(1)\|_2+\epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\|\mathbf{z}(d)\|_2+\epsilon} \end{pmatrix} \tag{4}$$

$\mathbf{Z}\,(j)$ represents the jth row of $\mathbf{Z}$, and ϵ is a small positive scalar.

By substituting the second term of the criterion by its trace form, problem (2) becomes:

$$f(\mathbf{Q}, \mathbf{P}) = Tr\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} Tr\,\left(\,(\mathbf{Q}^T\mathbf{X}_i)^T\;\mathbf{D}_i\;\mathbf{Q}^T\mathbf{X}_i\right) + \lambda_2\,\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\|_2^2 \tag{5}$$

$$\min_{\mathbf{Q},\mathbf{P}} f\,(\mathbf{Q}, \mathbf{P}) \quad s.t. \quad \mathbf{P}^T\,\mathbf{P} = \mathbf{I}$$

Eq. (5) presents the criterion of the proposed method. Thus, by looking for the minimum of this criterion, we are targeting a transformation matrix which jointly ensures: (i) class discrimination using Linear Discriminant Analysis (LDA), (ii) class-wise common sparsity, and (iii) energy preserving property of PCA. To find a solution for the proposed method, we used the alternating minimization method since we have two unknown matrices. The step that updates the linear transformation $\mathbf{Q}$ will deploy a gradient descent step. This deployment has two advantages: (1) It has a lower computational complexity compared to other methods. (2) It leads to accurate solutions. In case of dealing with small-sized datasets, where computing a costly matrix inversion is not targeted, closed-form approaches can be a good option for obtaining the solution of a minimization problem. When working with medium- to very large-sized datasets, the Gradient Descent approach is preferred. Furthermore, in such approaches, the unknowns are updated incrementally and smoothly at each iteration, which leads to more accurate solutions.

3.2. Solution steps to the proposed method

To solve the formulated problem, we have adopted the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). We calculated each variable while other variables are fixed. We proceed as follows:

- **Calculate the orthogonal matrix P:** This matrix can be calculated by fixing the matrix $\mathbf{Q}$ and solving the following problem:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \tag{6}$$

Using $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ and the fact that the squared norm of matrix $\mathbf{A}$ is given by $\|\mathbf{A}\|_2^2 = Tr(\mathbf{A}^T\mathbf{A}) = Tr(\mathbf{A}\,\mathbf{A}^T)$, problem (6) is equivalent to the following maximization problem:

$$\min_{\mathbf{P}^T\,\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \quad \longrightarrow \quad \max_{\mathbf{P}^T\,\mathbf{P}=\mathbf{I}} Tr\,(\mathbf{P}^T\,\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}) \tag{7}$$

We can find a solution for problem (7) by performing the singular value decomposition (SVD) of $\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}$. Suppose that the SVD factorization is given by $SVD\,(\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}) = \mathbf{U}\,\Sigma\,\mathbf{V}^T$. Then $\mathbf{P}$ is obtained by Zou et al. (2006):

$$\mathbf{P} = \mathbf{U}\,\mathbf{V}^T \tag{8}$$

- **Calculate the Projection matrix Q:** We have adopted a gradient descent scheme to calculate $\mathbf{Q}$ in each iteration of the proposed method. The orthogonal matrix $\mathbf{P}$ is fixed. We consider the trace form of the resulting criterion:

$$f(\mathbf{Q}, \mathbf{P}) = Tr\,\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} Tr\,(\,\mathbf{X}_i^T\mathbf{Q}\,\mathbf{D}_i\,\mathbf{Q}^T\mathbf{X}_i) + \lambda_2\,\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\|_2^2$$

We calculate the gradient of the objective function w.r.t. $\mathbf{Q}$ as follows:

$$\mathbf{G} = \frac{\partial\,f}{\partial\,\mathbf{Q}} = 2\,\mathbf{S}\,\mathbf{Q} + \lambda_1 \sum_{i=1}^{C} 2\,\mathbf{X}_i\,\mathbf{X}_i^T\,\mathbf{Q}\,\mathbf{D}_i + 2\lambda_2\,[\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q} - \mathbf{X}\,\mathbf{X}^T\,\mathbf{P}] \tag{9}$$

Using the computed gradient matrix, we can update $\mathbf{Q}$ by:

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \alpha\,\mathbf{G} \tag{10}$$

where $\mathbf{Q}_{t+1}$ and $\mathbf{Q}_t$ denote the projection matrix $\mathbf{Q}$ in iteration $t+1$ and iteration t respectively. α is the step length (learning rate).

- **Update the matrices $\mathbf{D}_i$**: We update $\mathbf{D}_i$, $(i = 1, \ldots, C)$ by:

$$\mathbf{D}_i = \begin{pmatrix} \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_i(1)\right\|_2+\epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_i(d)\right\|_2+\epsilon} \end{pmatrix} \tag{11}$$

where ϵ is a small positive scalar and $\mathbf{Q}^T\mathbf{X}_i\,(j)$ represents the jth row vector of $\mathbf{Q}^T\mathbf{X}_i$.

Algorithm 1 summarizes the proposed scheme and presents the main stages for optimizing problem (2).

Algorithm 1. Enhanced Discriminant Analysis with Class Sparsity using Gradient Method **RDA_GD**, **EDA_CS**

```
Input:           1. Data samples $\mathbf{X} \in \mathbb{R}^{d\times N}$
                 2. Labels of the training samples
                 3. The step length of the gradient descent $\alpha$
                 4. Parameters $\lambda_1$, $\lambda_2$
Output:          $\mathbf{P}$, $\mathbf{Q}$

Initialization:  $\mathbf{Q}^{(0)}$ obtained from RSLDA or RDA_FSIS (see Section 3.3).

Process:         set $t = 0$ and $\mathbf{Q} = \mathbf{Q}^{(0)}$

                 Repeat
                     Fix $\mathbf{Q}$, update $\mathbf{P}^{(t+1)}$ using Eq. (8).
                     Calculate the gradient matrix $\mathbf{G}$ using Eq. (9)
                     Fix $\mathbf{P}$, update $\mathbf{Q}^{(t+1)}$ using Eq. (10).
                     Update $\mathbf{D}_i$ using Eq. (11)
                     set $t = t + 1$
                 Until convergence
```

Once the transformation matrix, $\mathbf{Q}$, is estimated by **Algorithm 1**, the training and test samples are projected to the new subspace. This is carried out by $\mathbf{z}_{train} = \mathbf{Q}^T\,\mathbf{x}_{train}$ and $\mathbf{z}_{test} = \mathbf{Q}^T\,\mathbf{x}_{test}$ where $\mathbf{x}_{train}$ is a training data sample, and $\mathbf{x}_{test}$ is a test data sample.

3.3. Initialization of projection matrix Q

The linear transformation $\mathbf{Q}$ requires a good initial guess since it is estimated by a gradient descent scheme. We used two initialization procedures leading to two variants of the proposed algorithm. The first variant is called Robust Discriminant Analysis using Gradient Descent **RDA_GD**. In this variant, the initial guess $\mathbf{Q}^{(0)}$ for the linear transformation matrix $\mathbf{Q}$ is set to the solution of the RSLDA (Wen, Fang et al., 2018) method (solved using its own ADMM optimization). This initial transformation inherits the feature ranking delivered by RSLDA. The second variant, denoted as Enhanced Discriminant Analysis with Class Sparsity **EDA_CS**, sets the initial guess $\mathbf{Q}^{(0)}$ to the solution provided by RDA_FSIS (Dornaika & Khoder, 2020). This variant inherits the feature ranking and inter-class sparsity advantages exploited by RDA_FSIS method.

3.4. Computational complexity

This section is intended to analyze the computational complexity of the proposed method (see **Algorithm 1**).

Cost of Algorithm 1: This algorithm iteratively estimates the matrices $\mathbf{Q}$ and $\mathbf{P}$. The orthogonal matrix $\mathbf{P}$ requires a singular value decomposition. The computational cost of an SVD decomposition of a $d \times d$ matrix is $\mathcal{O}\left(d^3\right)$. $\mathbf{Q}$ is calculated in the second step of Algorithm 1. This step requires the calculation of the corresponding gradient matrix. Since these two steps consist of simple matrix operations, they have small computational cost and thus can be ignored. Also the step intended to update $\mathbf{D}_i$ (Eq. (11)) is a simple matrix operation that has a very small cost and can also be ignored. Hereby, the total computational cost of Algorithm 1 is $\mathcal{O}\left(\tau'(d^3)\right)$ where τ' denotes the number of iterations of Algorithm 1.

Cost of RDA_GD: In the first variant of our proposed method, we used the RSLDA (Wen, Fang et al., 2018) method for the initialization of the transformation matrix $\mathbf{Q}$ before feeding to our algorithm. Thus, the complexity of RSLDA method should be added to the complexity of Algorithm 1. Let τ represent the number of iterations of RSLDA. The latter has a complexity of $\mathcal{O}\left(\tau(d^2N + 4d^3)\right)$. In summary, the overall cost of the first proposed variant (RDA_GD) would be the sum of the RSLDA cost and the cost of Algorithm 1 which is equal to $\mathcal{O}\left(\tau(d^2N + 4d^3)\right)$ + $\mathcal{O}\left(\tau'(d^3)\right)$ where τ' denotes the number of iterations of Algorithm 1.

Cost of EDA_CS: For the second proposed variant EDA_CS, we have constructed the initial guess of the transformation matrix from the RDA_FSIS (Dornaika & Khoder, 2020) method. Knowing that the latter has a complexity of $\mathcal{O}(\tau\,(2d^3))$, the second proposed variant has an overall complexity of $\mathcal{O}\left(\tau\,(2d^3)\right)$ + $\mathcal{O}\left(\tau'(d^3)\right)$ where τ and τ' represent the number of iterations of RDA_FSIS and Algorithm 1 respectively.

4. Performance evaluation

To test the two proposed variants, we have conducted experiments on several datasets including faces, objects and handwritten datasets. In our work we have used the following six public datasets in addition to a large-scale dataset: **USPS**[1] digits dataset, **Honda**[2] dataset, **COIL20**[3] object dataset, **Extended Yale B**[4] face dataset, **FEI**[5] dataset, **Georgia**,[6] and the large scale **MNIST** dataset consisting of 60,000 images. Details about these datasets are summarized in Table 1.

In this section, we will present the classification performance when the projected spaces are obtained by the proposed scheme and the competing methods. The proposed method has two variants: **RDA_GD** and **EDA_CS** (see above text).

The two proposed variants were compared with the following approaches: K-nearest neighbors (KNN) (Kozma, 2008), Support Vector Machines (SVM) (Chang & Lin, 2011) (the Linear SVM was implemented using the LIBSVM library[7]), Linear Discriminant Analysis (LDA) (Tharwat et al., 2017), PCE (Peng, Lu, Yi, & Yan, 2016) (unsupervised method), ICS_DLSR (Wen, Xu et al., 2018), Robust sparse LDA (RSLDA) (Wen, Fang et al., 2018), Joint Robust Discriminant Analysis and Inter-class Sparsity (RDA_FSIS) (Dornaika & Khoder, 2020) and Local Discriminant Embedding (LDE) (Chen, Chang, & Liu, 2005). To construct the graph required by the LDE method, three main parameters are required. These are the number of homogeneous neighbors (K_1), the number of heterogeneous neighbors (K_2), and the regularization parameter in the LDE criterion denoted as β. In our experiments, we have adopted the values of 3, 5 and 0.02 for K_1, K_2, and β, respectively.

The proposed method alongside with the compared ones were tested under the same conditions in order to guarantee a fair comparison. Datasets were randomly split into a training part and a test part. First, for each compared embedding method, a transformation matrix was estimated from the training part. Then, training and test data were projected onto the new space using the computed transformation. Finally, the Nearest Neighbor classifier (NN) (Cunningham & Delany, 2007) was used to classify the test data, the value of K was set to 1 (K=1). Different percentages of training were tried. Moreover, for a given percentage of training data, the whole evaluation was repeated ten times. Indeed, we adopted ten random splits for every configuration and reported the average recognition rate (rate of correct classification of test part) over these ten random splits. We used PCA as a preprocessing technique with an energy preservation rate of 100%. The value for α was chosen from the set $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}\}$.

[1] https://www.kaggle.com/bistaumanga/usps-dataset.

[2] http://vision.ucsd.edu/~leekc/HondaUCSDVideoDatabase/HondaUCSD.html.

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[4] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.

[5] https://fei.edu.br/~cet/facedatabase.html.

[6] http://www.anefian.com/research/face_reco.htm.

[7] https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Table 1
Brief datasets description.

Dataset	Type	# Samples	# Features	# Classes	Descriptor
USPS	Digits	1100	256	10	RAW-brightness images
Honda	Face	2277	1024	22	RAW-brightness images
COIL20	Object	1440	177	20	Local Binary Patterns
Extended Yale B	Face	2414	1024	38	RAW-brightness images
FEI	Face	700	1024	50	RAW-brightness images
Georgia	Face	750	1024	50	RAW-brightness images
MNIST	Digits	60,000	2048	10	Deep features (ResNet-50)

Table 2
Mean classification accuracies (%) of different methods on the tested datasets.

Dataset	Method										
	Train./class	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	RDA_FSIS	**RDA_GD**	**EDA_CS**
USPS	30	87.01±1.5	88.21±1.2	84.91±1.7	83.54±1.3	72.01±1.1	88.46±0.8	89.45±1.2	90.05±0.8	89.50±1.2	**90.40**±0.8
	40	88.56±1.6	90.40±0.9	86.19±0.9	85.3±1.2	72.30±1.7	90.16±0.7	91.11±1.0	91.27±0.9	**91.81**±1.1	91.76±0.5
	55	90.51±1.4	92.09±0.8	88.64±1.0	87.16±1.7	73.32±2.2	91.25±1.2	92.65±1.1	92.56±1.2	93.07±1.0	**93.40**±1.0
	65	91.76±1.3	93.16±0.9	89.29±1.5	88.58±1.1	74.11±1.9	91.53±1.3	92.89±1.0	93.33±1.0	93.71±0.9	**93.73**±0.6
Honda	10	64.12±2.1	71.32±2.1	65.95±2.2	65.74±2.2	61.86±2.2	70.79±2.5	69.90±2.1	72.48±2.0	70.16±1.9	**72.73**±2.0
	20	77.69±1.2	83.60±1.0	79.39±1.4	79.25±1.7	75.33±1.4	82.95±1.2	83.03±1.3	84.19±1.4	83.60±1.2	**84.40**±1.4
	30	84.78±1.3	89.09±1.0	85.84±1.1	86.24±1.1	82.55±1.8	88.20±1.0	89.04±1.2	89.44±1.0	89.41±1.1	**89.66**±1.1
	50	91.36±0.9	94.15±1.2	92.28±1.1	92.34±0.8	90.03±0.7	93.53±0.6	94.13±0.8	**94.54**±1.0	94.53±0.8	94.45±0.9
FEI	5	88.98±2.5	91.18±2.3	92.60±3.6	90.67±2.6	86.04±3.2	92.16±2.7	93.19±2.5	94.01±2.3	93.81±2.6	**94.24**±2.7
	6	90.35±2.7	92.93±2.8	94.18±3.9	92.15±2.7	88.73±3.7	93.65±2.7	94.25±2.3	94.63±2.3	94.75±2.5	**94.80**±1.9
	7	92.60±3.6	94.31±2.5	95.60±3.5	94.26±3.0	91.09±4.2	95.20±2.2	95.66±1.5	96.09±1.5	96.20±1.5	**96.26**±1.8
	8	94.27±2.9	95.23±2.2	96.03±3.5	95.57±2.4	93.20±4.4	96.17±1.9	96.43±1.6	96.67±1.7	**96.97**±1.7	96.87±2.0
COIL20	20	94.58±0.9	97.65±1.3	96.19±0.8	95.00±0.7	94.87±1.6	98.04±0.5	96.73±0.6	97.85±0.6	96.89±0.6	**98.05**±0.6
	25	95.79±0.8	98.22±0.7	97.07±0.8	96.12±0.7	95.99±1.3	98.22±0.6	97.74±0.7	98.60±0.5	97.89±0.5	**98.74**±0.5
	30	96.65±0.6	98.70±0.8	97.81±0.5	97.01±0.6	97.49±0.7	98.75±0.1	98.26±0.7	99.10±0.4	98.52±0.6	**99.15**±0.5
	35	97.14±0.7	98.81±0.8	98.15±0.3	97.42±0.6	98.11±0.6	99.12±0.4	98.68±0.6	99.36±0.4	98.80±0.6	**99.55**±0.2
Georgia	3	52.57±1.4	56.22±2.3	48.18±2.8	52.77±2.3	46.43±2.3	59.73±2.1	62.32±2.2	62.67±2.0	62.35±2.2	**63.05**±1.6
	5	61.28±1.5	66.98±1.9	59.20±1.9	62.14±1.6	56.18±1.9	71.12±1.3	73.48±1.6	74.28±1.1	73.54±1.5	**74.68**±1.2
	7	66.73±1.5	72.83±1.2	67.83±2.4	67.10±2.0	62.15±1.8	78.38±1.4	78.82±1.1	79.98±1.7	79.42±1.7	**80.30**±1.3
	9	71.40±1.0	77.53±2.0	72.57±3.0	72.13±2.3	66.37±2.9	82.57±2.1	82.77±2.2	83.30±2.1	82.80±2.2	**83.33**±2.1
Extended Yale B	10	69.80±4.5	73.85±5.6	82.32±5.1	79.92±4.3	86.39±3.1	86.56±4.5	86.79±4.8	88.27±4.5	87.10±4.4	**88.59**±4.1
	15	75.20±4.5	80.02±4.6	86.76±4.7	83.77±4.9	89.23±3.4	89.53±3.8	89.93±3.8	91.73±3.6	90.04±3.8	**91.89**±3.6
	20	80.24±2.5	85.79±2.8	90.70±2.4	88.44±2.2	92.19±1.4	93.14±2.2	93.59±2.5	95.11±1.8	93.75±2.5	**95.22**±1.8
	25	82.24±3.3	89.03±1.5	92.17±1.3	90.43±2.1	93.35±1.0	94.50±1.1	94.92±1.2	96.23±0.8	95.02±1.2	**96.33**±0.7
MNIST	1000	91.75	97.58	85.74	93.22	93.77	98.02	97.95	98.25	98.21	**98.30**

The results are summarized in Table 2. This table depicts the classification rates as well as the standard deviations of the two proposed variants and the competing methods using the USPS, Honda, FEI, COIL20, Georgia and Extended Yale B datasets. The results are obtained using different training and testing percentages from the data and over 10 random splits. The last row in Table 2 illustrates the classification accuracy using the large scale MNIST dataset (60,000 images), results for the MNIST dataset are obtained using a **single split** when using 1000 samples from each class for training. Fig. 1 illustrates the obtained confusion matrix associated with the test part of the MNIST dataset using the second variant of our proposed method **EDA_CS**, this figure shows the distribution of the predicted samples over different classes.

Analysis of results: The first proposed variant of our method has slightly outperformed the RSLDA method. This is realistic since the first proposed method mainly provides a fine-tuning of the RSLDA transformation. The second variant outperformed the first variant and other competing methods, and gave the best performance in classification. This is because this variant was initialized through the solution of RDA_FSIS method and hence inherited its advantages. We have noticed that satisfactory performance can be achieved while choosing the values of λ_1 and λ_2 from the sets $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$ accordingly. Generally, a value of 0.1 for both λ_1 and λ_2 seems to be a good choice for the two variants.

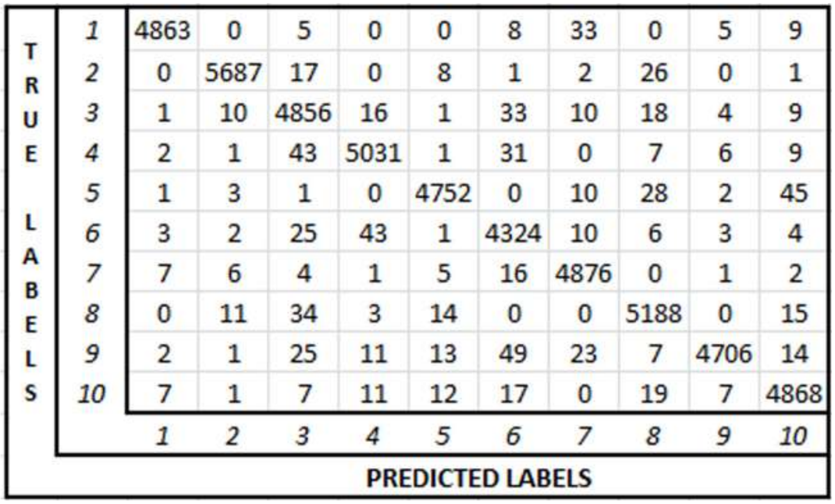
5. Conclusion

We introduced a novel criterion in order to obtain a discriminant linear transformation. Our proposed approach differs from competing methods where both the criterion and the optimization were different. We used the gradient descent method to find a solution for the proposed criterion which guaranteed a better and enhanced solution than the closed form one used by most of the stated competing methods.

Two different types of discrimination were imposed; namely, (i) inter-class sparsity and (ii) robust LDA. We deployed an iterative alternating minimization scheme to estimate the linear transformation and the orthogonal matrix associated with the robust LDA. The linear transformation was efficiently updated via the steepest descent gradient technique. We proposed two initialization schemes for the linear transformation. Our proposed method has been able to achieve higher performance and contributed in delivering a more discriminant transformation than the existing competing methods. Our method's strong point is that it could be used as a fine-tuning technique that is usable by many other feature extraction methods. The general aspect comes from the fact that our proposal is a gradient descent based refinement which is applied on a closed-form solution. The proposed framework can retain the advantages of multiple methods and lead to enhanced performances. Experiments conducted on several public image datasets have shown that the proposed scheme can outperform many discriminant methods as well as the iterative optimization based on closed-form solutions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



TRUE LABELS

	1	2	3	4	5	6	7	8	9	10
1	4863	0	5	0	0	8	33	0	5	9
2	0	5687	17	0	8	1	2	26	0	1
3	1	10	4856	16	1	33	10	18	4	9
4	2	1	43	5031	1	31	0	7	6	9
5	1	3	1	0	4752	0	10	28	2	45
6	3	2	25	43	1	4324	10	6	3	4
7	7	6	4	1	5	16	4876	0	1	2
8	0	11	34	3	14	0	0	5188	0	15
9	2	1	25	11	13	49	23	7	4706	14
10	7	1	7	11	12	17	0	19	7	4868

PREDICTED LABELS

Fig. 1. Confusion Matrix for the MNIST dataset.

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, *3*(1), 1–122.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and technology (TIST)*, *2*(3), 27.

Chen, H.-T., Chang, H.-W., & Liu, T.-L. (2005). Local discriminant embedding and its variants. In *2005 IEEE computer society conference on computer vision and pattern recognition: Vol. 2* (pp. 846–853). IEEE.

Clemmensen, L., Hastie, T., Witten, D., & Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, *53*(4), 406–413.

Cunningham, P., & Delany, S. J. (2007). K-nearest neighbour classifiers. *Multiple Classifier Systems*, *34*(8), 1–17.

Dornaika, F., & Khoder, A. (2020). Linear embedding by joint robust discriminant analysis and inter-class sparsity. *Neural Networks*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Kozma, L. (2008). K nearest neighbors algorithm (kNN). *Helsinki University of Technology*.

Langley, P. (1994). *Selection of relevant features in machine learning: Defense Technical Information Center*.

Li, Z., Liu, J., Yang, Y., Zhou, X., & Lu, H. (2013). Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, *26*(9), 2138–2150.

Peng, X., Lu, J., Yi, Z., & Yan, R. (2016). Automatic subspace learning via principal coefficients embedding. *IEEE Transactions on Cybernetics*, *47*(11), 3583–3596.

Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, *41*(1), 77–93.

Stańczyk, U., Zielosko, B., & Jain, L. C. (2018). Advances in feature selection for data and pattern recognition: An introduction. In *Advances in feature selection for data and pattern recognition* (pp. 1–9). Springer.

Tao, H., Hou, C., Nie, F., Jiao, Y., & Yi, D. (2015). Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(4), 796–808.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190.

Wan, M., Lai, Z., Yang, G., Yang, Z., Zhang, F., & Zheng, H. (2017). Local graph embedding based on maximum margin criterion via fuzzy set. *Fuzzy Sets and Systems*, *318*, 120–131.

Wan, M., Li, M., Yang, G., Gai, S., & Jin, Z. (2014). Feature extraction using two-dimensional maximum embedding difference. *Information Sciences*, *274*, 55–69.

Wang, D., Nie, F., & Huang, H. (2015). Feature selection via global redundancy minimization. *IEEE Transactions on Knowledge and Data Engineering*, *27*(10), 2743–2755.

Wen, J., Fang, X., Cui, J., Fei, L., Yan, K., Chen, Y., & Xu, Y. (2018). Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(2), 390–403.

Wen, J., Xu, Y., Li, Z., Ma, Z., & Xu, Y. (2018). Inter-class sparsity based discriminative least square regression. *Neural Networks*, *102*, 36–47.

Xue, Y., Zhang, L., Wang, B., Zhang, Z., & Li, F. (2018). Nonlinear feature selection using Gaussian kernel SVM-RFE for fault diagnosis. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*(10), 3306–3331.

Yang, J.-B., & Ong, C.-J. (2012). An effective feature selection method via mutual information estimation. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(6), 1550–1559.

Zang, S., Cheng, Y., Wang, X., & Ma, J. (2019). Semi-supervised flexible joint distribution adaptation. In *Proceedings of the 2019 8th international conference on networks, communication and computing* (pp. 19–27).

Zhao, Z., He, X., Cai, D., Zhang, L., Ng, W., & Zhuang, Y. (2015). Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, *28*(3), 689–700.

Zhu, R., Dornaika, F., & Ruichek, Y. (2019). Joint graph based embedding and feature weighting for image classification. *Pattern Recognition*, *93*, 458–469.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *15*(2), 265–286.

CHAPTER 3

A Hybrid Discriminant Embedding with Feature Selection: Application to Image Categorization

Applied Intelligence
https://doi.org/10.1007/s10489-020-02009-3

A hybrid discriminant embedding with feature selection: application to image categorization

A. Khoder[1] · F. Dornaika[1,2]

Accepted: 7 October 2020

Abstract

In recent times, feature extraction was the focus of many researches due to its usefulness in the machine learning and pattern recognition fields. Feature extraction mainly aims to extract informative representations from the original set of features. This can be carried out using various ways. The proposed method is targeting a hybrid linear feature extraction scheme for supervised multi-class classification problems. Inspired by recent robust sparse LDA and Inter-class sparsity frameworks, we will propose a unifying criterion that is able to retain these two powerful linear discriminant method's advantages. Thus, the obtained transformation encapsulates two different types of discrimination, the inter-class sparsity and robust Linear Discriminant Analysis with feature selection. We will introduce an iterative alternating minimization scheme in order to estimate the linear transform and the orthogonal matrix. The linear transform is efficiently updated via the steepest descent gradient technique. We will also introduce two initialization schemes for the linear transform. The proposed framework is generic in the sense that it allows the combination and tuning of other linear discriminant embedding methods. According to the experiments which have been carried out on several datasets including faces, objects and digits, the proposed method was able to outperform the competing methods in most cases.

Keywords Supervised learning · discriminant analysis · feature extraction · linear embedding · class sparsity · dimensionality reduction · image classification

1 Introduction

Different data types in various fields like images, videos, gaming and others are represented through a large number of features. Achieving a good representation of these data was thus the focus of many researchers. Deriving a representation can be carried out using different strategies, the most known of which being feature extraction.

Discovering the most relevant and informative features is very important. It can reduce the storage and computing requirements. More importantly, good data representation will lead to better classification performance. This explains why Representation Learning became a hot research topic (e.g., [23, 24, 32, 33, 45, 57, 61]). Feature extraction can be obtained via linear or nonlinear methods. Most feature extraction methods focus on the estimation of a linear transformation that maps the original features to another space where latent variables can be obtained.

A feature can be identified as one of the following: relevant, irrelevant or redundant. A feature is called irrelevant when it does not contribute to the predictive model's enhancement, in other words, it can sometimes worsen the classification accuracy when taken in consideration during the classification process. Relevant features contribute in achieving a more predictive model hence leading to a higher classification accuracy. Those are the ones that the model aims to extract and select among all others. A redundant feature does not lead the model to perform better in the classification process.

Many methods were proposed earlier in the purpose of extracting image features; these methods are referred to as image retrieval methods. An example of a novel Kernel based retrieval method is the method proposed in [40], where the authors proposed a novel approach to extract visual and textual features, and fuse them using a kernel based method. Using the same method, visual features were

✉ F. Dornaika
fdornaika@gmail.com

[1] University of the Basque Country UPV/EHU, San Sebastian, Spain

[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Published online: 09 November 2020

extracted using the SURF descriptor, then the embedded text within the images was detected using the Maximally Stable Extremal Region (MSER) [28] algorithm and by applying step filters. Another method was proposed based on multi-factors correlation [46]. In this method, three correlations have been used to extract the image feature, structure element correlation (SEC), gradient value correlation (GVC) and gradient direction correlation (GDC). Moreover, another text based imaged retrieval method was proposed, in [41], where the authors proposed a novel approach to detect the text in an image and exploit it as keywords and tags for automatic text-based image retrieval. The Method proposed by [39] have also drawn attention.

Another proposed method is the tensor local discriminant embedding (TLDE) [16] used for the Hyperspectral image (HSI)'s classification [15]. TLDE takes advantage of the spatial structure and spectral information. It maps a high dimensional space into a low dimensional space by three projection matrices. It can ensure a good data discrimination. However, its main limitation is that (HSI)'s classification is a small sample problem. Authors in [20] proposed a supervised method called double discriminant embedding (DDE), which uses two transformations for extracting features from the data. (DDE) performs very well using limited training samples.

Many linear techniques have been used in the pattern recognition community (e.g., LDA, LDE and LSR). All these techniques aim to obtain a discriminative projection space. The least square regression (LSR) method proved to be effective in the pattern recognition field [52]. This method's objective is to connect source and target data with minimal error. LSR frameworks are known to be very flexible; they allow the introduction of new regularization terms. One example of an LSR-based method is the Linear Regression (LR) which demonstrated a very good classification performance, as well as a good flexibility. However, LR-based methods are prone to have some issues [56], the most famous of which being that the (LR)-Based method's label matrices are too strict and inappropriate for classification. (LR)-based methods also ignore the relationship between samples. In order to solve this problem, authors in [50] proposed the discriminative LSR (DLSR) where a more relaxed label matrix was introduced instead of the strict (zero-one) label matrix. DLSR's performance is superior to that of LSR. However, the introduction of constraints that target the label matrix's relaxation have enlarged the distances of the regression responses between samples belonging to the same class. In order to fix that problem, authors in [48] have proposed ICS_DLSR, solving the addressed problem by introducing an inter-class sparsity constraint in the criterion.

Many extensions to the principal component analysis (PCA) have also been proposed, namely the locality preserving projections (LPPs) [18] and the neighborhood preserving embedding (NPE) [17]. The stated methods were proposed to solve the principal component analysis problem, namely being sensitive to outliers. Sparse coding or Representation extraction methods [49, 54, 58], and low-rank representation (LRR) [4, 60] have also performed well at pattern recognition. LRR works on the data's global structure but overlooks local structure. This issue was tackled by proposing latent low-rank representation (LatLRR) [25] where low-rank matrices were proposed to recover the data's space information. Despite its good discrimination ability, LatLRR is restricted by fixed feature dimensionality. The problem facing LatLRR was addressed with approximate low-rank projection learning (ALPL) [13]. Authors in [26] further proposed a low-rank 2-D preserving projection method which is more robust to noise and can reduce the computation complexity. It is true that all of the above mentioned methods provide good discrimination; however, none of them took advantage of both class-shared and class-specific information which limit their performance, a matter which was addressed by the authors in [1]. DSDPL serves to decompose original high dimensional training data via learned projection matrices into class-shared and class-specific subspaces. DSDPL ensured more freedom to capture the data's main energy which reduces information loss and provides better reconstruction properties. It is known that LDA can suffer from the small sample size (SSS) problem. Many LDA-based techniques were proposed to overcome this problem, namely: OLDA, ULDA and many others. Another issue is that LDA fail to deal with non-Gaussian distribution data. Sparse LDA (SLDA) [31] was proposed to overcome the issue of redundant features' presence in the data. SLDA imposed the sparse constraint and was able to learn a sparse discriminant space. It is true that SLDA performed well at most of the classification tasks, but it still lacks the ability to implicitly perform feature selection. This was addressed in the proposed RSLDA method [47] where the authors imposed the $\ell_{2,1}$ norm over the sought transformation matrix to ensure that their method performs feature selection. ℓ_1 norm is aso included in the purpose of dealing with the sparse noise.

Another method that has imposed the $\ell_{2,1}$ norm regularization over the sought linear transformation (for feature ranking) was the method proposed in [64]. The authors proposed a nonlinear method (FDEFS) and tested it for semi-supervised learning. It incorporated the Manifold Smoothness, Margin Discriminant Embedding and the Sparse Regression for feature selection. Nowadays, researches focus on deploying linear projection models that simultaneously perform feature extraction and ranking [47, 63].

Solving the optimization problems proposed in these methods could be implemented using different strategies.

One of the many used techniques is the gradient descent algorithm, which has been used for a long time in the optimization field and showed very good characteristics in solving unconstrained optimization problems due to its simplicity and low complexity. Variants for the gradient technique were proposed and referred to as ”Adaptive Gradient techniques (AGT) which are very good at dealing with sparse data.

Authors in [9] have founded that Adaptive gradients improve the data's robustness. Although using (AGT) eliminates the need to manually tune the learning rate, these algorithms were shown to have a few weaknesses. They will get to a point where they are no longer able to acquire additional knowledge due to the squared gradients' accumulation in the denominator. Since every added term is positive, the accumulated sum keeps growing during training. This in turn causes the learning rate to shrink and eventually become very small.

In this paper, we will present a unified and hybrid discriminant embedding method that minimizes the loss of discriminative information. The proposed method differs from the existing related methods at many levels in terms of the criterion design, optimization technique and initialization process. As for criterion design, the proposed method integrates LDA and a variant of PCA into a joint learning framework. It inherits LDA's excellent discriminative capability while at the same time allows the reconstruction of original data with minimal information loss. ICS_DLSR was the first method to integrate the inter-class sparsity constraint into LSR. In our proposed method, we have integrated the inter-class sparsity constraint into an LDA framework which pursued the transformed samples belonging to same classes to have the same row-sparsity structure. Our proposed method has many advantages due to its hybrid initialization capability. The proposed method differs from other methods as it can inherit many existing methods' advantages through its initialization process. Our framework is generic in the sense that it allows for the combination and tuning of other linear discriminant embedding methods, which allows the method to automatically inherit these methods' advantages. Unlike most of the other methods, we have used the gradient descent algorithm to find the solution to our proposed criterion rather than the closed-form solution used in ICS_DLSR and RSLDA for example. The gradient algorithm offers faster, less complex and more accurate solutions than the closed form solutions. Moreover, the proposed linear transformation is generic and can be used by many types of objects (signals, images and texts) and many types of descriptors (including both regular and stable image features). In our work, we have used and tested different types of image descriptors. Image raw brightness, Local Binary patterns and Deep features (provided by deep Convolutional Neural Networks) were used as image descriptors for the tested datasets. In addition to working with regular image features, many other recent works are focusing on working with stable image features. Image moments are a kind of stable image feature that provides a generic representation of objects with simple or complex shapes. Moments are often described by their robustness to noise and their good rotational invariant stability. An example of an image-moment based method that proved to be very efficient is the method proposed in [43] where the authors proposed Polar Harmonic Fourier Moments (PHMs). PHMs proved to be numerically stable and their RBF to be noticeably simpler that of other methods. Authors in [44] proposed the “Ternary Radial Harmonic Fourier moments based robust stereo image zero-watermarking algorithm” (TRHFM) in order to enhance the copyright protection of stereo images that are known to be easy to copy and modify. To be able to work with color images, authors in [42] proposed the Quaternion Polar Harmonic Fourier Moments (QPHFM), a method that proved to have the best image reconstruction performance and performed excellently in both noise-free and noisy conditions.

The proposed method retains the strengths of two recent discriminant methods: (i) RSLDA [47] and (ii) ICS_DLSR [48]. The former promotes Linear Discriminant Analysis with implicit feature selection, while the latter promotes inter-class sparsity, which means that projected features have a common sparse structure for the features in each class.

The main contributions are thus as follows. First, we will provide a novel objective function that allows the estimation of the linear transform. Second, we will provide an optimization algorithm where the linear transformation is estimated by the gradient descent method. Third, we will propose two initialization procedures for the linear transformation which lead to two variants of the proposed algorithm. The first procedure refines the RSLDA (transformation matrix $\mathbf{Q}$) solution using the proposed model's objective function. The second procedure sets the initial transformation matrix to a hybrid combination of transformation matrices obtained from two methods: Inter-class sparsity based discriminative least square regression denoted as ICS_DLSR [48] and RSLDA [47].

Our proposed method inherits the advantages of two powerful discriminant methods at two levels: (1) the hybrid transformation initialization, and (2) the refinement via the proposed single new criterion.

The proposed method is also able to obtain a well-constructed projection space that ensures high classification performance; it can be additionally used in tuning an already obtained projection matrix. The proposed method can be generic in the sense that any hybrid initial transformation matrix could be fed into our algorithm and then a more discriminant solution for the transformation matrix will be obtained, leading to a higher classification performance.

The paper's main contributions could be seen as follows:

- The proposed method inherits the advantages of two recent powerful discriminant methods. The obtained transformation encapsulates two different types of discrimination, namely the inter-class sparsity in addition to robust LDA.
- Introducing a hybrid initialization for the transformation matrix, where the initial matrix is created by combining two solutions of two different methods.
- Using the gradient descent method to find a solution for the proposed criterion, where the sought transformation matrix's gradient is calculated in each iteration and the unknowns are updated accordingly.

The conducted experiments show that the proposed method has led to an improvement in the classification accuracy and was able to outperform competing methods. The remainder of the paper is structured as follows. Section 2 describes related work and presents the notations used in our paper. Section 3 presents the proposed method's criterion and solution details. The obtained experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Work and notations

This section will describe methods that are relevant to our proposed work. We are going to briefly talk about the gradient descent method and how we used it to obtain a better embedding space. We will also be showing how the introduction of the $\ell_{2,1}$ [51] norm and the inter-class sparsity constraint were used for feature selection and helped in discrimination [37]. Additionally, we will numerate various recent methods that have used such a constraint by embedding it into their global criterion to insure that the method performs feature selection [12, 27].

2.1 Notations

We will start by introducing the notations that we will use in our paper. We will refer for the training set by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d\times N}$, with d the dimension of each sample.

Each sample $\mathbf{x}_i$ is a column vector with d features $\in \mathbb{R}^d$

The number of training samples will be denoted by N, in addition C will represent the total number of classes.

The $\ell_{2,1}$ norm of a matrix $\mathbf{Z} \in \mathbb{R}^{d\times N}$ is obtained through the following formula $\|\mathbf{Z}\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{N} z_{ij}^2}$, and the ℓ_2 norm for the vector $\mathbf{z} = [z_1, z_2, ..., z_d]$ is obtained as follows $\|\mathbf{z}\|_2 = \sqrt{\sum_{i=1}^{d} z_i^2}$.

Table 1 shows the main notations used in our paper.

2.2 Related Work

Many linear projection methods were recently proposed. The methods mainly aims to extract a discriminant embedding for the data. Some have integrated constraints that implement feature selection within the method and rank their projection matrices' features. Feature selection or ranking is becoming a trending problem in the machine learning field. Using all data features will not lead very often to a high classification performance. Feature selection is intended to efficiently select the most relevant features of the data that enhances discrimination [36, 53, 55].

One big problem for handling data is the high dimensionality. The most famous method used to tackle the high dimensionality curse is the Principle component analysis (PCA) [35] method. PCA is an unsupervised feature extraction method that transforms the original data features and projects them into a low dimensional space. Another well-known supervised linear method that was able to ensure good discrimination is the Linear Discriminant Analysis (LDA) [11, 38] method, a supervised technique (meaning that it requires label information for the data). LDA and its variants are some of the most used and discriminating algorithms in the machine learning field. LDA estimates a transformation matrix where the desired space maximizes the between-class variance and minimizes the within-class variance. The projection axis $\mathbf{w}$ would be the solution for the Fisher criterion [22]:

$$\mathbf{w} = \arg \min_{\mathbf{w}^T\mathbf{w}=1} \mathbf{w}^T\,(\mathbf{S}_w - \mu\,\mathbf{S}_b)\,\mathbf{w} \tag{1}$$

Table 1 Main notations used in the paper

Notation	Description
$\mathbf{X}$	Training data samples $\in \mathbb{R}^{d\times N}$
$\mathbf{P}$	Orthogonal matrix $\in \mathbb{R}^{d\times d}$
$\mathbf{Q}$	Projection matrix $\in \mathbb{R}^{d\times d}$
$\mathbf{D}$	Diagonal matrix
$\mathbf{S}_w$	Within-class scatter matrix
$\mathbf{S}_b$	Between-class scatter matrix
d	Dimensionality of data
N	Number of data samples
n_i	Number of samples in the i-th class
C	Number of classes
$\mathbf{x}_i$	The i-th data sample $\in \mathbb{R}^d$

where μ is a small positive constant that balances the effect of the two scatter matrices (Within-class scatter matrix $\mathbf{S}_w$ and between-class scatter matrix $\mathbf{S}_b$) which could be calculated as:

$$\mathbf{S}_b = \frac{1}{N} \sum_{i=1}^{C} n_i \left(\mu_i - \mu\right) \left(\mu_i - \mu\right)^T \tag{2}$$

$$\mathbf{S}_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{n_i} (\mathbf{x}_j{}^i - \mu_i) \, (\mathbf{x}_j{}^i - \mu_i)^T \tag{3}$$

where μ, μ_i are the mean of all data samples and the mean of samples of the i-th class, respectively. Many variants of LDA were proposed and still being proposed (e.g. [7, 65, 66]), as the linear discriminant analysis showed good interpretability for the data.

Review of Robust Sparse Linear Discriminant Analysis (RSLDA): RSLDA [47] was proposed to tackle many limitations of the classical LDA [38], RSLDA mainly adds the $\ell_{2,1}$ regularization of the projection matrix. This regularization term is inserted in the global criterion to insure that the method performs feature ranking and weighting. RSLDA minimizes the following criterion:

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{E}} Tr\left(\mathbf{Q}^T \mathbf{S} \mathbf{Q}\right) + \lambda_1 \, ||\mathbf{Q}||_{2,1} + \lambda_2 \, ||\mathbf{E}||_1 \tag{4}$$

$$s.t. \quad \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T \mathbf{X} + \mathbf{E}, \;\; \mathbf{P}^T \mathbf{P} = \mathbf{I}$$

where $\mathbf{S}$ is the difference matrix $\mathbf{S}_w - \mu\,\mathbf{S}_b$, λ_1 and λ_2 are two parameters that balance the importance of different terms. In the criterion of RSLDA the $\ell_{2,1}$ norm was imposed on the projection matrix to achieve feature selection.

Review of Inter Class Sparsity Least Square Regression: In [48], the authors proposed the Inter-class sparsity based discriminative least square regression ICS_DLSR [48]. This method provides a linear mapping to the soft labels' space where the latent space's dimension is set to the number of classes. This method was able to construct a model where the margins of samples pertaining to the same class is widely reduced while the one for the samples pertaining to different classes is enlarged. This was done by adding a class-wise row sparsity constraint to the transformed features.

Another similar method is the method described in [37] where the $\ell_{2,1}$ norm is applied on the original linear discriminant analysis transformation.

3 Proposed Method

In this section, we will present our problem formulation and show the steps applied for finding a solution to our problem. Our method is a linear projection method used for feature extraction and targeting a more discriminative transformation matrix. Two of the method's variants are proposed. These two variants differ in the initialization step. Our proposed method has inherited feature ranking by using the RSLDA solution as an initial guess for the sought transformation. The next step is to fine-tune the transformation matrix's initial guess by minimizing the proposed criterion with a gradient descent method aimed at finding the required solution for the transformation matrix $\mathbf{Q}$.

The gradient descent algorithm is one of the most simple and efficient algorithms used to solve unconstrained optimization problems. In our algorithm, We have used the gradient descent approach to calculate the transformation matrix $\mathbf{Q}$ and find the solution.

3.1 Formulation

We propose a novel method intended to obtain the two matrices: $\mathbf{Q} \in \mathbb{R}^{d\times d}$ projection matrix, in addition to the orthogonal matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$. Our proposed method aims to minimize the following objective function:

$$f(\mathbf{Q},\mathbf{P}) = Tr\left(\mathbf{Q}^T \mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} ||\mathbf{Q}^T \mathbf{X}_i||_{2,1} + \lambda_2 \, ||\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T \mathbf{X}||_2^2 \tag{5}$$

$$s.t. \quad \mathbf{P}^T \mathbf{P} = \mathbf{I}$$

where $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$ is the data matrix associated with the i-th class, n_i is the number of training samples in the i-th class, C is the number of classes.

The first term in the (5) is the LDA criterion where $\mathbf{S}$ represents the LDA scatter matrix which could be calculated as $\mathbf{S} = \mathbf{S}_w - \mu\,\mathbf{S}_b$ in which $\mathbf{S}_w$ being the within-class matrix and $\mathbf{S}_b$ the between-class matrix. These two matrices are given by (2) and (3). The second term of the criterion is imposed to ensure that transformed features of the same class, in the projected space, obtain common sparse structure. $\mathbf{Q}$ is the sought projection matrix. In addition, a variant of (PCA) constraint is introduced to guarantee that original data would be recovered well, presented in the third term of the proposed method criterion. λ_1 and λ_2 are two trade-off parameters to control the importance of the different terms. One knows that the $\ell_{2,1}$ norm of a matrix can be written as:

$$||\mathbf{Z}||_{2,1} = Tr\left(\mathbf{Z}^T \, \mathbf{D}\,\mathbf{Z}\right) \tag{6}$$

where $\mathbf{D}$ is a diagonal matrix that is given by:

$$\mathbf{D} = \begin{pmatrix} \frac{1}{\|\mathbf{z}(1)\|_2+\epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\|\mathbf{z}(d)\|_2+\epsilon} \end{pmatrix} \tag{7}$$

where $\mathbf{Z}\,(j)$ represents the j-th row of $\mathbf{Z}$.

By substituting the second term of the criterion by its trace form showed in (6), problem (5) can be viewed as:

$$f(\mathbf{Q},\mathbf{P}) = Tr\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} Tr\;(\,(\mathbf{Q}^T\mathbf{X}_i)^T\;\mathbf{D}_i\mathbf{Q}^T\mathbf{X}_i) + \lambda_2\,||\mathbf{X} - \mathbf{P}\mathbf{Q}^T\,\mathbf{X}||_2^2 \tag{8}$$

$$\min_{\mathbf{Q},\mathbf{P}} f\;(\mathbf{Q},\mathbf{P})\;\; s.t. \quad \mathbf{P}^T\,\mathbf{P} = \mathbf{I} \tag{9}$$

Equation (8) presents the criterion for the proposed method. The minimization of this criterion's first term is targeting a transformation matrix which ensures class discrimination using Linear Discriminant Analysis (LDA). The criterion's second term is introduced to obtain class sparsity. By introducing this constraint, transformed features from each class will obtain a common sparse structure. Finally, a variant of "Principle component analysis" constraint is introduced in our proposed criterion [14]. This last constraint was introduced for the purpose of retaining PCA's energy [35], this constraint will assure robustness for our data.

To find a solution for the proposed method, we have used the descent gradient algorithm, a mathematical process used for the minimization of a specific function. Using the gradient algorithm, one should know the function called the cost function in addition to the function's derivative. The gradient algorithm allows solving the optimization problem in a way that, from a given point, one knows the gradient and can move in that direction to obtain a solution. Using the descent gradient algorithm has many advantages, from which we shall state the most important, namely:

- Has a lower computational complexity compared to other methods. Finding the solution through the descent gradient algorithm is often less computationally demanding. Using the descent gradient to find a solution will lead to a faster model.
- Leads to accurate solutions. Not only is the descent gradient algorithm known to be fast, but it will also lead to a more accurate solution for the minimization problem than the closed form solution.

3.2 Solution steps to the proposed method

To solve the formulated problem above, we have adopted the alternating direction method of multipliers (ADMM) [2] and calculated each variable while other variables are fixed as follows:

- **Calculate the orthogonal matrix P:**

  $\mathbf{P}$ can be calculated by fixing the variable $\mathbf{Q}$ and through solving the following problem:

  $$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \tag{10}$$

  Using $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ the fact the squared norm of a matrix $\mathbf{A}$ is given by $\|\mathbf{A}\|_2^2 = Tr(\mathbf{A}^T\mathbf{A}) = Tr(\mathbf{A}\,\mathbf{A}^T)$, problem (10) is equivalent to the following maximization problem:

  $$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \;\longrightarrow\; \max_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} Tr\;(\mathbf{P}^T\,\mathbf{X}\mathbf{X}^T\,\mathbf{Q}) \tag{11}$$

  One can find a solution for problem (11) by performing singular value decomposition of $\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}$. Suppose the SVD decomposition is given by $SVD\;(\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}) = \mathbf{U}\,\Sigma\,\mathbf{V}^T$. Then $\mathbf{P}$ is obtained as [66]:

  $$\mathbf{P} = \mathbf{U}\,\mathbf{V}^T \tag{12}$$

- **Calculate the Projection matrix Q:**

  Gradient descent is an iterative optimization scheme used to minimize function by moving in the direction of steepest descent in each iteration. How to use gradient method differs through different fields, in machine learning and classification, the gradient is used to iteratively update the parameters of the desired model. We have adopted gradient descent method to calculate $\mathbf{Q}$ in each iteration of the proposed method like follows:

  The orthogonal matrix $\mathbf{P}$ is fixed. Let us consider the trace form of the criterion of our problem:

  $$f(\mathbf{Q},\mathbf{P}) \;=\; Tr\left(\mathbf{Q}^T\mathbf{S}\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} Tr(\mathbf{X}_i^T\mathbf{Q}\mathbf{D}_i\mathbf{Q}^T\mathbf{X}_i) + \lambda_2\,||\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\;\mathbf{X}||_2^2 \tag{13}$$

  We calculate the gradient of the objective function w.r.t. $\mathbf{Q}$ as follows:

  $$\mathbf{G} = \frac{\partial\, f}{\partial\,\mathbf{Q}} = 2\,\mathbf{S}\,\mathbf{Q} + \lambda_1 \sum_{i=1}^{C} 2\,\mathbf{X}_i\;\mathbf{X}_i^T\;\mathbf{Q}\,\mathbf{D}_i + 2\lambda_2\,[\mathbf{X}\,\mathbf{X}^T\;\mathbf{Q} - \mathbf{X}\,\mathbf{X}^T\;\mathbf{P}] \tag{14}$$

  Using the gradient matrix, we can update $\mathbf{Q}$ by:

  $$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \alpha\,\mathbf{G} \tag{15}$$

  where $\mathbf{Q}_{t+1}$ and $\mathbf{Q}_t$ denotes the projection matrix $\mathbf{Q}$ in iteration $t+1$ and iteration t respectively. α is the step length (learning rate).

- **Update Variable $\mathbf{D}_i$:** We update $\mathbf{D}_i$, $(i = 1, ..., C)$ by:

$$\mathbf{D}_i = \begin{pmatrix} \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_i(1)\right\|_2+\epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_i(d)\right\|_2+\epsilon} \end{pmatrix} \quad (16)$$

where ϵ is a small positive scalar and $\mathbf{Q}^T\mathbf{X}_i\ (j)$ represents the j-th row vector of $\mathbf{Q}^T\mathbf{X}_i$.

Algorithm 1 summarizes our proposed method and describes the main steps for solving the problem (5).

Algorithm 1 Feature extraction using gradient method **FE_GD**.

```
Input:           1. Data samples $\mathbf{X} \in \mathbb{R}^{d\times N}$
                 2. Labels of the training samples
                 3. The step length of the gradient descent $\alpha$
                 4. Parameters $\lambda_1, \lambda_2$
Output:          $\mathbf{P}, \mathbf{Q}$
Initialization:  $\mathbf{Q}^{(0)}$ obtained from RSLDA or using a
                 hybrid combination (see Section 3.3).
Process:         set $t = 0$ and $\mathbf{Q} = \mathbf{Q}^{(0)}$
                 Repeat
                 Fix $\mathbf{Q}$, update $\mathbf{P}^{(t+1)}$ using (11).
                 Calculate the gradient matrix $\mathbf{G}$ using (12)
                 Fix $\mathbf{P}$, update $\mathbf{Q}^{(t+1)}$ using (13).
                 Update $\mathbf{D}_i$ using (14)
                 set $t = t + 1$
                 Until convergence
```

The projection of the training and test samples is carried out using the estimated projection matrix $\mathbf{Q}$. This is given by $\mathbf{z}_{train} = \mathbf{Q}^T\mathbf{x}_{train}$ and $\mathbf{z}_{test} = \mathbf{Q}^T\mathbf{x}_{test}$ where $\mathbf{x}_{train}$ is a training data sample, and $\mathbf{x}_{test}$ is a test data sample.

3.3 Initialization of Projection Matrix Q

The linear transformation $\mathbf{Q}$ needs a good initial guess since it is estimated by a gradient descent update rule. In this section, we provide two initialization procedures leading to two variants of the proposed algorithm.

3.3.1 Using RSLDA [47] algorithm

In this variant, the initial guess $\mathbf{Q}^{(0)}$ for the linear transformation matrix $\mathbf{Q}$ is given by the solution of the RSLDA method (solved using its own ADMM optimization). We can note that this initial transformation inherits the feature ranking of RSLDA.

3.3.2 Hybrid combination of projection matrices obtained from the two embedding methods RSLDA and ICS_DLSR [48]

In our proposed algorithm's second variant, the initial transformation matrix $\mathbf{Q}^{(0)}$ is set to a hybrid combination of the transformation matrices obtained by the two embedding methods RSLDA [47] and ICS_DLSR [48].

The number of the hybrid transformation's rows $\mathbf{Q}^{(0)}$ should be d. On the other hand, the number of columns (projection axes) can be set to any arbitrary value. Without losing generality, in order to be consistent with the linear methods, we will assume that the total number of $\mathbf{Q}^{(0)}$ columns is d. Thus, $\mathbf{Q}^{(0)} \in \mathbb{R}^{d\times d}$. According to [48], the linear transformation $\mathbf{Q}_{ICS\_DLSR}$ obtained by the ICS_DLSR algorithm is $\in \mathbb{R}^{d\times C}$ where d and C represent the dimension of features and the number of classes, respectively. On the other hand, the RSLDA method [47] provides its own linear transformation $\mathbf{Q}_{RSLDA} \in \mathbb{R}^{d\times d}$. The sought initial hybrid projection matrix $\mathbf{Q}^{(0)}$ used in our algorithm is denoted by $\mathbf{Q}_{Hybrid}$. It is constructed by taking all the C columns of $\mathbf{Q}_{ICS\_DLSR}$ to which the first $d - C$ columns of $\mathbf{Q}_{RSLDA}$ are appended. The resulting transformation matrix $\mathbf{Q}_{Hybrid}$ is $\in \mathbb{R}^{d\times d}$. The strategy for hybrid initialization methodology is illustrated in Fig. 1.

In the above construction of the hybrid matrix $\mathbf{Q}_{Hybrid}$, featured within our work the number of projection axes for each type of projection was respectively fixed to C and $d - C$ for ICS_DLSR and RSLDA. We emphasize the fact that these dimensions can be changed.

In our experiments, according to Table 2, we can see that the value of C that represents the number of classes varies between 10 and 50 for the datasets used. d represents the number of features for each dataset is also shown in the same table.

3.4 Computational complexity

This section is intended to analyze the proposed method's computational complexity (see **Algorithm 1**). Matrices $\mathbf{Q}$, $\mathbf{P}$, are sought to be calculated. The orthogonal matrix $\mathbf{P}$ requires singular value decomposition. The computational cost for a decomposition of a $d \times N$ matrix would be $\mathcal{O}\left(N^3\right)$. $\mathbf{Q}$ is calculated in the second step of the method, it requires the calculation of the corresponding gradient matrix, but since these two steps only consist of simple matrix operations, they have small computational costs, thus could be ignored. Also the step intended to update $\mathbf{D}_i$ coming from the (16) is a simple matrix operation that have a very small cost.

On the other hand, in our proposed method's first variant, we have used the RSLDA method for the transformation matrix $\mathbf{Q}$ initialization before it is fed to our algorithm.

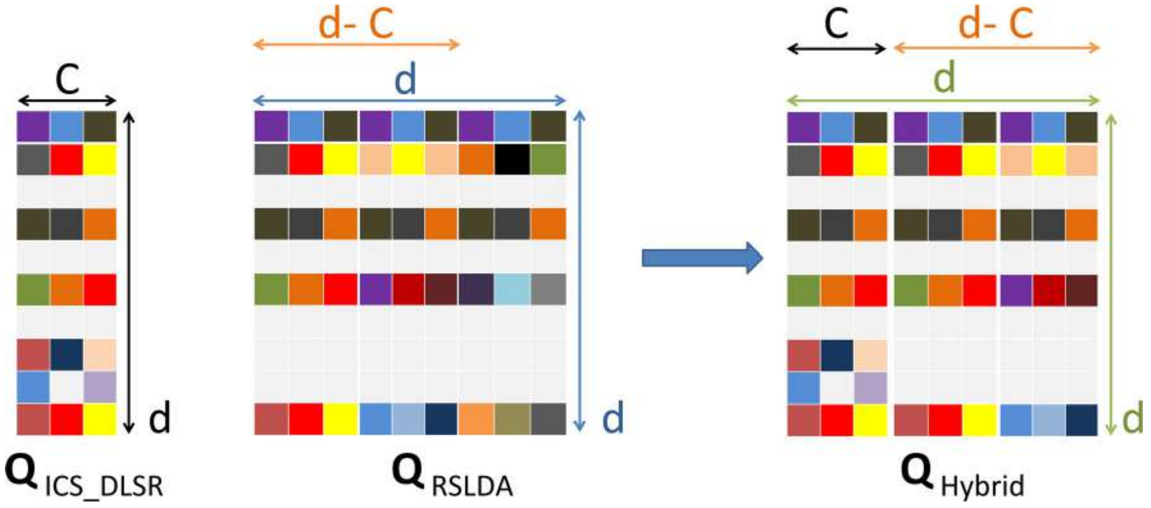


Fig. 1 Hybrid initialisation using the linear transformations associated with ICS_DLSR and RSLDA

Thus, the complexity of RSLDA method should be added to the complexity of our proposed method. Supposing τ represents the number of iterations of RSLDA. The latter has a complexity of $\mathcal{O}\left(\tau(d^2N+4d^3)\right)$. The proposed algorithm's main computational complexity takes place in the updating **P** step. The the proposed method's (first variant) complete cost is mainly $\mathcal{O}\left(\tau\left(N^3\right)\right)$. In summary, the overall cost would be the sum of RSLDA cost added to the cost of our proposed method which would be equal to $\mathcal{O}\left(\tau(d^2N+4d^3)\right)+\mathcal{O}\left(\tau'\left(N^3\right)\right)$ where τ' denotes the number of iterations of Algorithm 1.

For the second proposed variant, we have constructed the initial guess of the transformation matrix through a combination of the two solutions obtained from the two methods RSLDA[47] and ICS_DLSR[48] method. Knowing that ICS_DLSR algorithm has a complexity of $\mathcal{O}\left(\tau\left(d^3\right)\right)$, the second proposed variant have a total complexity of $\mathcal{O}\left(\tau\left(d^3\right)\right)+\mathcal{O}\left(\tau(d^2N+4d^3)\right)+\mathcal{O}\left(\tau'\left(N^3\right)\right)$

4 Performance Study

To test the two variants of our proposed method, we conducted experiments on several datasets including faces, object and handwritten datasets. Detailed information on these datasets are presented in this section, Next we are going to present the setups for the experiments and the results obtained.

4.1 Datasets

In our work we have conducted our experiments over the following five public datasets in addition to a large-scale dataset: **USPS**[1] digits dataset, **Honda**[2] dataset, **COIL20**[3] object dataset, **Extended Yale B**[4] face dataset, **FEI**[5] dataset, and the large scale **MNIST** dataset consisting of 60,000 images.

Table 2 presents a summary for all the information concerning the datasets used in our paper.

4.2 Results

In this section, we will present the classification performance when the projected spaces are obtained by the proposed schemes and some competing methods. The proposed method has two variants, namely:

- Feature Extraction Using Gradient Descent **FE_GD** : In this variant, our proposed method is implemented while the initial transformation matrix $\mathbf{Q}^{(0)}$ used in our proposed method's first iteration is set to the output of RSLDA [47] algorithm as presented in Section 3.3.1.
- Feature Extraction Using Gradient Descent With Hybrid initialization **FE_GD_HI**: The second variant of the proposed method consists of initializing the transformation matrix $\mathbf{Q}^{(0)}$ used in our proposed method's first iteration as a hybrid combination of two transformation matrices obtained from the two methods RSLDA [47] and ICS_DLSR [48] as shown in Fig. 1 and detailed in Section 3.3.2.

[1] https://www.kaggle.com/bistaumanga/usps-dataset

[2] http://vision.ucsd.edu/~leekc/HondaUCSDVideoDatabase/HondaUCSD.html

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

[5] https://fei.edu.br/~cet/facedatabase.html

Table 2 Brief datasets description

Dataset	Type	Number of samples	Number of features	Number of classes	Descriptor
USPS	Digits	1100	256	10	RAW-brightness images
Honda	Face	2277	1024	22	RAW-brightness images
COIL20	Object	1440	177	20	Local Binary Patterns
Extended yale B	Face	2414	1024	38	RAW-brightness images
FEI	Face	700	1024	50	RAW-brightness images
MNIST	Digits	60,000	2048	10	Deep features (ResNet-50)

Table 3 Mean classification accuracies (%) of different methods on the tested datasets. The best performance is bolded

Dataset \ Method	Training samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	FE_GD	FE_GD_HI
USPS	30	87.01	88.21	84.91	83.54	72.01	88.46	89.45	89.50	**90.29**
	40	88.56	90.40	86.19	85.3	72.30	90.16	91.11	**91.81**	91.46
	55	90.51	92.09	88.64	87.16	73.32	91.25	92.65	**93.07**	92.87
	65	91.76	93.16	89.29	88.58	74.11	91.53	92.89	**93.71**	93.49
Honda	10	64.12	71.32	65.95	65.74	61.86	70.79	69.90	70.16	**72.14**
	20	77.69	83.60	79.39	79.25	75.33	82.95	83.03	83.60	**84.64**
	30	84.78	89.09	85.84	86.24	82.55	88.20	89.04	89.41	**90.12**
	50	91.36	94.15	92.28	92.34	90.03	93.53	94.13	94.53	**95.10**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	93.19	93.81	**94.58**
	6	90.35	92.93	94.18	92.15	88.73	93.65	94.25	94.75	**95.08**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.66	96.20	**96.29**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.43	**96.97**	96.40
COIL20	20	94.58	97.65	96.19	95.00	94.87	**98.04**	96.73	96.89	97.66
	25	95.79	98.22	97.07	96.12	95.99	98.22	97.74	97.89	**98.59**
	30	96.65	98.70	97.81	97.01	97.49	98.75	98.26	98.52	**99.08**
	35	97.14	98.81	98.15	97.42	98.11	99.12	98.68	98.80	**99.39**

Table 4 Mean classification accuracies (%) on the Extended Yale B dataset. The best performance is bolded

Dataset \ Method	Training samples	KNN	SVM	LDA	LDE	ELDE	PCE	SULDA	MPDA	ICS_DLSR	RSLDA	FE_GD	FE_GD_HI
Ext. Yale B	10	69.80	73.85	82.32	79.92	85.85	86.39	84.61	83.67	86.56	86.79	87.10	**88.42**
	15	75.20	80.02	86.76	83.77	89.30	89.23	88.72	86.82	89.53	89.93	90.04	**91.21**
	20	80.24	85.79	90.7	88.44	93.07	92.19	91.66	90.38	93.14	93.59	93.75	**93.81**
	25	82.24	89.03	92.17	90.43	94.09	93.35	92.14	91.79	94.50	94.92	95.02	**95.09**

Table 5 Mean classification accuracies (%) of different methods on the MNIST dataset. The best performance is bolded

Dataset \Method	Training samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	FE_GD	FE_GD_HI
MNIST	1000	91.75	97.58	85.74	93.22	93.77	98.02	97.95	98.21	**98.33**

The two proposed variants have been compared with the following methods: K-nearest neighbors (KNN) [21], Support Vector Machines (SVM) [3] (the Linear SVM was implemented suing the LIBSVM library[6] Linear Discriminant Analysis (LDA) [38], Local Discriminant Embedding (LDE) [5], PCE [30] (unsupervised method) ICS_DLSR [48] and Robust sparse LDA (RSLDA) [47].

All experiments and compared methods used the same conditions in order to guarantee a fair comparison. For each compared embedding method, the whole dataset is randomly split into a training part and a test part.

First, for each compared embedding method, a transformation matrix is estimated from the training part, then, training and test data are projected onto the new space using the already computed transformation. Finally, the test data classification is performed using the Nearest Neighbour classifier (NN) [8].

Different percentages of training are used. Moreover, for a given percentage of training data, the whole evaluation is repeated ten times. That means we adopt ten random splits for every configuration and report the average recognition rate (correct classification rate for test part) over these ten random splits.

We used PCA as a pre-processing technique. In our experiments, PCA [35] is used as a dimensionality reduction technique to preserve (100%) entirety of the data's energy. As for the parameter α, we should set it to a small value. In our experiments, this value was chosen in $\{10^{-7}, 10^{-5}\}$.

The obtained results are summarized in Table 3. This table depicts the two proposed variants classification rates in addition to those of the competing methods when used with the USPS, Honda, FEI and COIL20 datasets. The results are obtained using different training and testing percentages from the data. Results shown in this table are obtained using the Nearest Neighbor classifier. Table 4 contains data about the obtained results for different competing methods using the Extended Yale B dataset. In this table, various training percentages corresponding to different numbers of samples used in the training process are shown. We should emphasize that more competing methods are presented in Table 4. These additional methods are ELDE, SULDA [59] and MPDA [62]. These are added to enrich the comparison using more methods. The depicted rates are the average over 10 random splits and correspond to different numbers of training samples. The first column inside the table depicts the number of training images per class.

Table 5 illustrates the classification accuracy for the competing methods alongside with the two proposed variants using the large scaled MNIST dataset that contains a total number of 60,000 images in total. Results shown in this table are obtained using one split, while using 1000 samples from each class for training and the remaining samples are used for testing(Fig. 2).

Figure 3 presents the obtained recognition rate (%) associated with the LDA [38], LDE [5], RSLDA [47] and our proposed method's two variants. The recognition rate is given as a function of the dimension of the projected features. Results are shown for (a) the COIL20 dataset, (b) the Extended Yale B and (c) the HONDA dataset. 30, 10 and 10 samples from each class are respectively used for training. The depicted results were obtained using the **Nearest Neighbor** (NN) Classifier.

We have used 21 evaluations using 6 different datasets from the experiments in this paper to study the statistical analysis of our proposed method's two variants alongside with those of the competing methods. We performed the Friedman test [10] and computed the critical distance CD. The obtained results of the conducted test lead to the conclusion that the tested methods do not have the same performance. Figure 2 shows the CD diagram for the 9 methods including our two proposed variants, where the average rank of each is marked along the axis.

Visualization of transformation matrix Q: Figure 4 visualizes the first 50 rows of the transformation matrix **Q** obtained from our proposed method's two variants. The

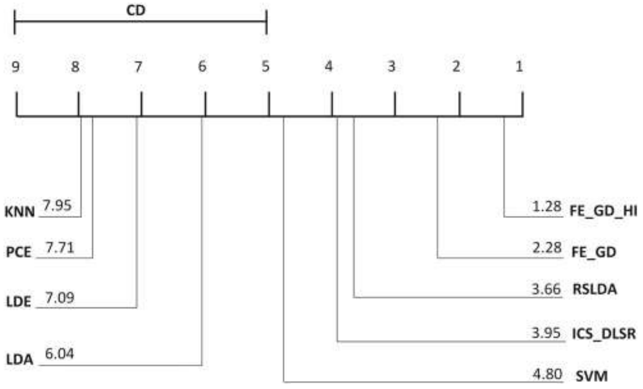
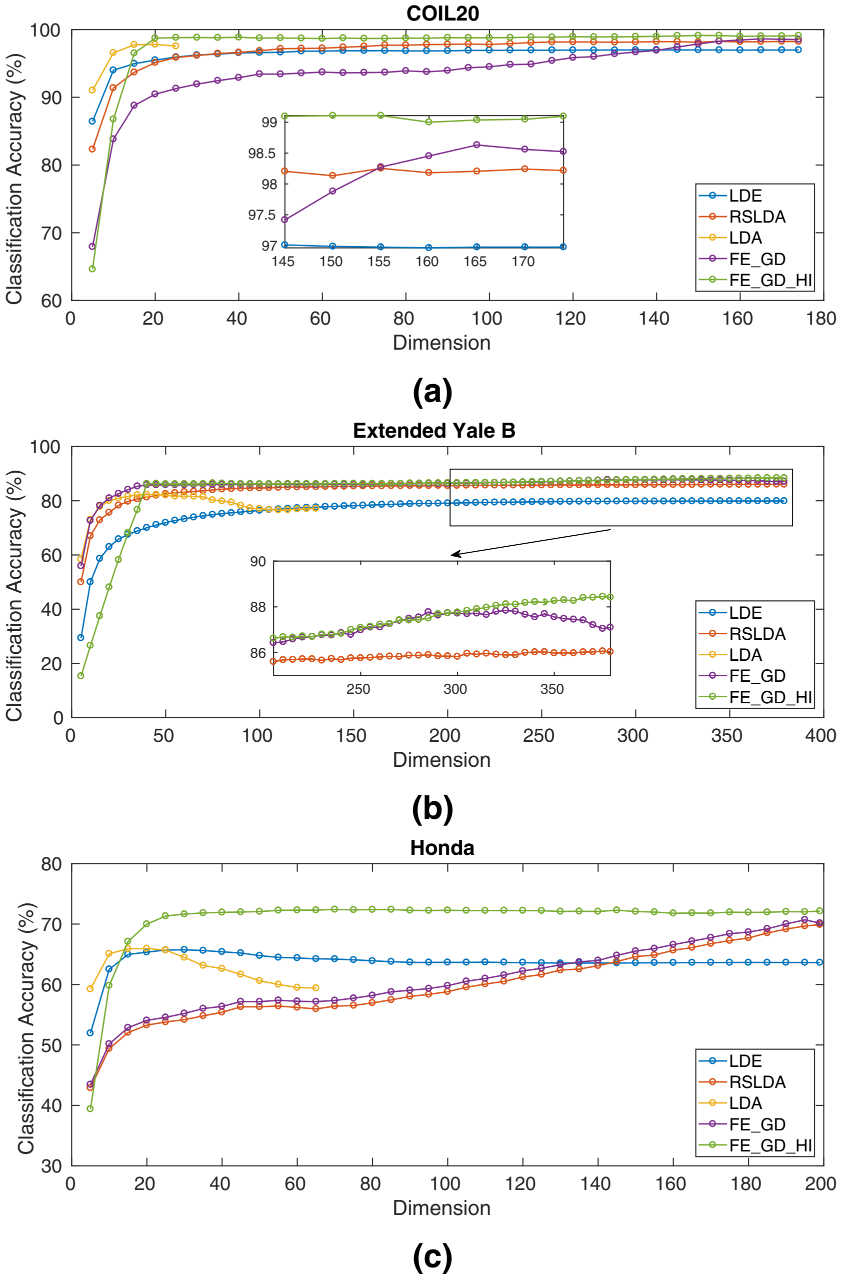


Fig. 2 Statistical analysis - CD diagram

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Fig. 3 Classification accuracy (%) vs. dimension for different datasets



dataset used to obtain this transformation matrix is the USPS digits dataset while using 30 samples from each class for training. Figure 4a and c depict the elements of the transformation matrix $\mathbf{Q}$ obtained by the proposed variants. Figure 4b and d show the features of the transformation matrices obtained from the two proposed variants according to the $\mathbf{Q}$ scores (row-norm) after being normalized to have values between 0 and 1. We can clearly see from this figure that most relevant features are placed at the top.

Implicit vs explicit feature selection: This experiment is intended to compare how the classification performance will vary when the data is submitted to pure feature selection and ranking techniques. Table 6 shows the classification performance when original data was ranked using the Fisher score, ReliefF score [34], Minimum redundancy maximum relevance (MRMR) [29], and Robust multi-label feature selection with dual-graph regularization (DRMFS) algorithm [19] compared to our proposed method (the Extended Yale B dataset is used). The MRMR algorithm uses the mutual information [6] as a proxy for computing relevance and redundancy among variables (features). In [19], authors proposed a criterion aimed to calculate the feature weight matrix. Authors imposed both $\ell_{2,1}$-norm and non-negative constraints onto the feature weight matrix to enhance the property of row-sparsity. Once the weight matrix is calculated, the scores of each row representing each feature can be calculated, and one can evaluate the desired top K-features.

The results show that our proposed method outperform the competing feature selection methods compared in Table 6. Despite the fact that our proposed method's main goal is to perform feature extraction and obtain a discriminant transformation, our method explicitly performs feature selection by imposing the $\ell_{2,1}$ norm over the transformation matrix $\mathbf{Q}$ in our objective function.

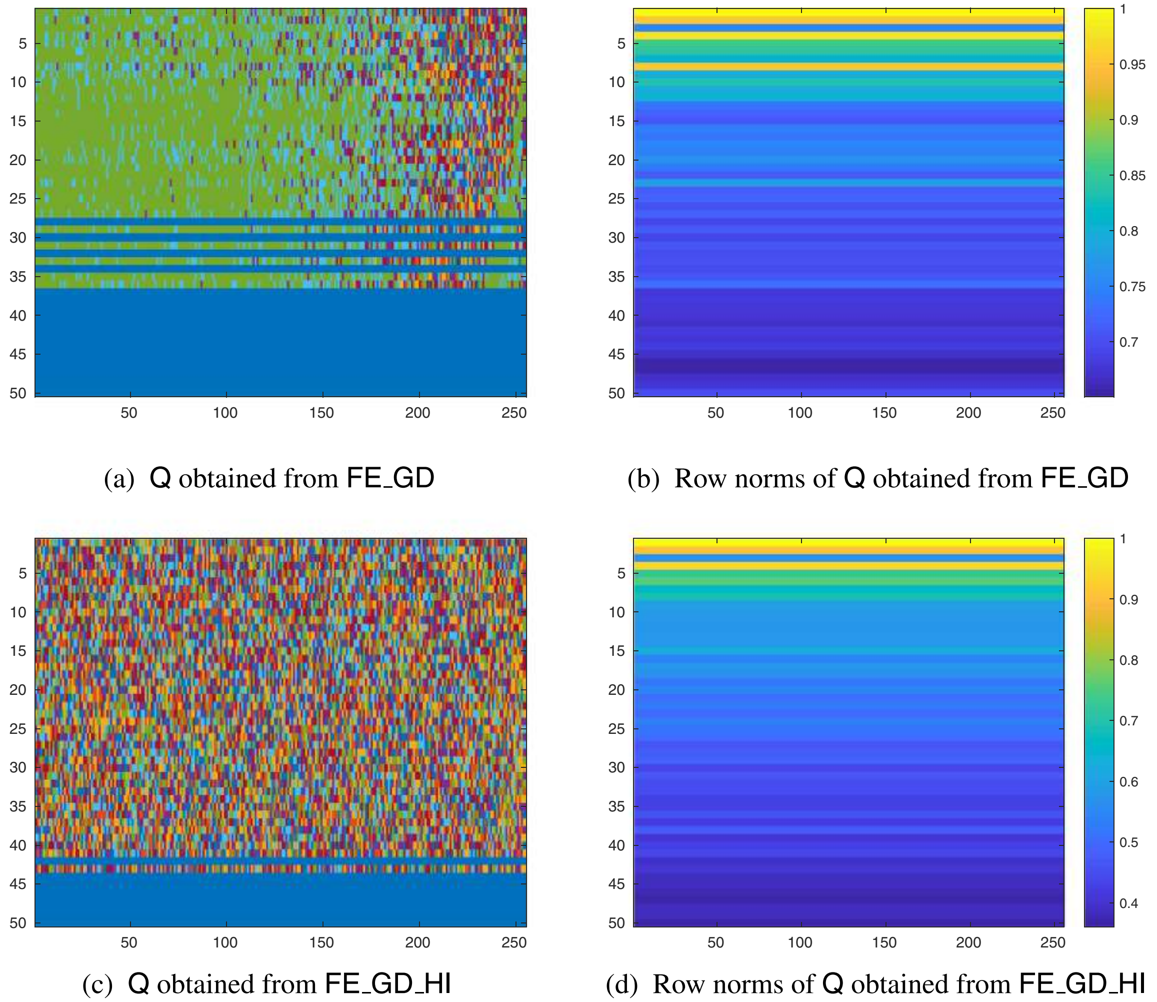
A variant of Principle component analysis constraint is introduced in our proposed criterion $\lambda_2 \, ||\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}||_2^2$. We introduced this constraint to retain PCA's energy preserving property [35]. This constraint will assure robustness for the obtained transformation. We studied the effect of removing this constraint from our objective function and how the PCA variant contributed to obtaining better outcomes. Table 7 presents the classification performance on the USPS, Extended Yale B and Honda datasets using different training percentages when the PCA constraint was removed from our objective function.

In this table, we have evaluated the performance of ICS_DLSR, RSLDA, and our proposed method on three datasets. Our proposed method's two variants classification performance are presented in the last two columns of Table 7 (i.e., columns 6 and 7). Columns 4 and 5 depict the performance of the proposed variants when the PCA constraint is removed from the global criterion.

One can observe that the classification performance obtained with the PCA variant constraint is better than that obtained without this constraint. This proves the contribution of the PCA variant in obtaining better outcomes.

4.3 Parameter sensitivity

In this section, we will investigate and demonstrate the effect of changing the proposed method parameters on the classification rates for different datasets. The proposed



(a) Q obtained from FE_GD

(b) Row norms of Q obtained from FE_GD

(c) Q obtained from FE_GD_HI

(d) Row norms of Q obtained from FE_GD_HI

Fig. 4 Transformation matrix $\mathbf{Q}$ visualization (USPS dataset) (First 50 rows)

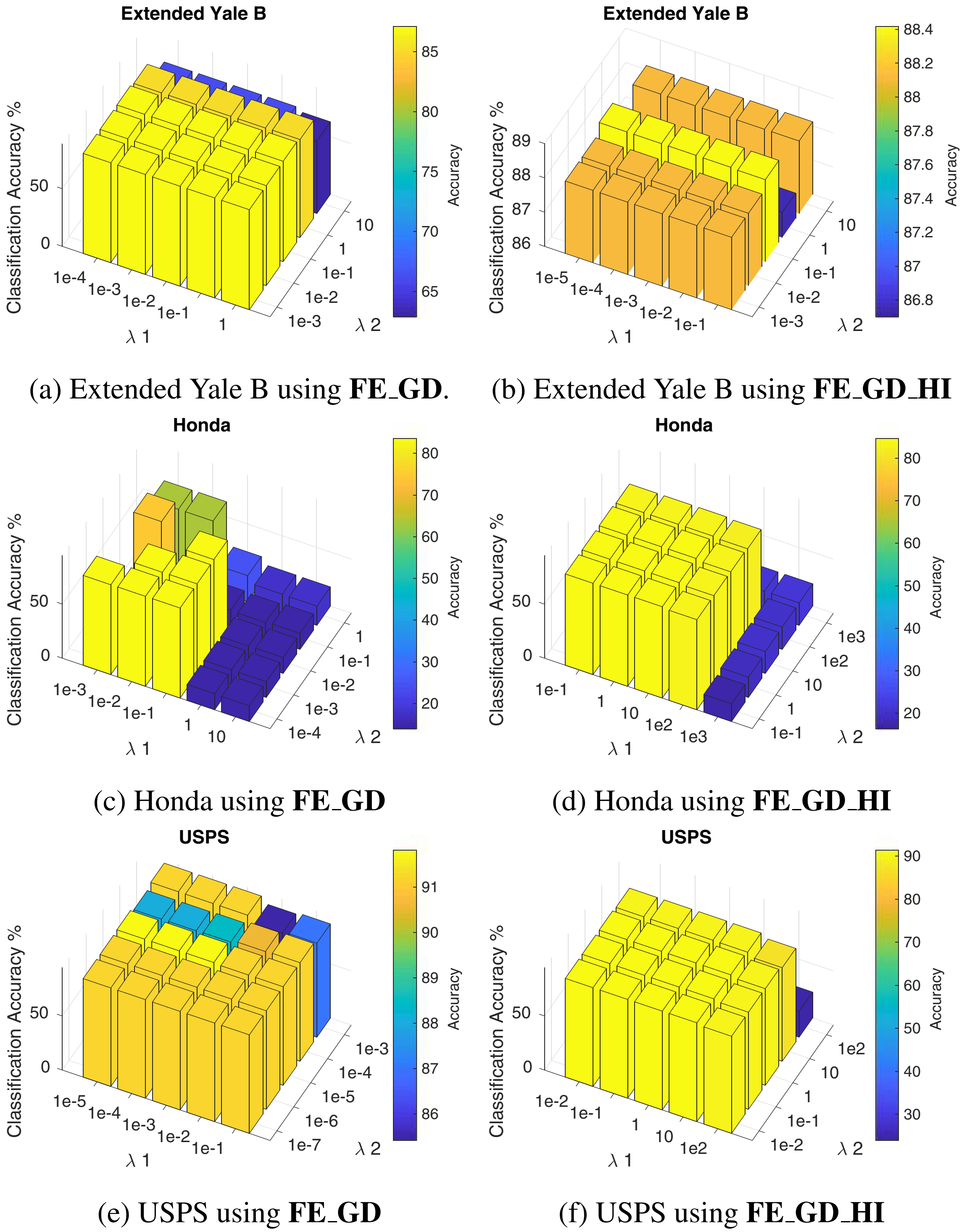
Table 6 Feature selection comparison. The extended Yale B dataset is used. The best performance is bolded

Method	Selection scheme	Training samples			
		10	15	20	25
KNN	None	69.8	75.2	80.24	82.24
	Fisher	72.17	76.93	81.93	84.10
	ReliefF	70.79	76.85	82.00	83.63
	MRMR	71.50	76.23	80.99	83.05
	DRMFS	71.59	76.33	81.11	83.10
LDA	None	82.32	86.76	90.70	92.17
	Fisher	83.02	86.60	91.31	92.65
	ReliefF	82.39	87.00	91.06	92.21
	MRMR	82.43	86.63	90.99	92.28
	DRMFS	82.65	87.02	91.16	92.68
LDE	None	79.92	83.77	88.44	90.43
	Fisher	80.32	84.13	88.89	90.59
	ReliefF	80.09	84.58	89.17	90.50
	MRMR	79.91	83.70	88.41	90.48
	DRMFS	80.21	83.89	88.70	90.52
SVM	None	73.85	80.02	85.79	89.03
	Fisher	76.09	81.44	87.24	90.17
	ReliefF	74.84	81.80	87.47	90.23
	MRMR	76.14	81.57	86.91	89.53
	DRMFS	75.15	80.86	87.14	89.89
Proposed	**FE_GD**	87.10	90.04	93.75	95.02
	FE_GD_HI	**88.42**	**91.21**	**93.81**	**95.09**

Table 7 Classification performance (%) without and with the PCA variant constraint

Dataset	Training samples	ICS_DLSR	RSLDA	Proposed method Without P		Proposed method With P	
				FE_GD	FE_GD_HI	FE_GD	FE_GD_HI
USPS	30	88.46	89.45	89.50	90.24	89.50	90.29
	40	90.16	91.11	91.11	91.31	91.81	91.46
	55	91.25	92.65	92.49	92.69	93.07	92.87
	65	91.53	92.89	93.51	93.36	93.71	93.49
Extended yale B	10	86.56	86.79	86.77	88.11	87.10	88.42
	15	89.53	89.93	89.90	90.95	90.04	91.21
	20	93.14	93.59	93.57	93.35	93.75	93.81
	25	94.50	94.92	94.92	94.62	95.02	95.09
Honda	10	70.79	69.90	69.88	72.12	70.16	72.14
	20	82.95	83.03	83.24	84.45	83.60	84.64
	30	88.20	89.04	88.91	90.12	89.41	90.12
	50	93.53	94.13	93.81	95.08	94.53	95.10

method mainly has two parameters to be configured, λ_1 and λ_2. Figure 5 shows the classification rates' variation using different parameter combinations of the proposed method. In other words, the same figure shows how changing λ_1 and λ_2's values affects the gradient method using the Extended Yale B, Honda and USPS datasets. Figure 5a, c and e show the classification performance variation of the Extended Yale B, Honda and USPS datasets when using 10, 20 and 40 samples for training from each class respectively using the first variant of the proposed method **FE_GD**. The classification rate is also studied on the same datasets using the same training percentages for the second variant of the proposed method **FE_GD_HI** and results are depicted in Fig. 5b, d and f.



(a) Extended Yale B using **FE_GD**.

(b) Extended Yale B using **FE_GD_HI**

(c) Honda using **FE_GD**

(d) Honda using **FE_GD_HI**

(e) USPS using **FE_GD**

(f) USPS using **FE_GD_HI**

Fig. 5 Classification accuracy (%) according to parameters

For the Extended Yale B dataset, we studied different values for λ_1 in the range of $[10^{-5}, 1]$ and values from $[10^{-3}, 10]$ for λ_2 in the two variants; we noticed that satisfactory rates for the Extended Yale B dataset can be obtained using λ_1 in the range of $[10^{-3}, 10^{-1}]$ and λ_2 in the range of $[10^{-2}, 10^{-1}]$.

With regards to the Honda dataset, we studied different values for λ_1 in the range of $[10^{-3}, 10^3]$ and values from $[10^{-4}, 10^3]$ for λ_2; we noticed that satisfactory rates for this dataset can be obtained using λ_1 in the range of $[10^{-1}, 10]$ and λ_2 in the range of $[10^{-3}, 10^2]$.

For the USPS dataset, satisfactory rates can be obtained when λ_1 lies in the range of $[10^{-5}, 10^-1]$ and $\lambda_2 \in [10^{-5}, 10^2]$. As a conclusion, we can say that in order to obtain a satisfactory rate using the proposed method, the parameters λ_1 and λ_2 should lie in the intervals shown in the figures above. A value of 0.1 for both λ_1 and λ_2 seems to be a good choice for the two variants.

Figure 5 shows the variation of the classification accuracy rates (%) according to the different values of the parameters λ_1 and λ_2 for the variants of the proposed method, for the Extended Yale B, Honda and USPS datasets using 10, 20 and 40 samples from each class for training respectively and the rest for testing.

4.4 Analysis of results

From the results depicted in this paper's tables and figures, we can have the following observations:

1. The proposed and competing methods' classification accuracy demonstrates that our method has out-performed competing methods in most of the cases.
2. The first proposed method **FE_GD** has slightly outperformed the RSLDA method. This seems to be very realistic since the proposed first method refines the RSLDA solution.
3. In general, the second proposed method **FE_GD_HI** is superior to the first proposed method **FE_GD**. It benefits from the hybrid combination of two different embedding methods as well as from the refinement provided by the gradient descent tool.
4. The proposed method has a superior performance when used with several types of image datasets, including faces, objects and digits (Tables 3 and 4).
5. From Fig. 5, we can see that the proposed method's optimal parameters, that gives the best classification rates have large ranges. In other words, the best classification rate is often guaranteed by searching a small number of parameter combinations.
6. From Fig. 4 and Table 6, we can clearly observe that our proposed method outperformed other pure feature selection methods. This is due to the fact that our proposed method implicitly performs feature ranking alongside its main objective, feature extraction.
7. Through the observation of the results presented in Table 7, we can see that the classification performance in the case of the PCA constraint's removal from the objective function is lower. Hence, we can conclude that the PCA variant has contributed in enhancing our proposed method's discrimination leading to better outcomes.

5 Conclusion

In this paper, we introduced a novel linear method aimed to obtain a discriminant linear transform. The obtained linear transformation encapsulates two different types of discrimination, namely the inter-class sparsity and robust LDA. We deployed an iterative alternating minimization scheme to estimate the linear transform and the orthogonal matrix associated with the robust LDA. The linear transform is efficiently updated via the steepest descent gradient technique.

We proposed two initialization scheme for the linear transform. The first scheme sets the initial solution by the linear transform obtained by the robust sparse LDA method (RSLDA). The second variant initializes the solution via the hybrid combination of the two transformations obtained by the RSLDA and ICS_DLSR methods. The proposed method's two variants have demonstrated superiority over competing methods and have led to a more discriminative transformation. The proposed framework is generic in the sense that it allows the combination and tuning of other linear discriminant embedding methods. Like any other supervised learning technique, our method requires all of the data labels to be collected in advance, which is hard in some real life scenarios, this is our proposed method's main limitation. As a future work, the proposed method may be transformed into a semi-supervised learning algorithm where labeled and unlabeled data are used for training. Another idea that can be implemented, is transferring our proposed model to a deep model.

References

1. Belous G, Busch A, Gao Y (2020) Dual subspace discriminative projection. Pattern Recognition, pp 107581
2. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends®, in Machine Learning 3(1):1–122
3. Chang C-C, Lin C-J (2011) Libsvm: a library for support vector machines. ACM Trans Intel Syst Technol (TIST) 2(3):27
4. Chen C-F, Wei C-P, Wang Y-CF (2012) Low-rank matrix recovery with structural incoherence for robust face recognition. In: 2012

IEEE conference on computer vision and pattern recognition, IEEE, pp 2618–2625

5. Chen H-T, Chang H-W, Liu T-L (2005) Local discriminant embedding and its variants. In: 2005 IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), vol 2, IEEE, pp 846–853
6. Chen W (2020) Mutualinfo(x, y,nBins, ifplot). MATLAB Central File Exchange
7. Clemmensen L, Hastie T, Witten D, Ersbøll B (2011) Sparse discriminant analysis. Technometrics 53(4):406–413
8. Cunningham P, Delany SJ (2007) k-nearest neighbour classifiers. Multiple Classifier Systems 34(8):1–17
9. Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Ranzato M, Senior A, Tucker P, Yang K et al (2012) Large scale distributed deep networks. In: Advances in neural information processing systems, pp 1223–1231
10. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
11. Duda RO, Hart PE, Stork DG (2012) Pattern classification. John Wiley & Sons
12. Fan Z, Xu Y, Zhang D (2011) Local linear discriminant analysis framework using sample neighbors. IEEE Trans Neural Netw 22(7):1119–1132
13. Fang X, Han N, Wu J, Xu Y, Yang J, Wong WK, Li X (2018) Approximate low-rank projection learning for feature extraction. IEEE Trans Neural Netw Learn Syst 29(11):5228–5241
14. Fang X, Teng S, Lai Z, He Z, Xie S, Wong WK (2017) Robust latent subspace learning for image classification. IEEE Trans Neural Netw Learn Syst 29(6):2502–2515
15. Gao L, Yang B, Du Q, Zhang B (2015) Adjusted spectral matched filter for target detection in hyperspectral imagery. Remote Sens 7(6):6611–6634
16. He L, Yang H, Zhao L (2019) Tensor subspace learning and classification: Tensor local discriminant embedding for hyperspectral image. In: Proceedings of the IEEE international conference on computer vision workshops, pp 0–0
17. He X, Cai D, Yan S, Zhang H-J (2005) Neighborhood preserving embedding. In: Tenth IEEE international conference on computer vision (ICCV'05) vol 1, vol 2, IEEE, pp 1208–1213
18. He X, Niyogi P (2004) Locality preserving projections. In: Advances in neural information processing systems, pp 153–160
19. Hu J, Li Y, Gao W, Zhang P (2020) Robust multi-label feature selection with dual-graph regularization. Knowledge-Based Systems, pp 106126
20. Imani M, Ghassemian H (2017) High-dimensional image data feature extraction by double discriminant embedding. Pattern Anal Applic 20(2):473–484
21. Kozma L (2008) k nearest neighbors algorithm (knn). Helsinki University of Technology
22. Lai Z, Xu Y, Jin Z, Zhang D (2014) Human gait recognition via sparse discriminant projection learning. IEEE Trans Circuits Syst Video Technol 24(10):1651–1662
23. Langley P (1994) Selection of relevant features in machine learning: Defense technical information center
24. Li Z, Liu J, Yang Y, Zhou X, Lu H (2013) Clustering-guided sparse structural learning for unsupervised feature selection. IEEE Trans Knowl Data Eng 26(9):2138–2150
25. Liu G, Yan S (2011) Latent low-rank representation for subspace segmentation and feature extraction. In: 2011 International conference on computer vision, IEEE, pp 1615–1622
26. Lu Y, Lai Z, Li X, Wong WK, Yuan C, Zhang D (2018) Low-rank 2-d neighborhood preserving projection for enhanced robust image representation. IEEE Trans Cybern 49(5):1859–1872
27. Martínez AM, Kak AC (2001) Pca versus lda. IEEE Trans Pattern Anal Mach Intel 23(2):228–233
28. Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. Image Vis Comput 22(10):761–767
29. Peng H, Ding C, Long F (2005) Minimum redundancy-maximum relevance feature selection
30. Peng X, Lu J, Yi Z, Yan R (2016) Automatic subspace learning via principal coefficients embedding. IEEE Trans Cybern 47(11):3583–3596
31. Qiao Z, Zhou L, Huang JZ (2009) Sparse linear discriminant analysis with applications to high dimensional low sample size data. Int J Appl Math 39(1):6
32. Quinlan JR (2014) *C4.* 5: programs for machine learning. Elsevier
33. Raileanu LE, Stoffel K (2004) Theoretical comparison between the gini index and information gain criteria. Ann Math Artif Intell 41(1):77–93
34. Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of relieff and rrelieff. Mach Learn 53(1-2):23–69
35. Smith LI (2002) A tutorial on principal components analysis. Technical report
36. Stańczyk U., Zielosko B, Jain LC (2018) Advances in feature selection for data and pattern recognition: an introduction. In: Advances in feature selection for data and pattern recognition, Springer, pp 1–9
37. Tao H, Hou C, Nie F, Jiao Y, Yi D (2015) Effective discriminative feature selection with nontrivial solution. IEEE Trans Neural Netw Learn Syst 27(4):796–808
38. Tharwat A, Gaber T, Ibrahim A, Hassanien AE (2017) Linear discriminant analysis: a detailed tutorial. AI Commun 30(2):169–190
39. Unar S, Wang X, Wang C, Wang Y (2019) A decisive content based image retrieval approach for feature fusion in visual and textual images. Knowl-Based Syst 179:8–20
40. Unar S, Wang X, Zhang C (2018) Visual and textual information fusion using kernel method for content based image retrieval. Information Fusion 44:176–187
41. Unar S, Wang X, Zhang C, Wang C (2019) Detected text-based image retrieval approach for textual images. IET Image Process 13(3):515–521
42. Wang C, Wang X, Li Y, Xia Z, Zhang C (2018) Quaternion polar harmonic fourier moments for color images. Inf Sci 450:141–156
43. Wang C, Wang X, Xia Z, Ma B, Shi Y-Q (2019) Image description with polar harmonic fourier moments. IEEE Transactions on Circuits and Systems for Video Technology
44. Wang C, Wang X, Xia Z, Zhang C (2019) Ternary radial harmonic fourier moments based robust stereo image zero-watermarking algorithm. Inf Sci 470:109–120
45. Wang D, Nie F, Huang H (2015) Feature selection via global redundancy minimization. IEEE Trans Knowl Data Eng 27(10):2743–2755
46. Wang X, Wang Z (2014) The method for image retrieval based on multi-factors correlation utilizing block truncation coding. Pattern Recogn 47(10):3293–3303
47. Wen J, Fang X, Cui J, Fei L, Yan K, Chen Y, Xu Y (2018) Robust sparse linear discriminant analysis. IEEE Trans Circuits Syst Video Technol 29(2):390–403
48. Wen J, Xu Y, Li Z, Ma Z, Xu Y (2018) Inter-class sparsity based discriminative least square regression. Neural Netw 102:36–47
49. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2008) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intel 31(2):210–227
50. Xiang S, Nie F, Meng G, Pan C, Zhang C (2012) Discriminative least squares regression for multiclass classification and feature selection. IEEE Trans Neural Netw Learn Syst 23(11):1738–1754

51. Xu J, Tang B, He H, Man H (2016) Semisupervised feature selection based on relevance and redundancy criteria. IEEE Trans Neural Netw Learn Syst 28(9):1974–1984
52. Xu Y, Fang X, Zhu Q, Chen Y, You J, Liu H (2014) Modified minimum squared error algorithm for robust classification and face recognition experiments. Neurocomputing 135:253–261
53. Xue Y, Zhang L, Wang B, Zhang Z, Li F (2018) Nonlinear feature selection using gaussian kernel svm-rfe for fault diagnosis. Appl Intell 48(10):3306–3331
54. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: 2009 IEEE Conference on computer vision and pattern recognition, IEEE, pp 1794–1801
55. Yang J-B, Ong C-J (2012) An effective feature selection method via mutual information estimation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42(6):1550–1559
56. Ye J (2007) Least squares linear discriminant analysis. In: Proceedings of the 24th international conference on machine learning, pp 1087–1093
57. Zang S, Cheng Y, Wang X, Ma J (2019) Semi-supervised flexible joint distribution adaptation. In: Proceedings of the 2019 8th international conference on networks, communication and computing, pp 19–27
58. Zhang L, Yang M, Feng X (2011) Sparse representation or collaborative representation: Which helps face recognition?. In: 2011 International conference on computer vision, IEEE, pp 471–478
59. Zhang X, Chu D, Tan RC (2015) Sparse uncorrelated linear discriminant analysis for undersampled problems. IEEE Trans Neural Netw Learn Syst 27(7):1469–1485
60. Zhang Y, Jiang Z, Davis LS (2013) Learning structured low-rank representations for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 676–683
61. Zhao Z, He X, Cai D, Zhang L, Ng W, Zhuang Y (2015) Graph regularized feature selection with data reconstruction. IEEE Trans Knowl Data Eng 28(3):689–700
62. Zhou Y, Sun S (2016) Manifold partition discriminant analysis. IEEE Trans Cybern 47(4):830–840
63. Zhu R, Dornaika F, Ruichek Y (2019) Joint graph based embedding and feature weighting for image classification. Pattern Recogn 93:458–469
64. Zhu R, Dornaika F, Ruichek Y (2019) Learning a discriminant graph-based embedding with feature selection for image categorization. Neural Netw 111:35–46
65. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2):301–320
66. Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 15(2):265–286

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A. Khoder received the Bachelor of Science degree in Computer Engineering and the Master of Science degree in Computer and Communication Engineering from the Lebanese International University, Lebanon, in 2017. Currently, he is pursuing his Ph.D. degree at the University of the Basque Country, San Sebastian. His research interests include machine learning, pattern recognition, computer vision and artificial intelligence.

F. Dornaika received his engineer degree in Electrical Engineering from the Lebanese University, in 1990, an M.S. degree in signal, image and speech processing from Grenoble Institute of Technology, France, in 1992, and a Ph.D. degree in computer science from Grenoble Institute of Technology, France and INRIA, in 1995. He is currently a Research Professor at IKERBASQUE (Basque Foundation for Science) and the University of the Basque Country. Prior to joining IKERBASQUE, he held numerous research positions in Europe, China, and Canada. He has published more than 300 papers in the field of computer vision and pattern recognition. His research covers a wide range of topics in computer vision. His current research interests include machine learning, pattern recognition, and data mining.

CHAPTER 4

Ensemble Learning via Feature Selection and Multiple Transformed Subsets: Application to Image Classification

Ensemble Learning via Feature Selection and Multiple Transformed Subsets: Application to Image Classification

A. Khoder[1] and F. Dornaika[1,2]

[1] University of the Basque Country UPV/EHU, San Sebastian, SPAIN
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, SPAIN

Abstract

In the machine learning field, especially in classification tasks, the model's design and construction are very important. Constructing the model via a limited set of features may sometimes bound the classification performance and lead to non-optimal results that some algorithms can provide. To this end, Ensemble learning methods were proposed in the literature. These methods' main goal is to learn a set of models that provide features or predictions whose joint use could lead to a performance better than that obtained by the single model. In this paper, we propose a new efficient ensemble learning approach that was able to enhance the classification performance of a linear discriminant embedding method. As a case study we consider the efficient "Inter-class sparsity discriminative least square regression" method. We seek the estimation of an enhanced data representation. Instead of deploying multiple classifiers on top of the transformed features, we target the estimation of multiple extracted feature subsets obtained by multiple learned linear embeddings. These are associated with subsets of ranked original features. Multiple feature subsets were used for estimating the transformations. The derived extracted feature subsets were concatenated to form a single data representation vector that is used in the classification process. Many factors were studied and investigated in this paper including (Parameter combinations, number of models, different training percentages, feature selection methods combinations, etc.). Our proposed approach has been benchmarked on different image datasets and achieved competitive results. The conducted experiments showed that the proposed approach can enhance the classification performance in an efficient manner compared to the single-model based learning and was able to outperform competing methods.

Keywords: Ensemble learning, feature subsets, multi-models, machine learning, feature selection, image classification, class sparsity least square regression.

1 Introduction

Image classification is a widely investigated task in the machine learning and computer vision fields. Many researchers worked and focused on the implementation of both linear and non-linear models designed for classification tasks. Achieving reliable discriminative data representations is the objective in all the cases. It is a known fact that a more discriminative data representation will lead to enhanced classification performance. This is where the importance of engaging relevant data features in the model creation rises. Nowadays, representation learning is becoming more and more investigated [30, 33, 42, 43, 49, 57, 58]. Data features are usually separated into three categories, important (relevant), irrelevant or redundant. A good model should always target relevant features of the data and work on constructing the desired model using these features. This will ensure optimal classification performance.

Generally, specific features will ensure better representation for the data rather than other ones. These are referred to as relevant features. Authors in [18, 39] has concluded that using the original data would not lead to the optimal classification performance in the learning applications. This should be addressed by extracting the most representative features from the original data. Data can then be analysed via the extracted features. In addition to the problem that original data are not the best to work with, there exist another problem namely: curse of dimensionality, referring to the large number of features in the data. In real life and in specific applications, the dimension of the data can be very large which makes their use very costly, both in time and computation wise. Various researchers focused on tackling this issue by using two main approaches namely: feature selection, and feature extraction. In these days, these schemes are highly targeted and play a major role in learning systems [28].

Researchers seek representation approaches that guarantee the delivery of a discriminative transformation matrix that has certain specifications and good discrimination abilities [13, 21, 50, 52]. After that, one can use this transformation matrix to project the training and test data to the new derived space in order to obtain a new and more representative set of features. These features will be used in the construction of the model that will be then used in the classification tasks.

In the literature, one can notice that most of the time single model based classifications were targeted and investigated. In other words, researchers work on proposing and implementing an algorithm in the purpose of achieving a good discriminative model that ensures good classification performance. Usually, in this process, what happens is that a model is created using the proposed algorithm, and then the output data is fed to a classifier for classification process to begin. In order to enhance the performance, one can use many known feature selection techniques (eg. Fisher score, ReliefF [26] and many more). Feature selection techniques have been widely used in the machine learning field [4]. In addition to that, one can perform a brutal search for the best features that are able to ensure the best classification performance provided by the proposed scheme, but still notice that the optimal performance was not achieved. In reality, it is not necessary that single model learning will always lead to the optimal performance provided by a proposed method.

To address this issue, and investigate how to improve the performance of different methods, few researches talked about the ensemble learning methods. An Ensemble learning combines the predictions from multiple machine learning models into a single model which can reduce the generalization error. They offer increased flexibility and can scale in proportion to the amount of training data available. A couple of widely used ensemble approaches are bagging [3] and boosting [36].

The main idea of ensemble learning is to blend and combine the predictions from multiple models. These models are usually very good models and each one of them, taken separately, provides a good discriminant characteristic. By combining these models, one will obtain a single model that is described by its enhanced discrimination ability. Thus, leading to a better classification. So, the hypothesis is that in the case where the models are correctly combined, this can lead to more accurate and/or robust models. A variety of ensemble learning methods have been used in classification tasks mostly with deep convolutional neural networks (CNN's) for image classification. The reason is that ensemble learning has shown promising and excellent contribution in enhancing the performance of neural networks [11].

The performance of one single model is usually measured by its ability of obtaining the best predictor for the data. This can only be derived after the classification process finishes. There is no way to realize this information prior to that by only exploiting the handled data and the optimization problem [29]. This has been addressed in [41, 29]. These researches focused on using a cross-validation strategy to evaluate the performance of each model individually. This strategy is referred to as the "discrete Super Learner selector".

One different view to ensure an enhanced performance can be the estimation of the optimal combination of the models that leads to the best predictor. This is well investigated in the literature. Brieman in [3] addressed and condensed several related works regarding the theoretical properties of ensemble learning [2, 14, 16, 44, 46]. Another well-known strategy used in ensemble learning is called ”stacking” [53], it involves combining the predictions from multiple models on the same dataset. Many researchers have proposed linear combination approaches that introduced stacking to the ensemble of models [53, 3].

In order to derive the most efficient combination of models, the work described in [3] investigated stacked regression by using cross-validation. The cross-validation based work has been expanded in the purpose of finding the best combination of predictors by proposing the ”Super Learner” approach [29]. This framework demonstrated superiority and very good contributions in multiple areas namely: online learning [1], medicine [37, 54], spatial prediction applications [10] in addition to mortality prediction [6, 40].

In this paper, we propose a new framework used for supervised classification tasks. Instead of using an ensemble of classifiers, we propose the use of an ensemble of data representations. Our proposed approach is based on ensemble learning. The proposed approach creates multiple subsets of original features; these subsets are carefully chosen by using a single or multiple feature selection techniques. For each subset, a projection model (feature extraction) is built in order to get the transformed features. At the final stage, all transformed features are concatenated and used as a single large data representation that feed a classifier.

We make sure that the features of the data are ranked according to their importance by subjecting them to multiple feature selection techniques. In the way we have chosen to construct the features subsets, the most relevant features of the data were taken into consideration every time. Every created subset that we have used contains the most relevant features of the data overlapped with different features every time. In this way, even in the case where the chosen feature subset contains less relevant features, these features are there alongside with the most relevant ones and not alone. Moreover, due to the adopted feature ranking, the most relevant features will be used in several projection models.

The main idea of the proposed approach is generic and can be used by various methods. However, we have chosen the ”Inter-class sparsity based discriminative least square regression” denoted as (ICS_DLSR) [52] as a backbone projection algorithm. This is motivated by (1) its remarkable discriminating ability, (2) efficient projection model computation, and (3) economic size of transformed features. The use of several feature selection techniques led to multiple variants of the proposed scheme. In brief, the paper has the following contributions:

- Proposing an ensemble of models based learning approach that improved the classification performance compared to single model learning.
- Studying the effect of the introduction of hybrid combination of multiple feature selection techniques into one single model.

The remainder of the paper is divided as follows: section 2 will show the preliminaries. Section 3 is intended to describe the methodology of our proposed scheme. Section 4 will present the experimental results and method evaluation. Finally section 5 concludes the paper.

2 Preliminaries

In current times, achieving an efficient data representation is the focus of many researches. Many studies are conducted for this purpose, and good methods have been delivered by various researchers [50, 52, 13, 57, 58]. To be able to test our ensemble learning based approach, we have chosen to use the " inter-class sparsity discriminative least square regression " (ICS_DLSR) [52] approach for multiple considerations. ICS_DLSR is an efficient method for both training and testing. It is flexible and has good discrimination properties. In this section, we will briefly describe some preliminaries. We will review the ICS_DLSR method and talk about the adopted feature selection techniques used for ranking the data features.

2.1 Notations

We will proceed with the presentation of the notations used in our article. The training set is denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, whith d being the dimension of the samples. Each sample $\mathbf{x}_i$ is represented by a column vector consisting of 'd' features $\in \mathbb{R}^d$. N denotes the number of training samples. The total number of classes is denoted by C. The projection matrix is denoted as $\mathbf{Q} \in \mathbb{R}^{C \times d}$, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N] \in \mathbb{R}^{C \times d}$ is the label matrix corresponding to the training set $\mathbf{X}$, where each column vector $\mathbf{y}_i \in \mathbb{R}^{C\times 1}$ is simply defined as follows: if training sample $\mathbf{x}_i$ belongs to the k-th class, then the k-th element of column vector $\mathbf{y}_i$ is 1 while the remaining elements are 0.

Table 1 illustrates the $\ell_{2,1}$ and Frobenius (ℓ_F) norm computation for a matrix $\mathbf{Z} \in \mathbb{R}^{C\times d}$, where $\mathbf{Z}_{ij}$ denotes the (i, j)-th element of matrix $\mathbf{Z}$.

Table 1: Matrix norms.

Type	Formula
$\ell_{2,1}$ norm	$\|\mathbf{Z}\|_{2,1} = \sum_{i=1}^{C} \sqrt{\sum_{j=1}^{d} Z_{ij}^2}.$
ℓ_F norm	$\|\mathbf{Z}\|_F = \sqrt{\sum_{i=1}^{C} \sum_{j=1}^{d} Z_{ij}{}^2}.$

2.2 Review of Inter-class sparsity discriminative least square regression (ICS_DLSR) [52]:

Original Least Square Regression (LSR) only focuses on fitting the input features to the corresponding output labels but still ignores the correlations among samples. LSR has been effective and proved very good contribution in many applications like gene classification [32], cancer classification [17], face recognition [55], image retrieval [15] and speech recognition [23].

Based on the LSR framework, the authors in [52] proposed the Inter-class sparsity discriminative least square regression (ICS_DLSR) method in order to obtain a more discriminative and compact projection space. This proposed framework imposed an inter-class sparsity constraint on the projected data which ensures that the derived projected data obtain common class structure. In addition, the authors introduced an error term with row-sparsity constraint to relax the strict zero–one label matrix. This allowed ICS_DLSR to be more flexible in the learning process. ICS_DLSR achieved superior performance and proved to be effective on many datasets. It aims to minimize the following problem:

$$\min_{\mathbf{Q},\mathbf{E}} \frac{1}{2}\|\mathbf{Y} + \mathbf{E} - \mathbf{QX}\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{Q}\|_F^2 + \lambda_2 \sum_{i=1}^{C} \|\mathbf{QX}_i\|_{2,1} + \lambda_3 \|\mathbf{E}\|_{2,1} \tag{1}$$

In Eq. (1), $\mathbf{Q}$, $\mathbf{X}$ $\mathbf{E}$ and $\mathbf{Y}$ represent the linear transformation matrix, the data samples matrix, the error matrix, and the label matrix, respectively. λ_1, λ_2 and λ_3 are three parameters that determine the effect of the corresponding terms. C denotes the total number of classes. The matrix

$\ell_{2,1}$ norm is used to promote the row-sparsity of a matrix. In this optimization problem, there are two unknown variables the linear transformation and the error matrix. To solve the problem, the authors adopted the alternating direction method of multipliers (ADMM) [34, 35, 56] to obtain the solution for **Q** and **E**.

2.3 Feature Selection techniques

In machine learning and computer vision, feature quality assessment is an important topic

In most of the learning problems, there exist hundreds or thousands of features describing each object. These features can either enhance the learning, or at particular occasions worsen it. For the purpose of ensuring the optimal learning performance, we should select the subset containing the most relevant features of the data. By doing so, one can enhance the performance and decrease the computational cost at the same time. Therefore, the problem of feature (attribute) selection has received much attention in the literature. Selecting the most relevant features of the data can be implemented using what is known by feature selection techniques.

- **Feature selection using Fisher score:**

  Generally, feature selection approaches main objective is selecting and highlighting the set of the relevant features of the original data. This selected subset of features is normally used to construct a more robust and compact model. Hence, leading to superior classification performance. Fisher score is one of the most famous algorithms used for feature selection, it works by computing the score of each data feature and then selects each feature accordingly.

  Fisher algorithm computes the score of the i-th feature S_i by the following formula:

$$S_i = \frac{\sum_{j=1}^{C} n_j (\mu_{ij} - \mu_i)^2}{\sum_{j=1}^{C} n_j \rho_{ij}^2} \tag{2}$$

  where ρ_{ij} and μ_{ij} represent the variance and the mean of the i-th feature associated with the j-th class. The number of instances in the j-th class is denoted by n_j and μ_i is the mean of the i-th feature. C is the number of classes.

- **Feature selection using ReliefF score:**

Original Relief Algorithm Another well-known algorithm that enables features ranking is the Relief algorithm. The majority of the approaches used for approximating the reliability of the attributes presume the conditional independence of the attributes and are thus less suitable for problems that might involve more feature interaction. Relief based algorithms (Relief, ReliefF and RReliefF) do not simply make this assumption [24, 26, 25].

These algorithms are reliable, conscious of the contextual information, and can effectively estimate the quality and the relevance of attributes in problems with high attribute dependency. Relief algorithms are based on the concept of local margins for each feature. These margins should be large enough for relevant features. These algorithms are widely considered as feature subset selection methods used in the pre-processing phase before the model is trained [24]. They are still one of the most popular pre-processing algorithms to date [12]. They are actually general feature estimators which have been successfully used in a multitude of environments. Inspired by instance-based learning, the authors in [24] proposed the classical Relief algorithm. Relief is optimized for two-class problems. The basic principle of the algorithm is to consider not just the disparity in features values and the variance in the classes but also the distance between the instances.

Let us consider the feature vector **v** and the feature vectors of the instance closest to **v** from each class. The closest instance belonging to the same group is referred to as near-hit (NH), and the closest instance with a different group is denoted as near-miss (NM).

Relief Algorithm [26] iteratively computes the weight for the *i*-th feature by:

$$W_i = W_i - (V_i - \mathrm{NH}_i)^2 + (V_i - \mathrm{NM}_i)^2 \tag{3}$$

ReliefF Algorithm Authors in [26] improved the Relief algorithm. They developed an extension of the original Relief, called ReliefF, that improves the original algorithm by estimating margins more reliably. Irrelevant attributes either the redundant or noisy ones may affect the selection of the nearest neighbors. Thus, the estimation of the margins becomes unreliable. To address this problem, ReliefF searches for the ”k” nearest (NH’s) and (NM’s) rather than a single (NH and NM) and averages the contribution of all k nearest (NH’s) and (NM’s). The selection of the nearest neighbors is very important in Relief-F. The purpose is to find the nearest neighbors with respect to important attributes. In all our experiments, ”k” was set to 10 which, empirically, gives satisfactory results. In some problems significantly better results can be obtained in case of tuning ”k” (as is typical for the majority of machine learning algorithms). Many studies were conducted to explore the feature selection ability using ReliefF algorithm [45]. More deails about Relief variants can be found in [19].

- **Feature selection using Robust multi-label feature selection with dual-graph regularization:**

Authors in the [20] proposed a novel dual-graph regularization based feature selection method called ”Robust multi-label feature selection with dual-graph regularization” (DRMFS). The proposed algorithm differ from the existing methods by incorporating only a single unknown variable (feature weight matrix) in its global criterion. In addition, the designed approach is described by its capability of achieving a global optimal solution, compared to most of the competing methods with multiple unknown variables and their ability of only achieving local optimal solutions. DRMFS was designed based on feature graph regularization and label graph regularization, jointly. The former preserves the geometric structure of features, while the latter addresses the correlations of the data labels. Authors imposed the $\ell_{2,1}$ norm constraint on both the loss function and the weight matrix to improve the robustness of the method and ensure the row sparsity property. The objective function of the DRMFS algorithm is as follows:

$$\min_{\mathbf{W}} ||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_{2,1} + \alpha Tr(\mathbf{W}^T\mathbf{L}^X\mathbf{W}) + \beta Tr(\mathbf{W}\mathbf{L}^Y\mathbf{W}^T) + \gamma||\mathbf{W}||_{2,1} \qquad s.t. \mathbf{W} \geq 0. \qquad (4)$$

where $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{Y}$ denote the data, feature weight and label matrices, respectively. α, β and γ are three regularization parameters. $\mathbf{L}^X$ and $\mathbf{L}^Y$ represent the feature graph and label graph Laplacian matrices, accordingly.

Once the feature weight matrix $\mathbf{W}$ is computed, the score of each feature is given by $||\mathbf{W}_{i*}||_2$ $(1 \leq i \leq d)$, where d denotes the dimensionality. It is possible to retrieve the most relevant top-k features according to the highest scores $(k \leq d)$.

Additional detailed information about this proposed method is presented at [20].

3 Proposed Ensemble Class Sparsity Discriminative Regression

In this section, we will describe our ensemble learning based approach. We will present the different phases of the process and the model construction.

3.1 Steps and Methodology

Let us consider the data matrix $\mathbf{X} \in \mathbb{R}^{d \times N}$ where d and N represents the dimension (number of features) of the original data and the total number of samples, respectively. First, we apply one of the feature selection techniques over the original data.

- The score of each feature is computed (by one of the selection techniques stated above) and then features are ranked according to their scores. In this way, most relevant features, which are usually the ones with highest scores are placed at the top while the ones with lower scores are placed at the bottom. A graphical illustration of this weighting and ranking process is shown in Figure 1. We denote the ranked features data matrix by $\mathbf{X}_s \in \mathbb{R}^{d \times N}$.

- Subsequent to the feature ranking process, we start by constructing our subsets of features. We construct multiple feature subsets in a way that each one is unique (coming from taking different percentages of features from the data matrix with ranked features) as it is shown in the upper part of Figure 2. In its simplest implementation, the number of percentages defines the number of models, M. According to this scheme, the most relevant features of the data are taken into consideration in more than one subsets. Every created subset contains the most relevant features of the data overlapped with different features every time. Thus, even in the case where the chosen feature subset contains less relevant features, these features are there alongside with the most relevant ones and not alone. This ensures that no feature subset taken into consideration would harm the learning process.

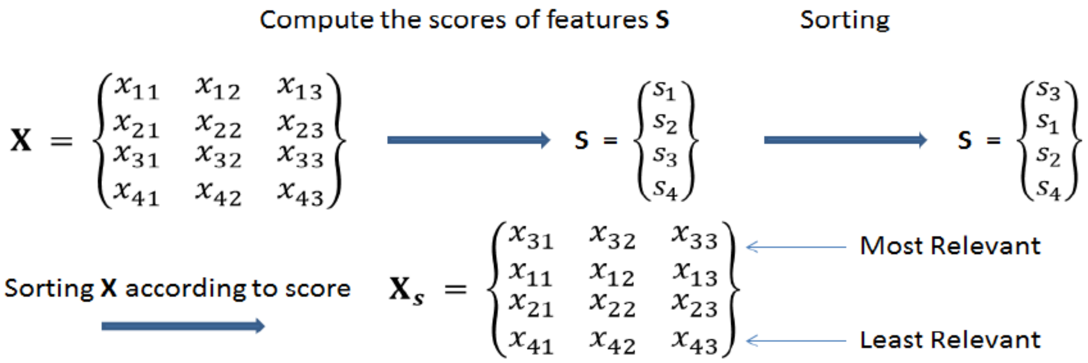

Compute the scores of features S
Sorting
$\mathbf{X} = \begin{Bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{Bmatrix}$
$\mathbf{S} = \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{Bmatrix}$
$\mathbf{S} = \begin{Bmatrix} s_3 \\ s_1 \\ s_2 \\ s_4 \end{Bmatrix}$
Sorting X according to score
$\mathbf{X}_s = \begin{Bmatrix} x_{31} & x_{32} & x_{33} \\ x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{41} & x_{42} & x_{43} \end{Bmatrix}$
Most Relevant
Least Relevant

Figure 1: Features Ranking General Methodology.

- Let us consider creating M models. After generating the M subsets, the ICS_DLSR algorithm is applied on each subset that is fed as input data for the algorithm. In the ICS_DLSR algorithm process, each input generates a linear transformation matrix $\mathbf{Q}_n$ associated with this input. We have $n = 1, ..., M$.

- After obtaining the projection matrices $\mathbf{Q}_n$ delivered by ICS_DLSR, we can create our targeted data representations. We proceed by projecting each feature subset using the corresponding transformation $\mathbf{Q}_n$. Assuming that $\mathbf{X}$ represents the original data, after sorting according to the features scores this will be denoted as $\mathbf{X}_s$. Let $\mathbf{S}_n$ represents the data formed by the n-th subset of features, $\mathbf{S}_n \subset \mathbf{X}_s$. It worth noting that the training and test data are submitted to the same procedure. Projecting training and test samples using $\mathbf{Q}_n$ is implemented by $\mathbf{A}_n = \mathbf{Q}_n \mathbf{S}_n$ and $\mathbf{B}_n = \mathbf{Q}_n \mathbf{T}_n$, where $\mathbf{S}_n$ corresponds to the training data formed by the $n-$th feature subset and $\mathbf{T}_n$ represents the test samples having the same subset of features. This leads to M models formed by the obtained descriptors (projected data vectors) with $n = 1, ..., M$.

- In the final stage of the proposed approach, the obtained M models are concatenated to form a single data representation which is finally fed to a given classifier (e.g., the Nearest Neighbor classifier). Since ICS_DLSR is used as a projection model, the dimension of the projection space provided by each model $\mathbf{Q}_n$ is C, the dimension of the final representation is $M \times C$.

```
Algorithm. 1. ICS_DLSR Based Ensemble Learning for Image Classification

Inputs:   1. Data samples $\mathbf{X} \in \mathbb{R}^{d\times N}$
          2. Labels vector
          3. Number of models, M
          4. Percentages of subsets
          5. Parameters $\lambda_1$, $\lambda_2$, $\lambda_3$
          6. Feature selection technique

Steps:    1. Compute the scores and rank the features using one of
          the feature selection techniques (Fisher score, ReliefF, DRMFS, or other).

          2. Select subsets of features according to the pre-defined percentages.

          3. Apply the ICS_DLSR algorithm using each one of the extracted subsets of
          features as an input and derive the corresponding transformation matrices.

          4. Project the training and test data on the new space using the obtained projection
          matrices associated with each input and construct the targeted models out of the.
          transformed subsets.

          5. Concatenate the obtained transformed subsets to form a single data representation vector.

Output:   Data representation vector obtained by the concatenated models.
```

Figure 2 depicts a graphical illustration of the main steps of the proposed approach. For simplicity, the case of **three models** creation was adopted in the example provided by this figure. This figure demonstrates the full process which includes: ranking the original features of the data, subsets construction, model creation, concatenation, and classification. Algorithmic steps of the proposed approach are illustrated in Algorithm 1.

3.2 Proposed Variants

We have proposed three variants of our approach namely: (i) Ensemble of models Class sparsity based discrimination using Fisher score EM_ICS_FS, (ii) Ensemble of models Class sparsity based discrimination using Combined score EM_ICS_HS and (iii) Ensemble of models Class sparsity based discrimination using the ”Robust multi-label feature selection with dual-graph regularization” (DRMFS) algorithm [20] EM_ICS_DRMFS.

- Ensemble of models Class sparsity based discrimination using Fisher score **EM_ICS_FS**:

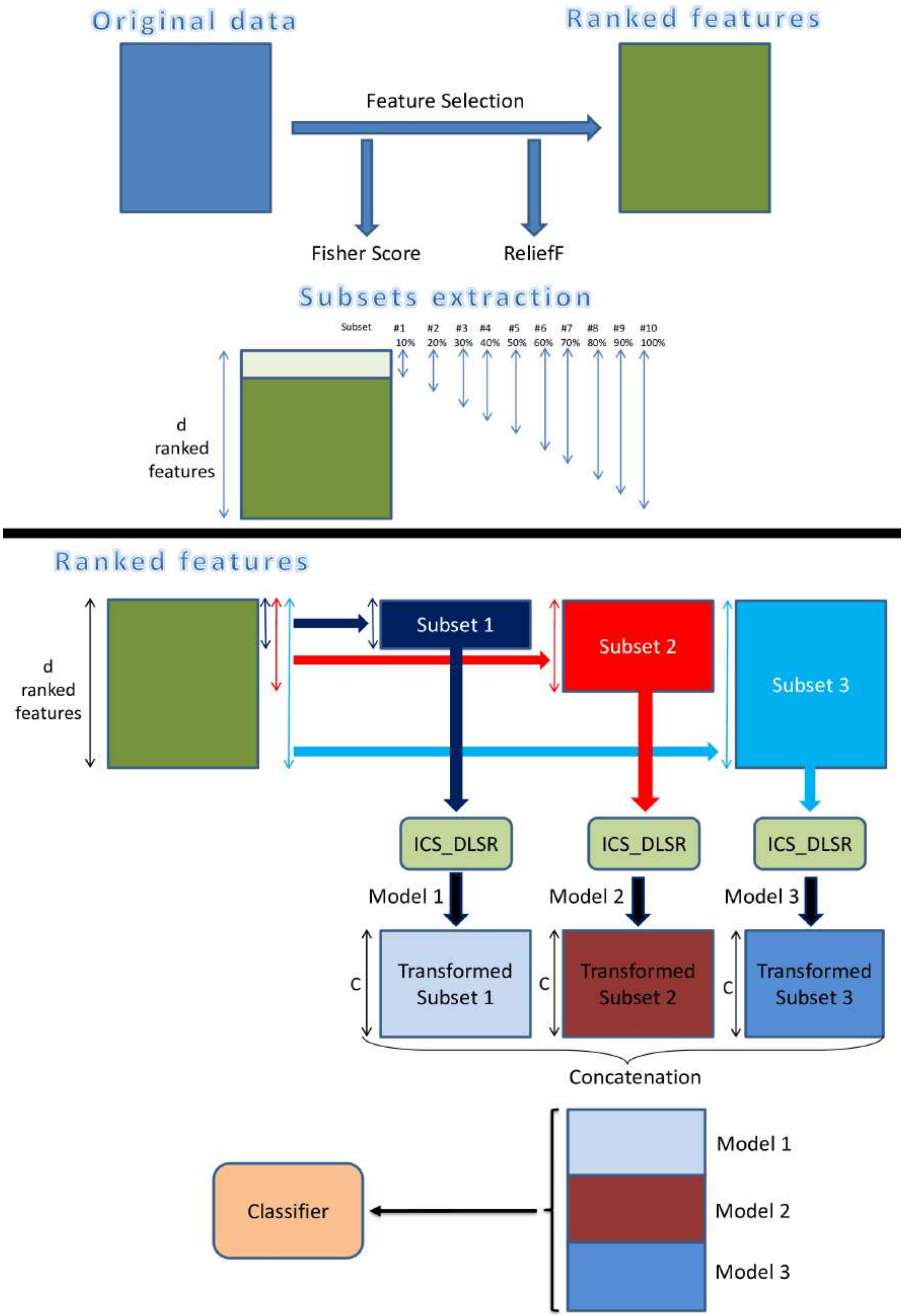


Figure 2: Proposed Ensemble Learning Methodology.

In this variant of the approach, we have constructed a total of **10 models** in which the proportions of the data features taken from the original data are 10%, 20%, 30%,...,100%, respectively. The data contained in these models were obtained after original features are ranked via the **Fisher Score** feature selection technique only. The methodology of the model creation procedure is described in Figure 2.

- Ensemble of models Class sparsity based discrimination using Combined score **EM_ICS_HS**: In this second variant, we have constructed a total of **10 models**. The main difference of this variant comes from the fact that the created models were obtained when the subsets of features were ranked using multiple feature techniques. In our experiments, **5 models** were created when the applied feature selection technique is the **Fisher Score** and the other **5 models** were constructed when we have applied **ReliefF** feature selection technique on the original data features. The proportions of the features taken from the data to construct the subsets for this variant are as follows [20%, 40%, 60%, 80%, 100%]. The methodology for the combined model creation is described in Figure 3.

- Ensemble of models Class sparsity based discrimination using DRMFS algorithm **EM_ICS_DRMFS**: We have constructed a total of **10 models** in which the proportions of the data features taken from the original data are 10%, 20%, 30%,...,100%, respectively. The data contained in these models were obtained after original features are ranked via the recently proposed **DRMFS** algorithm.

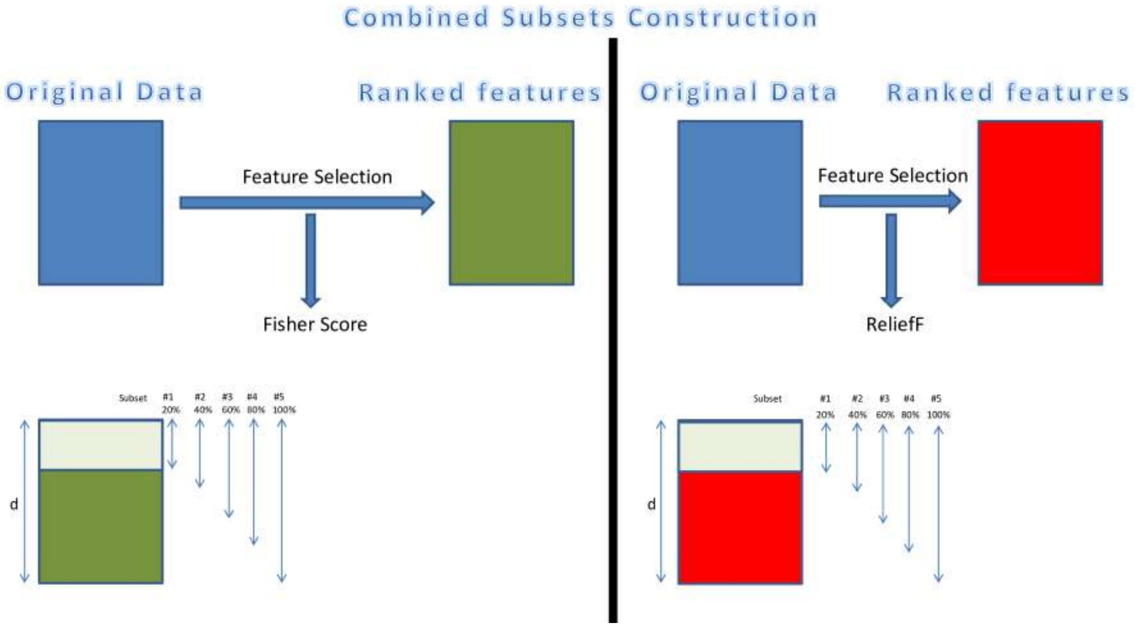


Figure 3: Combined Model Construction Methodology.

4 Experiments and Analysis

4.1 Datasets

This section will provide detailed information regarding the datasets used in the experiments presented in this paper. Faces, objects and scene image datasets with different sizes were tested using our proposed approach.

- **Extended Yale B Face Dataset[1]:** The database used in this paper in the condensed version of the original Extended yale B dataset. Images in this dataset represent the faces of 38 different individuals while each one of these individuals has between 58 and 64 image. These face images were taken in various illuminations conditions and with different facial expressions for each person. A total number of 2414 images were used, each image is rescaled to 32×32 pixels. Raw brightness images of dimension 1024 are used in the experiments for this dataset. Results were derived while using different training percentages. 10, 15, 20, and 25 samples from each class were used as training samples and the remaining are used for testing.

[1] *http : //vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html*

- **LFW-a Dataset[2]:** ”The Labeled Faces in the Wild-a (LFW-a)” is constructed from the images of the original LFW database after alignment using a commercial face alignment software. Images in this dataset maintained the same structure as in the original LFW dataset. This dataset contains a total of 3,408 image samples representing 141 classes. Raw brightness images of dimension 1,024 are used in the experiments. The reported results were obtained after we had varied the training percentage while using 5,6,7 and 8 image samples from each class as training samples. Remaining samples were used as test samples.

- **COIL20 Object Dataset[3]:** With the full name ”The Columbia Object Image Library”, COIL20 dataset contains images representing various objects. Each object is rotated around a vertical axis. It contains the images of 20 objects in which each object has 72 images, leading to a total number of 1,440 images. Local Binary Patterns (LBP) [31] are used as image descriptors in this dataset. We adopted the uniform LBP histogram (59 values). Three LBP descriptors are constructed from the image using 8 points and three values for the radius (R=1, 2, and 3 pixels). As a result, the final concatenated descriptor has 177 values. We varied the training samples percentage, in our experiments we took 20, 25, 30, and 35 image samples from each class for training and the remaining were used as testing portions.

- **Georgia Face dataset[4]:** This dataset contains face images corresponding to 50 persons, each individual is represented by 15 images describing frontal and tilted faces with different facial expressions, lighting conditions and scale. The total number of images included in this dataset is 750 images. The images used are cropped and resized to 32×32 pixel for each image. Raw-brightness images of dimension 1024 are used in the experiments. The reported results are obtained after we used 3, 5, 7, and 9 image samples from each class as training samples and the remaining are used as test samples.

- **FEI dataset[5]:** The stated dataset contains pictures of the students and staff members at FEI. It is a face dataset that contains a set of colorful face images taken against a white background. The images are in an upright frontal position with profile rotation of up to about 180 degrees. This dataset contains a total number of 700 images, 14 images for each

[2] https://talhassner.github.io/home/projects/lfwa/index.html

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4] $http://www.anefian.com/research/face_reco.htm$

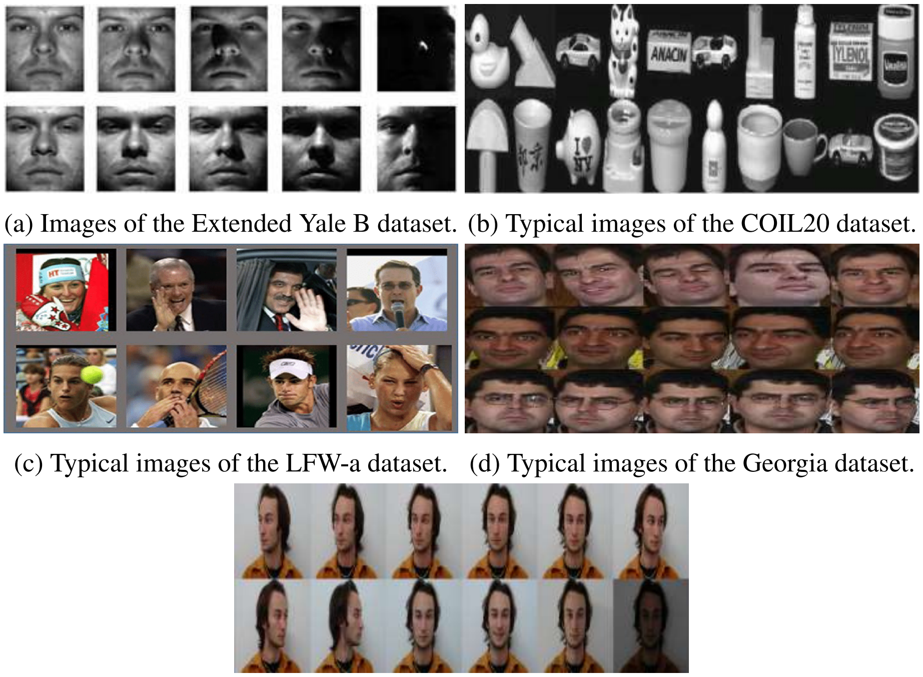
[5] $https://fei.edu.br/ cet/facedatabase.html$

one of the 50 people. Raw brightness images of dimension 1024 are used. The reported results are obtained after we used 5, 6, 7, and 8 image samples from each class for training samples and the rest was used for testing.

- **Outdoor Scene dataset[6]:** This scenes dataset contains 2,688 images belonging to 8 groups. The descriptor used consists of 256 HOG features.

Table 2: Brief datasets description.

Dataset	Type	Number of Samples	Number of features	Number of classes	Descriptor
Extended Yale B	Face	2414	1024	38	RAW-brightness images
LFW-a	Face	3408	1024	141	RAW-brightness images
COIL20	Object	1440	177	20	Local Binary Patterns
Georgia	Face	750	1024	50	RAW-brightness images
FEI	Face	700	1024	50	RAW-brightness images
Outdoor Scene	Scene	2688	256	8	HOG features



(a) Images of the Extended Yale B dataset. (b) Typical images of the COIL20 dataset.

(c) Typical images of the LFW-a dataset. (d) Typical images of the Georgia dataset.

(e) Typical images of the FEI dataset.

Figure 4: Typical images of various datasets.

Table 2 presents a brief description of the datasets used in our paper, more information about these datasets can be found in the provided links presented in the footnotes. Figure 4 shows some of the typical images included in the tested datasets.

[6] https://github.com/sudalvxin/SMSC/tree/master/data

4.2 Experimental Setup

In the conducted experiments, the proposed approach is contrasted with many methods. We state from these: K-nearest neighbors (KNN) [27], Support Vector Machines (SVM) [7], Linear Discriminant Analysis (LDA) [48], Local Discriminant Embedding (LDE) [8], PCE [39], ICS_DLSR [52] and Robust sparse LDA (RSLDA) [50]. We note that the SVM used in the experiments is the **Linear SVM**, it was implemented using LIBSVM library[7]. To further investigate the discrimination ability of the suggested approach, we have added some additional compared methods to the table of the Extended Yale B results (6). Robust Discriminant Analysis using Gradient Descent RDA_GD [22] , Linear Regression Based Classification (LRC) [38], Low-rank Linear Regression (LRLR) [5], Low-rank Ridge Regression (LRRR) [5], Sparse Low-rank Regression (SLRR) [5], Low-rank Preserving Projection via Graph Regularized Reconstruction (LRPP_GRR) [51] and Manifold Partition Discriminant Analysis (MPDA) [59] were added to table 6 in the purpose of widening the comparison among competing methods.

For a rational and accurate contrast, tests are carried out following the same experimental setup for all compared methods (eg, pre-processing and dimensionality reduction techniques). The classification performances presented in the tables are achieved using **10 splits** which were chosen **randomly** for each dataset, unless specified otherwise in the table's caption. We report the average classification accuracy over the 10 splits.

In the conducted simulations, various training and test proportions were used for each dataset as detailed in section 4.1. For each dataset and each compared approach, the targeted embedding matrix is first computed using the training data components. After that, the training and test data are projected onto the new space using the predicted embedding. And for the final step, classification of the test data is then performed using the Nearest Neighbour classifier (NN) [9]. The results presented in the tables were found with K=1 (1-NN).

In our testing phase, we invoked dimensionality reduction of the raw features before feeding them to the learning models and classifiers most of the time. The Principal Component Analysis (PCA) was used as a pre-processing technique used for this purpose [47]. For the competing methods, PCA was used to preserve 100% of the data's energy. We note that, in some conducted experiments and for some methods e.g. (ICS_DLSR, in addition to the proposed approach), the

[7] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

original dimensionality was preserved and no pre-processing techniques were applied in order to highlight on the ability of the proposed approach in selecting the most relevant original features.

The reported classification rates of the methods are chosen from the best parameter configurations and correspond to the average over 10 randomly selected splits as mentioned before.

4.3 Experimental Results

In this section, we will present the results derived through our experiments. We will compare our proposed method with the others mentioned in section 4.2.

4.3.1 Feature selection techniques comparison

In this section, we study the performance of the proposed ensemble approach in the case of using three different feature selection methods to select the subsets of features that we are going to work with. Adopting multiple selection techniques have led to multiple variants of the proposed scheme. The main goal is to enhance the classification performance obtained by the original ICS_DLSR algorithm. In our experiments we have chosen the subsets of features that we are going to use after the original features have been ranked using Fisher score, a combination of ReliefF and Fisher score, in addition to ranking with the Robust multi-label feature selection with dual-graph regularization (DRMFS) [20] algorithm. The reason we have selected Fisher score and ReliefF feature selection techniques is that these algorithms have shown stability, very good performance and have been used widely in the machine learning field. We have also worked with the DRMFS algorithm in order to enrich the experiments.

The proposed variants denoted as **EM_ICS_FS** and **EM_ICS_DRMFS** represent our method where the features were ranked via the Fisher score and the DRMFS algorithm, respectively. The third variant denoted as **EM_ICS_HS** represents the case where the features were ranked via a hybrid combination using both ReliefF and fisher score algorithms.

Table 3: Comparison of the mean classification performance (%) of different variants using LFW-a dataset .

LFW-a			
Training Samples	**Methods**		
	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_HS**
5	22.56	**27.38**	25.92
6	25.72	**31.75**	30.12
7	29.04	**36.07**	34.60
8	31.92	**39.71**	38.57

Table 4: Comparison of the mean classification performance (%) of different variants using the COIL20 dataset .

COIL20			
Training Samples	**Methods**		
	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_DRMFS**
20	98.04	98.36	**98.51**
25	98.22	98.61	**98.63**
30	98.75	98.92	**99.11**
35	99.12	99.21	**99.39**

Table 5: Comparison of the mean classification performance on the Outdoor Scene dataset.

Outdoor Scene				
Training Samples	**Methods**			
	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_HS**	**EM_ICS_DRMFS**
50	68.19	68.75	**68.84**	68.80
70	69.41	**70.51**	70.15	70.11
90	69.64	**70.60**	70.41	70.45
110	70.21	71.03	**71.05**	70.78

Table 3 compares the classification performance of two variants of the proposed scheme alongside with the performance of the single model learning using the ICS_DLSR algorithm. Results presented in this table were obtained using the LFW-a dataset.

Table 4 presents the performance achieved by the proposed approach using two different feature selection algorithms. Classification rates presented in this table are obtained in case of using 10 models where the original data is ranked via the different algorithms. Results presented in this table were obtained using the COIL20 dataset.

Table 5 presents the classification performance obtained by the proposed variants compared to the performance associated with the single model ICS_DLSR algorithm over the Outdoor Scene dataset.

4.3.2 Method comparison

Table 6: Mean classification accuracies (%) of compared methods on the Extended Yale B dataset.

						Ext. Yale B			
Training Samples	Method	KNN	SVM	LDA	LDE	PCE	SULDA	RSLDA	RDA_GD
10		69.80	73.85	82.32	79.92	86.39	84.61	86.79	87.10
15		75.20	80.02	86.76	83.77	89.23	88.72	89.93	90.04
20		80.24	85.79	90.7	88.44	92.19	91.66	93.59	93.75
25		82.24	89.03	92.17	90.43	93.35	92.14	94.92	95.02
	Method	LRC	LRLR	LRRR	SLRR	LRPP_GRR	MPDA	ICS_DLSR	**EM_ICS_FS**
10		81.65	84.63	87.76	87.95	84.82	83.67	86.56	**88.46**
15		88.92	86.31	91.09	89.75	89.07	86.82	89.53	**91.43**
20		91.74	88.93	93.19	92.58	91.42	90.38	93.14	**94.49**
25		93.78	90.98	95.51	94.24	92.25	91.79	94.50	**95.88**

Table 7: Mean classification accuracies (%) of compared methods on the tested datasets using the first proposed variant **EM_ICS_FS**.

Dataset \ Method	Training Samples	KNN	SVM	LDA	LDE	PCE	RSLDA	RDA_GD	ICS_DLSR	**EM_ICS_FS**
LFW-a	5	9.90	12.72	20.51	9.98	9.44	24.70	25.11	22.56	**27.38**
	6	10.57	13.61	25.28	10.49	10.26	28.42	28.61	25.72	**31.75**
	7	11.06	14.70	28.62	11.24	10.98	31.50	31.82	29.04	**36.07**
	8	11.35	15.72	32.42	11.71	11.73	32.48	32.69	31.92	**39.71**
COIL20	20	94.58	97.65	96.19	95.00	94.87	96.73	96.89	98.04	**98.36**
	25	95.79	98.22	97.07	96.12	95.99	97.74	97.89	98.22	**98.61**
	30	96.65	98.70	97.81	97.01	97.49	98.26	98.52	98.75	**98.92**
	35	97.14	98.81	98.15	97.42	98.11	98.68	98.80	99.12	**99.21**

Table 8: Mean classification accuracies (%) of compared methods on the tested datasets using **EM_ICS_HS**.

Dataset \ Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	**EM_ICS_FS**	**EM_ICS_HS**
Georgia	3	52.57	56.22	48.18	52.77	46.43	59.73	59.37	**59.95**
	5	61.28	66.98	59.20	62.14	56.18	71.12	71.40	**72.02**
	7	66.73	72.83	67.83	67.10	62.15	78.38	77.83	**79.03**
	9	71.40	77.53	72.57	72.13	66.37	82.57	81.93	**82.67**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	92.20	**92.56**
	6	90.35	92.93	94.18	92.15	88.73	93.65	93.88	**94.20**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.14	**95.43**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.00	**96.27**

Table 6 presents the classification performance of the proposed approach alongside with the competing methods using the first proposed variant over the Extended Yale B face dataset. Various training percentages were used. This table contains an extended number of compared methods, these methods were added to extend the comparison of the proposed method among other methods. Table 7 presents the obtained classification performance using the first proposed variant alongside with the competing methods over the LFW-a and COIL20 datasets.

Table 8 shows the obtained performance associated with two variants of the proposed scheme **EM_ICS_FS** and **EM_ICS_HS** next to the compared competing methods. Results presented in this table are noted over Georgia and FEI datasets.

4.4 Parameters sensitivity analysis

This section's main objective is to describe and study the effect of the main parameters of our proposed approach. We will show how the variation of the proposed approach's parameters affects the overall performance.

Like we have stated above, the ICS_DLSR algorithm minimizes the following objective function:

$$\min_{\mathbf{Q},\mathbf{E}} \frac{1}{2}\|\mathbf{Y} + \mathbf{E} - \mathbf{Q}\mathbf{X}\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{Q}\|_F^2 + \lambda_2 \sum_{i=1}^{C} \|\mathbf{Q}\mathbf{X}_i\|_{2,1} + \lambda_3\|\mathbf{E}\|_{2,1}$$

where $\mathbf{Q}$, $\mathbf{X}$ and $\mathbf{E}$ represent the transformation matrix, data samples and error matrix respectively. λ_1, λ_2 and λ_3 are three parameters to measure the effect of the corresponding terms. We have used the ICS_DLSR algorithm in our ensemble learning process. In our proposed approach, first we have selected multiple subsets of features using one or more feature selection techniques, then each subset of features was fed as an input to the ICS_DLSR algorithm to derive the associated transformation. Finally, we create the model out of the projected features.

Let us consider the subsets of features $\mathbf{Z}$, where $\mathbf{Z}_n \in \mathbb{R}^{m\times N}$ with $m \leq d$ represents the $n-$th features subset. $\mathbf{Z}_n^i$ denotes the $n-$th features subset corresponding to the $i-$th class. d and N denote the dimensionality of the data samples and the total number of the training data samples, respectively. Each feature subset is fed to the algorithm, our proposed approach work on minimizing the following problem:

$$\min_{\mathbf{Q},\mathbf{E}} \frac{1}{2}\|\mathbf{Y} + \mathbf{E} - \mathbf{Q}\,\mathbf{Z}_n\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{Q}\|_F^2 + \lambda_2 \sum_{i=1}^{C} \|\mathbf{Q}\,\mathbf{Z}_n^i\|_{2,1} + \lambda_3\|\mathbf{E}\|_{2,1} \qquad (5)$$

According to experimental evaluations which we have conducted, we found that most of the time the optimal performance is obtained when the value of λ_3 is set to 1. Thus, we can set λ_3 to 1 and study the effect of changing the values of the two parameters λ_1 and λ_2 on the classification performance over different datasets. Figures 5 and 6 illustrate our findings, while using the first proposed scheme EM_ICS_FS and the second proposed scheme EM_ICS_HS, respectively.

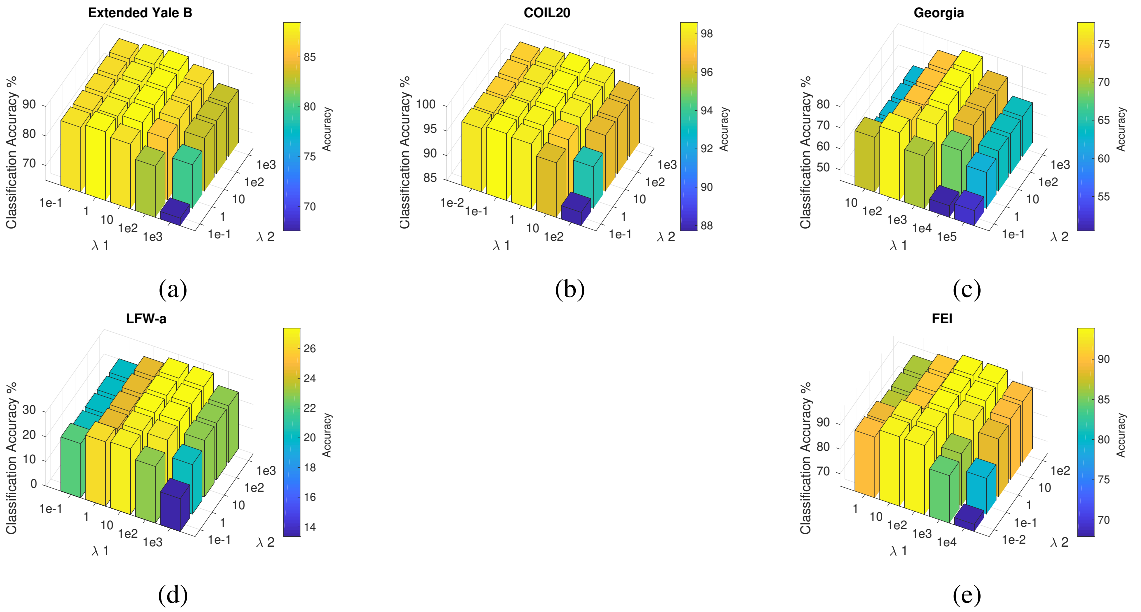


Figure 5: Classification performance vs Parameters sensitivity of the proposed method using **EM_ICS_FS**

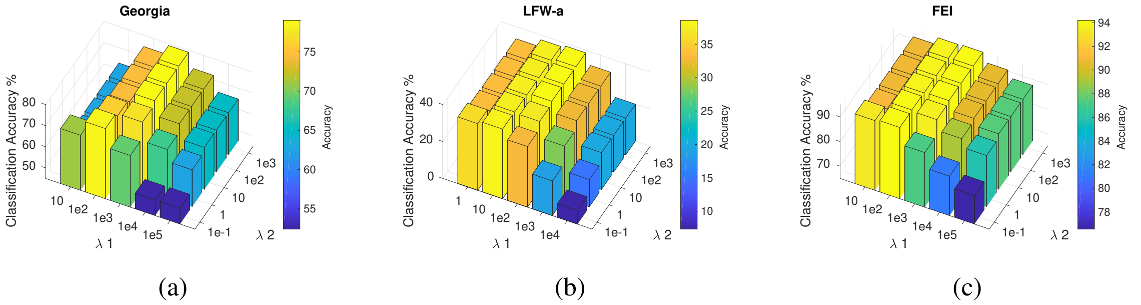


Figure 6: Classification performance vs Parameters sensitivity of the proposed method using **EM_ICS_HS**

Figures 5 and 6 illustrate the variation of the classification performance obtained as a function of different parameter combinations using EM_ICS_FS and EM_ICS_HS. In general, our proposed method achieved satisfactory classification performance using a wide range for the parameters used. For the tested dataset, the optimal performance was obtained when λ_1 and λ_2 are in the ranges $[1, 10^3]$ and $[1, 10^2]$, , respectively (incremental step is 10).

Another important factor in the ensemble learning, is the chosen number of created models, M, used for training. We have investigated about how the variation of the number of the created models affects the overall performance of the proposed scheme over the Extended Yale B dataset. Results presented in figure 7 are obtained while using 10 samples from each class from the Extended Yale B dataset for training and the remaining samples were used for testing.

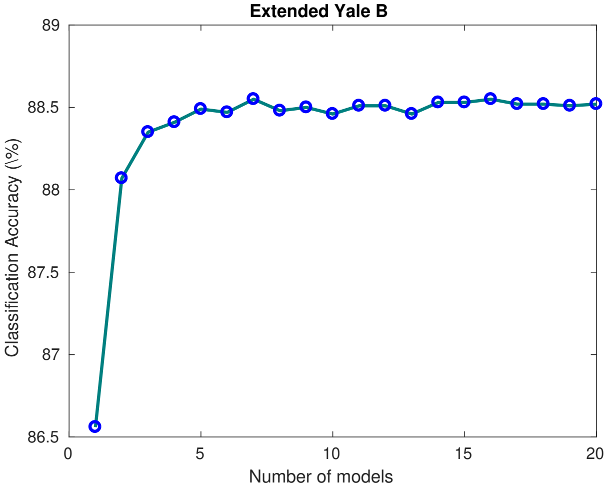


Figure 7: Classification performance variation according to the number of models.

4.5 Analysis of the Results

The experimental results illustrated in the previous figures and tables demonstrate the superiority of the suggested approach in comparison to other competing methods. Many observations can be made.

- The Proposed approach proved the superiority that ensemble learning can provide over single models. Conducted experiments have shown that by training multiple subsets of ranked features of original data, we can achieve better classification performance.

- We have proposed three variants for the proposed approach. All have shown very good discrimination properties and a remarkable enhancement over the baseline compared method, namely the ICS_DLSR method.

- For the datasets where the first variant of the proposed scheme failed to ensure an enhancement over the single model-based learning, other variants were able to enhance the classification performance and ensure the superiority of the proposed approach (e.g., the Georgia dataset using 3,7 and 9 training samples per class for training, and the FEI dataset when 7 and 8 training samples were used).

- The proposed approach is flexible in the sense that many other linear embedding approaches and feature selection techniques can be used and mixed to construct the desired models which may lead to a further better result.

- By analyzing the experimental results, we can observe that there is no specific feature selection technique that always leads to the best performance. The best option is to test multiple combinations to reach the optimal result. This in line with the literature of feature selection paradigms where the performance highly depends on the dataset used.

- Superior classification performance can be achieved if the parameters are accurately tuned. Very promising performance was obtained using a wide range for the used parameters, this is shown in Figures 5 and 6.

- The studied ensemble learning approach can achieve noticeably better classification performance using a small number of models (refer to Figure7) and different training/testing portions of the data.

- The performance improvement brought by the proposed scheme with respect to the single model highly depends on the dataset used and the adopted feature ranking technique. For instance, on the Extended Yale B and LFW-a datasets, we obtained significant performance enhancement compared to the single model while using Fisher score as the feature ranking scheme. Fair classification improvement was also noted when using the Outdoor Scene dataset with the second proposed variant. For other datasets, less enhancement was observed using the ensemble learning.

5 Conclusion

In this paper, we have proposed three variants of an ensemble learning approach that have been able to enhance the classification performance of the class-sparsity based least-square regression (ICS_DLSR) method. Multiple feature subsets were used in the training process with the ICS_DLSR algorithm and their corresponding outputs were used to construct multiple models. These models are concatenated to form a single data representation which is used in the classification process. The targeted models were created by using various subsets of the original data. Our proposed approach's design ensures that each created model contains the most relevant features that describes the data efficiently. Relevant features are taken into consideration each time in a way that even if less relevant features are found they will not harm the classification performance. Original data features have been ranked using different and combined feature selection techniques. Many factors were studied and investigated in this paper including (parameter combinations, different number of models, different training percentages, hybrid methods combinations, etc..). The obtained findings proved that the proposed approach enhanced the classification performance compared to the single-model and was able to outperform competing methods. Our proposed approach has been benchmarked on different datasets and achieved competitive results.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

[1] D. Benkeser, C. Ju, S. Lendle, and M. van der Laan. Online cross-validation-based ensemble learning. *Statistics in medicine*, 37(2):249–260, 2018.

[2] J. O. Berger and M. Bock. Combining independent normal mean estimation problems with unknown variances. *The Annals of Statistics*, pages 642–648, 1976.

[3] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] J. Cai, J. Luo, S. Wang, and S. Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[5] X. Cai, C. Ding, F. Nie, and H. Huang. On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1124–1132, 2013.

[6] A. Chambaz, W. Zheng, and M. Van Der Laan. Data-adaptive inference of the optimal treatment rule and its mean reward. the masked bandit. 2016.

[7] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[8] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 846–853. IEEE, 2005.

[9] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34(8):1–17, 2007.

[10] M. M. Davies and M. J. Van Der Laan. Optimal spatial prediction using ensemble machine learning. *The international journal of biostatistics*, 12(1):179–201, 2016.

[11] L. Deng and J. C. Platt. Ensemble deep learning for speech recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[12] T. G. Dietterich. Machine-learning research. *AI magazine*, 18(4):97–136, 1997.

[13] F. Dornaika and A. Khoder. Linear embedding by joint robust discriminant analysis and inter-class sparsity. *Neural Networks*, 2020.

[14] B. Efron and C. Morris. Combining possibly related estimation problems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 35(3):379–402, 1973.

[15] Q. Feng, Y. Zhou, and R. Lan. Pairwise linear regression classification for image set retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4865–4872, 2016.

[16] E. J. Green and W. E. Strawderman. A james-stein type estimator for combining unbiased and possibly biased estimators. *Journal of the American Statistical Association*, 86(416):1001–1006, 1991.

[17] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[18] N. Han, J. Wu, Y. Liang, X. Fang, W. K. Wong, and S. Teng. Low-rank and sparse embedding for dimensionality reduction. *Neural Networks*, 108:202–216, 2018.

[19] S. Hijazi. *Semi-supervised Margin-based Feature Selection for Classification*. PhD thesis, Université du Littoral Côte d'Opale; Université Libanaise, école doctorale , 2019.

[20] J. Hu, Y. Li, W. Gao, and P. Zhang. Robust multi-label feature selection with dual-graph regularization. *Knowledge-Based Systems*, 203:106126, 2020.

[21] A. Khoder and F. Dornaika. A hybrid discriminant embedding with feature selection: application to image categorization. *Applied Intelligence*, pages 1–17, 2020.

[22] A. Khoder and F. Dornaika. An enhanced approach to the robust discriminant analysis and class sparsity based embedding. *Neural Networks*, 2021.

[23] D. Kim and M. Gales. Noisy constrained maximum-likelihood linear regression for noise-robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(2):315–325, 2010.

[24] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pages 249–256. Elsevier, 1992.

[25] I. Kononenko and M. R. Šikonja. Non-myopic feature quality evaluation with (r) relieff. *Computational methods of feature selection*, pages 169–191, 2008.

[26] I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.

[27] L. Kozma. k nearest neighbors algorithm (knn). *Helsinki University of Technology*, 2008.

[28] N. Kwak and C.-H. Choi. Input feature selection for classification problems. *IEEE transactions on neural networks*, 13(1):143–159, 2002.

[29] M. J. Laan. van der, eric c. polley, and alan e. hubbard. “super learner.”. *Statistical applications in genetics and molecular biology*, 6, 2007.

[30] P. Langley. Selection of relevant features in machine learning: Defense technical information center, 1994.

[31] L. Li, P. W. Fieguth, and G. Kuang. Generalized local binary patterns for texture classification. In *BMVC*, volume 123, pages 1–11, 2011.

[32] Y. Li and A. Ngom. Nonnegative least-squares methods for the classification of high-dimensional biological data. *IEEE/ACM transactions on computational biology and bioinformatics*, 10(2):447–456, 2013.

[33] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu. Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2138–2150, 2013.

[34] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[35] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *arXiv preprint arXiv:1109.0367*, 2011.

[36] H. Lu and R. Mazumder. Randomized gradient boosting machine. *SIAM Journal on Optimization*, 30(4):2780–2808, 2020.

[37] A. R. Luedtke and M. J. van der Laan. Super-learning of an optimal dynamic treatment rule. *The international journal of biostatistics*, 12(1):305–332, 2016.

[38] I. Naseem, R. Togneri, and M. Bennamoun. Linear regression for face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(11):2106–2112, 2010.

[39] X. Peng, J. Lu, Z. Yi, and R. Yan. Automatic subspace learning via principal coefficients embedding. *IEEE transactions on cybernetics*, 47(11):3583–3596, 2016.

[40] R. Pirracchio, M. L. Petersen, M. Carone, M. R. Rigon, S. Chevret, and M. J. van der Laan. Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

[41] E. C. Polley and M. J. Van der Laan. Super learner in prediction. *U.C. Berkeley Division of Biostatistics Working Paper Series. Working Paper 266*, 2010.

[42] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[43] L. E. Raileanu and K. Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[44] J. Rao and K. Subrahmaniam. Combining independent estimators and estimation in linear regression with unequal variances. *Biometrics*, pages 971–990, 1971.

[45] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[46] D. B. Rubin and S. Weisberg. The variance of a linear combination of independent estimators using estimated weights. *Biometrika*, 62(3):708–709, 1975.

[47] L. I. Smith. A tutorial on principal components analysis. Technical report, 2002.

[48] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.

[49] D. Wang, F. Nie, and H. Huang. Feature selection via global redundancy minimization. *IEEE transactions on Knowledge and data engineering*, 27(10):2743–2755, 2015.

[50] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen, and Y. Xu. Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):390–403, 2018.

[51] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan. Low-rank preserving projection via graph regularized reconstruction. *IEEE Transactions on Cybernetics*, 49(4):1279–1291, Apr. 2019.

[52] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu. Inter-class sparsity based discriminative least square regression. *Neural Networks*, 102:36–47, 2018.

[53] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[54] R. Wyss, S. Schneeweiss, M. van der Laan, S. D. Lendle, C. Ju, and J. M. Franklin. Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*, 29(1):96–106, 2018.

[55] Y. Xu, X. Fang, Q. Zhu, Y. Chen, J. You, and H. Liu. Modified minimum squared error algorithm for robust classification and face recognition experiments. *Neurocomputing*, 135:253–261, 2014.

[56] J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 82(281):301–329, 2013.

[57] S. Zang, Y. Cheng, X. Wang, and J. Ma. Semi-supervised flexible joint distribution adaptation. In *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, pages 19–27, 2019.

[58] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang. Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):689–700, 2015.

[59] Y. Zhou and S. Sun. Manifold partition discriminant analysis. *IEEE Transactions on Cybernetics*, 47(4):830–840, 2016.

CHAPTER 5

A Supervised Discriminant Data Representation: Application to Pattern Classification

A Supervised Discriminant Data Representation: Application to Pattern Classification

A. Khoder[1], F. Dornaika[1,2,*], and A. Moujahid[1]

[1] University of the Basque Country UPV/EHU, SPAIN
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, SPAIN

Abstract

The performance of machine learning and pattern recognition algorithms generally depends on data representation. That is why, much of the current effort in performing machine learning algorithms goes into the design of preprocessing frameworks and data transformations able to support effective machine learning. The method proposed in this work consists of a hybrid linear feature extraction scheme to be used in supervised multi-class classification problems. Inspired by two recent linear discriminant methods: robust sparse linear discriminant analysis (RSLDA) and inter-class sparsity based discriminative least square regression (ICS_DLSR), we propose a unifying criterion that is able to retain the advantages of these two powerful methods. The resulting transformation relies on sparsity-promoting techniques to both select the features that most accurately represent the data, and to preserve the row-sparsity consistency property of samples from the same class. The linear transformation and the orthogonal matrix are estimated using an iterative alternating minimization scheme based on steepest descent gradient method and different initialization schemes. The proposed framework is generic in the sense that it allows the combination and tuning of other linear discriminant embedding methods. According to the experiments conducted on several datasets including faces, objects and digits, the proposed method was able to outperform competing methods in most cases.

Keywords: Supervised learning, discriminant analysis, feature extraction, linear embedding, class sparsity, dimensionality reduction, image classification.

1 Introduction

Modern systems of interest based on computer vision, such as driver-assistance systems, healthcare or surveillance systems, may be characterized as high-dimensional systems generally embedded onto low-dimensional manifolds that preserve the intrinsic properties of the original data. Learning good representations of the data able to extract and organize the discriminative information is of great interest. It may reduce the memory and computational requirements, and more importantly, tends to improve the performance of classifiers or other predictors. This explains why Representation Learning is becoming a hot research topic (e.g. [14, 15, 20, 21, 30, 39, 37]).

Among the various ways of learning representations, this work focuses on feature selection and feature extraction. Feature extraction can be performed using linear or nonlinear methods. Most feature extraction methods look for a linear transformation that maps the original features to another space where latent variables can be obtained. In these methods, feature ranking or selection can be imposed by adding a $\ell_{2,1}$-norm constraint on the transformation matrix in the global criterion [31]. Nowadays, researches focus on deploying linear projection models that perform feature ranking and extraction simultaneously [31, 40]. An interesting approach recently reported by Zhang et al. [38] gives a more discriminating feature representation which consists in transforming tree-structured data into vectorial representations. They authors implemented a clustering technique in order to develop a node allocation process which aims at describing the global embedded information. They introduced an additional model to preserve the local information hidden among child nodes for a parent node, which led to very good discrimination characteristics. Other methods, use least square regression frameworks to achieve a discriminative feature extraction [33].

A feature can be identified as one of the following: relevant, irrelevant or redundant. Usually, a feature is called irrelevant if it does not contribute in enhancing the prediction model, in other words, it degrades the classification accuracy when considered in the classification process. Relevant features are the features that contribute to a better predictive model and thus to higher classification accuracy. These features are the ones that the model should extract and select among all others. A redundant feature does not make the model perform better in the classification process.

In this paper, we present a unified and hybrid discriminant embedding method that can retain the strengths of two recent discriminant methods: (i) RSLDA [31] and (ii) ICS_DLSR [33]. The former promotes Linear Discriminant Analysis with implicit feature selection, and the latter promotes inter-class sparsity, which means that the projected features share a common sparse structure for the samples in each class.

Thus, the main contributions are as follows. First, we deduce a novel objective function to estimate the linear transformation which has proven to refine the solution of RSLDA (transformation matrix **Q**).

Second, we provide an optimization algorithm in which the linear transformation is estimated by a gradient descent method. This allow to sets the initial transformation matrix to a hybrid combination of transformation matrices obtained from both ICS_DLSR [33] and RSLDA [31]) methods.

Finally, we propose two initialization procedures for the linear transformation, which lead to two variants of the proposed algorithm.

Indeed, our approach inherits the advantages of two powerful discriminant methods at two levels: (1) the initialization of the hybrid linear transformation, and (2) the refinement via the proposed single new criterion. The proposed method is also capable of obtaining a well-constructed projection space that ensures high classification accuracy, it can additionally be used in tuning an already obtained projection matrix. Our approach can be generic in the sense that any hybrid initial transformation matrix can be fed into our algorithm and then a more discriminative solution for the transformation matrix is obtained, resulting in higher classification performance.

The main contributions of this work can be seen as follows:

- The proposed method inherits the advantages of two recent powerful discriminant methods. The obtained transformation encapsulates two different types of discrimination, namely inter-class sparsity in addition to robust LDA.

- A hybrid initialization for the transformation matrix is introduced, where the initial matrix is created by combining two solutions of two different methods.

- Using the gradient descent method to find a solution for the proposed criterion instead of the closed-form solution, where the gradient for the sought transformation matrix is calculated in each iteration and the unknowns are updated accordingly.

The experiments conducted show that the proposed method resulted in an improvement in classification accuracy in the majority of tested cases and was able to outperform several competing methods. The rest of the paper is organised as follows. Section 2 describes related work and presents the notations used. Section 3 presents the criterion and solution details of the proposed method along with two initialization procedures. The obtained experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Work and notations

In this section we describe some related works, and briefly introduce the gradient descent method and how we used it to obtain a better embedding space by selecting the best and most relevant features of the data. In addition, we will show how the introduction of the $\ell_{2,1}$ [34] norm and inter-class sparsity constraint was used for feature selection and helped in discrimination [25], and enumerate some recent methods that have used such a constraint by embedding it in their global criterion to ensure that the method performs feature selection [17, 9].

2.1 Notations

We will start by introducing the notations that we use in our paper. We will refer for the training set by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, with d the dimension of each sample.

Each sample $\mathbf{x}_i$ is a column vector with d features $\in \mathbb{R}^d$.

The number of training samples will be denoted by N, in addition C will represent the total number of classes. The $\ell_{2,1}$ norm of a matrix $\mathbf{Z} \in \mathbb{R}^{d \times N}$ is obtained through the following formula $||\mathbf{Z}||_{2,1} = \sum\limits_{i=1}^{d} \sqrt{\sum\limits_{j=1}^{N} z_{ij}^2}$, and the ℓ_2 norm for the vector $\mathbf{z} = [z_1, z_2, ..., z_d]$ is obtained as follows $||\mathbf{z}||_2 = \sqrt{\sum\limits_{i=1}^{d} z_i^2}$.

Table 1 shows the main notations used in our paper.

Table 1: Main notations used in the paper.

Notation	Description
$\mathbf{X}$	Training data samples $\in \mathbb{R}^{d\times N}$
$\mathbf{P}$	Orthogonal Matrix $\in \mathbb{R}^{d\times d}$
$\mathbf{Q}$	Projection Matrix $\in \mathbb{R}^{d\times d}$
$\mathbf{D}$	Diagonal Matrix
$\mathbf{S}_w$	Within-class scatter matrix
$\mathbf{S}_b$	Between-class scatter matrix
d	Dimensionality of data
N	Number of data samples
n_i	Number of samples in the i-th class
C	Number of classes
$\mathbf{x}_i$	The i-th data sample $\in \mathbb{R}^d$

2.2 Related Work

Recently, many feature extraction methods have been proposed. Some of these methods have built-in constraints that implement feature ranking/selection in the method and rank the features of their projection matrices. Feature selection or ranking is becoming more and more a trending problem in machine learning. Very often, using all data features does not lead to high classification performance. Feature selection aims to efficiently select the most relevant features of the data that best describe the data and improve discrimination. [24, 35, 36]. On the other hand, feature extraction aims to derive new sets of features from the original ones. The derived features are usually more discriminative than the original ones.

The best known method to tackle the curse of high dimensionality is the principal component analysis (PCA) [23] method. PCA is an unsupervised feature extraction method that transforms the features of the original data and projects them into a low-dimensional space. In the well-known supervised Linear Discriminant Analysis (LDA) [26, 8] method, label information is required for the data. LDA and its variants are among the most widely used and discriminative algorithms in machine learning. LDA estimates a transformation matrix in which the desired space maximizes the variance between classes and minimizes the variance within classes. The projection axis $\mathbf{w}$ would be the solution to the Fisher criterion

[13]:

$$\mathbf{w} = \arg \min_{\mathbf{w}^T\mathbf{w}=1} \mathbf{w}^T \left(\mathbf{S}_w - \mu\, \mathbf{S}_b\right) \mathbf{w} \tag{1}$$

where μ is a small positive constant that balances the effect of the two scatter matrices (Within-class scatter matrix $\mathbf{S}_w$ and between-class scatter matrix $\mathbf{S}_b$) which could be calculated as:

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C} n_i \left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T \tag{2}$$

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i} (\mathbf{x}_j{}^i - \mu_i)\,(\mathbf{x}_j{}^i - \mu_i)^T \tag{3}$$

where μ, μ_i are the mean of all data samples and the mean of samples of the i-th class, respectively. Many variants of LDA were proposed and still being proposed (e.g.[42, 41, 5]), as the linear discriminant analysis showed good interpretability for the data.

2.2.1 Review of Robust Sparse Linear Discriminant Analysis (RSLDA):

RSLDA [31] was proposed to address many limitations of classical LDA[26], RSLDA mainly adds $\ell_{2,1}$ regularization to the projection matrix. This regularization term is added to the global criterion to ensure that the method performs feature ranking and weighting. RSLDA minimizes the following criterion:

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{E}} Tr\left(\mathbf{Q}^T\mathbf{S}\mathbf{Q}\right) + \lambda_1 \,\|\mathbf{Q}\|_{2,1} + \lambda_2 \,\|\mathbf{E}\|_1 \tag{4}$$

$$s.t. \qquad \mathbf{X} = \mathbf{P}\,\mathbf{Q}^T\mathbf{X} + \mathbf{E}, \;\; \mathbf{P}^T\mathbf{P} = \mathbf{I}$$

where $\mathbf{Q} \in \mathbb{R}^{d\times d}$ and $\mathbf{P} \in \mathbb{R}^{d\times d}$ denote the projection matrix and the orthogonal matrix, respectively. $\mathbf{E}$ is an error matrix. $\mathbf{S}$ is the difference matrix $\mathbf{S}_w - \mu\,\mathbf{S}_b$, λ_1 and λ_2 are two parameters that balance the importance of the different terms. In the criterion of RSLDA, the $\ell_{2,1}$ norm was imposed on the projection matrix to achieve feature selection.

2.2.2 Review of Inter Class Sparsity Least Square Regression:

In [33], the authors propose the Inter-class sparsity based discriminative least square regression ICS_DLSR [33]. This method provides a linear mapping to the soft label space, where the dimension of the latent space is set to the number of classes. This method was able to construct a model in which the margins of samples from the same class are greatly reduced, while the margins for samples from different classes are increased. This was achieved by adding a class-wise row sparsity constraint for the transformed features. ICS_DLSR minimizes the following problem:

$$\min_{\mathbf{Q},\mathbf{E}} \frac{1}{2}||\mathbf{Y} + \mathbf{E} - \mathbf{Q}\mathbf{X}||_F^2 + \frac{\lambda_1}{2}||\mathbf{Q}||_F^2 + \lambda_2 \sum_{i=1}^{C} ||\mathbf{Q}\mathbf{X}_i||_{2,1} + \lambda_3||\mathbf{E}||_{2,1} \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the training set with n samples from C classes, and m is the feature dimension for each sample. $\mathbf{Y} \in \mathbb{R}^{C\times n}$ is a binary label matrix corresponding to the training set. $\mathbf{Q}$ is the transformation matrix and $\mathbf{E}$ denotes the errors. λ_1, λ_2 and λ_3 are three regularization parameters.

Another similar method is the one described in [25], where the $\ell_{2,1}$-norm is applied to the transformation of the original linear discriminant analysis.

3 Proposed Method

In this section we present our problem formulation and show the steps applied to solve it. Our method is mainly considered as a linear projection method used for feature extraction, aiming at finding a more discriminative transformation matrix. Two variants of the method are proposed. These two variants differ in the initialization step. Our proposed method has adopted feature ranking by using the solution of RSLDA as the initial estimate for the sought transformation. The next step is to fine tune the initial guess for the transformation matrix by minimising the proposed criterion with a gradient descent method, which aims to find the required solution of the transformation matrix $\mathbf{Q}$.

The gradient descent algorithm is one of the simplest and most efficient algorithms for solving unconstrained optimization problems. In our algorithm, we have used the gradient descent approach to compute the transformation matrix $\mathbf{Q}$ and find the solution.

3.1 Formulation

The main goal of our approach is to obtain both the projection matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ and the orthogonal matrix $\mathbf{P} \in \mathbb{R}^{d \times d}$ using a unique criterion. In fact, the main contribution consists of the following objective function:

$$f(\mathbf{Q}, \mathbf{P}) = Tr\left(\mathbf{Q}^T \mathbf{S} \mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} \|\mathbf{Q}^T \mathbf{X}_i\|_{2,1} + \lambda_2 \|\mathbf{X} - \mathbf{P} \mathbf{Q}^T \mathbf{X}\|_2^2 \tag{6}$$

$$s.t. \qquad \mathbf{P}^T \mathbf{P} = \mathbf{I}$$

Where $\mathbf{X}_i \in \mathbb{R}^{d \times n_i}$ is the data matrix belonging to the ith class, n_i is the number of training samples in the ith class, C is the number of classes.

The first term in the equation (6) is the LDA criterion, where $\mathbf{S}$ represents the LDA scatter matrix, which can be calculated as $\mathbf{S} = \mathbf{S}_w - \mu \, \mathbf{S}_b$, where $\mathbf{S}_b$ being the between-class matrix and $\mathbf{S}_w$ is the within-class matrix. These two matrices are given by the equations (2) and (3) respectively. The second term of the criterion is imposed to ensure that transformed features of the same class in the projected space share a common sparse structure. $\mathbf{Q}$ is the projection matrix we are looking for. In addition, a variant of the (PCA) constraint is introduced to guarantee that the original data is well recovered, which is presented in the third term of the proposed procedure criterion. λ_1 and λ_2 are two trade-off parameters to control the importance of the different terms. It is known that the $\ell_{2,1}$-norm of a matrix can be written as:

$$\|\mathbf{Z}\|_{2,1} = Tr\left(\mathbf{Z}^T \mathbf{D} \mathbf{Z}\right) \tag{7}$$

where $\mathbf{D}$ is a diagonal matrix that is given by:

$$\mathbf{D} = \begin{pmatrix} \frac{1}{\|\mathbf{Z}(1)\|_2 + \epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\|\mathbf{Z}(d)\|_2 + \epsilon} \end{pmatrix} \tag{8}$$

where $\mathbf{Z}\,(j)$ represents the j-th row of $\mathbf{Z}$.

By substituting the second term of the criterion by its trace form showed in equation

(7), problem (6) can be viewed as:

$$f(\mathbf{Q},\mathbf{P}) = Tr\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \sum_{i=1}^{C} Tr\left(\,(\mathbf{Q}^T\mathbf{X}_i)^T\,\mathbf{D}_i\,\mathbf{Q}^T\mathbf{X}_i\right) + \lambda_2\,\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\|_2^2 \qquad (9)$$

$$\min_{\mathbf{Q},\mathbf{P}} f\,(\mathbf{Q},\mathbf{P}) \quad s.t. \quad \mathbf{P}^T\,\mathbf{P} = \mathbf{I}$$

Equation (9) represents the criterion for the proposed method. The minimization of the first term of this criterion is targeting a transformation matrix that ensures class discrimination with Linear Discriminant Analysis (LDA). The second term of the criterion is introduced to obtain class sparsity. By introducing this condition, the transformed features from each class obtain a common sparse structure. Finally, a variant of the ”principal component analysis” constraint is introduced in our proposed criterion [10]. This last constraint was introduced to maintain the energy preserving property of PCA, and this constraint ensures the robustness of our data.

To find a solution for the proposed method, we used the gradient descent algorithm. Gradient descent algorithm is a mathematical process used for minimising a particular function. When using the gradient algorithm, in addition to knowing the derivative of the function, we should also know the function, which is called the cost function. The gradient algorithm allows the person to solve the optimization problem in such a way that one knows the gradient from a particular point and can move in that direction to get a solution. The use of gradient algorithm has many advantages, we mention the most important of them as:

- It has less computational complexity compared to other methods. Finding the solution by the descent gradient algorithm is usually less computationally expensive. Using the descent gradient to find a solution results in a faster model.

- It leads to accurate solutions. The gradient algorithm leads to a more accurate solution to the minimization problem than the closed form solution.

3.2 Solution steps to the proposed method

To solve the problem formulated above, we adopted the alternating direction method of multipliers (ADMM) [1] and calculated each variable while other variables are fixed as follows:

- **Calculate the orthogonal matrix P:**

  $\mathbf{P}$ can be calculated by fixing the variable $\mathbf{Q}$ and through solving the following problem:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \tag{10}$$

  Using $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ the fact the squared norm of a matrix $\mathbf{A}$ is given by $||\mathbf{A}||_2^2 = Tr(\mathbf{A}^T\mathbf{A}) = Tr(\mathbf{A}\,\mathbf{A}^T)$, problem (10) is equivalent to the following maximization problem:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \left\|\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}\right\|_2^2 \quad \longrightarrow \quad \max_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} Tr\,(\mathbf{P}^T\,\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}) \tag{11}$$

  One can find a solution for problem (11) by performing singular value decomposition of $\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}$. Suppose the SVD decomposition is given by $SVD\,(\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q}) = \mathbf{U}\,\Sigma\,\mathbf{V}^T$. Then $\mathbf{P}$ is obtained as [42]:

$$\mathbf{P} = \mathbf{U}\,\mathbf{V}^T \tag{12}$$

- **Calculate the Projection matrix Q:**

  Gradient descent is an iterative optimization technique used to minimize a function by moving in the direction of steepest descent in each iteration. The way the gradient method is used differs in different areas. In machine learning and classification, gradient is used to iteratively update the parameters of the desired model. We adopted the gradient descent method to compute $\mathbf{Q}$ in each iteration of the proposed method as follows:

  The orthogonal matrix $\mathbf{P}$ is fixed. Let us consider the trace form of the criterion of our problem:

$$f(\mathbf{Q},\mathbf{P}) = Tr\,\left(\mathbf{Q}^T\,\mathbf{S}\,\mathbf{Q}\right) + \lambda_1 \textstyle\sum_{i=1}^{C} Tr\,(\,\mathbf{X}_i^T\mathbf{Q}\,\mathbf{D}_i\,\mathbf{Q}^T\mathbf{X}_i) + \lambda_2\,||\mathbf{X} - \mathbf{P}\,\mathbf{Q}^T\,\mathbf{X}||_2^2$$

  We calculate the gradient of the objective function w.r.t. $\mathbf{Q}$ as follows:

$$\mathbf{G} = \frac{\delta\,f}{\delta\,\mathbf{Q}} = 2\,\mathbf{S}\,\mathbf{Q} + \lambda_1 \sum_{i=1}^{C} 2\,\mathbf{X}_i\,\mathbf{X}_i^T\,\mathbf{Q}\,\mathbf{D}_i + 2\lambda_2\,[\mathbf{X}\,\mathbf{X}^T\,\mathbf{Q} - \mathbf{X}\,\mathbf{X}^T\,\mathbf{P}] \tag{13}$$

Using the gradient matrix, we can update $\mathbf{Q}$ by:

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \alpha\, \mathbf{G} \tag{14}$$

where $\mathbf{Q}_{t+1}$ and $\mathbf{Q}_t$ denotes the projection matrix $\mathbf{Q}$ in iteration $t+1$ and iteration t respectively. α is the step length (learning rate).

- **Update Variable $\mathbf{D}_i$:** We update $\mathbf{D}_i, (i = 1, ..., C)$ by:

$$\mathbf{D}_i = \begin{pmatrix} \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_{i(1)}\right\|_2 + \epsilon} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\left\|\mathbf{Q}^T\mathbf{X}_{i(d)}\right\|_2 + \epsilon} \end{pmatrix} \tag{15}$$

where ϵ is a small positive scalar and $\mathbf{Q}^T\mathbf{X}_i\,(j)$ represents the j-th row vector of $\mathbf{Q}^T\mathbf{X}_i$.

Algorithm 1 summarizes our proposed method and describes the main steps for solving the problem (6).

```
Algorithm. 1.    Supervised discriminant analysis using gradient (SDA_G_1)
                 Supervised discriminant analysis using gradient via combined initialization (SDA_G_2)
Input:           1. Data samples $\mathbf{X} \in \mathbb{R}^{d\times N}$
                 2. Labels of the training samples
                 3. The step length of the gradient descent $\alpha$
                 4. Parameters $\lambda_1$, $\lambda_2$
Output:          $\mathbf{P}$, $\mathbf{Q}$

Initialization:  $\mathbf{Q}^{(0)}$ obtained from RSLDA or using a hybrid combination (see section 3.3).

Process:         set $t = 0$ and $\mathbf{Q} = \mathbf{Q}^{(0)}$

                 Repeat
                 Fix $\mathbf{Q}$, update $\mathbf{P}^{(t+1)}$ using Eq. (12).
                 Calculate the gradient matrix $\mathbf{G}$ using Eq. (13)
                 Fix $\mathbf{P}$, update $\mathbf{Q}^{(t+1)}$ using Eq. (14).
                 Update $\mathbf{D}_i$ using Eq. (15)
                 set $t = t + 1$
                 Until convergence
```

The projection of the training and test samples is carried out using the estimated projection matrix $\mathbf{Q}$. This is given by $\mathbf{z}_{train} = \mathbf{Q}^T\, \mathbf{x}_{train}$ and $\mathbf{z}_{test} = \mathbf{Q}^T\, \mathbf{x}_{test}$ where $\mathbf{x}_{train}$ is a training data sample, and $\mathbf{x}_{test}$ is a test data sample.

3.3 Initialization of Projection Matrix Q

The linear transformation **Q** needs a good initial estimate, since it is estimated by a gradient descent update rule. In this section, we present two initialization procedures that lead to two variants of the proposed algorithm.

3.3.1 Using RSLDA algorithm

In this variant, the initial estimate $\mathbf{Q}^{(0)}$ for the linear transformation matrix **Q** is given by the solution of the RSLDA [31] method (solved by a separate ADMM optimization). RSLDA was able to provide implicit feature selection by imposing the $\ell_{2,1}$ norm over the sought transformation matrix. Moreover, the introduction of the error matrix helped in tracking and modelling the random noise. These introduced terms have resulted in RSLDA obtaining a discriminative and efficient transformation. The solution of our proposed method is computed using the gradient approach, which requires a good initial estimate to ensure good performance. By adopting the solution derived from RSLDA method, our proposed variant could adopt the advantages of this method. Figure (1) describes the initialization process using the transformation matrix provided by RSLDA.

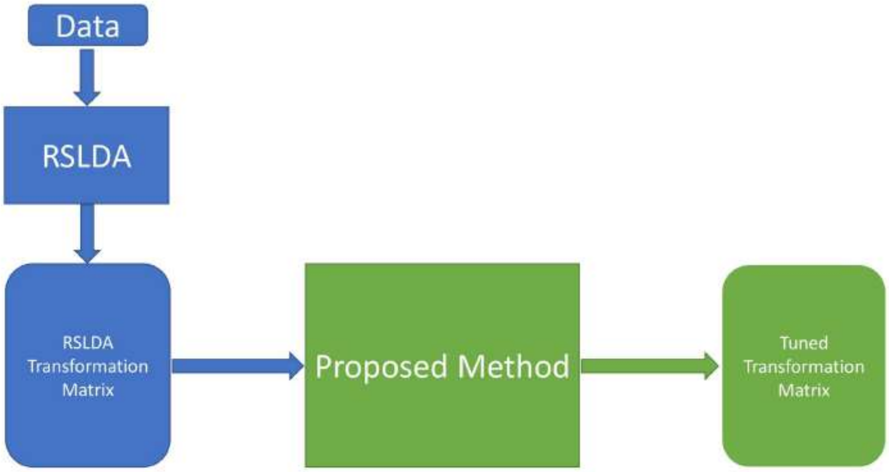


Figure 1: The output transformation provided by RSLDA is fed as an input to our proposed approach as an initial guess for the transformation matrix

3.3.2 Hybrid combination of projection matrices obtained from the two embedding methods RSLDA and ICS_DLSR

In the second variant of our proposed algorithm, the initial transformation matrix $\mathbf{Q}^{(0)}$ is set to a hybrid combination of the transformation matrices obtained by the two embedding methods RSLDA [31] and ICS_DLSR [33].

Let the number of rows of the hybrid transform $\mathbf{Q}^{(0)}$ be d. The number of columns (projection axes), on the other hand, can be set to any arbitrary value. Without loss of generality, to be consistent with linear methods, we will assume that the total number of columns of $\mathbf{Q}^{(0)}$ is d. Thus, $\mathbf{Q}^{(0)} \in \mathbb{R}^{d\times d}$. According to [33], the linear transformation $\mathbf{Q}_{ICS\_DLSR}$ obtained by the ICS_DLSR algorithm is $\in \mathbb{R}^{d\times C}$, where d and C represent the dimension of the features and the number of classes, respectively. On the other hand, the RSLDA method [31] its own linear transformation $\mathbf{Q}_{RSLDA} \in \mathbb{R}^{d\times d}$. The sought initial hybrid projection matrix $\mathbf{Q}^{(0)}$ used in our algorithm is denoted by $\mathbf{Q}_{Hybrid}$. It is constructed by taking all C columns of $\mathbf{Q}_{ICS\_DLSR}$ to which the first $d - C$ columns of $\mathbf{Q}_{RSLDA}$ are attached. The resulting transformation matrix $\mathbf{Q}_{Hybrid}$ is $\in \mathbb{R}^{d\times d}$. The strategy for the hybrid initialization methodology is shown in Figure 2.

In the above construction of the hybrid matrix $\mathbf{Q}_{Hybrid}$, our work fixed the number of projection axes for each projection type to C and $d - C$ for ICS_DLSR and RSLDA, respectively. We emphasize the fact that these dimensions can be changed.

In our experiments, according to Table 2, we can see that the value of C that represents the number of classes varies between 10 and 50 for the datasets used. d represents the number of features for each dataset is also shown in the same table.

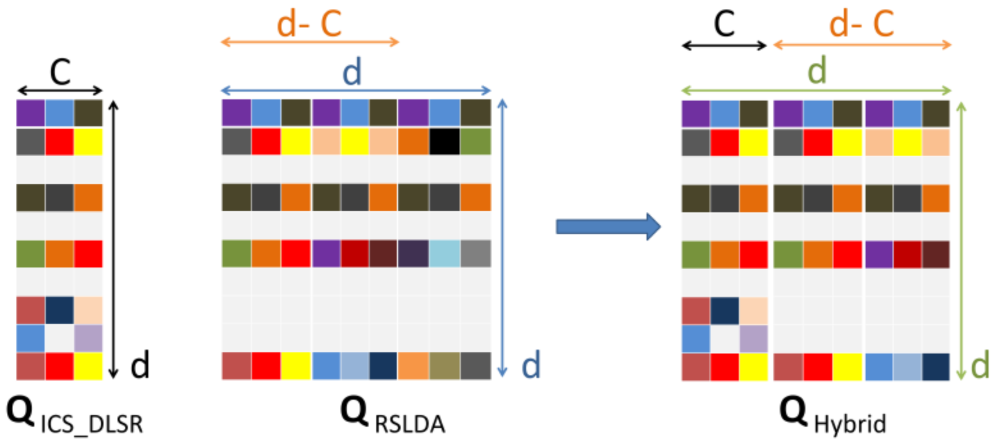


Figure 2: Combined initialisation using the linear embeddings derived from ICS_DLSR and RSLDA.

3.4 Computational complexity

In this section, the computational complexity of the proposed method will be analyzed (see **Algorithm 1**). Matrices **Q**, **P**, are sought to be calculated. The orthogonal matrix **P** requires a singular value decomposition. The computational cost of a decomposition of a $d \times N$ matrix would be $O\left(N^3\right)$. **Q** is computed in the second step of the procedure, it requires the computation of the corresponding gradient matrix, but since these two steps consist only of simple matrix operations, they have low computational cost and can therefore be ignored. Also, the step provided for updating $\mathbf{D}_i$ from the equation (15) is a simple matrix operation which has very low cost.

On the other hand, in the first variant of our proposed method, we have used the RSLDA method for the initialization of the transformation matrix **Q** before it is fed to our algorithm. Thus, the complexity of the RSLDA method should be added to the complexity of our proposed method. Supposing τ represents the number of iterations of RSLDA. The latter has a complexity of $O\left(\tau(d^2N + 4d^3)\right)$. The main computational complexity of the proposed algorithm takes place in the step for updating **P**. The complete cost of the proposed method (first variant) is mainly $O\left(\tau\left(N^3\right)\right)$. In summary, the overall cost would be the sum of RSLDA cost added to the cost of our proposed method which would be equal to $O\left(\tau(d^2N + 4d^3)\right) + O\left(\tau'\left(N^3\right)\right)$ where τ' denotes the number of iterations of Algorithm 1.

For the second proposed variant, we have constructed the initial guess of the transformation matrix through the combination of solutions obtained by the RSLDA[31] and ICS_DLSR[33] methods. Knowing that the ICS_DLSR algorithm has a complexity of $O(\tau(d^3))$, the overall cost of the second suggested variant would become $O\left(\tau(d^3)\right) + O\left(\tau(d^2N + 4d^3)\right) + O\left(\tau'(N^3)\right)$

4 Performance Study

To test both variants of our proposed method, we conducted experiments on several datasets including faces, objects, and handwritten datasets. Detailed information on these datasets is presented in this section. Next, we are going to present the experimental setup and the results obtained.

4.1 Datasets

In our work we have conducted our experiments over the following five public datasets in addition to a large-scale dataset: **USPS** [1] digits dataset, **Honda** [2] dataset, **COIL20** [3] object dataset, **Extended Yale B** [4] face dataset, **FEI** [5] dataset, and the large scale **MNIST** dataset consisting of 60,000 images.

1. **USPS Digits Dataset**[6]**:** The US Postal Service or abbreviated as (USPS) [22] is a well-known handwritten digits dataset used for digit recognition. This dataset represents 10 digits (from 0 to 9), it contains a total of 110 images for each digit, thus a total number of 1100 images in which the dimension of each one is 256. Raw-brightness images are used for classification. Popular training percentages for this dataset are used as we use 30, 40, 55, and 65 image samples from each class as training samples and set the rest as test samples.

2. **Honda dataset**[7]**:** Honda dataset contains a total of 2277 face images that represent the faces of 22 different individuals in different conditions. Each class contains approximately 97 images. Popular training percentages are used as we use 10, 20, 30,

[1] *https : //www.kaggle.com/bistaumanga/usps − dataset*

[2] *http : //vision.ucsd.edu/ leekc/HondaUCS DVideoDatabase/HondaUCS D.html*

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4] *http : //vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html*

[5] *https : //fei.edu.br/ cet/facedatabase.html*

[6] *https : //www.kaggle.com/bistaumanga/usps − dataset*

[7] *http : //vision.ucsd.edu/ leekc/HondaUCS DVideoDatabase/HondaUCS D.html*

and 50 image samples from each class as training samples and set the rest as test samples, Raw brightness images are used for the classification process.

3. **COIL20 Object Dataset**[8]**:** Columbia Object Image Library (COIL20) [18] dataset used for evaluation in our experiments consists of a total of 1440 images representing 20 different classes, each class has 72 images. Different images of this dataset represent various objects in which each object is rotated around a vertical axis. As a training set, we used 20, 25, 30, and 35 image samples from each class and set the rest for testing. The image descriptor used is the Local Binary Patterns (LBP) [16]. We used the uniform LBP histogram (59 values). Three LBP descriptors are constructed from the image using 8 points and three values for the radius (R=1, 2, and 3 pixels). Thus, the final concatenated descriptor has 177 values.

4. **Extended Yale B Face Dataset**[9]**:** This dataset is a popular dataset used for image classification purposes [11]. The Extended Yale dataset is constructed from facial taken in different illuminations and facial expressions for each subject. The dataset we have used is the cropped version of the original Extended Yale B dataset, it contains between 58 and 64 images per class, each class contains images that represent one individual. The total number of classes in this dataset is 38 and the total number of image samples is 2414. An adequate percentage of the training data is adopted as we have used 10, 15, 20, and 25 samples from each class for training, and the remaining were used as test samples. Each image of this dataset is rescaled to 32×32 pixels and represented through grayscale representation. Raw brightness images of dimension 1024 are used in the experiments.

5. **FEI dataset**[10]**:** The FEI face dataset contains 700 images of 50 students and staff members of FEI (14 images for each person). It is a face dataset that contains a set of colorful face images (Images are resized to 32 × 32 pixels) taken against a white background, The images are in an upright frontal position with profile rotation of up to about 180 degrees. Raw brightness images of dimension 1024 are used. We used 5, 6, 7, and 8 image samples from each class as training samples. We should emphasize

[8] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[9] *http : //vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html*

[10] *https : //fei.edu.br/ cet/facedatabase.html*

that the choice for these training set sizes comes from the fact that the number of samples in each class of the FEI dataset is relatively low (only 14) compared to other datasets.

6. **MNIST dataset:** The large-scale MNIST digits dataset is challenging. It contains a total number of 60,000 images representing 10 different classes. The length of the used image descriptor is 2048. The descriptor is obtained from the (ResNet-50) [11] convolutional neural network.

7. **20 News text dataset**: This is a cropped version of the 20 newsgroups dataset, with binary occurrence data for 100 words across 16,242 postings. This dataset contains a total of 2000 samples belonging to 4 classes.

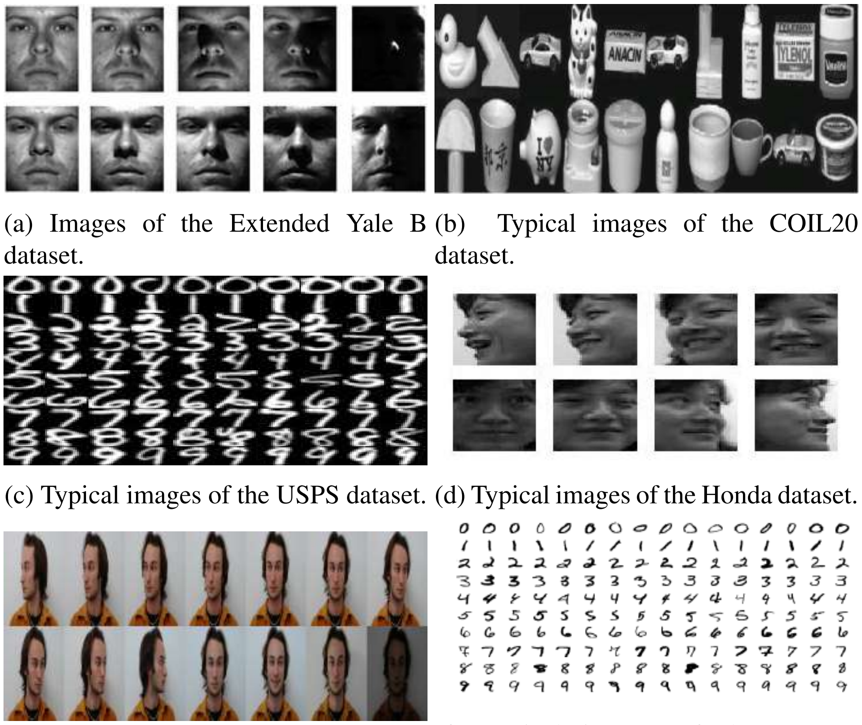
8. **Tetra synthetic dataset:** The terta dataset was defined in [28, 29]. This dataset consists of 400 data points belonging to four classes. The data points are in $\mathbb{R}^3$, this dataset presents the challenge associated with low inter-cluster distances.

Table 2 presents a summary for all the information concerning the datasets used in our paper.

Table 2: Brief datasets description.

Dataset	**Type**	**Number of Samples**	**Number of features**	**Number of classes**	**Descriptor**
USPS	Digits	1100	256	10	RAW-brightness images
Honda	Face	2277	1024	22	RAW-brightness images
COIL20	Object	1440	177	20	Local Binary Patterns
Extended Yale B	Face	2414	1024	38	RAW-brightness images
FEI	Face	700	1024	50	RAW-brightness images
MNIST	Digits	60,000	2048	10	Deep features (ResNet-50)
20 News	Text	2,000	100	4	Term Frequency times Inverse Document Frequency

[11] *https : //www.mathworks.com/help/deeplearning/ref/resnet50.html*



(a) Images of the Extended Yale B dataset.

(b) Typical images of the COIL20 dataset.

(c) Typical images of the USPS dataset.

(d) Typical images of the Honda dataset.

(e) Typical images of the FEI dataset.

(f) Typical images of the MNIST dataset.

Figure 3: Some Images of datasets

4.2 Results

As already reported, the proposed method has two variants, namely:

- Supervised discriminant analysis using gradient technique (**SDA_G_1**): In this variant, our proposed method is implemented in the case that the initial transformation matrix $\mathbf{Q}^{(0)}$ is set to the output of the RSLDA [31] algorithm as presented in section 3.3.1.

- Supervised discriminant analysis using gradient via combined initialization (**SDA_G_2**): The second variant of the proposed method consists of initializing the transformation matrix $\mathbf{Q}^{(0)}$ as a hybrid combination of the solutions derived from the RSLDA [31] and ICS_DLSR [33] methods. The initial transformation construction is shown in Figure 2 and detailed in section 3.3.2.

The proposed variants have been compared with the following methods: K-nearest neighbors (KNN) [12], Support Vector Machines (SVM) [3] (the Linear SVM was implemented suing the LIBSVM library[12] Linear Discriminant Analysis (LDA) [26], Local Discriminant Embedding (LDE) [4], PCE [19] (unsupervised method) ICS_DLSR [33] and Robust sparse LDA (RSLDA) [31].

All experiments for all compared methods were conducted under the same conditions to guarantee a fair comparison. For each compared embedding method, the whole dataset is randomly split into a training part and a test part.

First, for each compared method, a transformation matrix is estimated from the training part. Then, training and test data are projected onto the new space using the already computed transformation. Finally, the classification of the test data is then performed using the Nearest Neighbour classifier (NN) [6].

Different sizes of training sets were used. Moreover, for a given percentage of training data, the whole evaluation is repeated ten times. That means that we adopt ten random splits for every configuration and report the average recognition rate (rate of correct classification of test part) over these ten random splits.

We used PCA as a preprocessing technique. In our experiments, PCA [23] is used as a dimensionality reduction technique and used to preserve 100% of the data's energy. Concerning the parameter α we should set it to a small value. In our experiments, this value was chosen in $\{10^{-7}, 10^{-5}\}$.

[12] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 3: Mean classification performance (%) of the competing methods on the tested datasets.

Dataset \ Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**SDA_G_1**	**SDA_G_2**
USPS	30	87.01	88.21	84.91	83.54	72.01	88.46	89.45	89.50	**90.29**
	40	88.56	90.40	86.19	85.3	72.30	90.16	91.11	**91.81**	91.46
	55	90.51	92.09	88.64	87.16	73.32	91.25	92.65	**93.07**	92.87
	65	91.76	93.16	89.29	88.58	74.11	91.53	92.89	**93.71**	93.49
Honda	10	64.12	71.32	65.95	65.74	61.86	70.79	69.90	70.16	**72.14**
	20	77.69	83.60	79.39	79.25	75.33	82.95	83.03	83.60	**84.64**
	30	84.78	89.09	85.84	86.24	82.55	88.20	89.04	89.41	**90.12**
	50	91.36	94.15	92.28	92.34	90.03	93.53	94.13	94.53	**95.10**
FEI	5	88.98	91.18	92.60	90.67	86.04	92.16	93.19	93.81	**94.58**
	6	90.35	92.93	94.18	92.15	88.73	93.65	94.25	94.75	**95.08**
	7	92.60	94.31	95.60	94.26	91.09	95.20	95.66	96.20	**96.29**
	8	94.27	95.23	96.03	95.57	93.20	96.17	96.43	**96.97**	96.40
COIL20	20	94.58	97.65	96.19	95.00	94.87	**98.04**	96.73	96.89	97.66
	25	95.79	98.22	97.07	96.12	95.99	98.22	97.74	97.89	**98.59**
	30	96.65	98.70	97.81	97.01	97.49	98.75	98.26	98.52	**99.08**
	35	97.14	98.81	98.15	97.42	98.11	99.12	98.68	98.80	**99.39**

Table 4: Mean classification performance (%) of using the Extended Yale B dataset.

Dataset \ Method	Training Samples	KNN	SVM	LDA	LDE	ELDE	PCE	SULDA	MPDA	ICS_DLSR	RSLDA	**SDA_G_1**	**SDA_G_2**
Ext. Yale B	10	69.80	73.85	82.32	79.92	85.85	86.39	84.61	83.67	86.56	86.79	87.10	**88.42**
	15	75.20	80.02	86.76	83.77	89.30	89.23	88.72	86.82	89.53	89.93	90.04	**91.21**
	20	80.24	85.79	90.7	88.44	93.07	92.19	91.66	90.38	93.14	93.59	93.75	**93.81**
	25	82.24	89.03	92.17	90.43	94.09	93.35	92.14	91.79	94.50	94.92	95.02	95.09
		LRLR	SLRR	LRPP_GRR	LRRR								
	10	84.63	87.95	84.82	87.76								
	15	86.31	89.75	89.07	91.09								
	20	88.93	92.58	91.42	93.19								
	25	90.98	94.24	92.25	**95.51**								

The obtained results are summarized in Table 3. This table depicts the classification performance of the proposed variants in addition to the competing methods using the USPS, Honda, FEI, and COIL20 datasets. The results are obtained using different training and testing percentages from the data. Results shown in this table are obtained using the Nearest Neighbor classifier. Table 4 presents the obtained classification performance using the Extended Yale B dataset. In this table, various training percentages corresponding to a different number of samples used in the training process are shown. We should emphasize that more competing methods are presented in table 4, these additional methods are ELDE, in addition to SULDA and MPDA. These methods were added to enrich the comparison using more methods. To further improve the comparison over the Extended Yale B dataset, we have added more methods to the comparison table, based on low rank representations. The added methods are the Low-rank Linear Regression (LRLR) [2], Low-rank Ridge Regression (LRRR) [2], Sparse Low-rank Regression (SLRR) [2], and the Low-rank Preserving Projection via Graph Regularized Reconstruction (LRPP GRR) [32]. Low rank based methods findings can be found in the bottom part of table 4. The depicted rates are the average over 10 random splits and correspond to different numbers of training samples. The first column inside the table depicts the number of training images per class.

Table 5: Mean classification accuracies (%) of different methods on the tested datasets.

Dataset \ Method	Training Samples	KNN	SVM	LDA	LDE	PCE	ICS_DLSR	RSLDA	**SDA_G_1**	**SDA_G_2**
MNIST	1000	91.75	97.58	85.74	93.22	93.77	98.02	97.95	98.21	**98.33**

Table 5 illustrates the classification performance for the competing methods alongside the proposed variants using the large-scaled MNIST dataset that contains a total number of 60,000 images in total. Results shown in this table are obtained using one split while using 1000 samples from each class for training and the remaining samples were used for testing.

Table 6: Classification Performance (%) on the News20 text dataset.

News20			
Training Percentage			
20%		30%	
Method	Classification accuracy	Method	Classification accuracy
LDA	68.04	LDA	68.70
RSLDA	68.11	RSLDA	68.88
SDA_G_1	68.38	**SDA_G_1**	69.10
SDA_G_2	**68.87**	**SDA_G_2**	**69.58**

Table 6 depicts the obtained the classification performance using the News20 text dataset. Results presented in this table are the mean classification obtained using 10 split while using 20% and 30% of the data samples from each class for training and the remaining samples were used for testing.

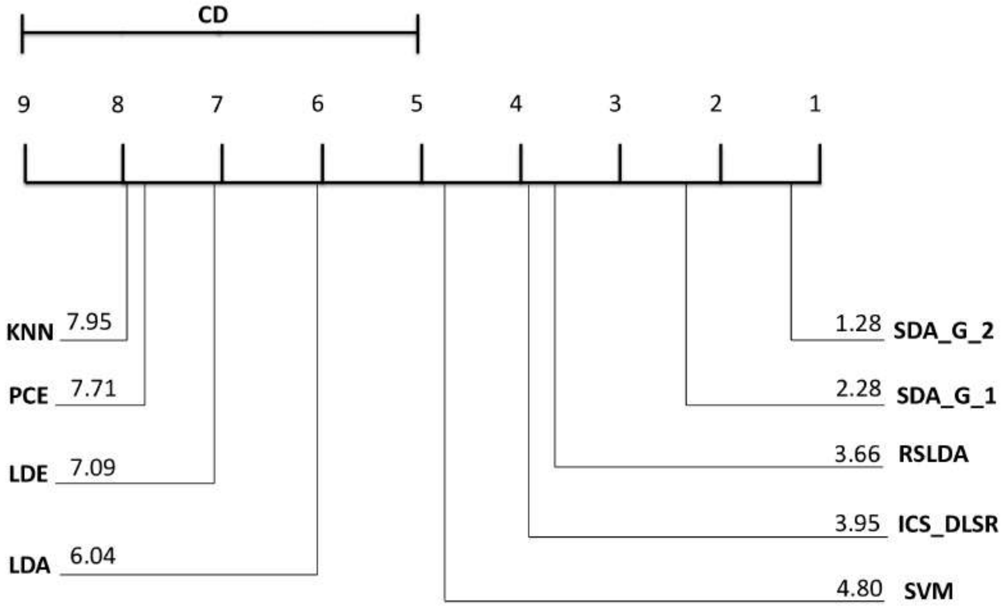
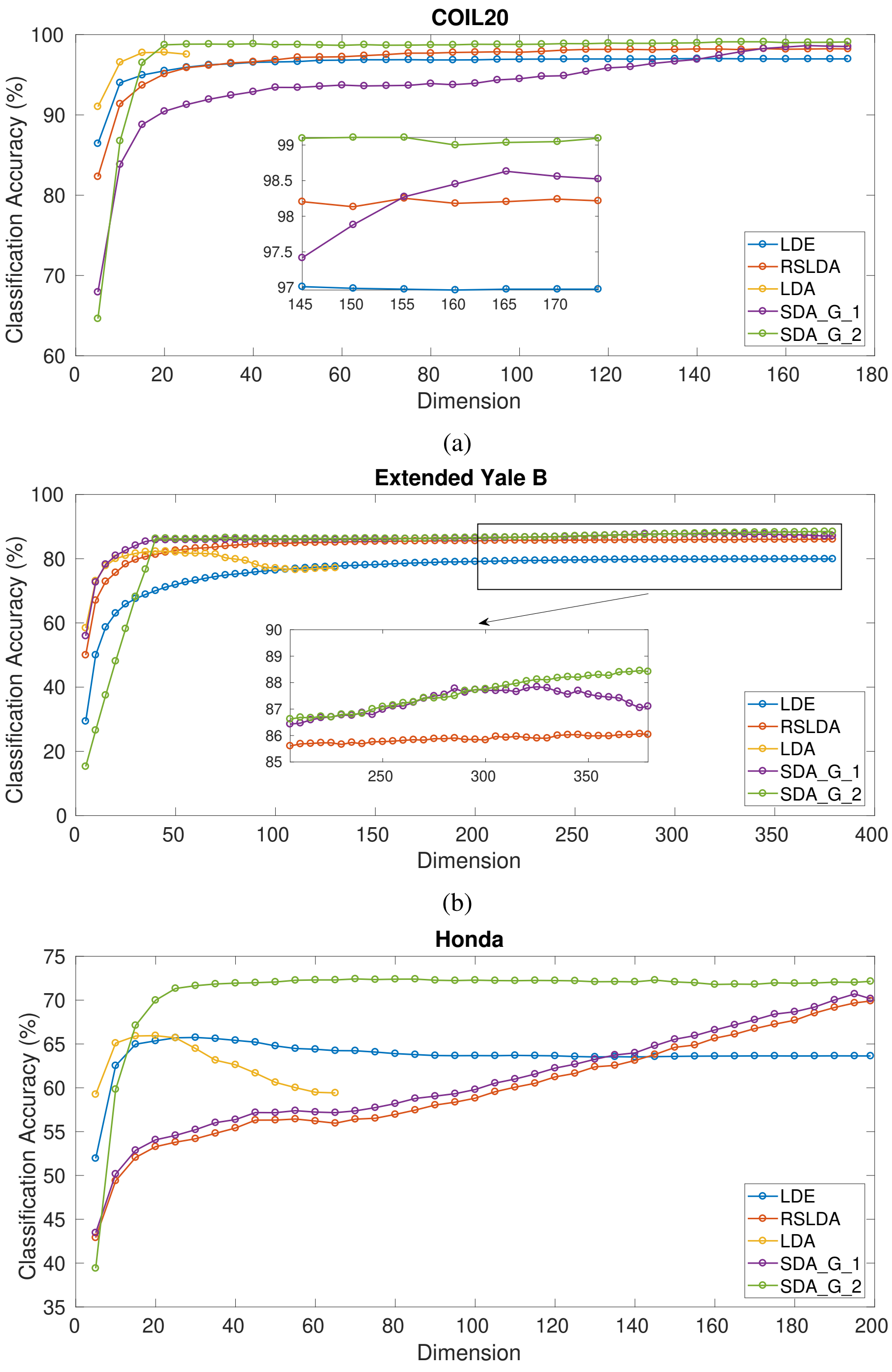


Figure 4: Statistical Analysis - CD diagram.



(a)

(b)

(c)

Figure 5: Classification accuracy (%) vs. dimension for different datasets.

Figure 5 presents the obtained recognition rate (%) associated with the LDA [26], LDE [4], RSLDA [31] in addition to the two proposed variants of our method. The recognition rate is plotted as a function of the dimension of the projected features. Results are shown for (a) the COIL20 dataset, (b) the Extended Yale B, and (c) the HONDA dataset. 30, 10, and

10 samples from each class are used for training, respectively. The depicted results were obtained using the **Nearest Neighbor** (NN) Classifier.

We have used the results obtained from 21 evaluations and using 6 different datasets from the experiments conducted in this paper to study the statistical analysis of our proposed method. We performed the Friedman test [7] and computed the critical distance CD. The obtained results of the conducted test yield to the conclusion that the tested methods do not have the same performance. Figure 4 shows the CD diagram for the 9 methods including our two proposed variants, where the average rank of each is marked along the axis.

Experiments using synthetic data:

In addition to the image datasets, we also conducted some experiments on the synthetic Tetra dataset [27]. This dataset consists of 400 data points belonging to four classes. The original data points of this dataset are in $\mathbb{R}^3$, but in our experiments, the dimension was augmented to 100 so each data sample is represented by 100 features. The 3-dimensional dataset is transformed to a high dimensional dataset $\in \mathbb{R}^{100}$ using a random projection matrix.

This dataset was chosen because it presents the challenge associated with low inter-cluster distances. The distance between the clusters is minimal. Data points of Tetra are visualized in Figure 6. One can see that the clusters nearly touch each other.

Figure 7 illustrates the TSNE visualization of the projected samples of the Tetra dataset using the original Linear discriminant Analysis LDA, RSLDA in addition to the first variant of our suggested method **SDA_G_1**. By looking at that figure, it is noticeable that our method provides very good class separation properties and lead to the most compact representation among competing methods. The proposed method ensured superior performance when it is applied on datasets with low inter-cluster distances.

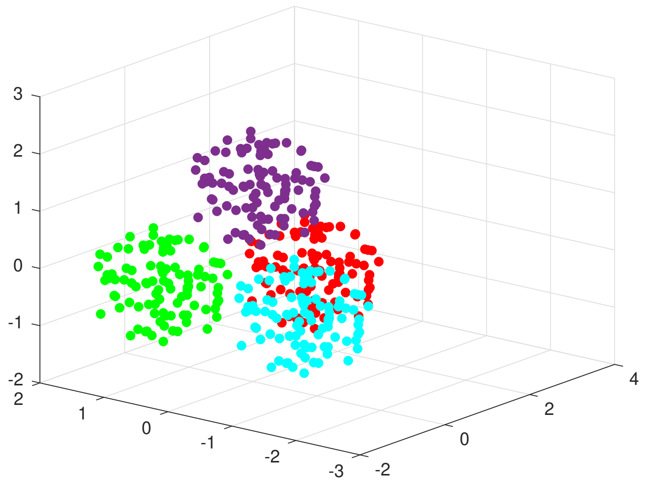
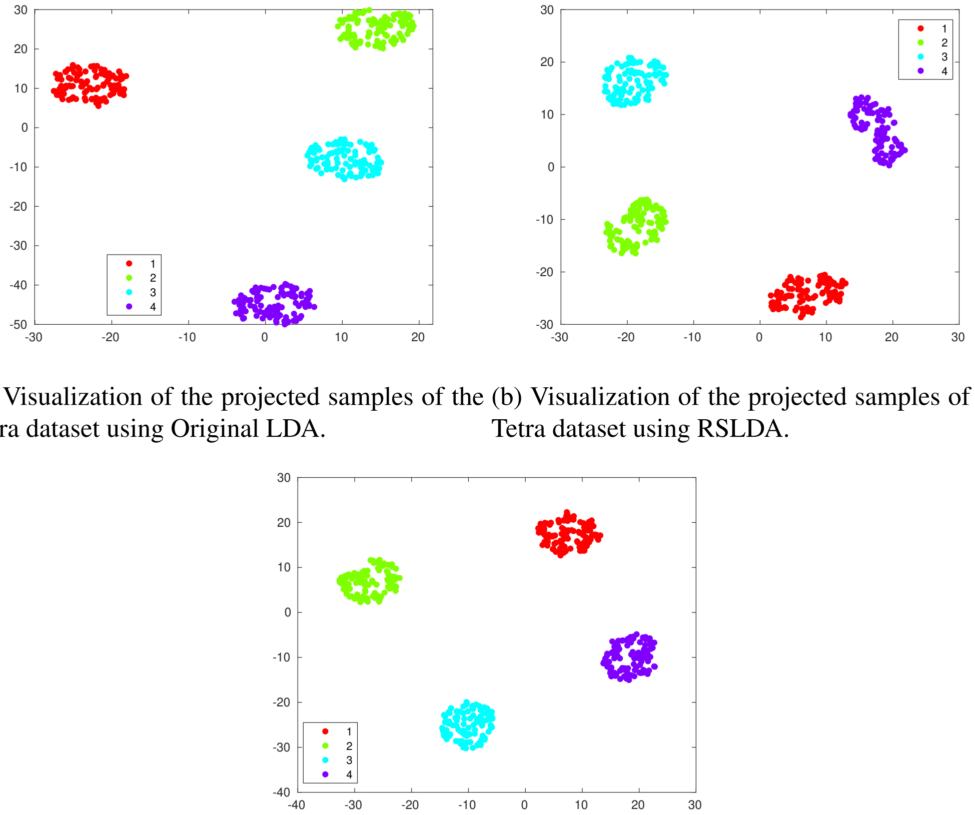


Figure 6: Visualization of the Tetra dataset points in the original space. These 3D points belong to four large full spheres close to each other.



(a) Visualization of the projected samples of the Tetra dataset using Original LDA.

(b) Visualization of the projected samples of the Tetra dataset using RSLDA.

(c) Visualization of the projected samples of the Tetra dataset using **SDA_G_1**.

Figure 7: TSNE Visualization of the projected samples of the Tetra dataset using LDA, RSLDA, and the first proposed variant **SDA_G_1**.

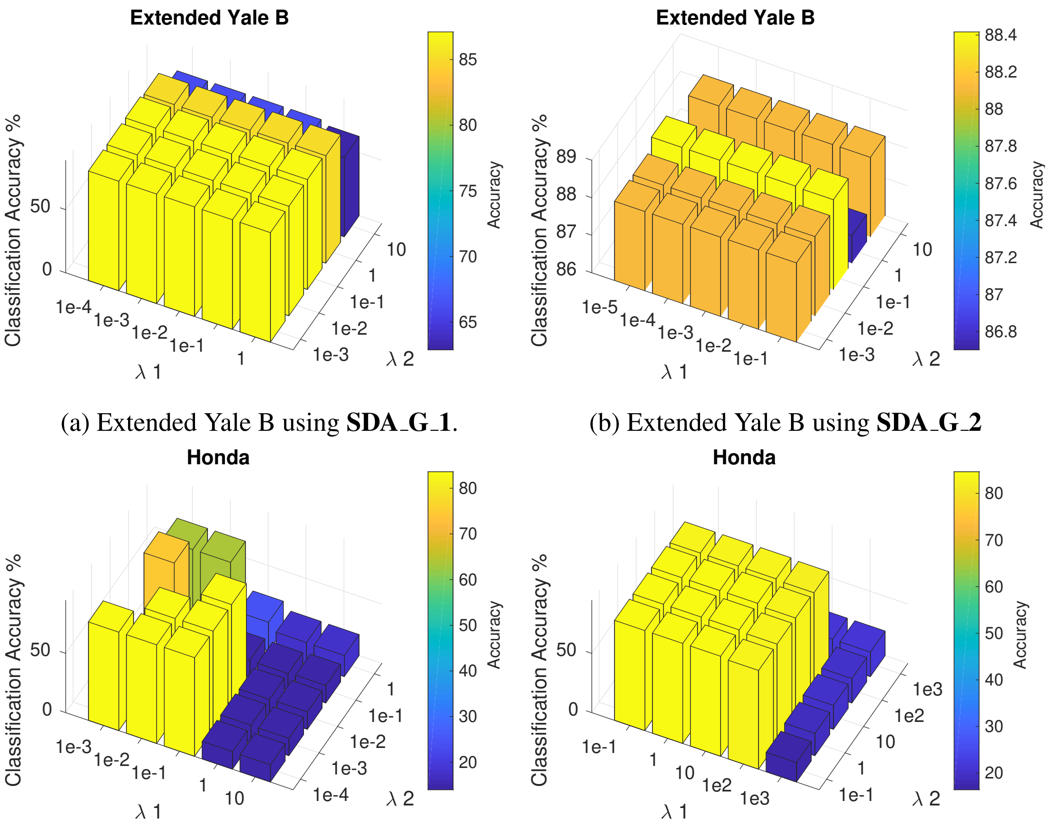
4.3 Analysis of Parameter sensitivity

In this section, we investigate the effect of changing the proposed method's parameters on the classification performance using different datasets. The proposed method has mainly two parameters to be configured, λ_1, and λ_2.

Figure 8 shows the variation of the classification performance when adopting different parameter combinations of the proposed method using the Extended Yale B and Honda datasets. Subfigures (8a) and (8c) shows the variation of the classification rates using the Extended Yale B and Honda datasets in the cases of using 10 and 20 training samples from each class, respectively, using the first variant of the proposed method **SDA_G_1**. Also, the classification performance is studied on the same datasets with adopting the same training percentages using the second variant of the proposed method **SDA_G_2**. Corresponding results are depicted in subfigures (8b) and (8d).

For the Extended Yale B dataset, we monitored the classification performance obtained by both of our proposed variants using different values for λ_1 and λ_2. λ_1 and λ_2 were varied for the ranges from $[10^{-5}, 1]$ and $[10^{-3}, 10]$ respectively. We noticed that satisfactory rates for the Extended Yale B dataset can be obtained when λ_1 was chosen from the range $[10^{-3}, 10^{-1}]$ and λ_2 whithin the range of $[10^{-2}, 10^{-1}]$.

Similar to the Extended Yale B experiment, we studied the classification performance of the proposed schemes over the honda dataset. We varied λ_1 in the range of $[10^{-3}, 10^{3}]$ and λ_2 in the range $[10^{-4}, 10^{3}]$. We noticed that satisfactory rates using Honda dataset can be obtained by choosing the value λ_1 from the range of $[10^{-1}, 10]$ and λ_2 from the range of $[10^{-3}, 10^{2}]$. We concluded that the values of the parameters λ_1 and λ_2 should lie in the intervals shown in the figures above to obtain satisfactory results using the proposed method. A value of 0.1 for both λ_1 and λ_2 seems to be a good choice for the two variants and the two datasets.



(a) Extended Yale B using **SDA_G_1**.

(b) Extended Yale B using **SDA_G_2**

(c) Honda using **SDA_G_1**

(d) Honda using **SDA_G_2**

Figure 8: Classification accuracy (%) according to parameters

Figure 8 shows the variation of the classification performance according to the change of the parameters λ_1 and λ_2. This figure corresponds to the variants of the proposed method when applied on the Extended Yale B and Honda dataset using 10 and 20 samples from each class for training respectively and the rest for testing.

4.4 Analysis of results

From our analysis of the experiments conducted, we can make the following observations:

1. The classification performance obtained by the proposed method alongside the competing methods demonstrates that our proposed approach has out-performed competing methods in the majority of the cases.

2. The first proposed variant **SDA_G_1** has slightly outperformed the RSLDA method. This seems to be very realistic since the first proposed method mainly provides a fine-tuning of the RSLDA transformation.

3. In general, the second proposed scheme **SDA_G_2** is superior to the first proposed one **SDA_G_1**. This is explained by the fact that the second scheme benefits from the hybrid combination of two different powerful embedding methods as well as from the refinement provided by the gradient descent tool.

4. The proposed method proved superior performance using several types of image datasets, including faces, objects, and digits. Also, our approach demonstrated superior performance using a text dataset.

5. The proposed method showed superiority and lead to very good class separation properties when it is applied on datasets with low inter-cluster distances.

6. The optimal parameters of the proposed method, which gives the best classification performance, have large ranges. In other words, the best classification performance is guaranteed most of the time by searching a small number of parameter combinations.

7. The competing method ICS_DLSR has performed better than our proposed method in a particular case using the COIL20 dataset while using 20 images from each class as training samples. On the other hand, the proposed method generally outperformed it using all other training percentages for the same dataset.

8. When the hybrid initialization was used in our algorithm, we adopted a combination of the two best-tuned transformation matrices obtained from the two methods RSLDA and ICS_DLSR as the initial transformation. In the majority of the tested cases, this has led to a noticeable enhancement in classification performance. The two best-tuned transformation matrices refer to the transformation matrices computed by two methods using the best parameter combination, which leads to the optimal performance of the method.

   It is worthy noting that the use of the combination of the two tuned transformation matrices is not necessarily the best option for a combination in our framework. Other combinations may lead to better discrimination. Thus, the obtained classification performance using the second variant of our suggested approach (Table 3) could be further enhanced if other combinations for the initialization are used.

5 Conclusion

In this work, we introduced a novel criterion to obtain a discriminant linear transformation. This transformation efficiently integrates two different mechanisms of discrimination which are the inter-class sparsity and robust discriminant analysis. We deployed an iterative alternating minimization scheme to estimate the linear transformation and the orthogonal matrix associated with the robust LDA. The linear transformation is efficiently updated via the steepest descent gradient technique.

We proposed two initialization variants for the linear transformation. The first scheme sets the initial solution to the linear transformation obtained by robust sparse LDA method (RSLDA). The second variant initializes the solution to a hybrid combination of the two transformations obtained by RSLDA and ICS_DLSR methods.

The two variants of the proposed method have demonstrated superiority over competing methods and led to a more discriminative transformation matrix, hence better classification performance.

The proposed framework is generic in the sense it allows the combination and tuning of other linear discriminant embedding methods.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] X. Cai, C. Ding, F. Nie, and H. Huang. On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1124–1132, 2013.

[3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and technology (TIST)*, 2(3):27, 2011.

[4] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 846–853. IEEE, 2005.

[5] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

[6] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34(8):1–17, 2007.

[7] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[9] Z. Fan, Y. Xu, and D. Zhang. Local linear discriminant analysis framework using sample neighbors. *IEEE Transactions on Neural Networks*, 22(7):1119–1132, 2011.

[10] X. Fang, S. Teng, Z. Lai, Z. He, S. Xie, and W. K. Wong. Robust latent subspace learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2502–2515, 2017.

[11] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):643–660, 2001.

[12] L. Kozma. k nearest neighbors algorithm (knn). *Helsinki University of Technology*, 2008.

[13] Z. Lai, Y. Xu, Z. Jin, and D. Zhang. Human gait recognition via sparse discriminant projection learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1651–1662, 2014.

[14] P. Langley. Selection of relevant features in machine learning: Defense technical information center. 1994.

[15] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu. Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2138–2150, 2013.

[16] L. Liu, P. Fieguth, and G. Kuang. Generalized local binary patterns for texture classification.

[17] A. M. Martínez and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[18] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20). 1996.

[19] X. Peng, J. Lu, Z. Yi, and R. Yan. Automatic subspace learning via principal coefficients embedding. *IEEE Transactions on Cybernetics*, 47(11):3583–3596, 2016.

[20] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[21] L. E. Raileanu and K. Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[22] A. K. Seewald. Digits-a dataset for handwritten digit recognition. *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*, 2005.

[23] L. I. Smith. A tutorial on principal components analysis. Technical report, 2002.

[24] U. Stańczyk, B. Zielosko, and L. C. Jain. Advances in feature selection for data and pattern recognition: An introduction. In *Advances in Feature Selection for Data and Pattern Recognition*, pages 1–9. Springer, 2018.

[25] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi. Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks and Learning Systems*, 27(4):796–808, 2015.

[26] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2):169–190, 2017.

[27] M. C. Thrun and A. Ultsch. Clustering benchmark datasets exploiting the fundamental clustering problems. *Data in Brief*, 30:105501, 2020.

[28] A. Ultsch. Kohonen's self organizing feature maps for exploratory data analysis. *Proc. INNC90*, pages 305–308, 1990.

[29] A. Ultsch. Self-organizing neural networks for visualisation and classification. In *Information and classification*, pages 307–313. Springer, 1993.

[30] D. Wang, F. Nie, and H. Huang. Feature selection via global redundancy minimization. *IEEE Transactions on Knowledge and Data Engineering*, 27(10):2743–2755, 2015.

[31] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen, and Y. Xu. Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):390–403, 2018.

[32] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan. Low-rank preserving projection via graph regularized reconstruction. *IEEE Transactions on Cybernetics*, 49(4):1279–1291, Apr. 2019.

[33] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu. Inter-class sparsity based discriminative least square regression. *Neural Networks*, 102:36–47, 2018.

[34] J. Xu, B. Tang, H. He, and H. Man. Semisupervised feature selection based on relevance and redundancy criteria. *IEEE Transactions on Neural Networks and Learning Systems*, 28(9):1974–1984, 2016.

[35] Y. Xue, L. Zhang, B. Wang, Z. Zhang, and F. Li. Nonlinear feature selection using gaussian kernel svm-rfe for fault diagnosis. *Applied Intelligence*, 48(10):3306–3331, 2018.

[36] J.-B. Yang and C.-J. Ong. An effective feature selection method via mutual information estimation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6):1550–1559, 2012.

[37] S. Zang, Y. Cheng, X. Wang, and J. Ma. Semi-supervised flexible joint distribution adaptation. In *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, pages 19–27, 2019.

[38] H. Zhang, S. Wang, X. Xu, T. W. Chow, and Q. J. Wu. Tree2vector: learning a vectorial representation for tree-structured data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5304–5318, 2018.

[39] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang. Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):689–700, 2015.

[40] R. Zhu, F. Dornaika, and Y. Ruichek. Joint graph based embedding and feature weighting for image classification. *Pattern Recognition*, 93:458–469, 2019.

[41] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 67(2):301–320, 2005.

[42] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.