eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

Master in Computational Engineering and Intelligent Systems

Computer Science

Master Thesis

Analysis of facial expressions: Experiments on multiple databases

Author

Leire Roa Barco

Supervisors

Dr. Fadi Dornaika Dr. Ignacio Arganda Carreras

informatika fakultatea facultad de informática

2017

Summary

This master thesis compares different face descriptors using classification techniques in order to classify emotions in images of faces of people of different ethnicities and ages, male and female. The comparison is done between hand-crafted features such as LBP and HOG and more modern features such as some pre-trained neural networks. The proposed methods were used on different databases, using different image sizes and cropping and standardizing all the images. The experimental results showed that some of the hand-crafted features were better that the pre-trained neural networks. To facilitate replication of our experiments the MATLBAB source code will be available at https://github.com/nagwlei/FaceEmotions.

Contents

List of Figures

List of Tables

1. CHAPTER

Introduction

Facial Expression Recognition is a field that has been actively studied in the last years for multpile application areas such as avatar animation, neuromarketing and sociable robots. However, it is a difficult problem, since the way people show their expressions can vary in brighntness, background and pose. Even more, these variations are bigger if we consider different subjects with variations in shape or etnicity.

We know that facial expression recognition is very studied, but few works perform fair evaluation avoiding mixing subjects while training and testing the proposed algorithms. So, in this work we aim to do an analysis of the different techniques used to do so in multiple databases that contain people from different etnicities and ages. To do so, in the second and third chapter the state of the art and used techniques are explained. After this we explain the different experiments we have carried out in the different databases, and finally the obtained conclusions are explained.

2. CHAPTER

Related Work

Facial expression is probably the best way of human emotion recognition, because the facial changes are linked to internal emotional state, intentions or social communication. Expression recognition is a task that humans perform easily, except some people with disabilities, like autistic persons; but it is not an easy work for computers. Although recent methods present accuracies larger than 95% in conditions like frontal face, controlled environments and high-resolution images, it is true that in many works in the literature we can see that there is not a consistent evaluation methodology, and most of the face expression recognition problems do not represent real scenarios. Even more, in cross-database evaluations and on databases with uncontrolled environments it has been reported low accuracy [Lopes et al., 2017].

Nevertheless, in the last decades it has been an increasing progress in recognition performance. It has been possible in part thanks to the emergence of Deep Learning methods, specifically with Convolutional Neural Networks. These approaches are due to a larger amount of data available to train learning methods and due to de advances in GPU technology. The larger amount of data is necessary for training networks with deep architectures, and the GPU technology permits a low cost high-performance numerical computation.

Where is the problem for computers? It is a challenge very hard for them to separate the feature space of expressions: in one hand facial features from one subject in two different expressions may be very similar in the feature space, and in the other hand features from two subjects with the same expression may be very different from each other. Even more, some expressions, like fear and sad, in some cases, are very similar [Jain and Li, 2011].

There is another challenge related to the facial expression recognition, which is that the training-testing scenarios can be very far from the testing images in terms of environmental conditions and subject ethnicity. One way to solve this problem is to train the method with one database and to test it with another database from different ethnic groups.

We can divide the systems of facial expression recognition into two categories: one is the one that works with static images, and the other is the system that works with dynamic image sequences. In the first case, with static images, they do not use temporal information, and the feature vector has information about the current input image only. In the second case, sequence based methods use temporal information of images to recognize the expression captured from one or more frames.

In both cases automated systems for facial expression recognition receive the input and give as output one of six basic expressions (anger, sad, surprise, happy, disgust and fear), and some systems also recognize a neutral expression.

Li and Jain [Jain and Li, 2011] describe automatic facial expression analysis in three steps: face acquisition, facial data extraction and representation, and facial expression recognition. The step of face acquisition can be divided in two steps: one is face acquisition and the other is head pose estimation. After the face acquisition we need to extract the facial changes caused by facial expressions. These changes are usually extracted using geometric feature-based methods or appearance-based methods. Geometric feature-based methods use shape and location of facial components like mouth, eyes, nose and eyebrows. The vector that represents the face geometry is composed of facial components or facial feature points. Appearance-based methods use feature vectors extracted from the whole face, or from specific regions; these vectors are achieved using image filters applied to the whole face image.

Expression recognition can be performed when feature vectors related to the facial expression are available. Following Liu et al. we know that expression recognition systems use a procedure with three stage: the first stage is feature learning, the second is feature selection and the third stage is classifier construction [Liu et al., 2014].

In the first stage the feature learning is responsible for the extraction of all features to represent the facial expression. They minimize intra-class variation and maximize inter-class variation. Both works are difficult, because to minimize the intra-class variation of expressions of different individuals with the same expression we can find images far from each other in the space of pixels; similarly maximizing the inter-class variation we can find expressions very close to one another in the space of pixels. At the end of the process we have a classifier or a set of classifiers with one for each expression, and it is used to infer the facial expression of the selected features.

There is a successful technique applied to the facial expression recognition problem and it is called the deep multi-layer neural network. It groups the three steps: learning, selection of features and classification in one single step [Fasel, 2002].

Some recent approaches for facial expression recognition have focused on uncontrolled environments such as not frontal face or spontaneous expressions, which is a difficult problem. The work of Teixeira et al. [Lopes et al., 2017] focuses on more controlled environments and discusses recent methods that achieve high accuracy in facial expression recognition using a comparable experimental methodology or methods that are based on deep natural networks.

There is another approach called Boosted Deep Belief Network (BDBN), proposed by Liu et al., and is composed by a set of classifiers, named as weak classifiers. Each weak classifier is responsible for classifying one expression. They perform the three learning stages in a unique framework: feature learning, feature selection and classifier construction [Liu et al., 2014].

Song et al. created a facial expression recognition system that uses a deep CNN and runs on a smartphone. They use a network composed of five layers and 65.000 neurons. They say that it is common to have an overfitting when using a small amount of training data and a big network. For that reason they applied data augmentation techniques to increase the amount of training data and used the drop-out during the network training [Song et al., 2014].

Burkert et al. proposed a method based on CNNs and their network architecture consists of four parts: the first part is responsible for the automatic data preprocessing, and the remaining parts carried out the feature extraction process. The extracted features are classified into a given expression by a fully connected layer at the end of the network. The architecture comprises 15 layers (7 convolutions, 5 poolings, 2 concatenations and one normalization layer). They did not guarantee that subjects used in training were not used in test. This is an important restriction in order to perform a fair evaluation of facial expression recognition methods [Burkert et al., 2015].

Liu et al. proposed an action unit (AU) inspired Deep Networks (AUDN) trying to explore a psychological theory that says that expressions can be decomposed into multiple facial expression action units. In their opinion the method is able to learn informative local appearance variation, an optimal way to combine local variations and a high-level representation for the final expression recognition [Liu et al., 2015].

Ali et al. proposed a collection of boosted neural network ensembles for multi-ethnic facial expression recognition. Their model has three steps: first a set of binary neural networks are trained, second the predictions of these neural networks are combined to compose the connexion of ensembles and third, these collections are used to detect the presence of an expression [Ali et al., 2016].

Shan et al. performed a study using Local Binary Patterns (LBP) as feature extractor. They combined and compared different machine learning techniques like template matching, Support Vector Machine (SVM), Linear Discriminant Analysis and liner programming to recognize facial expressions. They also conducted a study to analyse the impact of image resolution in the accuracy result and concluded that methods based on geometric features do not handle low-resolution images well, but the ones based on appearance, like Gabor Wavelets and LBP, are not so sensitive to the image resolution [Shan et al., 2009].

Byeon et al. proposed a video-based facial expression recognition system and developed a 3D-CNN with an image sequence (from neutral to final expression) using 5 successive frames as 3D input. They achieved an accuracy of 95% but the method relies on a sequence containing the full movement from the neutral to the expression [Byeon and Kwak, 2014].

There is another video-based approach proposed by Fan and Tjahjadi, who use a spatial-temporal framework based on histogram of gradients and optical flow. Their method has three phases: pre-processing, feature extraction and classification. In the first phase they detect facial landmarks and make and face alignment trying to reduce variations in the head pose, in the second phase they employ a framework that integrates dynamic information extracted from the variation in the facial shape caused by the expressions, and in the last phase they make the classification by using a SVM classifier with a RBF kernel [Fan and Tjahjadi, 2015].

Finally, Andre Teixeira and others in their work [Lopes et al., 2017] propose a solution for facial expression recognition that uses a combination of CNN and specific image pre-processing steps. This approach is focused on methods based on statistic images, and considers the six basics expressions plus neutral for controlled and uncontrolled scenarios.

3. CHAPTER

Theoretical Neural Networks

A neural network is a classifier that has inputs and output; it is a classifier made by connected neurons in which every neuron calculates when it has to go on or go off depending on the threshold, and as well as with the classifiers, neural networks are trained too, and it is inspired by biological neural networks.

3.1 Perceptrons

Perceptrons are the most basic "neurons" that compose any Neural Networks. This "neurons" were developed in the 1950s and 1960s by the scientis Frank Rosenblatt based on the work by Warren McCulloch and Walter Pitts [McCulloch and Pitts, 1943].

The perceptrons can be understood in two different ways: as a neuron or as a classifier. If we understand the perceptron as a classifier, it is a type of linear classifier that takes several binary inputs $x_1, x_2, \ldots$, and produces a single binary output.

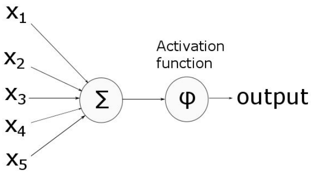

Figure 3.1: Perceptron [Nielsen, 2015].

These elements have to one or more inputs x_1, x_2, ..., x_n, where each of the inputs has a weight w_1, w_2, ..., w_n, and these weights are used to determine the output of the "neuron" which is a single binary value. If a perceptron has an output but it has not inputs the perceptron would simply output a fixed value, not the desired value. So, it is better to interpret the input perceptrons not as perceptrons themselves, but as units defined to output the desired values. The perceptron which can be understand as an algorithm for learning a binary classifier whose function can be defined as follows:

$$\text{f}(x) = \begin{cases} 1 & \text{if w} \cdot \text{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Where w is the vector of the weights, x is the input real value vector and $w \cdot x$ is the dot product $\sum_{i=0}^{n} w_i x_i$ being n the number of inputs and b the bias. This bias shifts the decision boundary away from the origin and does not depend on any input value. This function basically describes an hyperplane that separates the two regions we want to classify.

One problem with the perceptrons is that the perceptron learning algorithm does not determinate if the learning set is linearly separable, and because of this if the vectors are not linearly separable it will never reach a point where all the elements are classified. An example of this problem is the Boolean exclusive-or problem (XOR). However, according to Rosenblatt's theory, a Perceptron can also be understand as a neural network of perceptron (a multi-layer perceptron which will be explained later) and the perceptron algorithm is interpreted as a single-layer perceptron (which is the simplest feedfordward neural network). This concept of multilayer network perceptron (MLP) is able to implement all the logic arithmetic operators such as AND, OR, XAND. among others, that can lead to sophisticated decision making.

3.2 Sigmoid neurons

If we make a small change in the inputs of the perceptrons understanding them as functions, the output might be highly affected by this. The sigmoid neurons, also known as sigmoid functions, are the solution to this: They are a really similar classification function, but instead they can handle small changes in the input without the whole network getting strongly affected.

The sigmoid function as the perceptron function has as an input a vector x where each

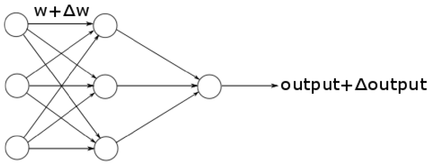

Figure 3.2: Sigmoid Neurons [Nielsen, 2015].

of the values of the vector has its own weight of $w_1, w_2, \ldots, w_n$. However, unless in the perceptron, the output is not a 0 or a 1, it is defined by a function called sigmoid function δ (also called sigmoid activation function) defined by:

$$\delta(\mathrm{z}) = \frac{1}{1+e^{-z}} \tag{3.1}$$

It is an S shape (sigmoid curve) function where the input z is w $z = w \cdot x + b$, being w the weight vector and x the input binary vector.

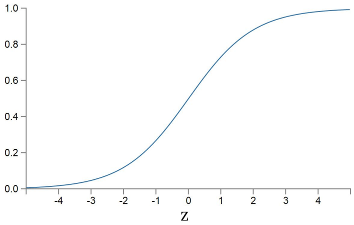

Figure 3.3: Logistic Function from [Nielsen, 2015].

As seen in the figure above, it has horizontal asymptotes in $z \rightarrow \infty$ that the value is 1 and in $z \rightarrow -\infty$. So, when the value of z is negative and small, the sigmoid function is close to the perceptron fucntion. In the other cases the functions are really different.

3.2.1 Rectifier

There is also a activation function close to the previously metioned ones called rectifier which is defined as

$$\mathrm{f(x) = max(0,x)} \tag{3.2}$$

where x is the input to a neuron. This is the most used activation function for deep neural networks and the units that use this method are called rectified liner units (ReLUs).

3.3 Architecture of Neural Networks

3.3.1 Feedfordward Neural Networks

Feedfordward Neural Networks are acyclic graphs usually composed at least of 3 layers: The input layer, the hidden layer and the output layer.

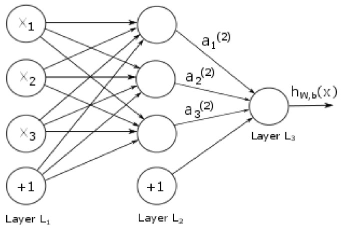

Figure 3.4: Neural Network model [Nielsen, 2015].

If there are more than 3 layers in a Neural Network, all these extra layers are hidden layers. Every layer is composed by units $x_1, x_2 ...$, and all the layers except the output layer also have a bias units, that is the intercept term and it is usually represented with a "+1".

For example in the layer in the image we have a Neural Network composed of 3 layers, and two of them the first and second layer are composed by 3 input units and a bias unit. These bias units do not have any input connection to them, because they always output the value "+1". So that, we have a neural network (Θ)=(Θ^0, Θ^1, Θ^2). The representation of this neural network is as follows:

$a_i^{(j)}$ = "activation" of unit i in layer j (a_0 represents the bias unit).

$\Theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j+1$.

$g(x)$ = logistic activation function.

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_\Theta(x) = a_1^3 = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

If we want to put it in a more compact way, it can be expressed in a vectorized way:

$$\begin{aligned} z^2 &= \Theta^{(1)}x & \qquad z^2 &= \Theta^{(1)}a^{(1)} \\ a^2 &= g(z^{(2)}) & a_0^{(2)} &= 1 \\ z^3 &= \Theta^{(2)}a^{(2)} \\ h_\Theta(x) &= a^{(3)} = g(z^{(3)}) \end{aligned}$$

This kind of Neural Networks instead of being constrained to fit the features x_1,x_2,x_3 to logistic regression, it has the flexibility to learn its own features $a_1^{(2)}$, $a_2^{(2)}$, $a_3^{(2)}$ to fit into logistic regression.

Depending on which parameters it chooses for z_1 it can learn some interesting features, and therefore we can end up with a better hypothesis than if we were constrained to use the features x_1, x_2, x_3 or contrained to use the polinomio terms $x1x2$, $x2x3$ [Turing Finance, 2014]. [1]

3.4 Training of Feedfordward Neural Networks

Any multilayer Neural Network needs to be trained in order to learn the features of the model, so that it is easier for the Network to solve the classification problem we want to solve.

3.4.1 Cost Function

The cost function of a Neural Network is a measure of how good the Neural Network is performing, and learning algorithms search through the solution in order to find a function that minimizes the cost.

Having a training set ($x_{(1)}$,$y_{(1)}$), ($x_{(2)}$,$y_{(2)}$), ... ,($x_{(n)}$,$y_{(n)}$), we define the regularized logistic regression cost as follows:

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\Theta(x^{(i)}) + (1-y^i)\log(1-h_\Theta(x^{(i)}))] + \frac{\lambda}{2m}\sum\frac{n}{j=1}\Theta_j^2$$

[1] If the network has s_j units in layer j, s_j+1 units in layer $j+1$, then Θ^j will be of dimension s_{j+1}x(s_j+1).

The first term of the sum represents the cost function and the second term is the regularization term (Note that we start the sumatory from j=1 beause we did not regularize the bias term Θ_0). If we are doing a binary classification with for example a tanh function, then we would have y=0 or y=1, and $h_\Theta \in \mathbb{R}$ and there would only be one layer. Whereas, if we have a multiclass classification, $y \in \mathbb{R}^k$, and we would have k outputs units, so $h_\Theta(x) \in \mathbb{R}^k$, always being $k \geq 3$.

The cost function for the whole Neural Network would be:

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1-y_k^{(i)})\log(1-(h_\Theta(x^{(i)}))_k)]+$$

$$\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

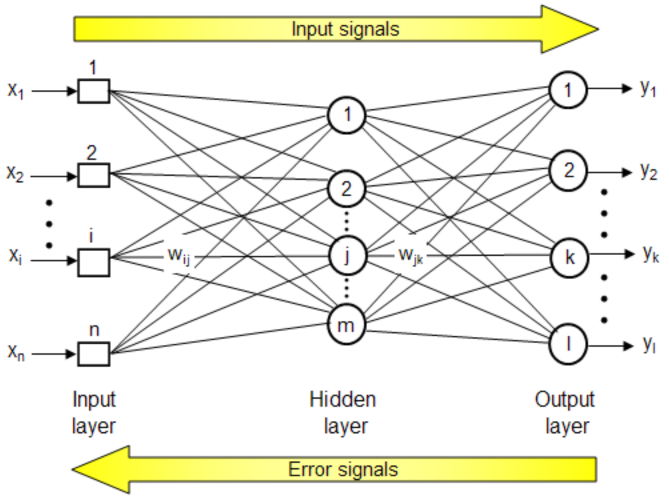

Figure 3.5: Feedforward NN with backpropagation [Buranajun et al., 2007].

As in the previous equation, the first term of the equation is the cost function itself and the second term is the regularization term, where we do not regularize the bias terms. $h_\Theta(x) \in \mathbb{R}^k$ and $h_\Theta(x)_i$ = i_{th} output.

3.4.2 Backpropagation

One of the most popular methods for training this layers is the backpropagation algorithm, an algorithm based on gradient descent (a first order optimization algorithm that minimizes functions by iteratively moving in the negative direction of the function gradient). It is a suppervised training algorithm used for training artificial network, that takes an input and forwards it through all the layer till the last layer. Once in there it compares the expected output with the real output and an error value is calculated for each one of the outpus. This errors are backpropagated till the input layers, but this layers do not obtain all the same error value, they receive a value depending on their contribution to the total error. The idea is that once the training is done, and each time the training is done again, the neurons will learn to recognize the features of the inputs and the intermediate layers will organize themselves.

The algorithm of this process would be:

```
Training set ($x^{(1)}$,$y^{(1)}$), ..., ($x^{(m)}$,$y^{(m)}$)
Set $\Delta^{(l)}_{i,j}$=0 (for all $l$, $i$, $j$) (used to compute $\frac{\partial}{\partial\Theta^{(l)}_{ij}}J(\Theta)$
for  m times do
    Set $a^{(1)}$=$x^{(i)}$
    Perform forward propagation to compute $a^{(l)}$ for $l = 2,3,\ldots,L$
    Using $y^(i)$, compute $\delta^{(L)}$=$a^{(L)}$-$y^{(i)}$
    Compute $\delta^{(L-1)}$,$\delta^{(L-2)}$,...,$\delta^{(2)}$
    Use backpropagation $\Delta^{(l)}_{ij}$:=$\Delta^{(l)}_{ij}$+$\delta^{(l+1)}_i(a^{(l)})^T$
end for
if $j \neq 0$ then
    $D^{(l)}_{ij}$:=$\frac{1}{m}\Delta^{(l)}_{ij}$ + $\lambda\Theta^{(}_{ij}l)$
else
    $D^{(l)}_{ij}$:=$\frac{1}{m}\Delta^{(l)}_{ij}$
end if
```

In the algorithm, first of all we have a training set of m elements. Then the deltas are set to 0 because these are going to be used as acumulators to compute the partial derivatives. Then we loop through the whole training set, and when we are in the i=1 that means we will be working with the sample (x^i,y^i). First of all we set the activations of the input layer to equal x^i, then we compute propagation to the rest of the layers till the last layer, layer L. After that y^i is used to compute the error $\delta^{(L)}$ for the output layer (it is the hypotesis output $a^(L)$ - what the target label was). Then, with backpropagation we are going to calculate $\delta^{(L-1)}$, $\delta^{(L-2)}$, ..., δ^2 (There is no δ^1 because we do not associate the error term with the input layer). Finally we use the delta terms to accumulate the derivative terms. At the end we compute $D_{i,j}$, where if j=0 it means that that is a bias term. And, that is the partial derivative of the cost function to each of the parameters, and that can be used in gradient descent or other optimization algorithms.

Depending on how it is implemented we might end up calculating δ also.[2] To solve it, we can use gradient checking and implement $\frac{d}{d\theta}J(\theta) \approx \frac{J(\theta+\varepsilon)-J(\theta-\varepsilon)}{2\varepsilon}$ [Yu et al., 2015] [Huang et al., 2015][Zhous et al.,].

3.4.3 Random Initialization

In order to work with a Neural Network it is neccesary to initialize the input Θ so that the Network starts to learn. One option might be to initialize the Θ to zeros, however, if we do this, after each update all the hidden layers are computing the exact some features, and this prevents the Neural Network form learning something interesting. So, it would be a better idea to intialize each $\Theta_{ij}^{(l)}$ to a radom value in $[-\varepsilon, \varepsilon]$. This process is called symetry breaking. However, this still fails to break the symetry in a full neural network, so we need to do the following steps:

- Implement backpropagation
- Do gradient checking
- Use gradient descent or one of the advanced optimization algorithims to minimize $J(\Theta)$ for the parameters starting from this randomly initialized parameters (symetry breaking)

Hopefully it will be able to find a good value of Θ.

[2] There is no $\delta^{(1)}$ because the first layer corresponds to the input layer and that is just the feature we observed in our trainning set, and it does not have any errors associated with it.

4. CHAPTER

Face Descriptors

In a lot of computer vision problems setting visual correspondences is really important, and one of the ways of doing this is by using feature-descriptors [Brown et al., 2005]. These are algorithms that take an image and output feature descriptors, also called feature vectors. These vectors encode information that can be used to differentiate one feature from another.

Feature vectors can be local or global; the local features describe the keypoints of the images, and usually are used for low level applications such as object detection and classification. Whereas the global features describe the image as a whole to generalize the entire objects, and some examples of their features include contour representations, shape descriptors and texture features. In general global descriptors are used for higher level applications such as object recognition.

In this case we are going to focus more in the local features. All of them are able to detect the same points independently in multiple images and are invariant to translation, rotation, scale, affine transformation, and presence of noise, blur etc. So, this kind of descriptors are robust to occlusion, clutter and illumination changes.

4.1 Local Binary Pattern (LBP)

It is a type of visual descriptor used for computer vision, that relies on thresholding the neighborhood and considers the result as a binary number [Ojala et al., 2002]. It is implemented in the following way

- **Division in cells:** Divide the examined window into cells.
- **Pixel comparision:** For each pixel in a cell compare the pixel to each of its 8 neighbors.

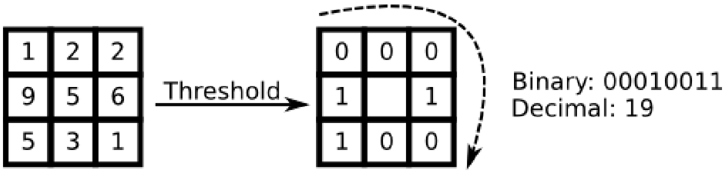

Figure 4.1: Pixel comparison example [OpenCVdocs, 2017].

- **Compute histogram:** Compute the histogram over the cell of the frequency of each number occurring. The histogram is a feature dimension whose dimension depends on the input parameters. It is recommendable to normalize the histogram.
- **Concatenate histogram:** Concatenate the histogram of all the cells.

There are multiple variations or extensions of the LBP, as for example Transition Local Binary Patterns (tLBP) [Trefnỳ and Matas, 2010], Direction coded Local Binary Patterns (dLBP) [Trefnỳ and Matas, 2010], Multi-block LBP or Pyramid-Based Multi-scale LBP [Qian et al., 2011].

4.2 Histogram of Oriented Gradients (HOG)

It is a feature descripor that relays on counting the occurrences of gradient orientation in a localized portion of an image [Dalal and Triggs, 2005].
In order to generate the approach the following steps are followed:

- **Gradient computation:** The image is divided into small connected regions called cells, and for each cell a histogram of gradient directions or edge orientations for the pixels within the cell is computed.
- **Orientation binning:** Each cell is discretized into angular bins according to the gradient orientation. Then, each cell's pixel contributes weighted gradients to its corresponding angular bin.

- **Descriptor blocks:** Groups of adjacent cells are considered as spacial regions called blocks, and the grouping of cells into a block is the basis for grouping and normalization of histograms.

- **Block normalization:** The block of histograms that are composed by a normalized group of histograms represent the descriptor.

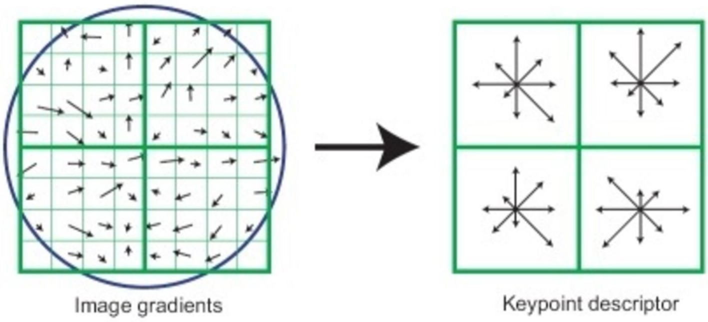

Figure 4.2: Gradient to keypoint [Lowe, 2004].

4.3 Pyramid-Based Multi-scale LBP

This is a variation of the conventional LBP method, where a image pyramid is used as an input, instead of the original image [Qian et al., 2011]. A image pyramid is a concatenation of images and images of lower size (usually the half of the previous image). For each of this images the LBP is calculated, and the feature vector that each of these images creates is returned. As seen in the following two images:

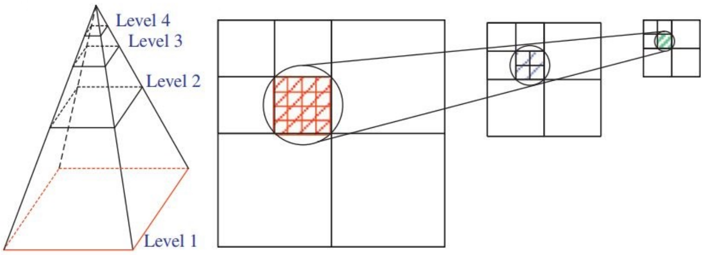

Figure 4.3: Visualization of Pyramid [Qian et al., 2011].

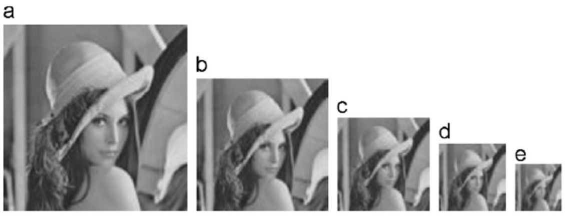

Figure 4.4: Example of pyramid image [Qian et al., 2011].

4.4 Multi-block LBP

It is another variation of the LBP where each of the images is divided into many blocks. For each of this block the LBP histogram is calculated, and finally all the histograms are concatenated to create the feature vector [Zhang et al., 2007].

4.5 Binarized Statistical Image Features (BSIF)

It is a feature descriptor that is similar to LBP in the way that it describes each pixel's neighborhood by a binary code which is obtained by first convolving the image with a set of linear filters and then binarizing the filter responses. Nevertheless, unlike in LBP these filters are learned by using statistics of natural images [Kannala and Rahtu, 2012].

The method is as following:

- The code value of a pixel is considered as a local descriptor of the image intesity pattern in the pixel's surroundings.
- The value of each element (each bit) in our binary code string is computed binarizing the response of a linear filter with a threshold at zero.
- Each bit is associated with a different filter and the desired length of the bit string determines the number of filters used.
- The filters are learnt from a training set of natural image patches by maximizing the statistical independence of the filter responses.

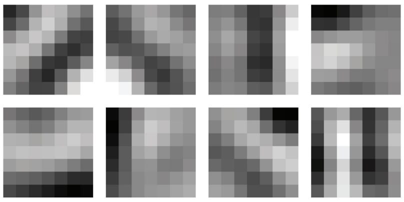

Figure 4.5: Example of learnt filter of size 9x9 [Kannala and Rahtu, 2012].

4.6 Hybrid features

These descriptors are the ones that are created mixing different kinds of descriptors, as for example concatenating the feature vectors obtained using the LBP and HOG.

4.7 VGG-F features

The VGG-F [Chatfield et al., 2014] is a pretrained Convolutional Neural Network (CNN) [LeCun et al., 1998]. The CNNs and are very effective in learning features with a high level of abstraction. They use deeper architectures, with a lot of layers, and new training techniques. This hierarchical network has alternating types of layers including convolutional layers, sub-sampling layers and fully connected layers.
Convolutional layers are characterized by the size of the kernels and by the number of maps generated. The kernel is shifted over the valid region of the input image and generates one map. Subsampling layers are used to increase the position invariance of the kernels and reduce the map size. There are two main types of sub-sampling layers called maximun-pooling and average pooling. The layers which are fully connected on CNNs are similar to the layers in general neural networks and their neurons are fully connected with the previous layer: convolution layer, subsampling layer or a fully connected layer. The CNN learns by finding the best weights of synapses. The learning can be performed using a gradient descent method, like the method proposed by LeCun et al. The CNN has one main advantage, because the input of model is a raw image instead of a set of hand-coded features.

In the case of the VGG-F, the CNN is composed of 8 learnable layers layers; 5 convolutional layers and 3 fully-connected layers. The input image has to be a 224x224 image, and the CNN was trained learning on ILSEVRC-2012 database using gradient descent with momentum.

Arch.	conv1	conv2	conv3	conv4	conv5	full6	full7	full8
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max

Figure 4.6: VGG-F architecture [Chatfield et al., 2014].

The hyperparameters used are the followying ones: momentum 0.9; weight decay 5x10^{-4}; initial learning rate 10^{-2}, which is decreased by a factor of 10, when the validation error stop decreasing. The layers are initialised from a Gaussian distribution with a zero mean and variance equal to 10^{-2}.

In order to work with this network, first of all, we have resized all the images of all the databases to 224x224, to match the entrance dimension. Also, we have not taken the last layer of the CNN, because it performed the classification for the ILSRV problem, and what we wanted to do was to take the 4096 dimension feature vector that was in this previous layer.

4.8 VGG-Face features

The VGG-Face [Parkhi et al., 2015] is a CNN trained using the IMDb database to recognise faces. It is formed by 11 blocks, each one containing a linear operator followed by one or more non-linearities such as ReLU and max pooling.

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Figure 4.7: VGG-Face architecture [Parkhi et al., 2015].

The first eight blocks are convolutional and the last three blocks are instead called Fully Connected (FC). All the convolution layers are followed by a rectification layer (ReLU). The input of the network is a 224x224 face image with the average face image substracted.

This CNN was trained using the IMDb database and was trained using 2,622,000 images (of 2622 different people). The hyperparameter setting used to train the network was the following one: It uses optimisation by stochastic gradient descent using min-batches of 64 samples and momentum conefficient of 0.9. The model is regularised using dropout with weight decay of 5x10^{-4}, and the dropout was applied after the two FC layers with a rate of 0.5. The learning was initially set 10^{-2} and then decreased by a factor of 10 when the validation set accuracy stopped increasing. The weights of the filters in the CNN were initialised by random sampling from a Gaussian distribution with zero mean and 10^{-2} standard deviation. Biases were initialized to zero.

In order to work with this network, first of all, we have resized all the images of all the databases to 224x224, to match the entrance dimension. Also, we have not taken the last layer of the CNN, because it performed the classification for the recognising which one of the 2622 people each images is of, and what we wanted to do was to take the 4096 dimension feature vector that was in this previous layer.

5. CHAPTER

Classification

In this project the objective is to classify the emotions of the faces in the pictures, and in order to do this classification, Support Vector Machines (SVMs) are used.

5.1 Support Vector Machines (SVMs)

Support Vector Machines [Cortes and Vapnik, 1995] are supervised learning algorithms used for classification and regression analysis. Given a set of examples, each one marked belonging to one of two categories, a SVM is a model that works as a non-probabilistic binary linear classifier.

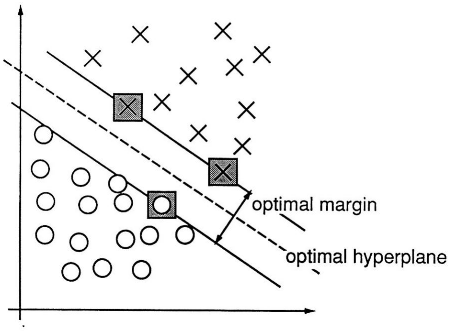

Figure 5.1: Separable problem in 2 dimensional space [Cortes and Vapnik, 1995].

It is based on the idea of finding a hyperplane that best divides the dataset into two classes. The data points that are nearest to the hyperplane are called support vectors, and this are the critical elements of a dataset, because if they are removed, they would alter the position of the dividing hyperplane.

The distance between the hyperplane and the nearest data point from either set is called the margin, and the goal is to choose a hyperplane with the widest possible margin between the hyperplane and any point within the training set.

SVM can also perform non-linear classification, using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

6. CHAPTER

Databases

In this project three databases that contain images of facial expression are used. Each of these databases contained at least the six basic emotions as refered by the psychologists Ekman and V.Friesen [Ekman and Friesen, 1971]: Anger, disgust, fear, happiness, sadness and suprise, and all of the images were taken in a controlled environment (a laboratory), with the subjects looking at the camera.

Some of the databases contain the FACS coding system [Pantic and Bartlett, 2007]. This is a widely used method for manual labeling of facial actions: it defines 44 different action units (AUs, which are the smallest visually discernable facial movmements) and sets rules for recognition of AU's temporal segment (onset, apex and offset) in a face video. It provides an objective and crompehensive language for facial expresssions descriptions, and it allows discovery of new patterns related to emotional states.

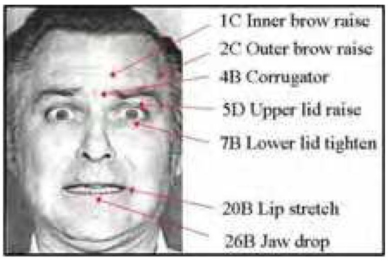

Figure 6.1: Example of FACS code for fear [Pantic and Bartlett, 2007].

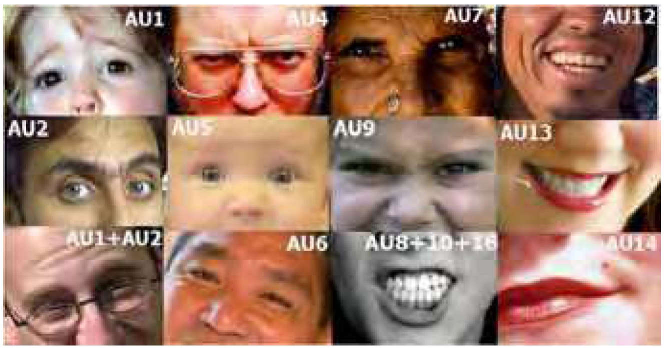

Figure 6.2: Example of facial action units (AUs) [Pantic and Bartlett, 2007].

6.1 The Child Affective Facial Expression (CAFE) set

This database [LoBue, 2014] is a set of photographs of 2-to 8-year old children with a mean M=5.3 years and with rho r=2.7-8.7 years. This databaset that belongs to the Department of Psychology at Rutgers University Newark in New Jersey contains children posing with one of the basic 6 emotions according to Eckman. The set is composed of a total of photos of 154 models: 90 female models and 64 male models (27 African American, 16 Asian, 77 Caucasian/European, 23 Latin and 11 South Asian.

Not all the children contain images of all the emotions, and the emotions are of two kinds: one is with the mouth open and the other one is with the mouth closed. Each of the models can have photos expressing emotions with the mouth open or closed, or both.

All of the photos of this dataset contain the image centered in the child's face, with their chin approximately 1/6 from the bottom of the image, and all of the images contain FACS codes because all the photos that did not include any were deleted from the original database. In order to make the images have these codes, the photographer made the children copy the faces and tell them which emotions to show in the face, and if any elements were missing, the photographer encouraged the children to revise their facial expressions [LoBue and Thrasher, 2014]. This photographer was trained in SPAFF (The Specific Affect coding system) [Gottman et al., 1996] which is a system that includes procedures for recognizing facial muscle movements associated with 17 codable emotional states in real time, and also incorporates the FACS coding system of Ekman and Colleges [Ekman, 1992]. This database originally was composed by two subsets: subset A which contains the stereotypical exemplars of the various facial expressions, the "basic" emotions we explained previously; and subset B contains expression that vary around the "basic" expression, but minimizing potential ceiling and floor effects. However, this division is not used in our experiments. [1]

6.2 The Japanese female facial expression (JAFFE) database

This database [Lyons et al., 1998] is a data set that contains 217 photos of 10 Japanese female models posing expressions of happiness, sadness, fear, anger, surprise, disgust and neutrality. All the expression were posed without any instructions and they were not compared to any standards for emotional facial expressions linke FACS.

Each of the pictures was took while looking through a semi-reflective plastic sheet towards the camera. The hair was tied away from the face in order to show all the expressive zones of the face, and Tungsten lights were positioned to create even illumination on the face. A box enclosed the region between the camera and the plastic sheet to reduce the back-reflection.

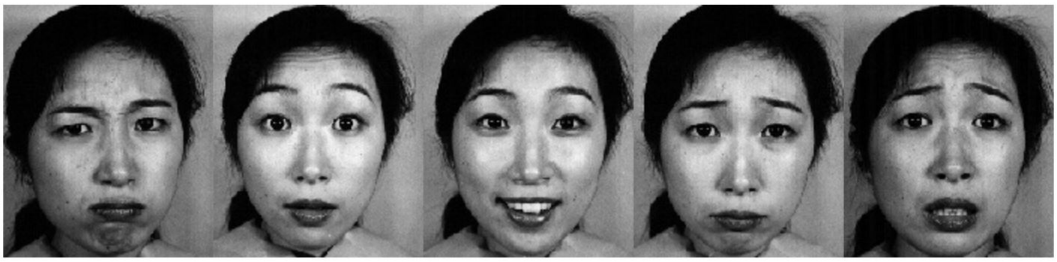

Figure 6.3: Example of JAFFE images [Lyons et al., 1998].

6.3 The Extended Cohn-Kanade Dataset (CK+)

The CK+ database [Lucey et al., 2010] is the second version of a database called Cohn-Kahade (CK). This database was released in 2000 in order to promote research into automatically detecting individual facial expressions. Originally the CK database had 486 sequences from 97 models. Each sequenced from a neutral expression to the peak expression, and each of the peak expressions was FACS coded and has an emotion label. However, in the CK+ database, the images are both posed and non-posed (spontaneous)

[1] An example of this images cannot be shown because in order to get access to the database an special permission is needed

expression. For the posed expressions, there are 22% more sequences and there are 27% more models, nevertheless as in the first version, each sequence is fully FACS coded and emotion labels have been included.

There is also a new launch planned for future release that contains all the original data collection of Cohn-Kanade synchronized frontal and 30-degree from frontal video.

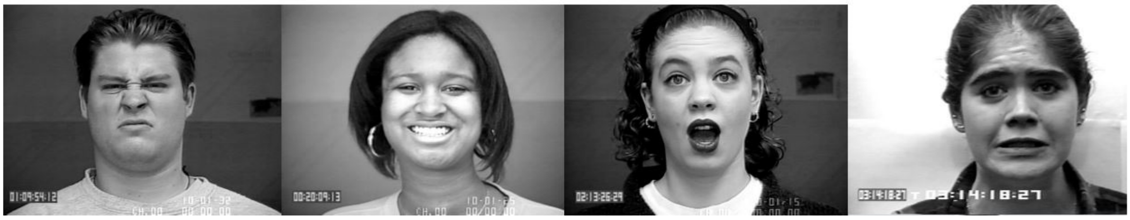

Figure 6.4: Example of CK+ database images [Lucey et al., 2010].

7. CHAPTER

Experiments and conclusions

7.1 Pre-processing of Data: face alignment

Some of the databases in the experiments were not aligned, so, in this cases, all the images were aligned. This is a procedure used to address the problem of the variation of the images in rotation, brightness and size even for images of the same subject. In order to solve this problem the face region is aligned with rotation normalization with the horizon and a centre point. To perform this alignment two information are needed, the facial image and the centre of both eyes. There are a lot of methods able to find the eyes and the others facial points with high precision, like a CUDA version of DRMF (Discriminative Response Map Fitting) developed by Cheng et al [Asthana et al., 2013].

In our case, the method we used to do so, was to first detect the face in the image, and crop this region. Detect the region of both eyes and divide this region into two to detect an eye on each of the regions. We chose the maximun area that is detected as an eye, and if it overlaped with one of the previous subregions, we accepted it that as the eye. If it was not detected, the threshold of the image was changed and the process was repeated. Then, the nose was detected. After this, in all the images a geometric transformation was done so that it fitted the previously set points of the eyes and nose center point [Nguyen et al., 2014].

Once this was done, in all of them, also in the ones that were not preprocessed, image cropping was done, which is a procedure used to increase the accuracy of the classifier discriminating between background and foreground. Some images have a lot of background information that is not important to the expression classification procedure. After the cropping all image parts that do not have expression specific information are removed, and facial parts that do not contribute for the expression are also removed. The region of interest was defined based on a ratio of the inter-eyes distance. Also all the images were resized to a size of 224x224, which is the needed enter size for the NNs.

For all of the images the following experiments weren done:

- HOG feature extraction varying the size of the HOG cell (that we call CellSize) between 6 and 25, and varying the number of orientation histogram bins (that we call NumBins) between 6 and 25.
- LBP feature extraction varying the size of LBP number of neighbors (NumNeighbors), which is selected from a circularly symmetric pattern around each pixel, between 2 and 32, and varying the number of the radius of circular patter to select neighbors (Radius) between 2 and 7. There are three versions of the LBP feature extraction:
  - Extract the LBP from the original images.
  - Extract the LBP feature vector from the original image, and concatenate it to the LBP of the image with half the size of the original one.
  - Extract the LBP feature vector from the original image, concatenate it to the LBP of the image with half the size of the original one and then concatenate it to the LBP of the image with quart the size of the original one.
- LBP Pyramid feature extraction, varying the number of neighbours between 2 and 32, and the radius between 2 and 7.
- LBP with fixed nNeighbours=8 and radius=1, varying the cellSize between 10 and 25, and trying with the original image, concatenating the LBP of the original image and half size image and concatenating the LBP of the original image, half size image and quart size image.
- BSIF feature extraction, with all the filters included in the original code and varying the bits, the code string correspondent to the binarized responses of different filters, between 5 and 12. There are three variants of the BSIF feature extraction:
  - Extraction in the original image.

  - Concatenation of the extraction of the original image and the extraction of the half-size image.
  - Concatenation of the extraction of the original image, the extraction of the half-size image and the extraction of the quart-size image.

- Hybrid LBP HOG BSIF feature extraction. There are two variants:
  - Extraction in the original image of the LBP, HOG and BSIF and concatenate the results.
  - Extraction in the image, half-size image and quart-size image of LBP, HOG and BSIF and concatenate the result.

- VGG-Face feature extraction.

- VGG-F feature extraction

In order to choose the range of the varyation of the experiments, at first, we choose some reasonable intervals, and if we had the suspicion that expanding the table was going to give better results, because values tended to be smaller in that direction, we extended the table.

7.2 CAFE experiments and results

This database is a database that was not intended to use for image analysis, its primary purpose was for psichology. So, there is not previous work with this database to compare it to.

7.2.1 CAFE original. CV by emotion.

At first we decided to work with the database as it is, with no alignement and in order to obtain the Mean Absolute Error (MAE) of each of the excutions a Cross Validation (CV) was done by emotion.

CellSize/ NumBins	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6x6	0.421	0.429	0.43	0.422	0.422	0.425	0.425	0.421	0.421	0.423	0.424	0.425	0.426	0.427	0.428	0.428	0.429	0.43	0.456	0.457
7x7	0.432	0.439	0.434	0.428	0.414	0.426	0.428	0.42	0.413	0.423	0.425	0.419	0.42	0.42	0.426	0.416	0.417	0.437	0.423	0.431
8x8	0.421	0.426	0.417	0.418	0.412	0.412	0.419	0.409	0.409	0.413	0.413	0.414	0.414	0.415	0.42	0.413	0.416	0.422	0.423	0.426
9x9	0.414	0.409	0.414	0.417	0.406	0.409	0.414	0.413	0.403	0.411	0.409	0.414	0.403	0.41	0.411	0.405	0.405	0.408	0.408	0.42
10x10	0.404	0.407	0.41	0.402	0.401	0.393	0.399	0.391	0.399	0.4	0.4	0.391	0.399	0.401	0.404	0.404	0.4	0.406	0.4	0.414
11x11	0.409	0.399	0.405	0.395	0.401	0.39	0.401	0.398	0.393	0.403	0.398	0.402	0.393	0.403	0.396	0.399	0.392	0.404	0.398	0.399
12x12	0.39	0.389	0.39	0.391	0.382	0.387	0.397	0.393	0.387	0.392	0.392	0.389	0.384	0.388	0.393	0.391	0.387	0.396	0.388	0.398
13x13	0.399	0.388	0.395	0.393	0.39	0.391	0.399	0.388	0.387	0.389	0.393	0.388	0.382	0.388	0.394	0.388	0.384	0.398	0.388	0.391
14x14	0.407	0.4	0.395	0.394	0.39	0.382	0.389	0.389	0.393	0.393	0.392	0.397	0.391	0.4	0.394	0.398	0.395	0.4	0.394	0.399
15x15	0.393	0.387	0.392	0.381	0.376	0.379	0.382	0.377	0.379	0.383	0.375	0.377	0.37	0.372	0.375	0.372	0.377	0.379	0.369	0.377
16x16	0.384	0.381	0.382	0.382	0.376	0.373	0.377	0.372	0.382	0.373	0.378	0.374	0.376	0.373	0.38	0.38	0.374	0.377	0.384	0.382
17x17	0.381	0.382	0.382	0.385	0.381	0.382	0.388	0.389	0.377	0.382	0.382	0.384	0.376	0.383	0.384	0.378	0.369	0.377	0.377	0.381
18x18	0.377	0.374	0.374	0.387	0.38	0.378	0.387	0.379	0.382	0.382	0.382	0.384	0.379	0.382	0.382	0.374	0.376	0.372	0.376	0.379
19x19	0.365	0.367	0.37	0.357	0.359	0.365	0.367	0.363	0.359	0.36	0.37	0.357	0.359	0.355	0.362	0.353	0.348	0.356	0.36	0.353
20x20	0.377	0.38	0.378	0.375	0.378	0.377	0.382	0.384	0.377	0.378	0.376	0.378	0.366	0.374	0.375	0.38	0.367	0.372	0.375	0.377
21x21	0.367	0.378	0.366	0.37	0.363	0.362	0.372	0.37	0.366	0.362	0.36	0.367	0.359	0.363	0.362	0.364	0.358	0.357	0.366	0.359
22x22	0.375	0.377	0.375	0.367	0.364	0.363	0.367	0.366	0.365	0.365	0.372	0.364	0.364	0.367	0.366	0.367	0.367	0.367	0.37	0.37
23x23	0.393	0.387	0.376	0.375	0.373	0.367	0.358	0.367	0.364	0.367	0.371	0.369	0.366	0.371	0.372	0.372	0.366	0.368	0.372	0.37
24x24	0.379	0.369	0.375	0.373	0.374	0.379	0.378	0.381	0.378	0.375	0.372	0.378	0.366	0.37	0.37	0.375	0.369	0.372	0.377	0.374
25x25	0.371	0.394	0.38	0.377	0.369	0.362	0.363	0.375	0.358	0.363	0.362	0.363	0.36	0.362	0.364	0.362	0.36	0.361	0.362	0.362

Table 7.1: MAEs aplying HOG in the CAFE db. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.539	0.566	0.586	0.558	0.575	0.554
4	0.486	0.488	0.526	0.525	0.531	0.534
8	0.436	0.439	0.46	0.46	0.481	0.482
16	0.426	0.416	0.432	0.439	0.447	0.44
32	0.439	0.414	0.438	0.446	0.451	0.451

Table 7.2: MAEs applying LBP in the CAFE db with images of 224x224 pixels. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.511	0.554	0.553	0.536	0.559	0.567
4	0.458	0.483	0.485	0.487	0.507	0.501
8	0.403	0.409	0.44	0.434	0.454	0.455
16	0.414	0.415	0.414	0.409	0.429	0.423
32	0.414	0.425	0.426	0.425	0.421	0.425

Table 7.3: MAEs applying LBP in the CAFE db with concatenating the same image in a resolution of 224x224 and 112x112 pixels. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.509	0.535	0.54	0.529	0.56	0.568
4	0.45	0.479	0.476	0.486	0.495	0.504
8	0.401	0.407	0.439	0.432	0.449	0.446
16	0.414	0.405	0.408	0.394	0.419	0.413
32	0.405	0.407	0.42	0.424	0.425	0.421

Table 7.4: MAEs applying LBP in the CAFE db with concatenating the same image in a resolution of 224x224, 112x112 and 56x56 pixels. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.68	0.663	0.665	0.7	0.736	0.778
4	0.596	0.589	0.612	0.64	0.655	0.752
8	0.585	0.581	0.542	0.59	0.612	0.695
16	0.554	0.549	0.544	0.576	0.63	0.7
32	0.563	0.534	0.534	0.597	0.67	0.723

Table 7.5: MAEs applying LBP Pyramid in the CAFE db. CV by emotion.

Concat/ CellSize	10x10	11x11	12x12	13x13	14x14	15x15	16x16	17x17	18x18	19x19	20x20	21x21	22x22	23x2	24x24	25x25
224	0.472	0.453	0.454	0.456	0.467	0.454	0.463	0.466	0.465	0.46	0.47	0.462	0.501	0.487	0.475	0.483
224 112	0.432	0.434	0.425	0.427	0.432	0.43	0.434	0.434	0.455	0.442	0.449	0.443	0.462	0.461	0.467	0.456
224 112 56	0.423	0.425	0.426	0.424	0.422	0.417	0.427	0.426	0.436	0.436	0.443	0.44	0.449	0.456	0.451	0.454

Table 7.6: MAEs aplying LBP with fixed nNeighbours=8 and radius=1 in the CAFE db. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.761	0.718	0.728	0.705				
5x5	0.782	0.746	0.719	0.69	0.666	0.659	0.619	0.608
7x7	0.761	0.714	0.704	0.679	0.666	0.638	0.598	0.558
9x9	0.76	0.739	0.689	0.662	0.613	0.588	0.561	0.512
11x11	0.759	0.753	0.697	0.67	0.61	0.591	0.522	0.492
13x13	0.764	0.758	0.695	0.674	0.605	0.554	0.531	0.495
15x15	0.79	0.774	0.697	0.664	0.603	0.554	0.512	0.48
17x17	0.78	0.763	0.723	0.659	0.62	0.546	0.496	0.468

Table 7.7: MAEs applying BSIF in the CAFE db with 224x224 images. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.743	0.707	0.705	0.673				
5x5	0.767	0.742	0.684	0.666	0.612	0.592	0.571	0.555
7x7	0.759	0.727	0.684	0.624	0.574	0.554	0.525	0.516
9x9	0.747	0.717	0.672	0.597	0.557	0.523	0.5	0.467
11x11	0.739	0.754	0.659	0.614	0.54	0.492	0.45	0.436
13x13	0.737	0.734	0.634	0.569	0.507	0.487	0.468	0.451
15x15	0.763	0.73	0.635	0.562	0.506	0.479	0.448	0.439
17x17	0.767	0.703	0.643	0.579	0.516	0.484	0.445	0.429

Table 7.8: MAEs applying BSIF in the CAFE db with concatenating 224x224 and 112x112 images. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.73	0.714	0.684	0.643				
5x5	0.769	0.702	0.664	0.617	0.575	0.555	0.549	0.542
7x7	0.756	0.703	0.628	0.569	0.552	0.512	0.479	0.489
9x9	0.747	0.725	0.641	0.566	0.502	0.481	0.465	0.439
11x11	0.743	0.682	0.61	0.529	0.492	0.456	0.447	0.435
13x13	0.729	0.706	0.603	0.513	0.491	0.474	0.445	0.449
15x15	0.716	0.706	0.617	0.524	0.497	0.454	0.452	0.436
17x17	0.73	0.695	0.607	0.541	0.502	0.473	0.446	0.448

Table 7.9: MAEs applying BSIF in the CAFE db with concatenating 224x224, 112x112 and 56x56 images. CV by emotion.

- **MAE of Hybrid LBP HOG BSIF:** 0.339
- **MAE of Hybrid concatenated LBP HOG BSIF:** 0.341
- **MAE of VGG-Face:** 0.428
- **MAE of VGG-F:** 0.552

7.2.2 CAFE Aligned CV by emotion

After obtaining the results of the CAFE database without the images being aligned, since they were not good, we though it might be a good idea to align them, which looking at the results turned out to be true. The CV we still did it by emotion.

CellSize/ NumBins	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6x6	0.278	0.278	0.273	0.28	0.275	0.274	0.276	0.274	0.273	0.274	0.274	0.273	0.273	0.275	0.276	0.278	0.275	0.278	0.274	0.279
7x7	0.273	0.273	0.269	0.264	0.268	0.262	0.26	0.264	0.261	0.265	0.261	0.265	0.263	0.262	0.266	0.269	0.268	0.269	0.269	0.27
8x8	0.267	0.263	0.264	0.265	0.269	0.265	0.269	0.268	0.262	0.261	0.264	0.257	0.262	0.26	0.263	0.259	0.258	0.261	0.258	0.262
9x9	0.26	0.263	0.262	0.261	0.269	0.259	0.257	0.257	0.26	0.258	0.255	0.26	0.258	0.258	0.257	0.261	0.261	0.259	0.259	0.259
10x10	0.262	0.256	0.25	0.252	0.248	0.246	0.247	0.251	0.244	0.249	0.247	0.246	0.247	0.242	0.243	0.243	0.245	0.242	0.247	0.246
11x11	0.264	0.256	0.256	0.261	0.255	0.25	0.256	0.252	0.251	0.248	0.252	0.253	0.249	0.25	0.252	0.252	0.251	0.249	0.245	0.248
12x12	0.25	0.252	0.255	0.257	0.248	0.256	0.251	0.249	0.246	0.243	0.243	0.244	0.247	0.247	0.248	0.248	0.245	0.248	0.248	0.247
13x13	0.258	0.264	0.25	0.252	0.257	0.251	0.252	0.254	0.253	0.252	0.254	0.254	0.252	0.248	0.256	0.254	0.254	0.252	0.253	0.252
14x14	0.244	0.249	0.24	0.244	0.248	0.247	0.247	0.243	0.248	0.248	0.244	0.247	0.246	0.243	0.244	0.245	0.246	0.243	0.245	0.241
15x15	0.252	0.249	0.244	0.239	0.245	0.244	0.245	0.241	0.241	0.243	0.247	0.249	0.243	0.239	0.243	0.245	0.243	0.244	0.248	0.24
16x16	0.256	0.252	0.249	0.243	0.243	0.242	0.242	0.244	0.24	0.237	0.237	0.232	0.235	0.235	0.239	0.232	0.234	0.233	0.237	0.233
17x17	0.252	0.252	0.252	0.246	0.242	0.245	0.237	0.247	0.242	0.243	0.24	0.234	0.235	0.237	0.238	0.232	0.235	0.234	0.231	0.229
18x18	0.262	0.259	0.258	0.252	0.257	0.25	0.25	0.244	0.249	0.248	0.248	0.243	0.247	0.25	0.248	0.249	0.246	0.249	0.247	0.248
19x19	0.258	0.253	0.254	0.249	0.247	0.247	0.243	0.25	0.252	0.248	0.241	0.25	0.245	0.241	0.24	0.243	0.243	0.243	0.24	0.232
20x20	0.247	0.247	0.237	0.242	0.242	0.242	0.242	0.242	0.239	0.239	0.237	0.237	0.237	0.234	0.236	0.237	0.233	0.237	0.24	0.232
21x21	0.27	0.263	0.273	0.263	0.257	0.255	0.249	0.25	0.246	0.249	0.243	0.242	0.243	0.244	0.244	0.24	0.239	0.248	0.247	0.245
22x22	0.253	0.247	0.252	0.246	0.243	0.243	0.244	0.242	0.242	0.242	0.242	0.24	0.247	0.238	0.244	0.247	0.247	0.245	0.25	0.247
23x23	0.259	0.251	0.256	0.25	0.244	0.247	0.242	0.242	0.232	0.233	0.231	0.234	0.237	0.237	0.237	0.243	0.242	0.242	0.239	0.234
24x24	0.267	0.263	0.267	0.251	0.248	0.247	0.249	0.247	0.242	0.242	0.24	0.242	0.24	0.237	0.239	0.237	0.24	0.239	0.236	0.238
25x25	0.268	0.273	0.265	0.27	0.263	0.258	0.262	0.261	0.254	0.253	0.253	0.254	0.256	0.249	0.248	0.245	0.247	0.242	0.244	0.247

Table 7.10: MAEs aplying HOG in the aligned CAFE db. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.398	0.356	0.339	0.347	0.366	0.382
4	0.341	0.286	0.292	0.281	0.296	0.309
8	0.329	0.297	0.295	0.297	0.28	0.278
16	0.315	0.303	0.274	0.294	0.279	0.274
32	0.329	0.303	0.284	0.289	0.286	0.282

Table 7.11: MAEs applying LBP in the aligned CAFE db with images of 224x224 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.346	0.343	0.32	0.331	0.336	0.347
4	0.29	0.273	0.274	0.255	0.267	0.279
8	0.304	0.275	0.257	0.267	0.254	0.259
16	0.29	0.266	0.25	0.264	0.255	0.248
32	0.297	0.272	0.257	0.264	0.279	0.274

Table 7.12: MAEs applying LBP in the aligned CAFE db with concatenating the same image in a resolution of 224x224 and 112x112 pixels. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.341	0.315	0.306	0.321	0.336	0.351
4	0.285	0.257	0.26	0.255	0.263	0.271
8	0.294	0.271	0.251	0.255	0.249	0.248
16	0.284	0.26	0.243	0.256	0.247	0.249
32	0.291	0.257	0.248	0.254	0.267	0.257

Table 7.13: MAEs applying LBP in the aligned CAFE db with concatenating the same image in a resolution of 224x224, 112x112 and 56x56 pixels. CV by emotion.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.48	0.471	0.473	0.499	0.519	0.571
4	0.356	0.366	0.388	0.414	0.451	0.501
8	0.324	0.325	0.36	0.364	0.394	0.463
16	0.315	0.3	0.331	0.341	0.379	0.451
32	0.33	0.31	0.329	0.347	0.4	0.491

Table 7.14: MAEs applying LBP Pyramid in the aligned CAFE db.CV by emotion.

Concat/ CellSize	10x10	11x11	12x12	13x13	14x14	15x15	16x16	17x17	18x18	19x19	20x20	21x21	22x22	23x23	24x24	25x25
224	0.365	0.36	0.351	0.359	0.362	0.35	0.367	0.369	0.361	0.36	0.382	0.361	0.38	0.381	0.386	0.396
224 112	0.315	0.321	0.319	0.312	0.322	0.323	0.324	0.325	0.328	0.327	0.351	0.341	0.352	0.352	0.345	0.352
224 112 56	0.299	0.303	0.305	0.309	0.305	0.317	0.306	0.315	0.315	0.325	0.341	0.339	0.334	0.352	0.336	0.341

Table 7.15: MAEs aplying LBP with fixed nNeighbours=8 and radius=1 in the aligned CAFE db. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.682	0.661	0.676	0.653				
5x5	0.603	0.576	0.596	0.589	0.538	0.533	0.519	0.534
7x7	0.586	0.546	0.544	0.502	0.503	0.441	0.447	0.43
9x9	0.61	0.554	0.512	0.483	0.452	0.399	0.394	0.367
11x11	0.61	0.574	0.534	0.458	0.409	0.38	0.359	0.354
13x13	0.613	0.581	0.532	0.489	0.419	0.377	0.361	0.336
15x15	0.612	0.578	0.559	0.467	0.413	0.382	0.353	0.332
17x17	0.632	0.58	0.522	0.452	0.413	0.381	0.339	0.341

Table 7.16: MAEs applying BSIF in the aligned CAFE db with 224x224 images. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.62	0.615	0.608	0.55				
5x5	0.564	0.526	0.488	0.455	0.432	0.429	0.409	0.425
7x7	0.559	0.515	0.476	0.404	0.417	0.354	0.379	0.356
9x9	0.552	0.535	0.441	0.403	0.373	0.334	0.332	0.321
11x11	0.572	0.55	0.456	0.381	0.334	0.321	0.31	0.302
13x13	0.6	0.566	0.445	0.373	0.371	0.322	0.308	0.297
15x15	0.586	0.557	0.468	0.362	0.325	0.317	0.295	0.277
17x17	0.574	0.568	0.45	0.361	0.323	0.315	0.297	0.289

Table 7.17: MAEs applying BSIF in the aligned CAFE db with concatenating 224x224 and 112x112 images. CV by emotion.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.586	0.581	0.536	0.498				
5x5	0.567	0.476	0.446	0.428	0.382	0.374	0.377	0.378
7x7	0.561	0.486	0.398	0.355	0.369	0.335	0.344	0.315
9x9	0.537	0.466	0.39	0.343	0.316	0.296	0.307	0.291
11x11	0.547	0.508	0.381	0.34	0.304	0.284	0.273	0.28
13x13	0.593	0.506	0.399	0.347	0.321	0.297	0.289	0.272
15x15	0.571	0.485	0.41	0.344	0.321	0.297	0.282	0.27
17x17	0.56	0.505	0.394	0.347	0.318	0.317	0.284	0.282

Table 7.18: MAEs applying BSIF in the aligned CAFE db with concatenating 224x224, 112x112 and 56x56 images. CV by emotion.

- **MAE of Hybrid LBP HOG BSIF:** 0.217
- **MAE of Hybrid concatenated LBP HOG BSIF:** 0.218
- **MAE of VGG-Face:** 0.385
- **MAE of VGG-F:** 0.295

7.2.3 CAFE Aligned. CV by subject.

We checked some up to date state of the art in facial expression, and realised that people were doing CV by subject instead of CV by emotion, so we proceed to do it that way in the aligned database, which gave better results than the original database.

CellSize/ NumBins	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6x6	0.266	0.27	0.265	0.272	0.266	0.266	0.266	0.266	0.267	0.266	0.266	0.265	0.266	0.262	0.264	0.263	0.265	0.266	0.265	0.271
7x7	0.252	0.255	0.256	0.259	0.262	0.256	0.256	0.253	0.253	0.254	0.254	0.256	0.258	0.256	0.258	0.254	0.26	0.26	0.26	0.259
8x8	0.248	0.254	0.25	0.249	0.25	0.245	0.252	0.248	0.245	0.249	0.248	0.248	0.248	0.247	0.252	0.248	0.254	0.249	0.25	0.252
9x9	0.257	0.254	0.255	0.252	0.249	0.254	0.255	0.25	0.25	0.254	0.25	0.253	0.251	0.251	0.252	0.253	0.254	0.251	0.252	0.254
10x10	0.259	0.258	0.251	0.252	0.255	0.249	0.25	0.254	0.25	0.249	0.25	0.245	0.248	0.247	0.25	0.25	0.247	0.249	0.249	0.252
11x11	0.247	0.253	0.252	0.253	0.249	0.25	0.249	0.248	0.244	0.246	0.246	0.246	0.241	0.243	0.243	0.245	0.242	0.245	0.244	0.249
12x12	0.259	0.251	0.254	0.252	0.254	0.25	0.248	0.246	0.245	0.247	0.245	0.248	0.251	0.249	0.247	0.249	0.249	0.243	0.247	0.243
13x13	0.248	0.248	0.252	0.254	0.251	0.249	0.248	0.247	0.249	0.243	0.248	0.249	0.246	0.246	0.247	0.244	0.249	0.244	0.249	0.245
14x14	0.252	0.24	0.243	0.243	0.243	0.246	0.243	0.242	0.24	0.24	0.243	0.237	0.244	0.237	0.237	0.237	0.24	0.238	0.239	0.234
15x15	0.248	0.244	0.246	0.25	0.245	0.245	0.243	0.238	0.242	0.243	0.24	0.243	0.243	0.242	0.24	0.245	0.243	0.242	0.24	0.243
16x16	0.258	0.257	0.249	0.244	0.24	0.247	0.243	0.239	0.239	0.233	0.236	0.232	0.233	0.231	0.227	0.229	0.226	0.229	0.228	0.226
17x17	0.254	0.257	0.238	0.237	0.237	0.235	0.237	0.235	0.232	0.23	0.229	0.229	0.229	0.231	0.222	0.226	0.223	0.227	0.226	0.221
18x18	0.258	0.258	0.258	0.255	0.248	0.25	0.25	0.248	0.245	0.242	0.245	0.244	0.243	0.243	0.24	0.24	0.243	0.24	0.238	0.239
19x19	0.26	0.254	0.251	0.25	0.249	0.249	0.241	0.244	0.244	0.243	0.24	0.235	0.235	0.235	0.237	0.237	0.237	0.231	0.234	0.233
20x20	0.25	0.245	0.235	0.239	0.243	0.237	0.235	0.239	0.233	0.232	0.231	0.233	0.232	0.229	0.232	0.234	0.231	0.228	0.228	0.226
21x21	0.263	0.252	0.26	0.26	0.243	0.244	0.24	0.241	0.241	0.237	0.237	0.241	0.24	0.235	0.236	0.235	0.238	0.238	0.237	0.236
22x22	0.256	0.25	0.248	0.255	0.247	0.245	0.246	0.237	0.237	0.237	0.235	0.235	0.233	0.233	0.229	0.232	0.239	0.236	0.235	0.234
23x23	0.266	0.249	0.255	0.244	0.242	0.242	0.243	0.233	0.232	0.232	0.233	0.233	0.235	0.233	0.233	0.237	0.237	0.237	0.236	0.235
24x24	0.261	0.25	0.245	0.243	0.242	0.236	0.24	0.235	0.234	0.239	0.234	0.231	0.3	0.231	0.226	0.236	0.231	0.231	0.226	0.229
25x25	0.268	0.26	0.258	0.256	0.244	0.248	0.245	0.245	0.244	0.244	0.237	0.237	0.237	0.234	0.233	0.232	0.233	0.235	0.232	0.232

Table 7.19: MAEs aplying HOG in the aligned CAFE db. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.402	0.357	0.35	0.348	0.35	0.373
4	0.347	0.311	0.29	0.288	0.307	0.302
8	0.306	0.295	0.29	0.297	0.294	0.281
16	0.323	0.307	0.278	0.271	0.266	0.271
32	0.343	0.3	0.286	0.277	0.281	0.272

Table 7.20: MAEs applying LBP in the aligned CAFE db with images of 224x224 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.353	0.337	0.326	0.334	0.336	0.364
4	0.292	0.271	0.264	0.258	0.278	0.283
8	0.289	0.258	0.255	0.261	0.26	0.256
16	0.289	0.254	0.242	0.243	0.251	0.254
32	0.299	0.26	0.249	0.257	0.27	0.26

Table 7.21: MAEs applying LBP in the aligned CAFE db with concatenating the same image in a resolution of 224x224 and 112x112 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.339	0.321	0.323	0.334	0.334	0.357
4	0.283	0.268	0.26	0.254	0.277	0.275
8	0.282	0.254	0.254	0.254	0.254	0.254
16	0.277	0.249	0.237	0.239	0.238	0.244
32	0.292	0.261	0.238	0.251	0.263	0.254

Table 7.22: MAEs applying LBP in the aligned CAFE db with concatenating the same image in a resolution of 224x224, 112x112 and 56x56 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.472	0.472	0.49	0.508	0.531	0.599
4	0.384	0.378	0.389	0.431	0.455	0.518
8	0.327	0.324	0.374	0.386	0.42	0.473
16	0.319	0.321	0.33	0.351	0.398	0.472
32	0.324	0.311	0.324	0.342	0.411	0.489

Table 7.23: MAEs applying LBP Pyramid in the aligned CAFE db. CV by subject.

Concat/ CellSize	10x10	11x11	12x12	13x13	14x14	15x15	16x16	17x17	18x18	19x19	20x20	21x21	22x22	23x23	24x24	25x25
224	0.365	0.36	0.351	0.359	0.362	0.35	0.367	0.369	0.361	0.36	0.382	0.361	0.38	0.381	0.386	0.396
224 112	0.315	0.321	0.319	0.312	0.322	0.323	0.324	0.325	0.328	0.327	0.351	0.341	0.352	0.352	0.345	0.352
224 112 56	0.299	0.303	0.305	0.309	0.305	0.317	0.306	0.315	0.315	0.325	0.341	0.339	0.334	0.352	0.336	0.341

Table 7.24: MAEs aplying LBP with fixed nNeighbours=8 and radius=1 in the aligned CAFE db. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.674	0.653	0.667	0.656				
5x5	0.615	0.596	0.603	0.578	0.541	0.543	0.518	0.53
7x7	0.6	0.547	0.549	0.503	0.506	0.461	0.448	0.42
9x9	0.613	0.549	0.524	0.488	0.465	0.393	0.394	0.385
11x11	0.622	0.579	0.545	0.456	0.431	0.393	0.354	0.357
13x13	0.631	0.591	0.547	0.494	0.439	0.382	0.354	0.34
15x15	0.635	0.622	0.597	0.467	0.44	0.375	0.346	0.322
17x17	0.643	0.603	0.546	0.47	0.422	0.384	0.328	0.325

Table 7.25: MAEs applying BSIF in the aligned CAFE db with 224x224 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.622	0.626	0.602	0.569				
5x5	0.572	0.551	0.515	0.467	0.447	0.411	0.433	0.431
7x7	0.551	0.532	0.491	0.41	0.421	0.369	0.373	0.352
9x9	0.556	0.549	0.445	0.399	0.375	0.334	0.342	0.333
1x11	0.594	0.552	0.467	0.395	0.362	0.33	0.298	0.303
13x13	0.622	0.568	0.457	0.395	0.375	0.339	0.3	0.295
15x15	0.597	0.586	0.487	0.388	0.351	0.317	0.289	0.276
17x17	0.586	0.549	0.457	0.371	0.335	0.317	0.287	0.274

Table 7.26: MAEs applying BSIF in the aligned CAFE db with concatenating 224x224 and 112x112 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.586	0.563	0.551	0.502				
5x5	0.547	0.527	0.472	0.435	0.38	0.362	0.382	0.392
7x7	0.571	0.486	0.424	0.365	0.37	0.344	0.327	0.319
9x9	0.551	0.492	0.406	0.357	0.339	0.31	0.31	0.3
11x11	0.574	0.494	0.404	0.369	0.306	0.289	0.283	0.275
13x13	0.591	0.529	0.412	0.365	0.342	0.295	0.283	0.287
15x15	0.582	0.535	0.445	0.356	0.348	0.306	0.28	0.271
17x17	0.574	0.52	0.422	0.371	0.323	0.301	0.273	0.271

Table 7.27: MAEs applying BSIF in the aligned CAFE db with concatenating 224x224, 112x112 and 56x56 images. CV by subject.

- **MAE of Hybrid LBP HOG BSIF:** 0.203
- **MAE of Hybrid concatenated LBP HOG BSIF:** 0.206
- **MAE of VGG-Face:** 0.391
- **MAE of VGG-F:** 0.303

7.3 JAFFE experiments and results

In this database we proceed to do the experiments using CV by subject, as we did before, since it seemed to give better results, and other authors were doing so too.

CellSize/ NumBins	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6x6	0.521	0.507	0.512	0.498	0.498	0.493	0.502	0.498	0.493	0.498	0.493	0.502	0.498	0.493	0.493	0.493	0.493	0.493	0.502	0.488
7x7	0.507	0.512	0.516	0.526	0.507	0.526	0.502	0.512	0.521	0.521	0.516	0.512	0.526	0.521	0.516	0.516	0.512	0.521	0.507	0.516
8x8	0.521	0.516	0.526	0.535	0.53	0.521	0.516	0.521	0.521	0.535	0.53	0.53	0.512	0.516	0.516	0.516	0.521	0.526	0.516	0.521
9x9	0.507	0.507	0.493	0.502	0.484	0.493	0.507	0.488	0.479	0.502	0.493	0.498	0.484	0.502	0.498	0.493	0.488	0.498	0.493	0.493
10x10	0.493	0.488	0.493	0.484	0.46	0.479	0.479	0.484	0.488	0.493	0.488	0.493	0.488	0.498	0.488	0.488	0.498	0.512	0.498	0.502
11x11	0.502	0.507	0.502	0.498	0.493	0.502	0.498	0.507	0.507	0.516	0.521	0.521	0.516	0.526	0.526	0.526	0.53	0.53	0.53	0.521
12x12	0.46	0.47	0.479	0.465	0.465	0.488	0.484	0.479	0.493	0.502	0.498	0.498	0.502	0.521	0.516	0.516	0.526	0.521	0.526	0.526
13x13	0.46	0.451	0.441	0.465	0.474	0.47	0.474	0.474	0.474	0.488	0.484	0.498	0.493	0.498	0.498	0.502	0.507	0.498	0.507	0.512
14x14	0.46	0.451	0.479	0.474	0.484	0.488	0.484	0.502	0.493	0.512	0.502	0.507	0.507	0.502	0.507	0.507	0.507	0.512	0.512	0.512
15x15	0.422	0.437	0.441	0.451	0.455	0.46	0.474	0.47	0.47	0.474	0.474	0.47	0.47	0.474	0.474	0.474	0.474	0.484	0.474	0.479
16x16	0.441	0.422	0.437	0.446	0.432	0.451	0.446	0.455	0.455	0.46	0.465	0.46	0.46	0.455	0.46	0.479	0.46	0.47	0.47	0.474
17x17	0.484	0.446	0.437	0.455	0.437	0.451	0.474	0.46	0.47	0.465	0.479	0.479	0.474	0.474	0.474	0.484	0.479	0.498	0.493	0.488
18x18	0.432	0.441	0.432	0.432	0.441	0.427	0.427	0.427	0.432	0.432	0.422	0.446	0.437	0.432	0.437	0.446	0.441	0.446	0.432	0.446
19x19	0.451	0.427	0.441	0.437	0.437	0.413	0.427	0.432	0.432	0.422	0.432	0.441	0.437	0.437	0.446	0.446	0.446	0.446	0.451	0.441
20x20	0.46	0.441	0.446	0.451	0.465	0.465	0.465	0.465	0.46	0.465	0.455	0.469	0.474	0.465	0.479	0.465	0.469	0.474	0.474	0.469
21x21	0.451	0.441	0.446	0.455	0.455	0.465	0.465	0.446	0.451	0.455	0.465	0.455	0.474	0.465	0.469	0.446	0.455	0.46	0.451	0.446
22x22	0.432	0.437	0.427	0.437	0.455	0.446	0.441	0.446	0.451	0.441	0.427	0.437	0.423	0.455	0.427	0.437	0.437	0.441	0.437	0.432
23x23	0.423	0.408	0.404	0.408	0.385	0.366	0.408	0.385	0.394	0.408	0.404	0.404	0.413	0.408	0.418	0.427	0.423	0.437	0.427	0.441
24x24	0.474	0.455	0.474	0.46	0.455	0.432	0.446	0.465	0.446	0.451	0.465	0.469	0.46	0.465	0.474	0.46	0.451	0.46	0.455	0.455
25x25	0.432	0.404	0.408	0.385	0.423	0.394	0.39	0.399	0.413	0.413	0.404	0.413	0.39	0.418	0.446	0.427	0.423	0.432	0.441	0.432

Table 7.28: MAEs aplying HOG in the JAFFE db. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.671	0.568	0.502	0.516	0.526	0.54
4	0.573	0.578	0.54	0.516	0.563	0.554
8	0.568	0.535	0.53	0.554	0.507	0.559
16	0.563	0.535	0.573	0.53	0.521	0.521
32	0.582	0.559	0.54	0.507	0.516	0.516

Table 7.29: MAEs applying LBP in the JAFFE db with images of 224x224 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.648	0.563	0.54	0.554	0.516	0.568
4	0.502	0.587	0.526	0.488	0.512	0.484
8	0.563	0.559	0.54	0.512	0.512	0.545
16	0.582	0.559	0.573	0.521	0.54	0.516
32	0.568	0.559	0.549	0.526	0.507	0.521

Table 7.30: MAEs applying LBP in the JAFFE db with concatenating the same image in a resolution of 224x224 and 112x112 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.629	0.563	0.53	0.568	0.516	0.568
4	0.502	0.549	0.516	0.484	0.488	0.484
8	0.578	0.545	0.54	0.507	0.507	0.535
16	0.563	0.545	0.563	0.535	0.526	0.502
32	0.568	0.563	0.559	0.549	0.507	0.507

Table 7.31: MAEs applying LBP in the JAFFE db with concatenating the same image in a resolution of 224x224, 112x112 and 56x56 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.676	0.653	0.624	0.653	0.686	0.761
4	0.62	0.653	0.624	0.573	0.686	0.723
8	0.61	0.559	0.569	0.592	0.671	0.742
16	0.61	0.578	0.568	0.643	0.686	0.761
32	0.639	0.578	0.582	0.639	0.709	0.751

Table 7.32: MAEs applying LBP Pyramid in the JAFFE db. CV by subject.

Concat/ CellSize	10x10	11x11	12x12	13x13	14x14	15x15	16x16	17x17	18x18	19x19	20x20	21x21	22x22	23x23	24x24	25x25
224	0.549	0.592	0.577	0.596	0.601	0.535	0.582	0.587	0.592	0.596	0.61	0.61	0.582	0.563	0.615	0.592
224 112	0.54	0.577	0.573	0.577	0.587	0.516	0.592	0.577	0.502	0.563	0.568	0.563	0.568	0.549	0.582	0.596
224 112 56	0.535	0.563	0.554	0.568	0.582	0.516	0.587	0.568	0.512	0.545	0.545	0.559	0.568	0.554	0.577	0.573

Table 7.33: MAEs aplying LBP with fixed nNeighbours=8 and radius=1 in the JAFFE db. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.803	0.831	0.826	0.812				
5x5	0.807	0.779	0.746	0.728	0.728	0.732	0.751	0.728
7x7	0.793	0.812	0.775	0.789	0.751	0.718	0.77	0.765
9x9	0.793	0.807	0.784	0.704	0.742	0.751	0.718	0.671
11x11	0.798	0.77	0.756	0.723	0.732	0.709	0.671	0.676
13x13	0.779	0.761	0.747	0.742	0.747	0.676	0.653	0.643
15x15	0.793	0.784	0.779	0.737	0.681	0.639	0.643	0.591
17x17	0.845	0.817	0.714	0.699	0.62	0.62	0.573	0.582

Table 7.34: MAEs applying BSIF in the JAFFE db with 224x224 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.77	0.822	0.803	0.789				
5x5	0.803	0.747	0.775	0.709	0.69	0.723	0.751	0.742
7x7	0.761	0.765	0.737	0.699	0.728	0.69	0.667	0.728
9x9	0.817	0.775	0.742	0.686	0.686	0.653	0.639	0.606
11x11	0.822	0.761	0.737	0.704	0.695	0.657	0.563	0.596
13x13	0.803	0.728	0.718	0.681	0.667	0.596	0.582	0.535
15x15	0.765	0.751	0.775	0.662	0.606	0.62	0.596	0.521
17x17	0.812	0.779	0.737	0.653	0.582	0.559	0.526	0.535

Table 7.35: MAEs applying BSIF in the JAFFE db with concatenating 224x224 and 112x112 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.784	0.746	0.779	0.779				
5x5	0.789	0.723	0.746	0.7	0.685	0.685	0.695	0.714
7x7	0.718	0.69	0.714	0.685	0.69	0.643	0.61	0.685
9x9	0.775	0.695	0.704	0.648	0.62	0.587	0.592	0.601
11x11	0.77	0.761	0.7	0.653	0.624	0.577	0.516	0.559
13x13	0.761	0.657	0.615	0.587	0.559	0.54	0.54	0.488
15x15	0.704	0.676	0.676	0.596	0.54	0.559	0.507	0.502
17x17	0.7	0.69	0.746	0.596	0.592	0.545	0.549	0.493

Table 7.36: MAEs applying BSIF in the JAFFE db with concatenating 224x224, 112x112 and 56x56 images. CV by subject.

- **MAE of Hybrid LBP HOG BSIF:** 0.422
- **MAE of Hybrid concatenated LBP HOG BSIF:** 0.422
- **MAE of VGG-Face:** 0.577
- **MAE of VGG-F:** 0.615

7.4 CK+ experiments and results

In the case of the CK+ database, since it is a sequence of images, the first image was taken as the neutral emotion (in this database we assumed there were 7 emotion instead of 6; the six basic emotions and the neutral one), and the last 3 images which were the peak of the emotion were taken as the real emotion. The sequence of images went from a neutral face to an emotion, being the last image in the sequence the peak of the emotion. However, if the images were colored, were left colored. In here, we also did CV by subject, as we did with the JAFFE, because it gave better results and other authors were doing so.

CellSize/ NumBins	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6x6	0.224	0.236	0.237	0.239	0.239	0.243	0.243	0.242	0.245	0.242	0.242	0.245	0.242	0.244	0.247	0.245	0.246	0.248	0.249	0.249
7x7	0.218	0.223	0.219	0.226	0.229	0.232	0.23	0.237	0.234	0.24	0.237	0.242	0.237	0.242	0.241	0.239	0.238	0.24	0.242	0.239
8x8	0.216	0.221	0.221	0.223	0.225	0.225	0.226	0.23	0.23	0.229	0.232	0.232	0.231	0.23	0.233	0.231	0.23	0.232	0.231	0.232
9x9	0.207	0.211	0.21	0.213	0.215	0.215	0.218	0.218	0.218	0.219	0.219	0.22	0.22	0.219	0.219	0.221	0.22	0.222	0.22	0.224
10x10	0.203	0.21	0.208	0.212	0.21	0.213	0.212	0.213	0.216	0.214	0.215	0.216	0.216	0.216	0.219	0.216	0.218	0.215	0.219	0.219
11x11	0.204	0.203	0.205	0.209	0.21	0.206	0.21	0.208	0.212	0.209	0.21	0.209	0.212	0.21	0.211	0.21	0.215	0.212	0.212	0.215
12x12	0.206	0.203	0.202	0.206	0.207	0.206	0.209	0.206	0.21	0.207	0.21	0.206	0.209	0.21	0.211	0.21	0.212	0.21	0.211	0.211
13x13	0.196	0.197	0.197	0.2	0.197	0.197	0.194	0.195	0.196	0.196	0.199	0.199	0.195	0.201	0.199	0.199	0.197	0.197	0.2	0.199
14x14	0.192	0.191	0.194	0.198	0.197	0.199	0.198	0.198	0.199	0.197	0.206	0.197	0.198	0.197	0.203	0.198	0.196	0.2	0.206	0.199
15x15	0.198	0.193	0.192	0.19	0.19	0.187	0.19	0.189	0.188	0.19	0.187	0.187	0.186	0.187	0.189	0.186	0.187	0.186	0.187	0.185
16x16	0.194	0.19	0.195	0.196	0.195	0.193	0.194	0.195	0.191	0.197	0.197	0.195	0.191	0.193	0.195	0.192	0.19	0.19	0.193	0.191
17x17	0.187	0.181	0.188	0.189	0.192	0.191	0.191	0.19	0.184	0.187	0.186	0.186	0.186	0.186	0.186	0.189	0.186	0.188	0.188	0.187
18x18	0.19	0.184	0.185	0.185	0.184	0.187	0.186	0.189	0.188	0.19	0.192	0.191	0.19	0.192	0.192	0.193	0.192	0.191	0.193	0.193
9x19	0.172	0.18	0.18	0.178	0.175	0.178	0.182	0.18	0.179	0.178	0.181	0.18	0.178	0.18	0.182	0.178	0.179	0.178	0.181	0.18
20x20	0.183	0.183	0.187	0.178	0.182	0.176	0.177	0.174	0.179	0.177	0.179	0.175	0.176	0.178	0.18	0.177	0.177	0.18	0.182	0.179
21x21	0.183	0.174	0.181	0.18	0.179	0.177	0.18	0.176	0.179	0.181	0.18	0.18	0.181	0.182	0.182	0.182	0.183	0.182	0.183	0.183
22x22	0.174	0.166	0.174	0.174	0.179	0.174	0.177	0.172	0.174	0.173	0.181	0.174	0.176	0.177	0.177	0.174	0.177	0.176	0.177	0.175
23x23	0.18	0.181	0.18	0.182	0.178	0.179	0.182	0.178	0.181	0.181	0.184	0.181	0.18	0.181	0.183	0.183	0.183	0.182	0.185	0.181
24x24	0.175	0.178	0.177	0.177	0.177	0.177	0.181	0.178	0.181	0.177	0.183	0.176	0.179	0.178	0.18	0.175	0.175	0.177	0.18	0.175
25x25	0.18	0.194	0.186	0.187	0.181	0.187	0.188	0.187	0.186	0.19	0.189	0.19	0.188	0.193	0.191	0.191	0.189	0.19	0.19	0.192

Table 7.37: MAEs aplying HOG in the CK+ db. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.349	0.341	0.332	0.327	0.316	0.307
4	0.275	0.261	0.267	0.263	0.264	0.266
8	0.259	0.249	0.251	0.255	0.247	0.231
16	0.266	0.256	0.253	0.248	0.235	0.23
32	0.265	0.263	0.255	0.257	0.241	0.231

Table 7.38: MAEs applying LBP in the CK+ db with images of 224x224 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.345	0.326	0.318	0.3	0.3	0.293
4	0.267	0.245	0.242	0.242	0.242	0.247
8	0.241	0.233	0.236	0.241	0.229	0.216
16	0.249	0.233	0.237	0.24	0.225	0.223
32	0.246	0.243	0.244	0.241	0.23	0.229

Table 7.39: MAEs applying LBP in the CK+ db with concatenating the same image in a resolution of 224x224 and 112x112 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.337	0.315	0.298	0.288	0.293	0.285
4	0.256	0.242	0.242	0.232	0.229	0.232
8	0.242	0.229	0.232	0.234	0.221	0.214
16	0.242	0.232	0.23	0.233	0.226	0.22
32	0.24	0.238	0.238	0.238	0.231	0.23

Table 7.40: MAEs applying LBP in the CK+ db with concatenating the same image in a resolution of 224x224, 112x112 and 56x56 pixels. CV by subject.

N. Neighbours/ Radius	2	3	4	5	6	7
2	0.346	0.337	0.329	0.34	0.368	0.408
4	0.28	0.274	0.244	0.268	0.294	0.365
8	0.256	0.238	0.205	0.218	0.276	0.324
16	0.238	0.239	0.222	0.244	0.275	0.352
32	0.252	0.241	0.236	0.268	0.328	0.355

Table 7.41: MAEs applying LBP Pyramid in the CK+ db. CV by subject.

Concat/ CellSize	10x10	11x11	12x12	13x13	14x14	15x15	16x16	17x17	18x18	19x19	20x20	21x21	22x22	23x23	24x24	25x25
224	0.263	0.265	0.267	0.258	0.26	0.259	0.255	0.267	0.265	0.262	0.268	0.266	0.265	0.278	0.271	0.306
224. 112	0.265	0.261	0.266	0.258	0.261	0.254	0.251	0.252	0.253	0.258	0.252	0.248	0.243	0.267	0.266	0.298
224. 112. 56	0.265	0.261	0.266	0.258	0.261	0.254	0.251	0.252	0.253	0.258	0.252	0.248	0.243	0.267	0.266	0.298

Table 7.42: MAEs aplying LBP with fixed nNeighbours=8 and radius=1 in the CK+ db. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.561	0.512	0.471	0.479				
5x5	0.552	0.512	0.452	0.436	0.41	0.401	0.411	0.405
7x7	0.557	0.495	0.441	0.362	0.365	0.359	0.36	0.378
9x9	0.579	0.53	0.427	0.339	0.353	0.332	0.318	0.345
11x11	0.522	0.489	0.412	0.357	0.358	0.327	0.306	0.309
13x13	0.503	0.508	0.419	0.323	0.348	0.298	0.302	0.304
15x15	0.498	0.462	0.382	0.342	0.327	0.302	0.292	0.306
17x17	0.484	0.466	0.385	0.34	0.299	0.299	0.297	0.297

Table 7.43: MAEs applying BSIF in the CK+ db with 224x224 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.534	0.491	0.432	0.41				
5x5	0.531	0.483	0.382	0.372	0.359	0.356	0.352	0.359
7x7	0.494	0.43	0.365	0.314	0.297	0.29	0.3	0.322
9x9	0.477	0.442	0.327	0.271	0.283	0.283	0.288	0.306
11x11	0.463	0.413	0.322	0.282	0.289	0.275	0.278	0.297
13x13	0.442	0.41	0.331	0.275	0.28	0.268	0.286	0.295
15x15	0.46	0.385	0.3	0.284	0.284	0.284	0.278	0.293
17x17	0.456	0.378	0.325	0.303	0.282	0.255	0.277	0.277

Table 7.44: MAEs applying BSIF in the CK+ db with concatenating 224x224 and 112x112 images. CV by subject.

Filter Size/ bits	5	6	7	8	9	10	11	12
3x3	0.503	0.448	0.386	0.372				
5x5	0.461	0.407	0.348	0.32	0.326	0.317	0.328	0.339
7x7	0.448	0.352	0.306	0.281	0.272	0.267	0.281	0.303
9x9	0.427	0.368	0.29	0.243	0.259	0.277	0.277	0.289
11x11	0.401	0.339	0.293	0.254	0.272	0.269	0.269	0.28
13x13	0.404	0.349	0.299	0.254	0.277	0.258	0.288	0.286
15x15	0.404	0.363	0.288	0.265	0.267	0.262	0.264	0.294
17x17	0.423	0.344	0.304	0.27	0.261	0.253	0.261	0.28

Table 7.45: MAEs applying BSIF in the CK+ db with concatenating 224x224, 112x112 and 56x56 images. CV by subject.

- **MAE of Hybrid LBP HOG BSIF:** 0.19

- **MAE of Hybrid concatenated LBP HOG BSIF:** 0.192

- **MAE of VGG-Face:** 0.321

- **MAE of VGG-F:** 0.242

7.5 Summary

Filter Size/ bits	CAFE Original	CAFE Aligned	CAFE Aligned (CV People)	JAFFE	CK+ Aligned
HOG	0.348	0.229	0.221	0.413	0.172
LBP 224X224	0.414	0.274	0.266	0.502	0.23
LBP concat 224 112	0.403	0.248	0.242	0.484	0.216
LBP concat 224 112 56	0.394	0.243	0.237	0.484	0.214
LBP Pyramid	0.534	0.3	0.311	0.559	0.205
BSIF 224x224	0.468	0.332	0.322	0.573	0.292
BSIF concat 224 112	0.429	0.277	0.274	0.521	0.255
BSIF concat 224 112 56	0.435	0.27	0.271	0.488	0.243
Hybrid LBP HOG BSIF	0.339	0.217	0.203	0.422	0.19
Hybrid concat LBP HOG BSIF	0.341	0.218	0.206	0.422	0.192
VGG-Face	0.428	0.385	0.391	0.577	0.321
VGG-F	0.552	0.295	0.303	0.615	0.242

Table 7.46: BSIF 224x224

8. CHAPTER

Conclusions and Future Work

Experiments showed in these databases what was unexpected; that the Neural Networks does not seem to be the best option. More classical feature descriptors like HOG or the hybrid one combining multiple descriptors give by far better results. However, this might be due to using some pretrained neural networks that were trained with images of people, but not prepared to detect emotions. It was also surprising to see that if the cross-validation was done by emotion the results were worse than if the crossvalidation was done by people. However, it was not a surprise to see that in the databases where the images where not aligned, the results with those same databases when alighning the images were improved. Moreover, it was not a surprise either to see that with the database JAFFE which are images of Japanese women, the results were not as good as in the other two databases, because the facial expresions in asian people are more difficult to distinguish. For future work it might be a good idea to have more labelled emotion databases to do more fair experiments, since in this database one of the database was only children faces, and another one was only Japanese females. It might be a good idea too to try to retrain the VGG-Face and VGG-F CNNs, since they were not trained for the purpose we are using them.

Bibliography

[Ali et al., 2016] Ali, G., Iqbal, M. A., and Choi, T.-S. (2016). Boosted nne collections for multicultural facial expression recognition. *Pattern Recognition*, 55:14–27.

[Asthana et al., 2013] Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2013). Robust discriminative response map fitting with constrained local models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451.

[Brown et al., 2005] Brown, M., Szeliski, R., and Winder, S. (2005). Multi-image matching using multi-scale oriented patches. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 510–517. IEEE.

[Buranajun et al., 2007] Buranajun, P., Sasananan, M., and Sasananan, S. (2007). Prediction of product design and development success using artificial neural network.

[Burkert et al., 2015] Burkert, P., Trier, F., Afzal, M. Z., Dengel, A., and Liwicki, M. (2015). Dexpression: Deep convolutional neural network for expression recognition. *arXiv preprint arXiv:1509.05371*.

[Byeon and Kwak, 2014] Byeon, Y.-H. and Kwak, K.-C. (2014). Facial expression recognition using 3d convolutional neural network. *International journal of advanced computer science and applications*, 5(12):1–8.

[Chatfield et al., 2014] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

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

[Ekman, 1992] Ekman, P. (1992). Facial expressions of emotion: New findings, new questions.

[Ekman and Friesen, 1971] Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

[Fan and Tjahjadi, 2015] Fan, X. and Tjahjadi, T. (2015). A spatial-temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences. *Pattern Recognition*, 48(11):3407–3416.

[Fasel, 2002] Fasel, B. (2002). Robust face analysis using convolutional neural networks. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 40–43. IEEE.

[Gottman et al., 1996] Gottman, J. M., McCoy, K., Coan, J., and Collier, H. (1996). The specific affect coding system (spaff) for observing emotional communication in marital and family interaction. *What predicts divorce*, pages 112–195.

[Huang et al., 2015] Huang, G., Huang, G., Song, S., and You, K. (2015). Trends in extreme learning machines: A review. *Nural Networks*, 61:32–48.

[Jain and Li, 2011] Jain, A. K. and Li, S. Z. (2011). *Handbook of face recognition*. Springer.

[Kannala and Rahtu, 2012] Kannala, J. and Rahtu, E. (2012). Bsif: Binarized statistical image features. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1363–1366. IEEE.

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

[Liu et al., 2015] Liu, M., Li, S., Shan, S., and Chen, X. (2015). Au-inspired deep networks for facial expression feature learning. *Neurocomputing*, 159:126–136.

[Liu et al., 2014] Liu, P., Han, S., Meng, Z., and Tong, Y. (2014). Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812.

[LoBue, 2014] LoBue, V. (2014). The child affective facial expression (cafe) set.

[LoBue and Thrasher, 2014] LoBue, V. and Thrasher, C. (2014). The child affective facial expression (cafe) set: Validity and reliability from untrained adults. *Frontiers in psychology*, 5.

[Lopes et al., 2017] Lopes, A. T., de Aguiar, E., De Souza, A. F., and Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order. *Pattern Recognition*, 61:610–628.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Lucey et al., 2010] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE.

[Lyons et al., 1998] Lyons, M. J., Akamatsu, S., Kamachi, M., Gyoba, J., and Budynek, J. (1998). The japanese female facial expression (jaffe) database. In *Proceedings of third international conference on automatic face and gesture recognition*, pages 14–16.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[Nguyen et al., 2014] Nguyen, D. T., Cho, S. R., Shin, K. Y., Bang, J. W., and Park, K. R. (2014). Comparative study of human age estimation with or without preclassification of gender and facial expression. *The Scientific World Journal*, 2014.

[Nielsen, 2015] Nielsen, M. A. (2015). Neural networks and deep learning.

[Ojala et al., 2002] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987.

[OpenCVdocs, 2017] OpenCVdocs (2017). Local Binary Pattern lbp. `https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html`.

[Pantic and Bartlett, 2007] Pantic, M. and Bartlett, M. S. (2007). Machine analysis of facial expressions. In *Face recognition*. InTech.

[Parkhi et al., 2015] Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *BMVC*, volume 1, page 6.

[Qian et al., 2011] Qian, X., Hua, X.-S., Chen, P., and Ke, L. (2011). Plbp: An effective local binary patterns texture descriptor with pyramid representation. *Pattern Recognition*, 44(10):2502–2515.

[Shan et al., 2009] Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816.

[Song et al., 2014] Song, I., Kim, H.-J., and Jeon, P. B. (2014). Deep learning for real-time robust facial expression recognition on a smartphone. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, pages 564–567. IEEE.

[Trefnỳ and Matas, 2010] Trefnỳ, J. and Matas, J. (2010). Extended set of local binary patterns for rapid object detection. In *Computer Vision Winter Workshop*, pages 1–7.

[Turing Finance, 2014] Turing Finance (2014). 10 misconceptions about neural networks.

[Yu et al., 2015] Yu, W.and Zhuang, F., He, Q., and Shi, Z. (2015). Learning deep representations via extreme learning machines. *Neurocomputing*, 149:308–315.

[Zhang et al., 2007] Zhang, L., Chu, R., Xiang, S., Liao, S., and Li, S. Z. (2007). Face detection based on multi-block lbp representation. In *International Conference on Biometrics*, pages 11–18. Springer.

[Zhous et al.,] Zhous, S., Chen, Q., and Wang, X. Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, 120:536–546.