Intelligent road lane mark extraction using a Mobile Mapping System

By

Asier Izquierdo Pérez

PhD Advisors:

Prof. Dr. Manuel Graña Romay, Jose Manuel Lopez Guede

at The University of the Basque Country (UPV/EHU)

December 2022

Abstract

This Thesis deals with the task of detection of different road lane marks, using signals from the different sensors that a Mobile Mapping System (MMS) is composed of, namely image sensors (cameras) and LiDAR sensor. MMSs are the combination of various navigation and remote sensing technologies on a common moving platform. During the last years, road landmark inventory has raised increasing interest in different areas: the maintenance of transport infrastructures, road 3d modelling, GIS applications, etc. Several commercial sensors are available which include a set of high-resolution cameras already calibrated in order to generate panoramic images, and a LiDAR sensor that allows to capture up to 700,000 georeferenced points, plus other components that provide ancillary information. The lane mark detection is posed as a two-class classification problem over a highly class imbalanced dataset. To cope with this imbalance we have applied Active Learning approaches. This Thesis has been divided into two main computational parts. In the first part, we have evaluated different Machine Learning approaches using panoramic images, obtained from image sensor, such as Random Forest (RF) and ensembles of Extreme Learning Machines (V-ELM), obtaining satisfactory results in the detection of road continuous lane marks. In the second part of the Thesis, we have applied a Random Forest algorithm to a LiDAR point cloud, obtaining a georeferenced road horizontal signs classification. We have not only identified continuous lines, but also, we have been able to identify every horizontal lane mark detected by the LiDAR sensor.

Keywords Mobile Mapping System, Active Learning, Random Forest, Extreme Learning Machine, LiDAR, Image Sensor, Panoramic image.

Acknowledgements

As in a long distance race, the journey to reach the finish line has had good and not so good moments, difficult moments and disparate situations. Along the way we have lived through a pandemic and I have been able to witness the most beautiful moment of my life; the birth of my children. And as in any other race, the success of crossing the finish line often could not be possible without the unconditional support of those closest to me, which is why I would like to dedicate a few lines to those who have been by my side and have supported me at all times during this time.

That is why I would first like to express my gratitude to José Manuel López and Manuel Graña for knowing how to guide me through the whole process.

My acknowledgements to the company Airestudio for trusting me and encouraging me to take this path and for being able to count on them at all times.

Last but not least, I would like to make a special mention to my close circle; my family. My wife, who has been there for me at all times, in the good times and in the not so good times. My parents, for their unconditional support and my children who have been my light in the dark days.

The work has been partially supported by Airestudio Geoinformation Technologies Scoop and Basque Government's BIKAINTEK grant. The work has also been supported by FEDER funds for the MINECO and MICIN projects TIN2017-85827-P and PID2020-116346GB-I00, the grants IT1284-19 and IT1689-22 as university research group of excellence from the Basque Government, Elkartek projects KK-2022/00051 and KK-2021/00070, and project 7-AA-3091-EG of the *Consejería de Fomento, Infraestructuras y Ordenación del Territorio. Dirección General de Infraestructuras de la Junta de Andalucía.*

Asier Izquierdo Pérez

“No basta con tener ingenio, lo principal es aplicarlo bien”

-René Descartes-

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Problem definition 2

1.3 Active learning 5

1.4 Overall description of works in the Thesis 5

1.5 Publications 7

1.6 Contents of the Thesis 8

2 State of the Art **11**

2.1 Introduction 11

2.2 Processing Image data 13

2.3 Processing LiDAR 3D data 17

2.3.1 Method 1 19

2.3.2 Method 2 20

2.4 Fusing 3D LiDAR data and RGB data 22

2.5 Conclusions 23

3 Dataset for the experiments **25**

3.1 Mobile Mapping System 25

3.1.1 Positioning system 25

3.1.2 Image sensor 26

3.1.3 LiDAR system 26

3.2 Image sensor based data 28

3.3 LiDAR intensity image dataset 31

3.4 LiDAR point clouds dataset 33

4 METHODS 37

4.1 Experimental setups 38

4.2 Performance measures 38

4.3 Classification Methods 42

4.3.1 Random Forest Classifiers 42

4.3.1.1 RF construction 43

4.3.2 Ensemble of Extreme Learning Machines 43

4.4 Active Learning 46

4.4.1 Classification uncertainty in ensemble classifiers 48

4.4.2 Active learning for image segmentation 48

4.4.3 Active learning for LiDAR point cloud classification . 49

4.4.4 Active Learning and class imbalance 49

4.5 Raster image from LiDAR point cloud 51

4.6 LiDAR Point Cloud features 52

4.6.1 Surface Normal estimation in a Point Cloud 53

4.6.2 Point Cloud Slopes computation 55

4.7 Gabor texture features 57

5 Experimental Results 61

5.1 Results on RGB panoramas images 61

5.1.1 Model parameter exploration. 61

5.1.2 Experimental Results 62

5.2 Results on the intensity images from LiDAR 69

5.2.1 Model parameter exploration. 69

5.2.2 Experimental Results 69

5.3 Results on the LiDAR sensor Point Cloud data 76

5.3.1 Model parameter exploration. 76

5.4 Conclusions 85

6 Conclusions 87

6.1 Achievements 87

6.2 Limitations 89

6.3 Future work 89

List of Figures

1.1 The ladybug sensor mounted in a car. 4

1.2 An example panorama from the ladybug sensor, obtained by the composition of the view of the five cameras. 4

2.1 Shadows in the road due to sun position 14

2.2 Intensity of the LiDAR sensor point cloud. 18

2.3 RGB image composed by the corresponding RGB pixels to each sample in the LiDAR point cloud. 18

2.4 Left: RGB point cloud. Right: generated 2D image 20

2.5 Left: Point cloud represented as intensity values. Right: Lane marks extraction 21

2.6 Point cloud and image fusion proposal [64] 24

3.1 IP-S3 HD1 Mobile Mapping System 27

3.2 An example panorama image obtained from the ladybug sensor after stitching the 5 camera images, obtained by the composition of the views of the five cameras 28

3.2 An example panorama image obtained from the ladybug sensor after stitching the 5 camera images, obtained by the composition of the views of the five cameras 28

3.2 An example panorama image obtained from the ladybug sensor after stitching the 5 camera images, obtained by the composition of the views of the five cameras 28

3.3 Some of the experimental images (left column) and their corresponding manually delineated ground truth (right column) white=background, black=road signs. 29

3.4 Individual images captured by the different cameras of the sensor. Cameras 0 to 4 correspond to the individual plane. Camera 5 corresponds to the zenith plane. 29

3.5 Example of filtered images with Gabor filters of various orientation and scales. 30

3.6 Left: one of the experimental images, Right: its corresponding manually delineated ground truth (white is the background) 32

3.7 Example of filtered images with Gabor filters of various orientation and scales. 34

3.8 Left: one of the experimental LiDAR Point Cloud, Right: its corresponding manually extracted ground truth. 35

4.1 Pipeline of the experimental setup for the Active Learning segmentation process with panoramic images 39

4.2 Pipeline of the experimental setup for the Active Learning segmentation process with LiDAR intensity based images . . 40

4.3 Pipeline of the experimental setup for the Active Learning segmentation process with LiDAR Point Clouds 40

4.4 Orthoimage generation process 52

4.5 Graphical representation of the calculation of the normal vectors 56

4.6 Same intensity values between different elements. 57

4.7 Examples of filtered images with Gabor filters of various orientations and scales. 59

4.7 Examples of filtered images with Gabor filters of various orientations and scales. 59

4.7 Examples of filtered images with Gabor filters of various orientations and scales. 59

5.1 Some visual results of the trained RF (left column) and V-ELM (right column) ensemble classifiers using balanced training sample increments. Parameter settings for RF are NT=50, NVS=5, NS=100. Parameter settings for V-ELM are NELM=50, NHELM=10, NS=50. 63

5.2 Some visual results of the trained RF ensemble classifiers using balanced training sample increments. 70

5.3 Some visual results of the trained RF ensemble classifiers using balanced training sample different parameters settings. Left column top-view, right column front-isometry view . . . 78

6.1 Limitations of the current MMS system configuration. The low density of points capturing the elements in the vertical plane is marked by a yellow circle. 90

List of Tables

2.1 Algorithms summary 13

5.1 Summary of the experimental design for RF (left) and V-ELM (right) for panoramic imagery. 62

5.2 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced. . . 65

5.2 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced. . . 65

5.2 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced. . . 65

5.3 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performarce results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 66

5.3 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performarce results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 66

5.3 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performarce results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 66

5.4 Performance of AL using V-ELM ensemble classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of ELM components (NELM), the number of hidden units at each ELM (NHELM), the number of samples added in each iteration of the active learning algorithm (NS). 67

5.4 Performance of AL using V-ELM ensemble classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of ELM components (NELM), the number of hidden units at each ELM (NHELM), the number of samples added in each iteration of the active learning algorithm (NS). 67

5.4 Performance of AL using V-ELM ensemble classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of ELM components (NELM), the number of hidden units at each ELM (NHELM), the number of samples added in each iteration of the active learning algorithm (NS). 67

5.5 Assessment of impact of sample selection strategy (Balanced unbalanced) over performance results shown in Tables 5.2 and 5.3, we give the p values of Wt=Welch's t-test, WR=nonparametric Wilcoxon rank test. 68

5.6 Assessment of impact of the size of training data sampling increments over performance results shown in Tables 5.2 and 5.4, we give the p values of Wt=Welch's t-test, WR=nonparametric Wilcoxon rank test. 68

5.7 Summary of the experimental design for RF. 70

5.8 Assessment of impact of the parameter setting over results shown in Tables 5.9, 5.10. We give the p values of **Tt** = T-test, **WR** = Wilcoxon signed-rank test. Significance level $\alpha = 0.05$ 72

5.9 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced. . . 74

5.10 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 75

5.10 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 75

5.10 Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). 75

5.11 Summary of the experimental design for RF. 76

5.12 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI). 79

5.13 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI). 80

5.14 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI). 81

5.15 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI). 82

5.16 AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI). 83

5.17 Assessment of impact of the parameter setting over results shown in Tables 5.9, 5.10. We give the p values of **Tt** = T-test, **WR** = Wilcoxon signed-rank test. Significance level $\alpha = 0.05$. 84

5.18 Average performance measures according to the used parameters (SEN, SP, PPV, AUC). 85

Chapter 1

Introduction

This chapter provides an overall introduction to the Thesis. First, we provide a short motivation and context in Section 1.1. A summary of the Thesis contents and contributions are given in Section 1.2. We comment on Active Learning in Section 1.3. We provide a summary description of the works carried oout during the Thesis in Section 1.4. The publications achieved during the work of the Thesis are listed in section 1.5. Finally, the main structure of the Thesis is presented in section 1.6.

1.1 Motivation

Mobile Mapping system (MMS) is the combination of various navigation and remote technologies on a common moving platform. It can be considered a powerful "tool" to represent and measure the reality. Usually, this type of systems is composed of an image sensor and a LiDAR sensor. Common commercial software used with MMS, in case of images, work on a panoramic image that are the result of stitching together the different images obtained from each individual camera. Commonly, these images are not georeferenced. Regarding LiDAR sensors, the raw data coming from the LiDAR sensor is not coloured. The colour is obtained from a post-process where each LiDAR point is attributed the colour of the corresponding pixel in the panoramic image. The optimal situation should be that the image and the LiDAR are registered so that a point in the image corresponds univocally to a point in the LiDAR cloud, but this is not always the case. The main reason for this lack of registration is that image domains are discrete integer grids, while point cloud domains are continuous coordiantes in the 3D space. In order to obtain a map as close as possible from discrete data to continuous data, the solution should be to reduce the image pixel size, which would lead to increasing the size of our data (remember that we are dealing with several hundred kilometres). In these cases, no matter how small the pixel size is, we might have a miscoloured cloud due to systematic failures in the mutual calibratoin of all the sensors of the MMS system. To do extraction of road structures, the common approach is to work on the colour of the cloud points, not touching the colour of the images obtained by the image sensor or not taking full advantage of the potential of the LiDAR sensor, that allows to use information other than the of RGB colour values.

The Thesis has been carried out in a collaboration between the University and a company, which has been covered by a specific grant of the Basque Government. At the company, I am currently working on a project where we are extracting road markings for later inventory. The biggest problem we are facing is poor road marking extraction since most extraction methods are based on colour based segmentation. These difficulties throughout the projects have been one of my greatest motivations to start the adventure of research at the level of PhD Thesis in order to be able to look for alternative approaches improving the extraction of road lane marks.

1.2 Problem definition

In this chapter we deal with the extraction methodologies of the horizontal signals drawn on the road: lane limiting lines, arrows, and other horizontal signalling. We will work with the data collected by the LiDAR sensor as well as with the panoramic images generated from the Ladybug camera set. The capture conditions, especially the illumination, is wildly changing from one traveling capture to another, or during the same trip due to the changes in the position of the sun, the time of the year, the weather, etc. Besides, the road maintenance is often in bad condition, so the road marking may be fading or interrupted.

The image provides only radiometric information. Although we find that most of the algorithms are designed for image analysis, we will also focus on the data collected by the LiDAR sensor. This data does not consist only of points with X, Y, Z values. A point cloud is created from many returns with a number of characteristics and attributes that make each return practically unique. The following LiDAR point attributes are maintained for each recorded laser pulse, additional information is stored along with each X, Y, Z value: intensity, return number, number of returns, RGB values (information that is normally obtained from the image), GPS time, scan angle and scan direction.

Finally, for supervised classification approaches the construction of the labelled dataset would be costly so we are extremely interested in exploiting cheaper alternatives, such as the Active Learning strategy [56].

The task ahead is driven by the industrial exploitation of a car mounted sensor nicknamed “ladybug” (figure 1.1)

More precisely, the sensor is the IP-S3 HD11 product of Topcon, Japan. It is composed of a positioning system (wheel encoder, GPS receiver, Inertial Measurement Unit), five cameras pointing at regular arc intervals of the circumference, and a sixth one pointing up. With one pointing directly at the rear of the car, and a LiDAR sensor. The views of the cameras are composed into panoramas such as the one shown in figure 1.2, where the central view corresponds to the rear view of the car, so that the front view is splited left and right of the panorama. At the bottom of the panorama, it is possible to appreciate the scaffolding supporting the sensors. Notice the changes in hue and saturation in the image due to the different sensitivities of the cameras. The LiDAR data and the image capture has a frequency of 1Hz. The task ahead is to create an inventory of the road signals and landmarks using both LiDAR and image data. All images are tagged spatially with coordinates provided by onboard GPS.

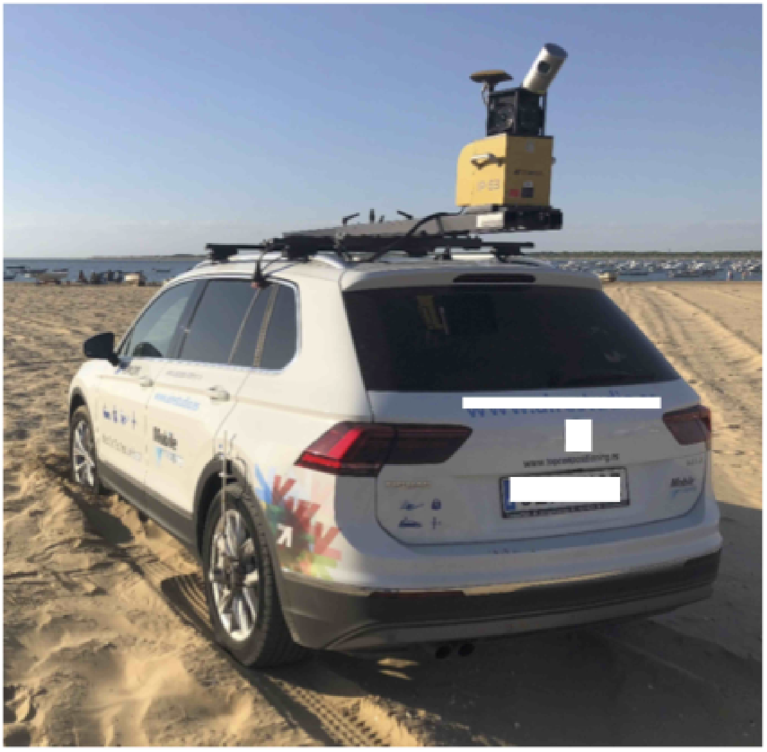


Figure 1.1: The ladybug sensor mounted in a car.

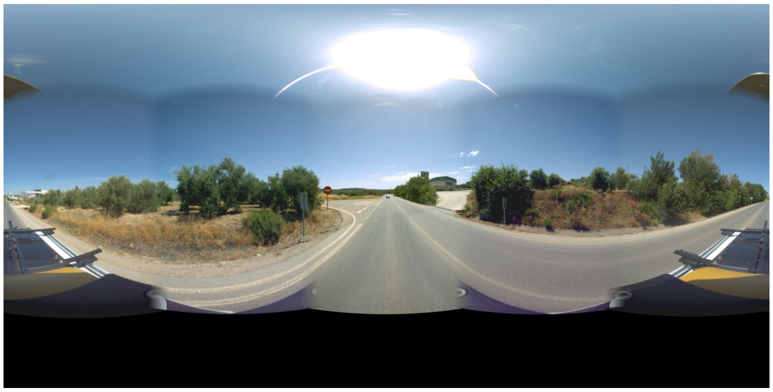


Figure 1.2: An example panorama from the ladybug sensor, obtained by the composition of the view of the five cameras.

1.3 Active learning

Classification approaches need careful selection and labelling of training data samples from the available data. In response to this issue, Active Learning (AL) [63] tries to achieve the most accurate classification using the smallest possible training set, minimizing the user interaction needed to label the training samples. Active Learning starts with a minimal training sample, adding new labelled samples in an iterative process. Aiming to provide the greatest increase in classifier accuracy [58], the additional samples are selected according to some classification uncertainty measure which does not require knowledge of the actual data label. Besides its benefits in economy of computation and data labelling, Active Learning is also useful when the underlying data statistics are non-stationary, so that the classifier built at one time instant may not be optimal later on. Active Learning has been successfully applied to classification of remote sensing images[63, 52, 62], and image retrieval based on semi supervised Support Vector Machines [37]. Active Learning inspiration for the selection of a minimal collection of training images is proposed in [40] for the development of combined generative and discriminative models in the segmentation of CT scans. An active feedback approach is used in [61] to improve the classification-based annotation of radiographs.[49]apply AL techniques in order to label Mobile Mapping system data reducing the requirements of manually annotated training samples for labelling point cloud scenes.

1.4 Overall description of works in the Thesis

In this chapter we formulate the road landmark segmentation problem [41] as classification of MMS data into road landmark and background classes. On the one hand we will work with the panoramic image obtained from the image sensor, and on the other hand we will deal with the point cloud obtained from the LiDAR sensor that is incorporated in the MMS system.

Image based methods

We collect the input panorama images provided by the so called *ladybug* sensor to be described later. The central part of the panorama contains most of the image useful information given by the rear view of the vehicle, so this central part is cropped for processing, discarding the remaining parts of the panorama. Gabor features are computed over the images collecting all the image features in a unique pool for the training of the classifiers and their validation. An Active Learning strategy is applied in order select the optimal training dataset. The classifier trained with the optimal training dataset is validated over the entire images, producing the performance report for the specific classifier. The validation task is repeated for the diverse classifiers and classifier parameters explored. The Active Learning oracle providing sample labels in the reported experiments is the ground truth provided by manual segmentation. In the practical application the oracle will be the user through some graphical interface for the selection of most informative pixels to be added to the training dataset.

For pixel classification we explore the results of Random Forest (RF) [17, 16] and Ensembles of Extreme Learning Machines (V-ELM) classifiers [23] based on texture features computed at pixel level. Specifically, a bank of Gabor filters is applied, so that the feature vector of each pixel is composed of the Gabor coefficients plus some spatial localization information. The position referred to are the pixel coordinates of the image. In case of needing to reference to reality, a conversion from pixel coordinate to terrain coordinate could be performed.

LiDAR based methods

It is decided to experiment with the LiDAR sensor, as it is considered that it provides information that the image sensor does not. The image only gives us RGB information. In a direct way, the image does not give us a georeferenced position in space, but from the image coordinate, we can approximate the ground coordinates with the help of the existing sensors in the vehicle (IMU, GPS/GNSS receiver and wheel encoder). We use The word approximate is used because images are discrete data and their georeferencing will be conditioned by the pixel size. LiDAR data does not consist only of points with X, Y, Z values. A point cloud is created from many returns with a number of characteristics and attributes that make each return practically unique. The following LiDAR point attributes are maintained for each recorded laser pulse, additional information is stored along with each X, Y, Z value: intensity, return number, number of returns, RGB values (information that is normally taken from the image), GPS time, scan angle and scan direction.

The LiDAR based methodology is further divided into two methodologies. On the one hand, orthophotos containing intensity values will be generated from the intensity data. Gabor features will be computed over these images. On the other hand, we will deal with LiDAR data directly, focusing on intensity values.

1.5 Publications

The Thesis is supported by the following achieved publications

1. zquierdo, A., Lopez-Guede, J.M., Graña, M. (2019). *Road Lane Landmark Extraction: A State-of-the-art Review.* In: Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E. (eds) Hybrid Artificial Intelligent Systems. HAIS 2019. Lecture Notes in Computer Science(), vol 11734. Springer, Cham. `https://doi.org/10.1007/978-3-030-29859-3_53`

2. Lopez-Guede, Jose, Izquierdo, Asier, Estevez, Julian, Graña, Manuel *Active learning for road lane landmark inventory with V-ELM in highly uncontrolled image capture conditions* (2021) Neurocomputing, Vol. 438 p. 259-269, `https://doi.org/10.1016/j.neucom.2020.07.151`

3. Izquierdo, A., Lopez-Guede, J.M. (2021). Active Learning for Road Lane Landmark Inventory with Random Forest in Highly Uncontrolled LiDAR Intensity Based Image. In: Herrero, Á., Cambra, C., Urda,

D., Sedano, J., Quintián, H., Corchado, E. (eds) 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020). SOCO 2020. Advances in Intelligent Systems and Computing, vol 1268. Springer, Cham. `https://doi.org/10.1007/978-3-030-57802-2_83` p. 862-871.

4. Izquierdo, Asier/Lopez-Guede, Jose/Graña, Manuel *'Extracción de Marcas Viales mediante Técnicas de Aprendizaje Activo a partir de Captura de Imágenes empleando Sensores MMS'* (2021). Virtual Conference of Doctoral Students in MSO 2021

5. Izquierdo, Asier/Lopez-Guede, Jose *'Active Learning for Road Lane Landmark Inventory Using Panoramic Images in Highly Uncontrolled Image Capture Conditions'*. (2022) Basque Conference on Cyber Physical Systems and Artificial Intelligence (https://www.ehu.eus/ccwintco/cybSPEED/bccpsai/

1.6 Contents of the Thesis

The contents of the Thesis are organised as follows:

- Chapter 2 contains a review of the literature concerning the main aspects of the Thesis, i.e. detection and extraction of road markings using different MMS sensors (LiDAR or image) or combining both sensors for detection and extraction.

- Chapter 3 contains a description of the dataset that has been used thorugout the Thesis. We comment on the components of the MMS system.

- Chapter 4 contains our contributions regarding the extraction of lane marks using different sensors from our MMS system. Active learning methods are applied successfully to images and LiDAR point clouds in order to extract road lane marks. For this purpose different methodologies are applied.

- Chapter 5 contains results obtained using different methodologies and sensors in order to extract horizontal lane marks.

- Finally, in chapter 6 some conclusions are considered. Achievements and shortcomings of this Thesis contribution are identified and a draft of future work is offered as an account for the pending *to-do* wish list in the future to cometh.

Chapter 2

State of the Art

In this Chapter we provide a short overview of the state of the art extracting road marker information from the data that is commonly provided by mounted monitoring systems used to obtain the inventory of road markers. Section 2.1 provisdes a short introduction. Section 2.2 reports on some methods that are based on image data provided by the cameras. Section 2.3 comments on methods based on 3D point clouds provided by the LiDAR sensor. Section 2.4 comments on methods that fuse 3D point clouds and RGB images to improve detectection. Section 2.5 gives some conclusions and motivation for the overall work in the Thesis as a departure from the state of the art.

2.1 Introduction

In the last few years, the need to elaborate and update road map databases has rapidly increased. These databases are used for different purposes, such as advanced driver assistance systems (ADAS), autonomous car navigation systems, roadway inventory, and any application based on real-time localization. Whatever the required application, an accurate model of the road is needed. Such an accurate model should contain information about horizontal and vertical road signs. In this paper, we focus on horizontal road signs, such as lane lines, crosswalks and arrow marks. Methodologies to extract horizontal lane image landmarks are heavily dependent on the sensor used to collect the road information. Depending on the sensor signal, different algorithms are used. Currently, it is possible to classify them into three big groups: high resolution optical image, Light Detection and Ranging (LiDAR) data, and a combination of optical images and LiDAR data. Regardless of the algorithm or the sensor used, the main classification problems are the following ones:

- False positives: signal features wrongly classified as road lane marks. Often the data is very badly balanced, with most of the datapoints belonging to the background. Most classifiers are biased towards the majority class, hence they tend to produce many false negatives. Correcting measures may tend to producethe contrary effect, many false positives.

- Image brightness issues: when the image is too bright, it is difficult to differentiate between different features. Brightness artifacts tend to be in the range of the white saturation, which is the range of values where we find the road markers.

- Lane landmark occlusions created by nearby vehicles. It is impossible to predict the oclusion by a passing by vehicle. The only solution is to try to register images in a sequence trying to remove the temporal effect of the passing by vehicle.

- Poor visibility: Inability to detect road features accurately due to visibility conditions, such as heavy rain or fog. Under extreme circunstances if often impossible to do any meaningful segmentation. Entire recordings will be lost because of these conditions. This issue falls completely out othe scope of this Thesis.

- Shadows may create misleading edges and texture on the road. Shadows ar3e another brithness artifact not that create discontinuities that can be misinterpreted as object contours.

	Imagery	LIDAR	Imagery+LIDAR
Template matching	[20]		
Hough transform	[10]		
SIFT	[48]		
SURF	[13]		
NURBS		[60]	
Otsu thresholding		[36]	
PCA		[15]	
RANSAC		[27]	

Table 2.1: Algorithms summary

In order to be able to extract road lane mark information, conventional computer vision and signal processing algorithms have been proposed depending on the sensor used. The main goal of the proposed algorithms is to extract road lane marks from a given road in an efficient way, solving noise and uncertainty issues. The most commonly used computer vision algorithms are referred in Table 2.1, according to the typology of the sensor used.

2.2 Processing Image data

Image sensors are routinely used by autonomous car navigation systems [47, 44]. Segmentation methods usually work on single images, so that the landmark extraction is repeated *de novo* for each image every s second, the time interval chosen between images. A CCD sensor is installed on the vehicle to obtain high quality images. Some authors think that the best way to capture the road surface and its lane marks is using vision-based approaches[45], since other technologies such as LiDAR based sensors are not capable to detect features in a flat plane accurately, such as road painted marks. Using imaging sensors to capture road landmarks can face the following problems:

- Applying a threshold on brightness, may lead to many false negatives and false positives due to occlusions or brightness artifacts.
- Occluded landmarks by other vehicles on the road.

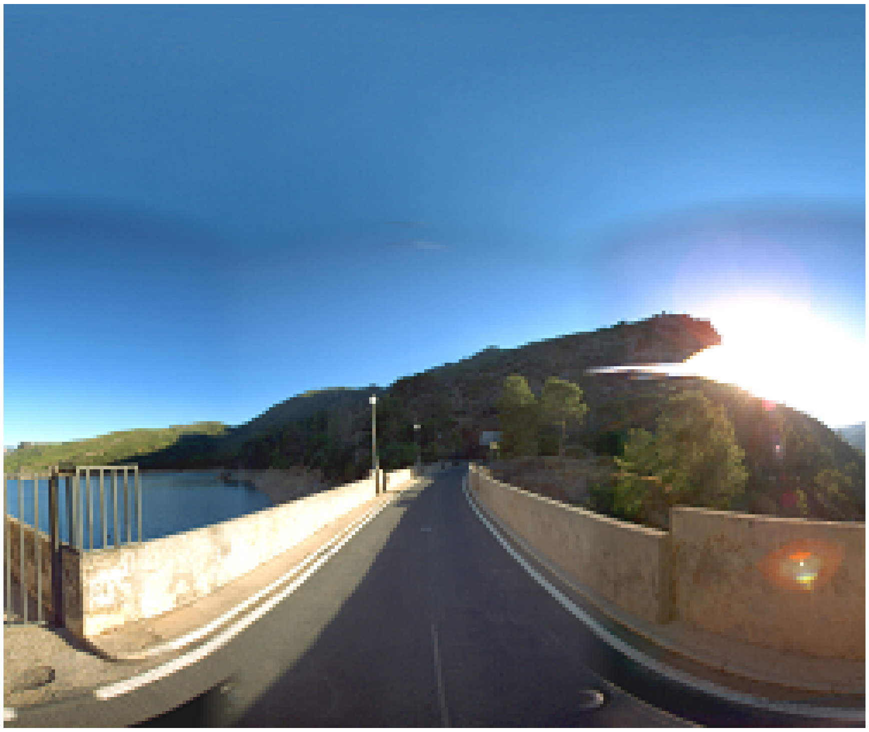


Figure 2.1: Shadows in the road due to sun position

- Image clarity issues (cast shadow on the image, saturated images, etc.), poor visibility conditions due to fog, heavy rain or reflections on wet roads at night time, or shadows in the road due to sun position in the sky, as shown in figure 2.1 [11].
- Other problems, such as different lane signs or a change in the width of the marks may also occur.

Many algorithms can be applied in order to extract horizontal lane marks. The most popular algorithms are the following ones:

- Template matching: [26] It is a technique in digital image processing for finding small parts of an image which match a template image [20]. The technique is widely used in object detection fields such as for vehicle tracking [12]. The crucial point is to adopt an appropriate measure to quantify similarity. However, this method also requires extensive computational cost since the matching process involves moving the template image to all possible positions in a larger target image and compute a numerical index which will dictate the most accurate position of the target image. This results in a significant optimization problem.

- Hough Transform: The purpose of the technique [**?**] is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maximum in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough Transform [10]. Hough transform has been notoriously used for line detection. The Hough transform is efficient if a large number of votes fall into the correct cell. For a cell to be easily detected regardless of errors introduced by noise in the image, it must stand out among its neighbours. This means that if we have many possible values for the parameters, the visibility of the cell could be compromised by its neighbouring cells; but if you have few values, erroneous results may appear, ignoring figures that are in the image. If we have many parameters the average number of votes for each cell is low and the cells corresponding to the real figures in the image do not necessarily appear to have a higher number of votes than the neighbouring cells. The complexity of the algorithm increases at a rate of $O(A^{p-2})$, where A is the number of points in the image and p is the number of parameters. Finally, the efficiency of the Hough transform depends on the quality of the algorithm's input data: edges must appear well delineated for the procedure to be efficient. Using the Hough transform on images with a lot of noise is usually a problem and some treatment must be applied to the image to remove it [59].

- SIFT: The Scale-Invariant Feature Transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images [48]. The essence of the algorithm is to find key points at different scales and calculate the direction of the key points. The key points found by SIFT are points that are very prominent and will not change due to factors such as illumination, affine transformation and noise, such as corner points, edge points, bright points in dark areas and dark points in bright areas. The drawback is that it is mathematically complicated and computationally heavy. SIFT is based on the Histogram of Gradients. That is, the gradients of each Pixel in the patch need to be computed and these computations cost time.

- SURF: Speeded-Up Robust Features (SURF) is a feature descriptor that extracts key points from different regions of a given image in order to find similarities between different images [13]. The SURF algorithm is similar to the SIFT algorithm explained above, with the difference that it is computationally faster because of the way it has been implemented, providing reliable matches among images under normal conditions [42].

The main issue of template matching is the need for precise templates and the need to overcome distortions and noise. Being a correlation-based algorithm, it can be robust for some kind of noises, but it is very sensitive to spatial distortions such as scale or perspective transformations. The Hough transform is quite robust but limited to specific features, such as lines. Some generalization has been proposed to detect objects under occlusion which can be useful in our problem given an adequate dataset of examples. Image feature extraction algorithms, such as SIFT and SURF, are quite robust against deformations and occlusions, because they extract salient image points that characterize the objects in the image. In fact, they were under patent until 2020, and therefore they could not be used freely in industrial applications until then. Alternative open and free algorithms are proposed in the literature that have been used extensively.

2.3 Processing LiDAR 3D data

As mentioned in vision-based sensor methodologies are susceptible to shadow, weather conditions, and brightness. Therefore, some authors proposed road lane mapping methods using LiDAR based sensors [51]. These methods are based on Mobile Mapping Systems (MMS) which use LiDAR technology, with scanners capable of collecting up to 106 measurements per second. The data acquired by MMSs is:

- Range and light intensity data from LiDAR scanners.
- Visual data (RGB) from panoramic images.
- Positioning data:
  - Absolute positioning by Differential Global Positioning Systems (DGPS) which are enhancements to the Global Positioning System (GPS) providing improved location accuracy, in the range of operations of each system, from the 10 ms nominal GPS accuracy to about 10 cm in the best implementations.
  - Inertial data from Inertial Measurement Unit (IMU).
  - Vehicle speed provided by odometer.

The data is fused by proprietary software of MMS, requiring the joint calibration of cameras and LiDAR sensors. The output usually consists of:

- A trajectory file, which contains information in terms of latitude, longitude, projected coordinates, altitude, sensor intrinsic rotations' angles (pitch, raw, yawn) and panoramic images information.
- Georeferenced panoramic images obtained from spherical image-based sensor(s).
- Georeferenced point cloud with RGB and intensity data, as shown in Figures 2.2 and 2.3.

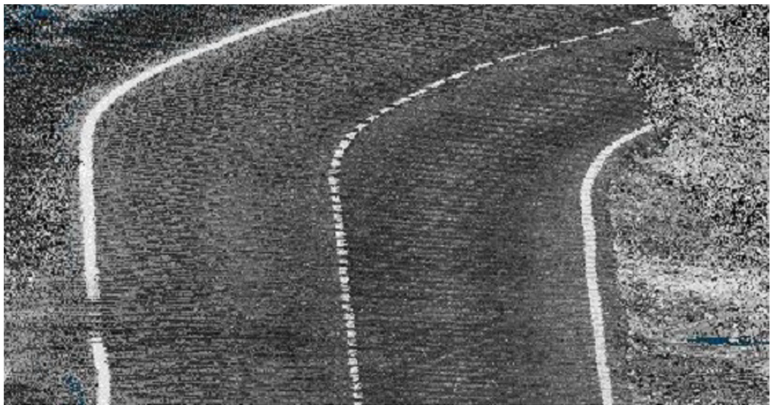


Figure 2.2: Intensity of the LiDAR sensor point cloud.

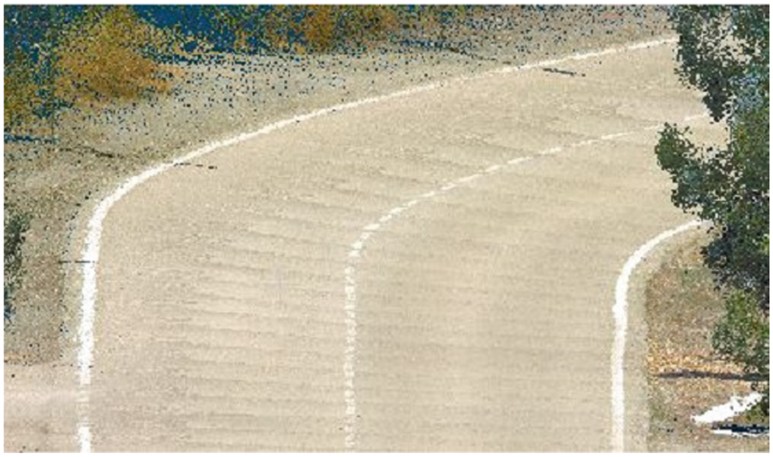


Figure 2.3: RGB image composed by the corresponding RGB pixels to each sample in the LiDAR point cloud.

Some studies [14, 57] propose to generate lane level maps by driving along the centreline of the road and then by analysing the trajectory obtained by Global Navigation Satellite System (GNSS). However, this approach is neither efficient nor accurate. Other authors use only georeferenced point clouds and trajectory file, in spite of the fact that MMS system images are available [57, 64]. Their methodology consists of a set of processing modules that allow them to obtain a digital image from the point cloud. This process is necessary since lane marks will be extracted from a nadir point of view, obtained by image processing.

Different workflows in order to extract information from the LiDAR point clouds are proposed. Here we review a couple of them.

2.3.1 Method 1

In order to obtain a digital image from a point cloud, Mancini et al. [51] proposed the following outline to extract objects that lie on the road surface:

1. Re-project point cloud coordinates from geographical to planar coordinates.

2. Geometric filtering.

3. Interpolation on a regular grid, where the generation of a raster 2D image is generated, as shown in figure 2.4

4. Apply a threshold to obtain a binary image.

5. Reduce noise applying morphological operations.

6. Isolate objects labelling and calculating morphological indicators.

7. Peak detector for lane line detection, including line validation and generation of attributes.

8. Apply a template matching for arrow marks.

9. Detect and extract crosswalks applying morphological indicators.

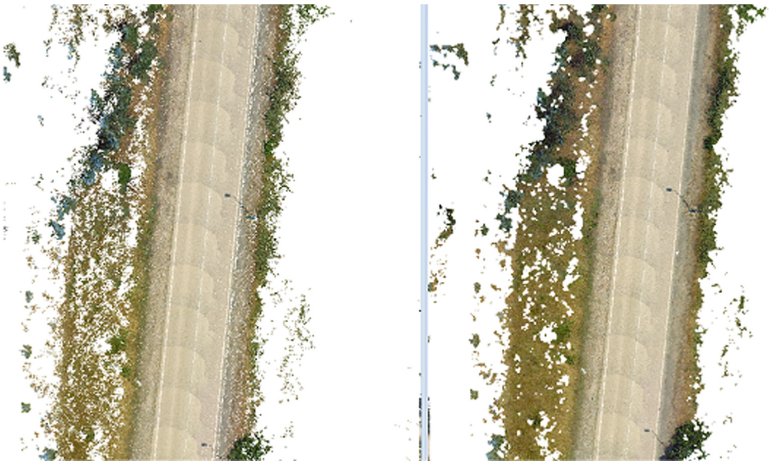


Figure 2.4: Left: RGB point cloud. Right: generated 2D image

2.3.2 Method 2

On the other hand, [64] proposes a different workflow to improve some problems such as time-consuming processing of point clouds (noise, unorganized points, big amount of points, etc.), variability in the reflective intensity values unevenness caused by dust of the road, intensity of lane marks or vehicle speed amongst others. As in the previous approach, a 2D image is generated from the point cloud, via different methods which include the following steps:

1. After the point cloud is gridded, unexpected changes in the normal vector are used in order to place the curb grids on both sides of the trajectory. Road edges are placed between the curb grids and used to segment the road points. So, to facilitate the following lane extraction, the inconsistencies of the reflective intensity of the road points are corrected.

2. Lane-marking extraction: after 3D road points are mapped into a 2D image, a self-adaptive thresholding method is developed to extract lane markings from the 2D image.

3. Lane mapping: In order to obtain the 3D lane lines, it is needed to

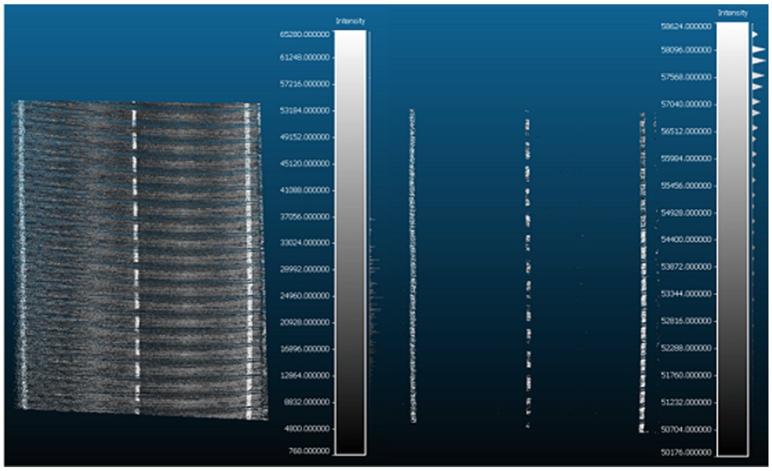


Figure 2.5: Left: Point cloud represented as intensity values. Right: Lane marks extraction

cluster and render points belonging to the same lane line type. To complement missing lines caused by occlusion, a global post-processing step that integrates all local extraction is used. Since lane signs are generally more reflective than the background surface, the intensity value can be useful to find lane-marking classified points. Some authors use a single-global threshold to extract lane signs [60]. Road boundaries are detected using a two-step method based on local road shape. Road sides and main axes are fitted by arc-length parameterised NURBS allowing to compute roads with a curvature at the desired resolution. Using the reflectivity information provided by LiDAR based sensors, road markings are extracted, as shown in figure 2.5.

- The Otsu thresholding method is proposed by [36] in order to find an intensity threshold value which maximizes the intracluster variance of road markings and road-surface LiDAR clusters, optimizing the segmentation of point clouds into asphalt and road marks.

2.4 Fusing 3D LiDAR data and RGB data

Only a few papers talk about combining images and 3D point clouds provided by MMSs to extract lane marks. According to [64] the following challenges exist using MMS based lane mapping methods;

- Time-consuming processes due to unorganized points or noisy classified points.
- Inconsistency of the reflective intensity measured point clouds due to different scanning ranges, incidence angles, surface characteristics, etc.
- Interferences caused by the unevenness due to the dirt or dust on the road.
- Instrument settings, velocity of the vehicle and quality of the MMS system.
- Within the lane-marks points surveyed, and particularly those which are further from the vehicle become sparse and difficult to recognize.
- Work with image texture and reflective intensity of the point cloud.

Generally, MMS-based lane mapping methods are divided in two steps. In a first stage, the road surface is extracted; in the following step the road lane-marks are extracted. To extract the road surface, many segmentation methods have been proposed:

- Planar-surface-based methods [53] generally use fitting algorithms. Most of them use the normal vectors of points which are usually estimated using RANdom SAmple Consensus (RANSAC) [68], Principal Components Analysis (PCA) [15], Robust and Diagnostic Principal Components Analysis or Hough transform [32].
- Edge-based methods [71], in a first process, linear road edges are detected and fitted using those edges in order to segment road surface [65]. Preliminary studies on the fusion of MMS images and point clouds have been carried out to detect tree species [66], pedestrian [43, 55] or

facades [33, 31], but not many studies have accomplished to extract lane marks from image and point cloud fusion.

- A validation method proposed by [64] fuses image texture saliency with the geometrical distinction of the lane-marking point cloud to improve the robustness of lane mapping. The proposed methodology is shown in figure 2.6, where the process mis divided in three sub-processes:

  1. Road surface extraction, based on the Normal vector using the RANdom SAmple Consensus (RANSAC) [54] algorithm, that is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers.
  2. Lane marking extraction based on a 2D image obtained from the 3D point cloud so as to obtain elevation values. The 2D image correlates with the 3D point cloud.
  3. Lane mapping: Lane markings in a local section are clustered and fitted into lane lines. Those point are clustered and refined according to the typology of the lane line belong to; continuous line, broken line, etc. To fill in the missing lanes caused by local occlusion, a global post-processing is adopted. In order to improve the robustness of lane mapping, textural saliency analysis is proposed using images obtained from MMS sensor. It is used to validate candidate lane lanes.

2.5 Conclusions

The methods exposed in this Chapter are conventional signal processing techniques which are not trained or learning from the data. Their parameters are set a priori in some sample data by manual observation of the results. There is no way to adapt the algorithms to new incoming data. For this reason, we approach the problem from a machine learning point of view, where a classification model is built up from the data available trying to minimize the detection error. This general approach allows to tune

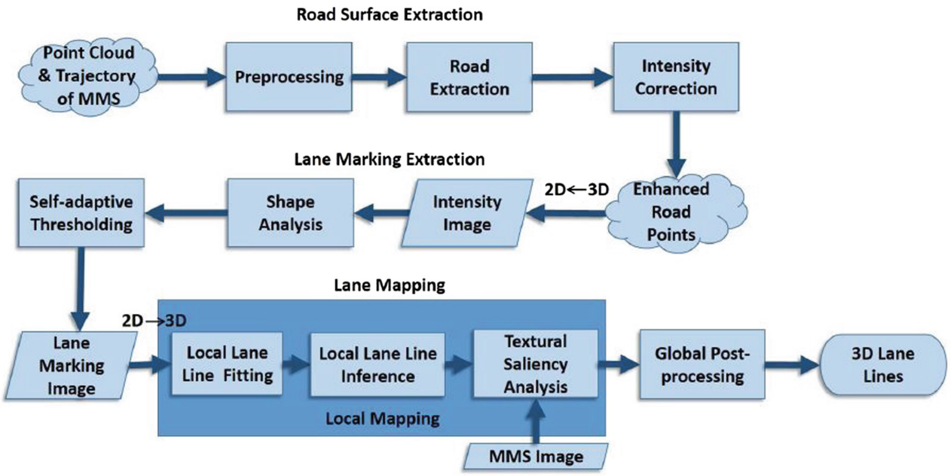


Figure 2.6: Point cloud and image fusion proposal [64]

the system to the data, and to have a method to carry out corrections or retuning of the classifier.

Chapter 3

Dataset for the experiments

In this chapter we introduce the actual datasets used for computational experiments. The scenarios will be divided according to the sensor used for data acquisition within the MMS system. Section 3.1 gives a short description of the mobile mapping system that has been used to capture the data. Section 3.2 describes the experimental dataset extracted from theRGB cameras. Section 3.3 describes the dataset consisting of images that have been composed from the intensity component of the LiDAR response. Section 3.4 describes the 3D LiDAR dataset without intensity information.

3.1 Mobile Mapping System

Mobile mapping is the integration of different methods of data capture mounted on a vehicle. The sensor used is the IP-S3 HD1 (Figure 3.1) product of Topcon, Japan. It is composed of a positioning system (wheel encoder, GPS/GNSS receiver, Inertial Measurement Unit), image sensor and a Lidar Sensor.

3.1.1 Positioning system

- **Wheel encoder:** Using an odometer we can calculate accurately the distance travelled by the vehicle.

- **GPS/GNSS receiver:** Receiver that collects a signal from the GPS costing of both GPS and GLONASS satellites, obtaining a vehicle coordinate at all times.

- **Inertial Measurement Unit**: An inertial measurement unit or IMU is an electronic device that measures and reports on the speed, orientation and gravitational forces of an apparatus, using a combination of accelerometers and gyroscopes. The IMU will calculate the current position within the trajectory, based on the speed, direction and time of the vehicle.

3.1.2 Image sensor

Five cameras pointing at regular arc intervals of the circumference, and a sixth one pointing up. Using the described configuration, the output is a panoramic image, a sum of the images obtained by the 6 cameras of the system, where the central view corresponds to the rear view of the car, so that the front view is split left and right of the panorama. At the bottom of the panorama, it is possible to appreciate the scaffolding supporting the sensors. Notice the changes in hue and saturation in the image due to the different sensitivities of the cameras. All images captured by the LadyBug are tagged spatially with coordinates provided by onboard GPS. The frequency used is 1HZ.

3.1.3 LiDAR system

This sensor is capable of measuring up to 700,000 points per second. During each turn, the 32 internal lasers cover the entire 360 degrees around the system, each from a slightly different viewing angle, minimizing the gaps in the cloud of points that arise from obstacles or dead angles. Points are also tagged spatially with coordinates. The frequency used for data capture is also 1 HZ.

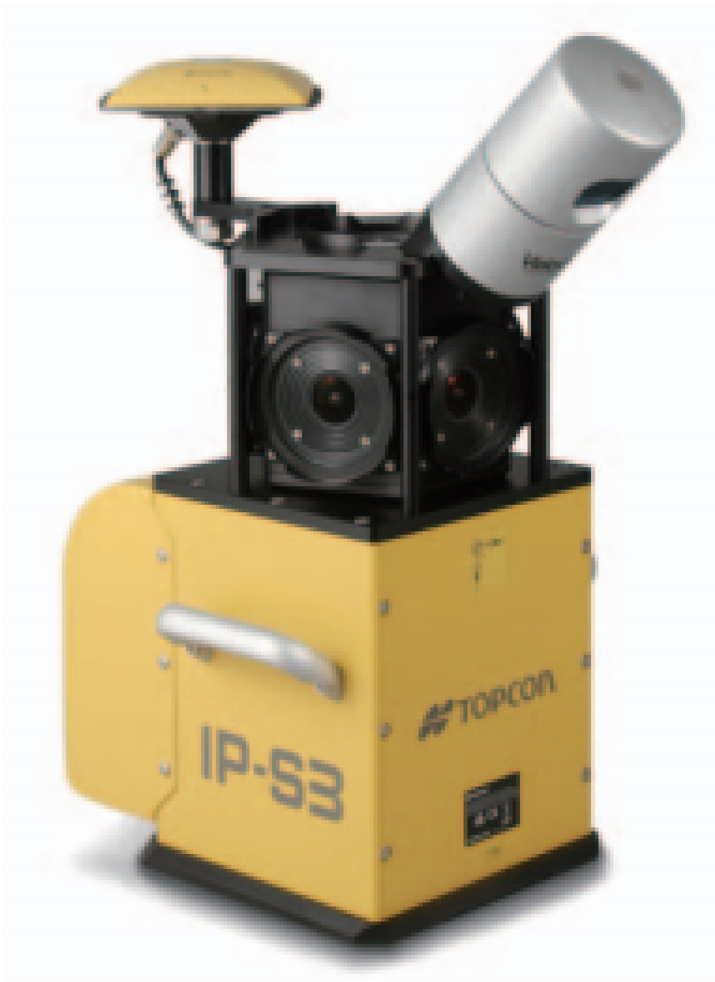


Figure 3.1: IP-S3 HD1 Mobile Mapping System

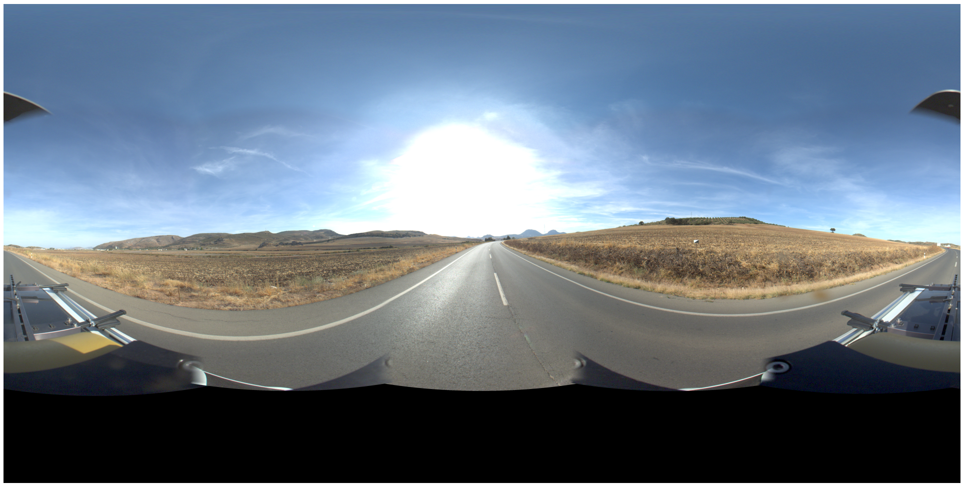


Figure 3.2: An example panorama image obtained from the ladybug sensor after stitching the 5 camera images, obtained by the composition of the views of the five cameras

Figure 3.2: An example panorama image obtained from the ladybug sensor after stitching the 5

3.2 Image sensor based data

In this section we will explain the dataset composed from the data obtained by the image sensor of the MMS system. The vehicle equipped with the MMS system made several data recording trips collecting a sequence of panorama images such as the one shown in Figure 3.2with a frequency of 1 per second, along with the LiDAR readings. We have selected 150 panorama images from different trips to compose the training and testing dataset. We extract the central region of the panorama images as the actual images in the experimental dataset. The left column of Figure 3.3 shows some of the images in the dataset (each of size 2841×489 × 3). We have hand delineated the ground truth of the images to provide the pixel labelling for the AL training sample selection and for the validation process computed at each step of the AL training dataset increment. The right column of Figure 3.3 shows the ground truth of the images in the left column.

The panoramic images are the result of the stitching of the six images (5 in the horizontal plane and one in the vertical plane) obtained as shown in

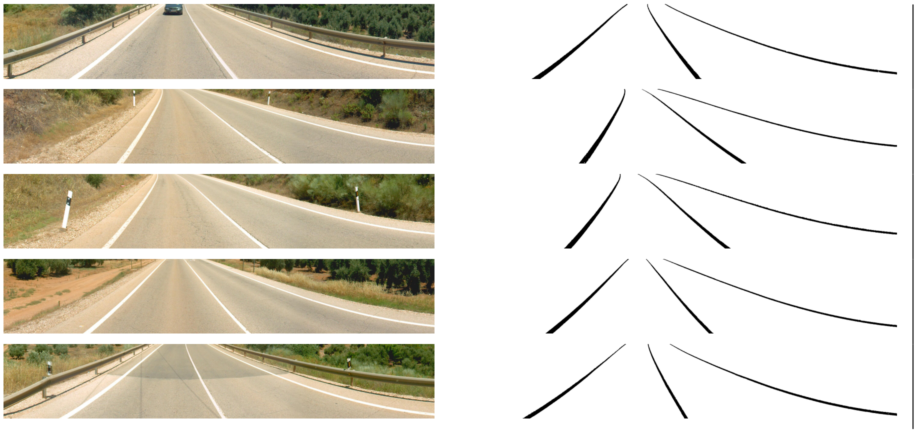
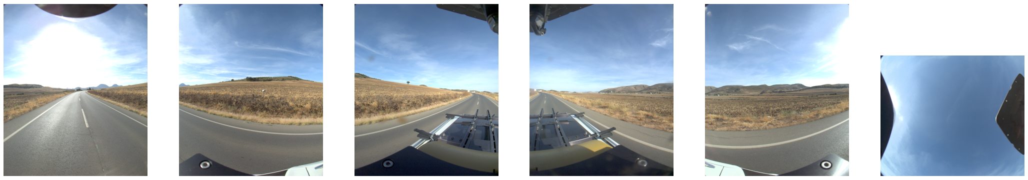


Figure 3.3: Some of the experimental images (left column) and their corresponding manually delineated ground truth (right column) white=background, black=road signs.



Camera 0 Camera 1 Camera 2 Camera 3 Camera 4 Camera 5

Figure 3.4: Individual images captured by the different cameras of the sensor. Cameras 0 to 4 correspond to the individual plane. Camera 5 corresponds to the zenith plane.

Figure 3.4. The image is only representative, i.e. due to the geometric and radiometric distortions it undergoes when it is generated, it is not possible to calculate the terrain coordinates directly from the pixel coordinates. Such calculation can be done with proprietary software that possess the complete calibration parameters of the entire system. In this case, as it is not a direct a direct measure and is not the subject of the Thesis, we do not pursue this issue any further. The coordinates to be used to reference the image will be only the pixel coordinates.

For the training and validation process we pool together the pixels from all selected images, their texture features obtained from the bank of Gabor

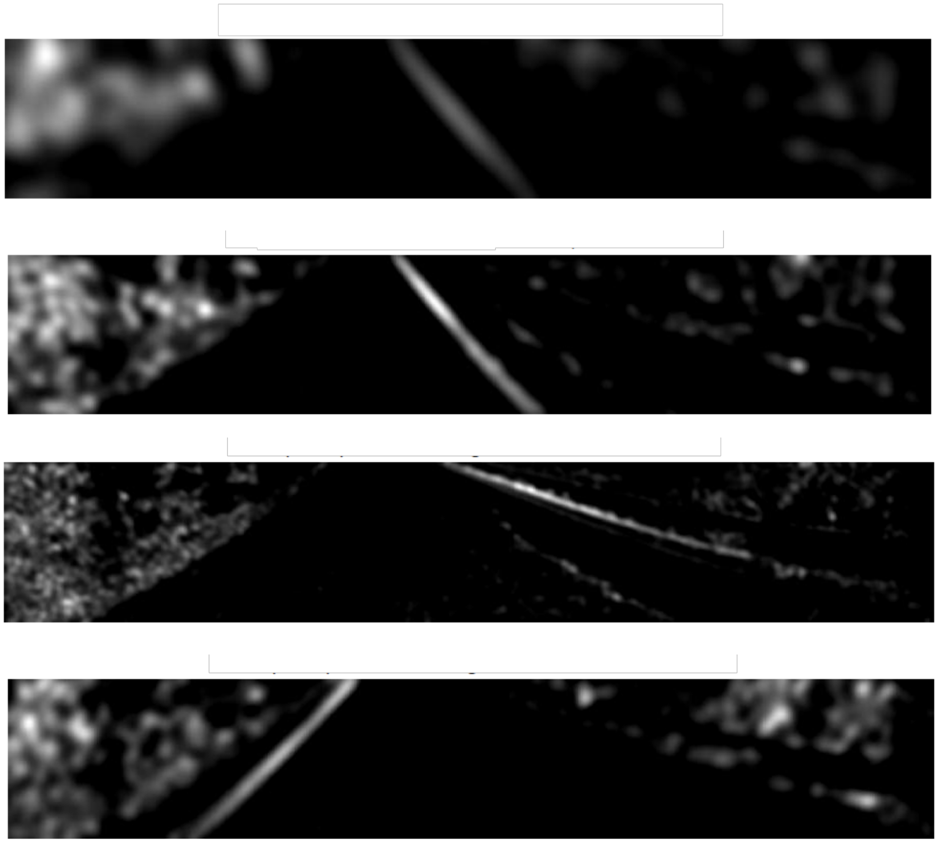


Figure 3.5: Example of filtered images with Gabor filters of various orientation and scales.

filters (that will be introduced formally in Chapter 4) that we can see in Figure 3.5, and their labels. The Gabor filter bank comprises 36 orientations (from $0^{\rm o}$ to $180^{\rm o}$ in steps of $5^{\rm o}$) and 6 wavelength values set following a linear rule $F = w_0 2^k$ with $k = 0, \ldots, 5$, where $w_0 = {}^4\!/_{2^{0.5}}$, hence we have 196 Gabor features per pixel. The dataset contains over 208 million pixels, each one is associated to a feature vector of dimension 198. The actual class imbalance ratio of the dataset is $IR = {}^1\!/_{39}$, i.e. we have 39 background pixels *per* each road signal pixel. In other words, the target minority class accounts for 2.45% of the dataset, the remaining 97.55% corresponds to the background, including the road and the environment.

3.3 LiDAR intensity image dataset

Each car trajectory generates a point cloud of more than 100 million LiDAR points. With proprietary software, more specifically the MMS system's own software, the different paths are extracted in divided point clouds of 10M. Thanks to the positioning system of the MMS system, the trajectory is georeferenced. No control-points are taken, so it is not possible to speak of relative positioning accuracy, but it is estimated that this accuracy is around 30-40 centimetres in real time (pre-processing). Through post-processing with the MMS system's own software, we achieve an average relative positioning accuracy of less than 10 centimetres. The post-processing consists of identifying the permanent GNSS stations closest to the route. These stations will provide us with corrections, which are stored in a file type called RINEX [30], giving us a file of corrections every 30s or every 1s. In our case we perform the post-processing tasks with a 1s RINEX file, applying these corrections to the measurements taken during the trajectory.

On the divided and post-processed point clouds, a dataset of 10 point clouds has been randomly chosen from which, using the intensity information, their respective raster images of intensities have been generated. These images, in addition to providing information on the reflectivity response, provide us with a georeferenced image, i.e. each pixel of the image has coordinates with a UTM coordinate system and an ETRS89 reference system.

The left column of Figure 3.6 shows some of the intensity images from in the dataset . We have hand delineated the ground truth of the images to provide the pixel labelling for the AL training sample selection and for the validation process computed at each step of the AL training dataset increment. The right column of Figure 3.6 shows the ground truth of the images in the left column.

For the training and validation process we pool together the pixels from all selected images, their texture features obtained from the bank of Gabor filters (Image 3.7 , and their labels. The Gabor filter bank comprises 36 orientations (from 0^{o} to 180^{o} in steps of 5^{o}) and 6 wavelength values set following a linear rule $F = w_0 2^k$ with $k = 0, \ldots, 5$, where $w_0 = {}^{4}\!/_{2^{0.5}}$, hence

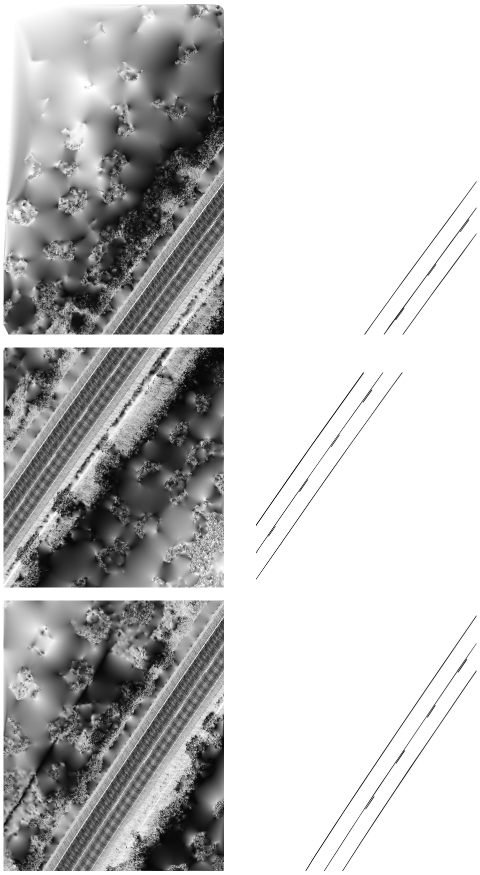


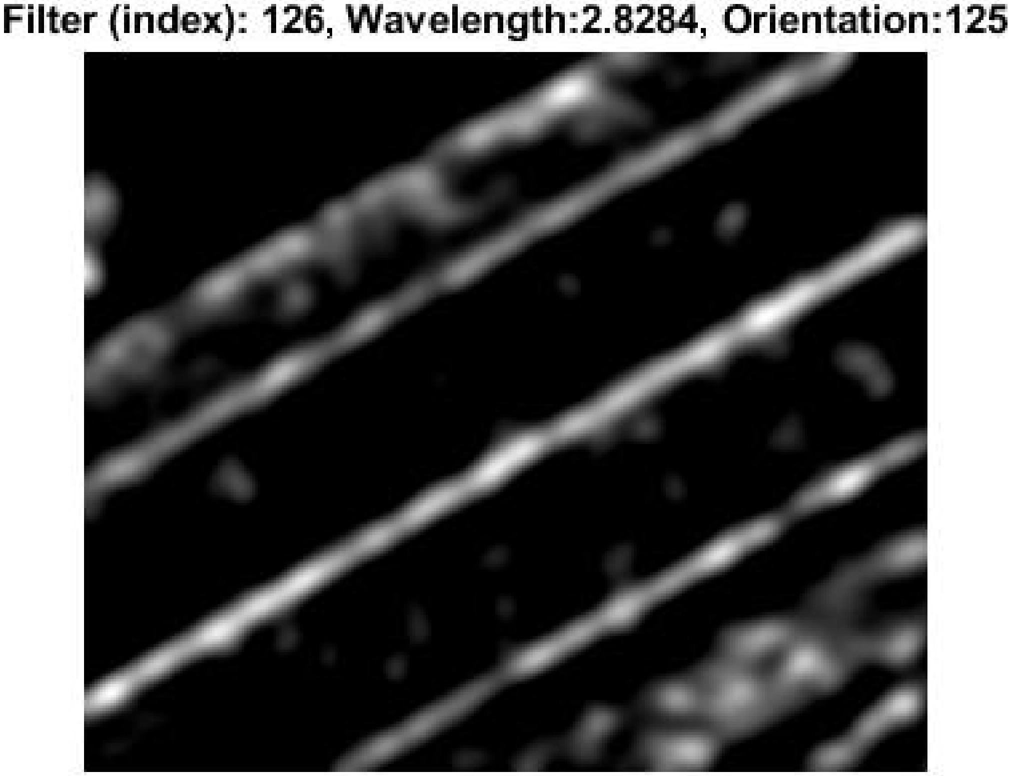
Figure 3.6: Left: one of the experimental images, Right: its corresponding manually delineated ground truth (white is the background)

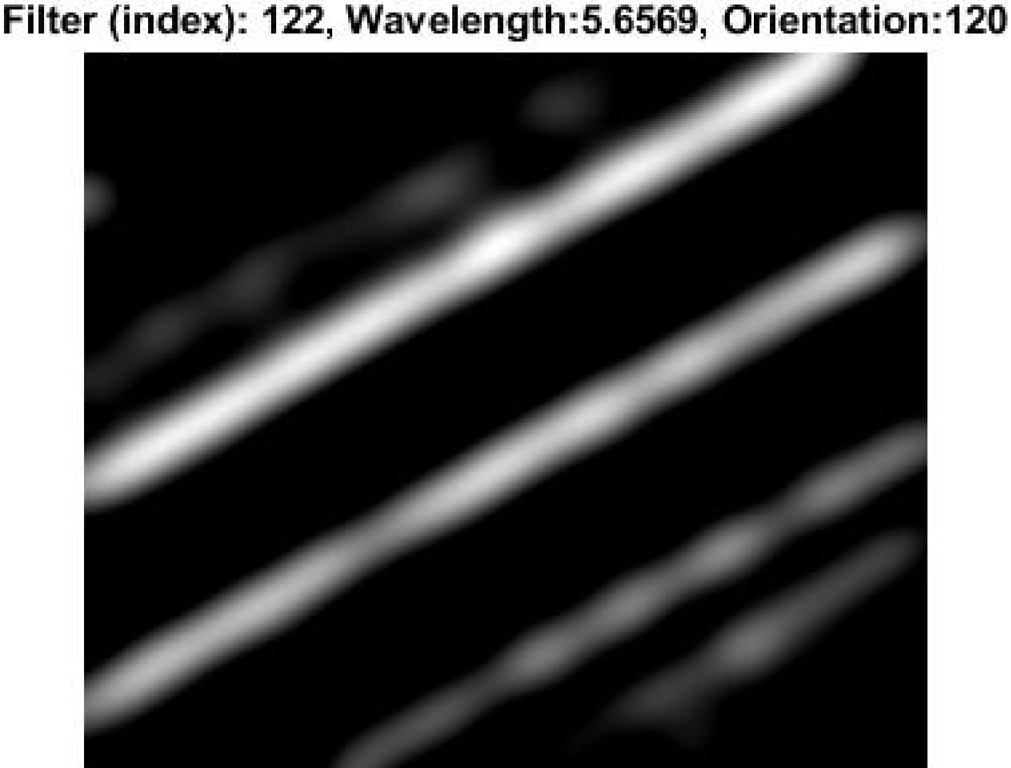
we have 196 Gabor features per pixel.

The actual class imbalance ratio of the dataset is $IR = 1/117$, i.e. we have 117 background pixels *per* each road signal pixel. In other words, the target minority class accounts for 0.85% of the dataset, the remaining 99.15% corresponds to the background, including the road and the environment.

3.4 LiDAR point clouds dataset

The configuration of the MMS system is as discussed in section **??**. Furthermore, the dataset used is the same as the one used in section **??** In order to provide the point labelling for the AL training sample selection and for the validation process computed at each step of the AL training dataset increment, we have hand extract all points that correspond to lane marks obtaining a ground truth for each of the point clouds to be studied. Figure 3.8 shows some Lidar Point Cloud and respective ground truth images. For the training and validation process we pool together the the points from all selected point clouds with their intensity values and their normal and deep degree, previosly computed, and their labels. The actual class imbalance ratio of the dataset is $IR = 1/28$, i.e. we have 28 background points *per* each road signal point. In other words, the target minority class accounts for 3.7% of the dataset, the remaining 96.3% corresponds to the background, including the road and the environment.


Filter (index): 126, Wavelength:2.8284, Orientation:125


Filter (index): 122, Wavelength:5.6569, Orientation:120

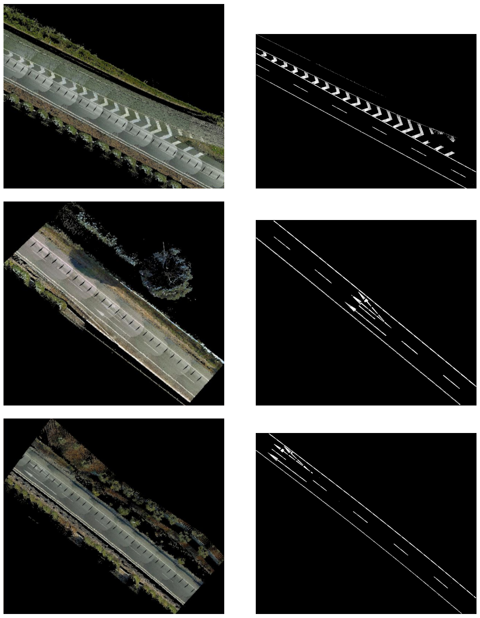


Figure 3.8: Left: one of the experimental LiDAR Point Cloud, Right: its corresponding manually extracted ground truth.

Chapter 4

METHODS

Road landmark inventory is a flourishing industry around the world, as the traffic becomes denser and the drivers must rely on a well-maintained infrastructure. Specifically, horizontal signals and lane landmarks, such as lines, arrows or other drawings on the asphalt, are of great public concern. Several commercial sensors are available which include optical cameras allowing to build panoramic images around the vehicle used for road inspection and LiDAR sensors, being able to collect geo-referenced data with additional information to planimetric and altimetric coordinates, such as intensity, number of returns, GPS time, etc. The landmark detection is posed as a two-class classification problem that may be solved by some approaches, such as random forest (RF) and ensembles of extreme learning machines (V-ELM). Besides model parameter selection, a central problem is the construction of the labeled training dataset due to human labor cost and the highly uncontrolled conditions of image capture. We have applied an open-ended Active Learning (AL) approach with a human operator in the loop who can start the AL process when detection quality is degraded by the change in imaging conditions in order to achieve adaptation to the new imaging conditions.

The content of the Chapter is as follows: Section 4.1 presents the schematics of the experimental setups carried out in this Thesis, Section 4.2 gives the definitions of the classification performance measures employed to assess the quality of the methods. Section 4.3 describes the main classification training methods applied along the Thesis. Section 4.4 describes the issues related to active learning strategies. Section 4.5 describes the construction of the raster image from the LiDAR point clouds. Section 4.6 describes the features extracted from LiDAR point clouds for classification. Finally, Section 4.7 gives a short review of Gabor features also used for texture based classification.

4.1 Experimental setups

We report performance results over a collection of road images in order to assess the most adequate classifier and parameter settings. Chapter 3 contains the description of the different experimental datasets extracted from the MMS system travels. The experimental setup for validation is illustrated in figure 4.1. For the classification of the points of interest, whatever the methodology, the results will be explored through Random Forest (RF) classifiers. The experimental setup for validation using different methodologies is illustrated in 4.2 and 4.3.

4.2 Performance measures

We report the Sensitivity (SEN), Specificity (SP), Accuracy (AC), Positive Predictive Value (PPV), and Area Under the receiver operating Curve (AUC) of the pixel-wise classification of the entire images using the classifiers built over the selected training datasets at the end of the AL process.TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively.

Sensitivity *(SEN)*: Metric used to detect positive instances (True positives). A model with high sensitivity will have few false negatives, which means that it is missing a few of the positive instances. In other words, sensitivity measures the ability of a model to correctly identify positive samples. The sum of sensitivity (true positive rate) and false negative rate would be 1. The higher the true positive rate,

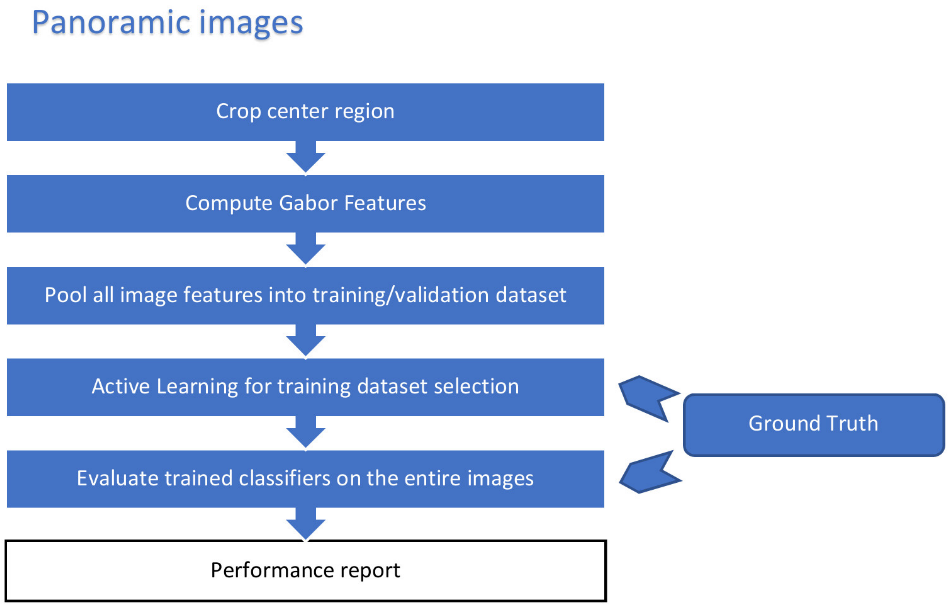

Panoramic images
Crop center region
Compute Gabor Features
Pool all image features into training/validation dataset
Active Learning for training dataset selection
Evaluate trained classifiers on the entire images
Performance report
Ground Truth

Figure 4.1: Pipeline of the experimental setup for the Active Learning segmentation process with panoramic images

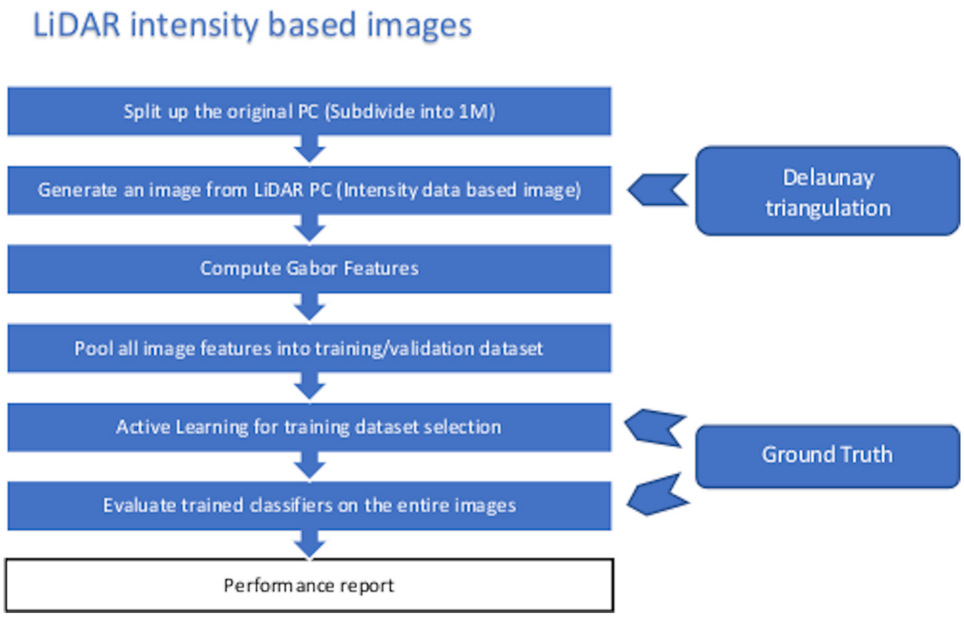

LiDAR intensity based images
Split up the original PC (Subdivide into 1M)
Generate an image from LiDAR PC (Intensity data based image)
Delaunay triangulation
Compute Gabor Features
Pool all image features into training/validation dataset
Active Learning for training dataset selection
Ground Truth
Evaluate trained classifiers on the entire images
Performance report

Figure 4.2: Pipeline of the experimental setup for the Active Learning segmentation process with LiDAR intensity based images

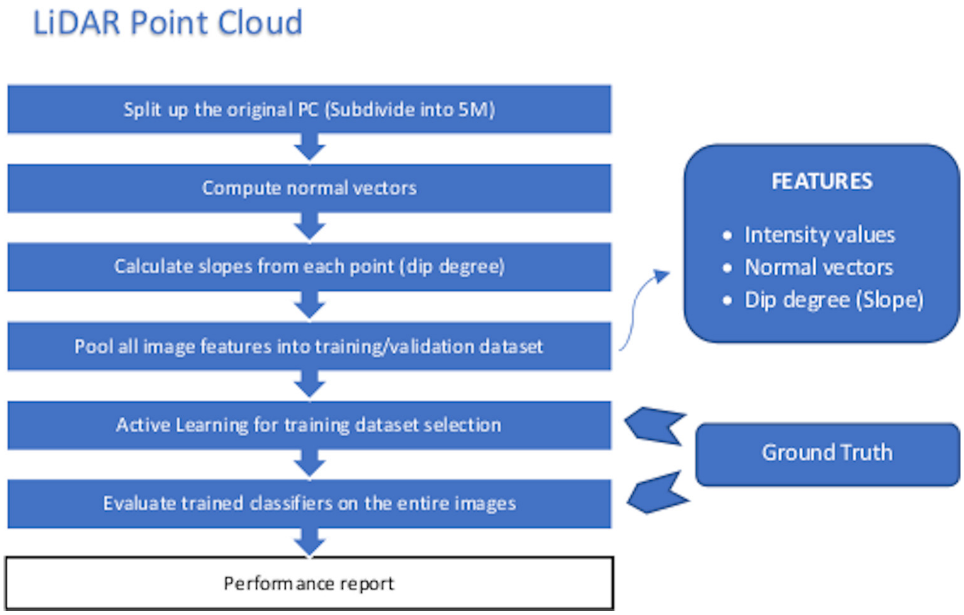

LiDAR Point Cloud
Split up the original PC (Subdivide into 5M)
Compute normal vectors
Calculate slopes from each point (dip degree)
Pool all image features into training/validation dataset
Active Learning for training dataset selection
Evaluate trained classifiers on the entire images
Performance report
FEATURES
Intensity values
Normal vectors
Dip degree (Slope)
Ground Truth

Figure 4.3: Pipeline of the experimental setup for the Active Learning segmentation process with LiDAR Point Clouds

the better the model is in identifying the positive cases in the correct manner. Mathematically, sensitivity is calculated as $SEN = \frac{TP}{TP+FN}$.

Specificity *(SP)*: It measures the proportion of true negatives that are correctly identified by the model.This implies that there will be another proportion of actual negative which got predicted as positive and could be termed as false positives. This proportion could also be called a True Negative Rate (TNR). High specificity means that the model is correctly identifying most of the negative results, while a low specificity means that the model is mislabelling a lot of negative results as positive. Mathematically, specificity is calculated as $SP = \frac{TN}{TN+FN}$.

Positive *Predictive Value (PPV):* It is defined as the proportion of positively classified cases that were truly positive. We report the PPV as $PPV = \frac{TP}{TP+FP}$.

Accuracy *(AC)*: It is used to quantify the number of correctly predicted data points out of all the data points. Formally is defined as $AC = \frac{TP+TN}{N}$ where $N = TP + TN + FP + FN$.

Area *Under the receiver operating Curve (AUC):* It is computed by approximate integration of the ROC over the classifier decision threshold. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:True Positive Rate False Positive Rate .

The most valuable metrics are the SEN, PPV, and AUC because they measure the recognition of the minority class and the overall balance of the recognition of both classes. Due to the strong class imbalance of the dataset AC is usually biased towards the recognition of the majority class.

The pool of pixels used for the selection of the training dataset is composed of pixels of all labelled images. The sample selection procedure tries to have pixel representatives from all images, in order to avoid over-training on one image. To this end, at each AL iteration we compute the classification uncertainty of all pixels from all images. However we do not ensure that the selection picks the same number of pixels from each image to be added to the training dataset. The separation of training and test data for validation is ensured as far as we report the performance measures over the pixels not included in the training set. Classifier training using AL never uses the labelling information of data not included in the training dataset [63].

4.3 Classification Methods

4.3.1 Random Forest Classifiers

Random Forest (RF) algorithm is a classifier [17] that encompasses bagging [16] and random decision forests [18], whose performance has been demonstrated in a variety of applications [19]. RF became popular due to its simplicity of training and tuning while offering a competitive performance to other machine learning approaches, such as support vector machines. Consider a RF as a collection of decision tree predictors, built so that they are as much decorrelated as possible, denoted:

$$\{h(\mathbf{x};\psi_t);t=1,...,T\}\,, \tag{4.1}$$

where x is a d-dimensional random sample of random vector X, ψ_t are independent identically distributed random vectors modelling the stochastic nature of the tree building process. Each tree $h(x;\psi_t)$; casts a unit vote in order to decide the class assignment of x. RF appear to capture complex correlations in data, and are proposed [17] to be resistant to both over-fitting of data when individual trees are very deep and not pruned, and under-fitting when individual trees are too shallow. Pruning is applied to mitigate potential over-fitting that might happen in a single tree. Overfitting is reduced in RF it is do by two means:

- A bootstrapped training: A statistical concept, that is a resampling method used to stimulate samples out of a data set using the replacement technique. The process of bootstrapping allows one to infer data about the population, derive standard errors, and ensure that data is tested efficiently. This technique involves repeatedly sampling a dataset with random replacement. A statistical test that falls under the category of resampling methods, this method ensures that the statistics evaluated are accurate and unbiased as much as possible.

- The multitude of random trees using random features allows individual trees to be strong classifiers but not so correlated with each other to induce overfitting.

4.3.1.1 RF construction

Given a dataset of N samples, a bootstrapped training dataset is used to grow a tree $h(x;\psi_t)$ on a randomly selected subset of data dimensions $\hat{d}$ such that $\hat{d}$<< d. Decision tree growing recursively picks the best data split of each node based on these information measure of each dimension. In RF pruning is not required. The RF training process picks randomly the dimension and the dataset bootstrapping according to independent identically distributed random vectors ψ_t. This randomness is the source of RF individual tree diversity ensuring the decorrelation of their outputs. Classification of a new input x is done by majority voting over the responses of the trees in the RF $C_u(x)$. The critical parameters of the RF classifier for the experiments reported below are the number of trees in the forest, the dimension of the random subspace, and the maximum tree depth. We report experiments assessing the effect of these parameters in our specific study. The 4.1 shows the process of Random Forest.

4.3.2 Ensemble of Extreme Learning Machines

Extreme Learning Machine (ELM) [39] was proposed for fast and efficient training of Single-Hidden Layer Feedfodward neural Networks (SLFN) It is composed of the random generation of the input-to-hidden layers weights followed by least squares estimation of the hidden-to-output layer weights. Random weight generation allows to solve the non-linear training problem as a linear problem, gaining in speed at the cost of some instability that

Algorithm 4.1 Random Forest algorithm for Classification.

```
Imputs
–Initial Dataset $N$ samples.
–Initial training set $X^t = \{\mathrm{x}_i,\ y_i\}_{i=1}^{l} (X \in \mathcal{X},\ t = 1)$.
–Considered RF: $h(\mathbf{x}; \psi_t); t = 1, ..., T$.
–Given a dataset of N samples, a bootstrapped training dataset is used to grow a tree $h(\mathbf{x}; \psi_t)$ on a randomly selected subset of data dimensions $\hat{d}$
–«d
for each tree $h(\mathbf{x}; \psi_t)$ do
    casts a unit vote in order to decide the class assignment of $x$
end for
Classification of a new input $x$ is done by majority voting over the responses of the trees in the RF $\mathrm{C}_u(x)$
```

has been shown empirically to be affordable. In general terms, a supervised classifier is a map from input feature space into a target value space, for the SFLN this map has the following form:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_{\mathbf{i}} \cdot h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}, \tag{4.2}$$

Where $\beta = [\beta_1, \ldots, \beta_L]^T$ is the matrix composed of the weights of the connections hidden-to-output units, and the transformation $h(x)$ is the ELM non-linear transformation from input to hidden space. The hidden units of ELM may be any piecewise continuous function that satisfies simple universal approximation conditions [38]. Given a sample of feature vectors and corresponding labels $\{(x_i, y_i)\}_{i=1}^{N}$, the estimation of the labels given the input features can be written in matrix notation as

$$\mathbf{H}\boldsymbol{\beta} = \hat{\mathbf{Y}}, \tag{4.3}$$

Where H is the matrix of responses of the hidden units to the input features in the sample, and $\widehat{y}$is the achieved approximate value of the labels of the samples.

The process of training an SLFN following ELM approach has two stages:

- Feature mapping implement by random sampling of the hidden units activation in response to the input features in order to build the feature kernel matrix H.

- Least square resolution of the following minimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{L\times m}} \left\|\mathbf{H}\boldsymbol{\beta}-\mathbf{Y}\right\|^2, \tag{4.4}$$

Whose optimal solution is given by $\beta = H^{\dagger}L$, where $H^{\dagger}$is the Moore-Penrose generalized inverse of H. Note that the definition of ELM allows naturally multivariate SFLN.

Neural network classifiers usually suffer from over-training which might degrade the generalization performance. During the training phase, all training samples are categorized into several classes by classifier and the learning error is used to evaluate the efficiency of training. Minimum training error is expected, but it cannot guarantee good classification results on unseen data. It is shown that combining a number of neural networks could solve the problem [35, 70].

ELMs can be combined very efficiently in ensembles which can be homogeneous or heterogeneous [8]. Applications on the prediction of readmissions in hospital environments [4], remote sensing image processing [7, 5, 6, 9] have already been reported. The simplest ensemble of ELM is the Voting ELM (V- ELM) [21], consisting of a collection of ELMs each trained on a bootstrap of the training dataset. Compared with the original ELM algorithm, the proposed V-ELM is able not only to enhance the classification performance and reduce the number of misclassified samples, but also to lower the variance among different realizations. Every SFLN has the same number of hidden neurons, and the decision on the class is performed by majority voting over the responses of the SFLN in the ensemble. Examples of applications of V-ELM are hyperspectral image classification [9], remote sensing data classification [34] and natural gas reservoir characterization [2], and wastewater quality index modelling [29]. It has been proved that accuracy converges to perfect classification as the size of the ensemble grows under the mild assumption that individual classifiers have performance better than random choice [21]. The V-ELM has spurred some variations in the literature, such as the use of soft-class dependent voting schemes [22] trying to increase model reliability. We apply the basic scheme in our work; Given a learning set L consisting of samples $(xn, yn), n = 1, 2, ..., N$, where $y_n$is the class label. It is assumed[46] that x is the input and y is predicted by $\varphi(x, L)$. In V-ELM, the aim is to better predict y using multiple ELMs than a single one. Suppose that $\varphi(x, L)$ predicts a class label $j \in 1, 2, ..., C$ and the prediction of kth classifier is $D_{k,j} \in 0, 1$ where $k = 1, 2, 3, ...k$, the ensemble can be defined as

$$\varphi(x, L) = \arg \underset{j=1}{\overset{c}{max}} \sum_{k=1}^{k} D_{k,j} \tag{4.5}$$

The voting is pure majority version which means that the output is the value with highest number of votes whether or not the sum of votes exceeds half.

4.4 Active Learning

The motivation of the Active Learning approach [24, 25, 62] is to facilitate the task of data labelling for supervised classification. The two objectives pursued by this approach are to reduce as much as possible the manual effort of data sample labelling, and the selection of those data samples which are more informative towards the building a robust and efficient classifier. Data labelling is a costly process involving manual labour and some interactive data visualization tool that facilitates the labelling process. In some cases, data labelling involves quite complex field work, so minimizing the number of required labelled samples and ensuring that each labelled sample contributes significantly to the quality of the classifier is of paramount importance. Moreover, manual data labelling may introduce erroneous classifications, which further interfere and degrade the training process. Such kind of errors are less likely in small datasets. Additionally, in the case of highly class imbalanced datasets, the guided selection of balanced training datasets may enhance the results as we have found in our experiments reported later. Let us denote $X = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$ and $U = \{\mathbf{x}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^d$ the training set containing labelled samples, and the unlabelled samples in the *pool of candidates*, respectively. We have that $u \gg l, \mathbf{x}_i \in \mathbb{R}^d$, and $y_i \in \{1, \ldots, N\}$. Active Learning is an iterative Algorithm. At iteration t, the algorithm selects q candidates from U^t to be added to the current training set X^t, aiming at maximal gain in performance of the classifier trained with the incrementing training dataset, while reducing the classification model uncertainty. An oracle provides the labels $\{y_m\}_{m=1}^{q}$ to the selected samples $S^t = \{\mathbf{x}_m\}_{m=1}^{q} \subset U$. The oracle can be a human carrying out computational experiences, such as we do in this chapter. The current training set is increased with the candidates selected from $(U^{t+1} = U^t \backslash S^t)$. The process sops when some criterion is met, such as the accuracy over a pre-set threshold θ_{max}. Algorithm 4.2 summarizes the Active Learning process.

Algorithm 4.2 Active learning general algorithm

Imputs

–Initial training set $X^t = \{\mathrm{x}_i,\ y_i\}_{i=1}^{l}(X \in \mathcal{X},\ t = 1)$.

–Pool of candidates $U^t = \{\mathrm{x}_i\}_{i=l+1}^{l+u}(U \in \mathcal{X},\ t = 1)$.

–Number of voxels q to add at each iteration (defining the batch of selected voxels S).

```
repeat
    Train a classifier with current training set $X^t$
    for each candidate in $U^t$ do
        Evaluate a user-defined heuristic
    end for
    Rank the candidates in $U^t$ according to the score of the heuristic
    Select the $q$ most interesting voxels $S^t = \{\mathrm{x}_k\}_{k=1}^{q}$
    The system assigns a label to the selected voxels $S^t = \{\mathrm{x}_k,\ y_k\}_{k=1}$
    Add the batch to the training set $X^{t+1} = X^t \cup S^t$
    Remove the batch from the pool of candidates $U^{t+1} = U^t \backslash S^t$
    $t = t + 1$
until $t > T$
```

4.4.1 Classification uncertainty in ensemble classifiers

RF and V-ELM classifiers implement committee approaches to the decision of the class corresponding to an unlabelled sample. This distributed cooperative decision can be used for classification uncertainty estimation. Given a committee of k base classifiers, then we have k labels for each candidate unlabelled sample$\mathbf{x}_i \in U$. The data sample class label is provided by the majority voting. Heuristically, the measure of the classification uncertainty of xi can be equated with the standard deviation $\sigma\left(\mathbf{x}_i\right)$ of the class labels generated by the committee. The pool of candidates can be ordered $U^* = \{\mathbf{x}_{j_i}\}_{i=l+1}^{l+u}$, according to $\sigma\left(\mathbf{x}_{j_i}\right) > \sigma\left(\mathbf{x}_{j_{i+1}}\right)$.The standard deviation query-by-bagging heuristic selection of samples to be added to the train set is stated as the following selection:

$$S^t = \{\mathbf{x}_{j_m}\}_{m=1}^{q} \tag{4.6}$$

The standard deviation of the ensemble label predictions is a powerful heuristic measure of classification uncertainty. A candidate sample that is classified equally by all ensemble members has a zero prediction standard deviation, hence its inclusion in the training set does not add any information. On the other hand, if a candidate has uniformly distributed responses from the ensemble, its standard deviation is maximal, hence it contributes maximal information when included in the training dataset.

4.4.2 Active learning for image segmentation

Image pixels are classified into the target and the background classes [50]. Target pixels correspond to the lane marks and other landmarks in the road. In a nutshell, an AL system returns to the user an image whose intensity value corresponds to the degree of uncertainty in the classification of the pixel. Upon this image, the user, in its role as the oracle will pick some of the pixels with greatest intensity, and will be assigning labels to them for their insertion in the training dataset. Then, a new instance of the classifier is trained [58]. The features of each pixel are the result of the application of a bank of Gabor filters, the pixel intensity and its coordinates. Though the feature vector dimensionality is relatively high, we do not carry out any feature selection procedure because we prefer to leave open the possibility that a certain orientation or scale may be meaningful in future images. The stream of images collected during commercial operation may contain images that are quite different from the ones used for training. Hence, the final commercial implementation will allow to restart the AL process when the human operator detects deviations from optimal segmentation. It is an open-end learning process with a human in the loop. The computational exploration reported in this chapter, does not resort to a human oracle. Instead, it is based on the hand delineated ground truth of a collection of images. Hence, the labelling process consists in the selection of the pixels with maximal uncertainty, applying random selection to solve ties.

4.4.3 Active learning for LiDAR point cloud classification

The point clouds are classified as well as the processed images into target and background classes. The points belonging to target correspond to horizontal road markings. Using AL techniques, a point cloud is returned classified into points corresponding to road landmarks and everything that does not belong to the target class. In order to achieve this classification, the normal to each point and its deep angle must first be calculated. In addition, the intensity response obtained by the LiDAR sensor will be taken into account. These features, as well as the coordinates of each point, are stored in a database. The computational exploration presented, as described with the image, does not require manual techniques, but is based on the ground truth manually segmented in a preprocess. Therefore, the labelling process consists of selecting those points with the highest uncertainty, applying random sampling to resolve ties.

4.4.4 Active Learning and class imbalance

In supervised classification, a dataset is imbalanced when the a priori probabilities of the classes are significantly different, i.e. there exists a minority (positive) class that is under-represented in the dataset in contrast to the majority (negative) class. Often we want to predict the minority class because it is related to the highest cost/reward events. Dataset class imbalance is a strong issue in machine learning because it induces very strong bias effect towards the majority class in classifier training by almost any modelling approach, therefore the accuracy performance metric is not adequate. Preferred performance metrics measure the predictive power on the minority class. The imbalance ratio (IR) is defined as the ratio of the number of data samples from the majority class to the number of data samples from the minority class. Conventional classifier performance deteriorates even with moderate imbalance ratios. Class imbalance effect has been also recognized in digital mapping applications. There are two basic approaches to deal with imbalanced data. The first is to manipulate the data, either by under-sampling the majority class [67] or by oversampling the minority class (SMOTE is the best known oversampling approach), achieving a balanced training dataset. Usually, using naive subsampling to produce balanced training datasets does not achieve competitive test performance results, so that some kind of modelling is needed to do it correctly. The second kind of approaches try to tailor the classifier model to cope with the imbalance data, such as applying conformal transformations to Support Vector Machines (SVM) [69]. Many approaches rely on ensemble classifiers, such as the Adaboost and its combinations/variations, or through the decomposition of multiple class imbalanced problem into one-to-one decision problems. Recently, AL approaches to deal with class imbalanced data are getting momentum, for example posing the active sample selection as a constrained optimization problem. However, this approach does not scale to large datasets, such as the one we are dealing with. We have carried out comparative experiments of balanced versus imbalanced class sample selection for the incremental additions to the training dataset. The imbalanced selection preserves the a priori distributions of the class through stratified sampling of the most uncertain samples. The balanced selection forces that the same number of samples of each class is added to the training set among, regardless of differences in the (order of magnitude of the) uncertainty measure.

We carried out the comparative on the computational experiments using RF classifiers, with positive results. On the basis of the RF comparative, we apply the balanced AL strategy to the V-ELM experiments (Only applies to panoramic images).

4.5 Raster image from LiDAR point cloud

In our case we generate the raster images from the different point clouds. The generation is done with proprietary software [3]. Point clouds captured with laser scanner (LiDAR) have a high information density, but their spatial distribution is not uniform since they depend on their distance from the measurement pole, i.e. where the scanner is located at the moment of the measurement. Thus, areas close to the scanner will have a high information density, and areas further away will have a lower information density. Point clouds can also come from a photogrammetric multi-correlation, with similar densities; but, due to the process of obtaining the cloud from the images, their homogeneity is greater than for LiDAR point clouds.

If it is desired to obtain a regular information model, i.e. a raster structure, an interpolation could be performed using various filtering operations such as lowest, highest, arithmetic mean, weighted, etc. points. This process is suitable when the structure is based on a cell size higher than the density present in the model. In this way, points will always be obtained for the measurement area.

However, if it is desired to obtain a raster image with a higher density than the measured one, the process does not solve what happens in those areas where no points have been found. This is usually the case when obtaining a georeferenced image of the point cloud, in which the pixel information is some characteristic of the measured points: their intensity, colour, altitude, classification value, etc. In this case, the image would present areas where there is no information because there are no points nearby.

Therefore, to avoid this situation, a continuous surface must first be calculated that relates all the points measured, avoiding the generation of gaps in areas where the density of points is lower due to their distance from the

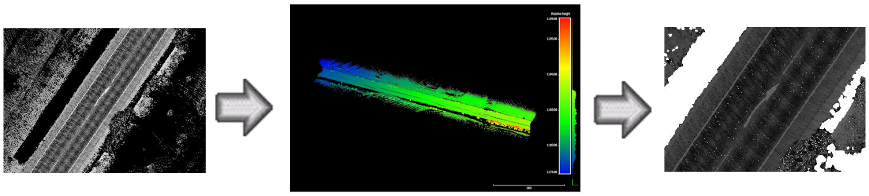


Figure 4.4: Orthoimage generation process

measurement pole. This surface can be obtained from a triangulation algorithm, such as Delaunay triangulation.The algorithm of this triangulation is based on the establishment of some points outside the data point cloud and external to it (fictitious points), marking the limits of action. These points will generate triangles with the points outside the cloud on the four sides (north, south, east and west), in such a way that a total surface is generated that is always convex (convex hull). Using this information, points that are connected with these fictitious points can be determinate and use the sides of the triangles generated with these for the formation of the outer boundary of the point cloud. The result will be a continuous surface divided by irregular triangles relating neighbouring points. In addition, a maximum distance can be established, which would avoid generating triangles that join points that are far apart in reality.

From this continuous surface, a raster structure can be interpolated, ensuring that no cells without information are generated. The process takes more computational time, but ensures a continuous image within the measured area.The process is fully automatic and returns an orthorectified product with no distortions, neither geometric nor radiometric, as well as a georeferenced image in space. The 4.4 shows the process of how from a point cloud we obtain an orthoimage from the digital elevation model (DEM).

4.6 LiDAR Point Cloud features

In a LiDAR file a number of attributes are stored along with each position value (X, Y, Z) for each recorded laser pulse; intensity, return number,

number of returns, point classification values, points that are on the edge of the flight line, RGB values (red, green and blue), GPS time, scan angle and scan direction. For our study we are only interested in the X, Y, Z information, plus the intensity, which represents the response of the laser when bouncing against an object. Depending on the type of scanner used in the point cloud capture, this parameter will be recorded as an 8-bit or 16-bit field. In the former case, it will appear as a value between 1 and 256 within the return attributes, in the latter case, it will be values between 1 and 65536. In order to have a LiDAR point Cloud features to train our model, in addition to the coordinates of each point and its intensity value, we will estimate the normals to the plane of the point cloud and calculate the slopes.

4.6.1 Surface Normal estimation in a Point Cloud

To calculate the normal vector to a surface, we first approximated the surface of the point cloud to a plane. This vector is perpendicular to the surface at a given point. When normals are considered on closed surfaces, the inward-pointing normal (pointing towards the interior of the surface) and outward-pointing normal are usually distinguished. The unit vector obtained by normalizing the normal vector (i.e., dividing a non-zero normal vector by its vector norm) is the unit normal vector, often known simply as the *unit normal.*

The normal vector at a point (x_0, y_0) on a surface $z = f(x, y)$ is given by

$$N = \begin{bmatrix} f_x(x_0, y_0) \\ f_y(x_0, y_0) \\ -1 \end{bmatrix}, \tag{4.7}$$

A normal vector to plane specified by

$$f(x, y, z) = ax + by + cz + d = 0 \tag{4.8}$$

is given by

$$N = \nabla f = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{4.9}$$

where ∇f denotes the gradient. The equation of a plane with normal vector n=(a,b,c) passing through the point (x_0,y_0,z_0) is given by

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = a(x - c) + b(y - y_0) + c(z - z_0) = 0. \tag{4.10}$$

For a plane curve, the unit normal vector can be defined by

$$\hat{N} \equiv \frac{d\hat{T}}{d\phi}, \tag{4.11}$$

where $\hat{T}$is the unit tangent vector and $\phi$is the polar angle. Give a unit tangent

$$\hat{T} \equiv u_1\hat{x} + u_2\hat{y} \tag{4.12}$$

with $\sqrt{u_1^2 + u_2^2} = 1$, the normal is

$$\hat{N} \equiv -u_1\hat{x} + u_1\hat{y} \tag{4.13}$$

For a plane curve given parametrically, the normal vector relative to the point $(f(t), g(t))$ is given by

$$x(t) = -\frac{g'}{\sqrt{f'^2 + f'^2}} \tag{4.14}$$

$$y(t) = \frac{f'}{\sqrt{f'^2 + f'^2}} \tag{4.15}$$

The actual place for the normal to the curve must be displaced by $(f(t), g(t))$. For a curve in space, the normal unit vector is given by:

$$\hat{N} \equiv \frac{\frac{d\hat{T}}{ds}}{\left|\frac{d\hat{T}}{ds}\right|} \tag{4.16}$$

$$= \frac{\frac{d\hat{T}}{dt}}{\left|\frac{d\hat{T}}{dt}\right|} \tag{4.17}$$

$$= \frac{1}{k}\frac{d\hat{T}}{ds}, \tag{4.18}$$

where $\hat{T}$ is the tangent vector, s is the arc length, and k is the curvature. It is also given by:

$$\hat{N} = \hat{B} \times \hat{T}, \tag{4.19}$$

where $\hat{B}$ is the binormal vector [1]

For a surface with parametrization $x(u, v)$, the normal vector is given by

$$N = \frac{\partial x}{\partial u} \times \frac{\partial x}{\partial v}, \tag{4.20}$$

Given a three-dimensional surface defined implicitly by $F(x, y, z) = 0$,

$$\hat{n} = \frac{\nabla F}{\sqrt{F_x^2 + F_y^2 + F_z^2}}. \tag{4.21}$$

The 4.5 shows a graphical representation of the calculation of the normal vectors of a small area of one of the point clouds under study.

4.6.2 Point Cloud Slopes computation

Another important data needed to train the model are the slopes to the planes of the calculated normals that will be used as another parameter to predict the points classified as road lane marks. Sometimes, elements that we do not want to classify as horizontal road lane can have the same response in LiDAR intensity values. This is the case shown in 4.6, in which we obtain the same intensity values for both the gate and the line marking the lane delimitation. The slope is calculated from the horizontal, expressing

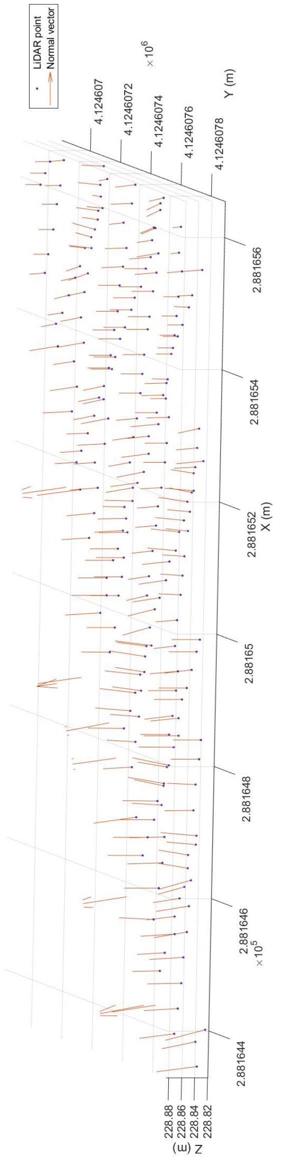
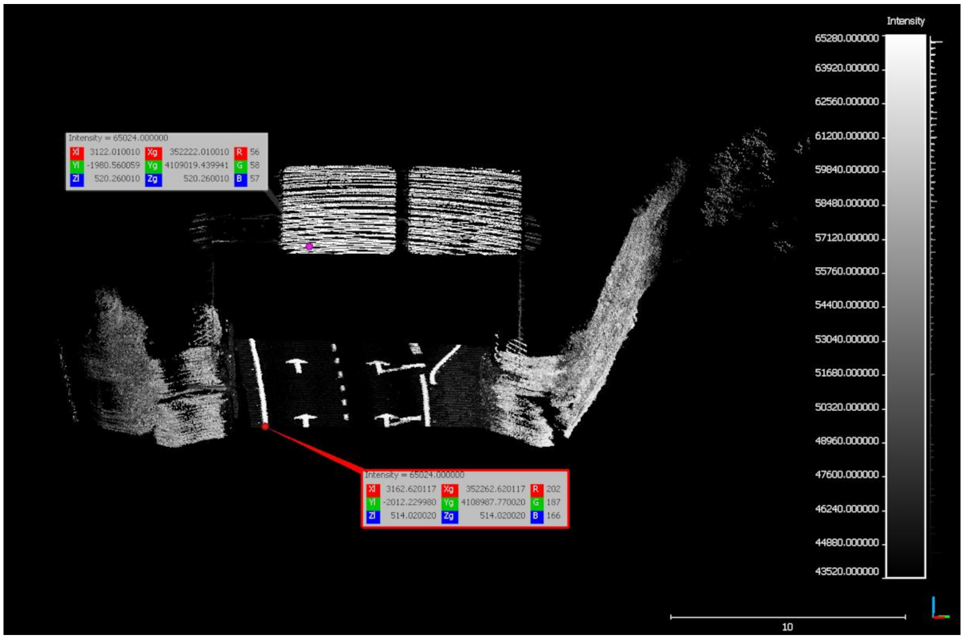


Figure 4.5: Graphical representation of the calculation of the normal vectors



```
Intensity = 65024.000000
Xl   3122.010010  Xg   352222.010010  R  56
Yl  -1980.560059  Yg  4109019.439941  G  58
Zl    520.260010  Zg      520.260010  B  57
```

```
Intensity = 65024.000000
Xl   3162.620117  Xg   352262.620117  R  202
Yl  -2012.229980  Yg  4108987.770020  G  187
Zl    514.020020  Zg      514.020020  B  166
```

Figure 4.6: Same intensity values between different elements.

the angle in positive values. Thus, all values greater than approximately 5 sexagesimal degrees should be discarded when predicting the classification.

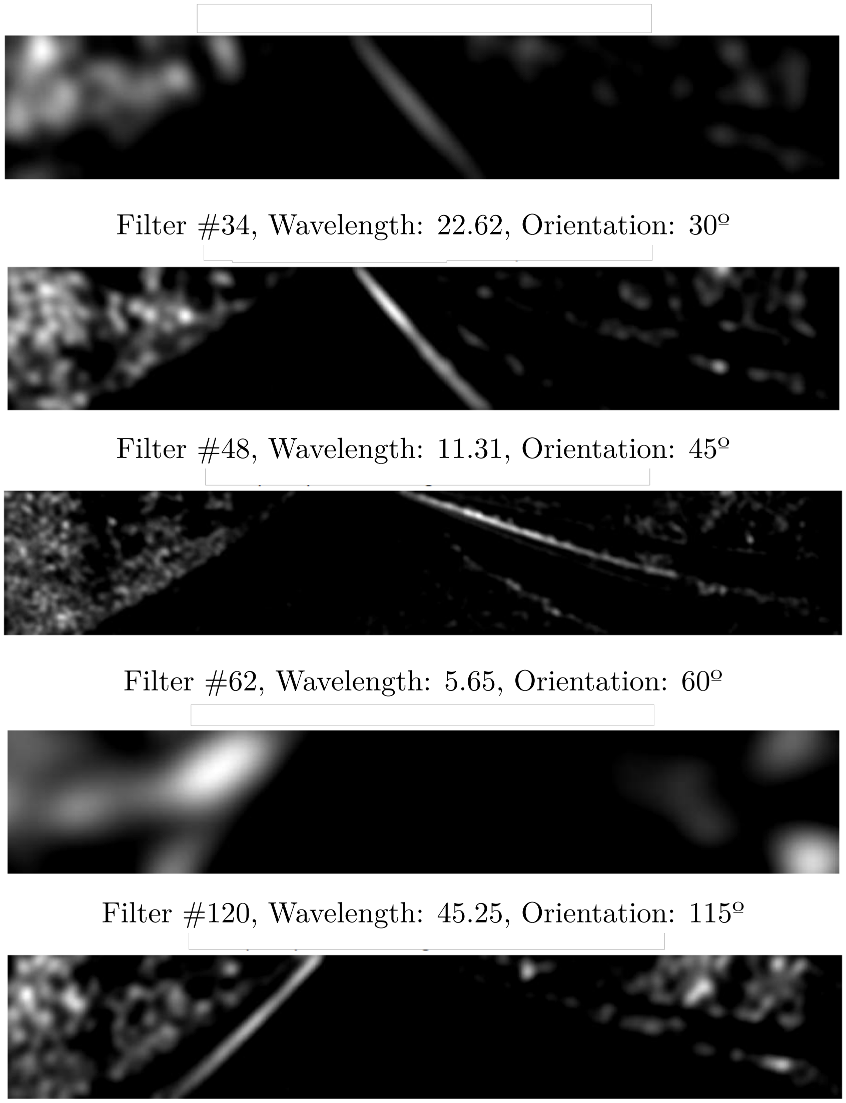
4.7 Gabor texture features

In order to have a systematic characterization of the surroundings of each pixel we use a bank of Gabor filters, which are used on both the RGB panoramic images and the intensity orthoimages obtained from LiDAR points. The magnitude of the responses of the Gabor filters are used a the feature vector for classification. In other words, we use the local texture descriptor of the image as features [28?] for classification. Formally, a Gabor filter is defined by the product of a sinusoidal wave, i.e. a plane wave in 2D, and a Gaussian function. The Gaussian component modulates the scale of the filter, while the wave component acts as a selector of the orientation and spatial frequency of the detected objects. In many implementations, Gaussian scale and wavelength are linked, so only the wavelength is specified. The

Gabor filter provides a complex valued response, so its magnitude and phase can be used as features. In this paper we use only the phase. Formally, a the impulse response of single filter is defined as follows:

$$g\left(x, y\right) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right] \exp\left[2\pi i\left(Ux + Vy\right)\right], \quad (4.22)$$

where we rotate the Euclidean coordinates by $x' = x\cos\left(\theta\right) + y\sin\left(\theta\right)$, and $y' = x\sin\left(\theta\right) + y\cos\left(\theta\right)$. Parameters σ_x, σ_y that define the spatial support and bandwidth of the filter are determined as a function of the wavelength F so that the Gaussian covers three periods of the sinusoid. The complex exponential factor is a 2D sinusoid wave of frequency $F = \sqrt{U^2 + V^2}$ and orientation $\gamma = \tan^{-1}\left({}^{V}\!/_{U}\right)$. Figure 4.7 shows some examples of the magnitude response of different filters over the same road image. Diverse orientations highlight different road lane marks, and diverse wavelengths produce crisp or blurred images, where details are highlighted or removed.



Filter #34, Wavelength: 22.62, Orientation: 30º

Filter #48, Wavelength: 11.31, Orientation: 45º

Filter #62, Wavelength: 5.65, Orientation: 60º

Filter #120, Wavelength: 45.25, Orientation: 115º

Filter #143, Wavelength: 11.31, Orientation: 140º

Figure 4.7: Examples of filtered images with Gabor filters of various orientations and scales.

Figure 4.7: Examples of filtered images with Gabor filters of various orientations and scales.

Chapter 5

Experimental Results

In this Chapter we concentrate the results of the computaitonal experiments carried out over each of the datasets described in Chapter 3. For each dataset we describe the model parameter exploration methodology, followed by the tables containing the actual results and some discussion about the relevance of the results. Sectionn 5.1 provides the results obtained over the panomic RGB images. Section 5.2 discusses the results achieved over the rater images obtained from the intensity response of the LiDAR sensor. Section 5.3 presents the results on the features extracted from the LiDAR point cloud. Finally, Section 5.4 gives some short conclusions.

5.1 Results on RGB panoramas images

5.1.1 Model parameter exploration.

We have carried out an exhaustive exploration of the model parameter settings in order to assess robustness of the approach and to find out which parameter is more influential.We can provide some guidance on the best parameter setting and the most promising model. Regarding RF computational experiments, we have explored the influence of the number of trees (NT), the number of variables taken into account at node split (NVS), and the size of the sample (NS) that is added to the current training dataset. We have also made the comparative between ensuring that the dataset sub-

RF	
Number of images	15
Balanced dataset	[0,1]
NS	[50,100]
NVS	[5,10,15,20]
NT	[50,100,150,200]
TPR training threshold	0,8
Number of iterations	10

V-ELM	
Number of images	15
Balanced dataset	[1]
NELM	[50,100,150,200]
NHELM	[25,50,100]
NS	[50,100]
TPR training threshold	0,8
Number of iterations	10
Activation function	Sigmoid

Table 5.1: Summary of the experimental design for RF (left) and V-ELM (right) for panoramic imagery.

sample added to the current training dataset is class balanced and not ensuring class balance. This experience is not repeated on the ELM experiments due to the negative results in RF experiments with unbalanced samples. Regarding ELM computational experiments, we have explored the influence of the ensemble size (NELM), the number of hidden units (NHELM) and the size of the sample (NS) added to the current training dataset. We have used the standard sigmoid activation function. In all reported experiments we have performed a fixed number of iterations (T=10) of the AL algorithm. Table 5.1 shows a summary of the configuration of the selected parameters for both RF and V-ELM experiments.

5.1.2 Experimental Results

For a qualitative appraisal of the results, we show in Figure 5.1 some visual results of the detections achieved by RF (left) and V-ELM (right) classifiers trained on the final training datasets after an AL processes. Notice that these results do not include post-processing and noise removal over the classification results. Removing noisy detections may be easily achieved applying morphological operators and some standard computer vision tools. These processes will be asily implemented during the industrial implementation of the approach, and including them here would somehow obscure the actual contribution of the machine learning approaches. The quantitative

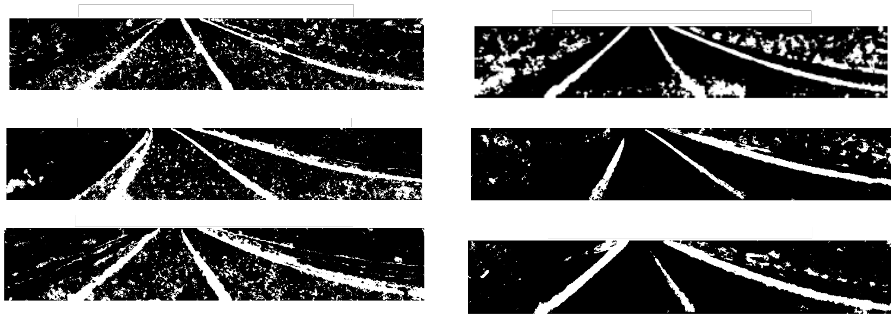


Figure 5.1: Some visual results of the trained RF (left column) and V-ELM (right column) ensemble classifiers using balanced training sample increments. Parameter settings for RF are NT=50, NVS=5, NS=100. Parameter settings for V-ELM are NELM=50, NHELM=10, NS=50.

analysis of the results have been organized into several research questions.

Impact of the sample selection strategy We compare the performance achieved by the RF classifiers when the training datasets are increased by candidate samples with or without ensuring class balancing, as shown in Table 5.2 and Table 5.3, respectively. Welch's one-sided t-tests comparing the performance measures (SEN, SP, PPV, AUC) obtained by the balanced versus unbalanced selection of samples to be added to the training dataset confirms that there is a significant increase in performance.Wilcoxon rank sum tests give even more significant differences($p < 1e - 11$ for PPV, $p < 1e - 10$ for SEN, $p < 1e - 9$ for SP). For clarity, we have summarized in Table 5.5 the p values of both tests. Notice that for SP the sign of the change is negative, because it decreases with the balanced design. This result provides an alternative approach to the correction of the bias induced by strong class imbalance in the dataset. In this situation, if we examine the AC performance, we are mislead for ignoring the minority class lead to very high AC values However, very poor SEN and PPV values as can be seen in Table 5.3 where unbalanced training dataset growth lead to high AC and SP but very low SEN and PPV values. In our computational experiments, the average PPV in Table 5.3 is 0.051 while in Table 5.2 it is 0.552. There is one order of magnitude increase of the PPV value. Consequently, we have not carried out the comparison on the V-ELM classifiers, which have been trained only with balanced incremental training datasets.

For clarity, we have summarized in Table 5.5 the p values of both tests. Notice that for SP the sign of the change is negative, because it decreases with the balanced design. This result provides an alternative approach to the correction of the bias induced by strong class imbalance in the dataset. In this situation, if we examine the AC performance we are mislead, because ignoring the minority class lead to very high AC values, but very poor SEN and PPV values as can be seen in Table 5.3 where unbalanced training dataset growth lead to high AC and SP but very low SEN and PPV values. In our computational experiments, the average PPV in Table 5.3 is 0.051 while in Table 5.2 it is 0.552. There is one order of magnitude increase of the PPV value. Consequently, we have not carried out the comparison on the V-ELM classifiers, which have been trained only with balanced incremental training datasets.

Effect of parameter settings The size of the sample (NS) that is added to the training dataset at each iteration is a global parameter of the AL process. In Table 5.6 we have summarized the p values from the tests of statistical significance related to NS. In the RF experiments, we find moderate but significant effect of the sample size 100 versus 50 in Welch's one-sided t-test ($p < 0.004$ for PPV, $p < 0.005$ for SP) and in non-parametric Wilcoxon rank sum test($p < 0.02$ for PPV, $p < 0.007$ for SP) on the data of Table 5.2. In the V-ELM experiments reported in Table 5.4 we find even lower significance in Welch's one-sided t-test ($p < 0.5$ for PPV, $p < 0.05$ for SP) and in non-parametric Wilcoxon rank sum test($p < 0.5$ for PPV, $p < 0.09$ for SP). However, we think that a greater sample increment is preferable, for instance, for V-ELM the mean value of PPV for NS=100 is 0.82, while for NS=50 it is 0.77. Similar values for RF are 0.67 and 0.42.

Regarding the effect of the RF parameters, we find that NT has a stronger effect than NVS in an analysis of variance ($p < 1e-5$ for PPV, $p < 0.005$ for SP). It is very interesting to observe a clear over-fitting effect,

NS	NVS	NT	SEN	SP	AC	PPV	AUC
100	5	50	0.901	0.172	0.894	0.981	0.*624*
100	5	100	0.898	0.168	0.891	0.969	0.616
100	5	150	0.906	0.167	0.899	0.966	0.628
100	5	200	0.919	0.150	0.912	0.957	0.633
100	10	50	0.890	0.174	0.883	0.968	0.608
100	10	100	0.878	0.179	0.871	0.930	0.595
100	10	150	0.870	0.181	0.863	0.949	0.584
100	10	200	0.851	0.194	0.844	0.951	0.569
100	15	50	0.914	0.159	0.907	0.964	0.634
100	15	100	0.917	0.158	0.909	0.958	0.637
100	15	150	0.922	0.153	0.914	0.962	0.640
100	15	200	0.910	0.161	0.902	0.948	0.628
100	20	50	0.870	0.186	0.863	0.956	0.590
100	20	100	0.884	0.176	0.877	0.961	0.600
100	20	150	0.870	0.185	0.863	0.955	0.589
100	20	200	0.856	0.196	0.850	0.963	0.580
50	5	50	0.927	0.154	0.919	0.950	0.651
50	5	100	0.924	0.151	0.917	0.957	0.644
50	5	150	0.910	0.159	0.903	0.952	0.626
50	5	200	0.922	0.150	0.915	0.953	0.638
50	10	50	0.855	0.199	0.849	0.973	0.582
50	10	100	0.870	0.186	0.863	0.955	0.588
50	10	150	0.837	0.205	0.831	0.947	0.563
50	10	200	0.907	0.141	0.900	0.890	0.598
50	15	50	0.904	0.165	0.896	0.964	0.622
50	15	100	0.914	0.159	0.906	0.958	0.632
50	15	150	0.889	0.181	0.882	0.966	0.614
50	15	200	0.912	0.156	0.905	0.951	0.626
50	20	50	0.869	0.185	0.862	0.969	0.587
50	20	100	0.877	0.184	0.870	0.941	0.598
50	20	150	0.887	0.170	0.880	0.948	0.597
50	20	200	0.892	0.165	0.885	0.925	0.600

Table 5.2: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced.

Table 5.2: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. P

NS	NVS	NT	SEN	SP	AC	PPV	AUC
100	5	50	0.997	0.012	0.972	0.078	0.692
100	5	100	0.999	0.008	0.973	0.032	0.729
100	5	150	0.996	0.014	0.971	0.108	0.685
100	5	200	0.999	0.020	0.973	0.057	0.739
100	10	50	0.996	0.000	0.970	0.090	0.773
100	10	100	0.998	0.000	0.972	0.043	0.750
100	10	150	0.992	0.011	0.967	0.172	0.575
100	10	200	0.998	0.017	0.973	0.064	0.729
100	15	50	0.998	0.005	0.972	0.059	0.661
100	15	100	1.000	0.000	0.974	0.015	0.750
100	15	150	0.998	0.015	0.973	0.067	0.726
100	15	200	0.996	0.001	0.970	0.079	0.740
100	20	50	0.998	0.012	0.973	0.043	0.718
100	20	100	0.999	0.011	0.974	0.030	0.738
100	20	150	0.999	0.000	0.973	0.039	0.542
100	20	200	0.999	0.018	0.974	0.043	0.742
50	5	50	1.000	0.004	0.974	0.015	0.718
50	5	100	1.000	0.004	0.974	0.009	0.737
50	5	150	0.995	0.032	0.970	0.132	0.716
50	5	200	0.999	0.000	0.973	0.024	0.542
50	10	50	0.999	0.005	0.973	0.037	0.685
50	10	100	0.995	0.008	0.970	0.127	0.595
50	10	150	0.999	0.001	0.973	0.040	0.500
50	10	200	0.998	0.002	0.973	0.040	0.579
50	15	50	0.997	0.007	0.972	0.055	0.655
50	15	100	1.000	0.000	0.974	0.005	0.750
50	15	150	0.998	0.004	0.973	0.046	0.643
50	15	200	1.000	0.000	0.974	0.013	0.750
50	20	50	1.000	0.000	0.974	0.009	0.750
50	20	100	1.000	0.003	0.974	0.014	0.716
50	20	150	0.999	0.000	0.973	0.034	0.750
50	20	200	0.999	0.006	0.973	0.027	0.716

Table 5.3: Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performarce results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS).

Table 5.3: Performance of AL using RF classifiers without ensuring that set of samples added at

NHELM	NELM	SEN	SP	AC	PPV	AUC
25	50	0.45	0.952	0.94	0.45	0.824
25	50	0.805	0.853	0.852	0.806	0.886
25	100	0.828	0.821	0.82	0.828	0.881
25	100	0.524	0.928	0.919	0.524	0.829
25	150	0.827	0.792	0.827	0.827	0.869
50	50	0.639	0.902	0.895	0.639	0.850
50	50	0.844	0.839	0.84	0.844	0.896
50	100	0.828	0.815	0.815	0.828	0.879
50	100	0.846	0.939	0.937	0.846	0.937
50	150	0.8	0.842	0.841	0.8	0.879
50	150	0.897	0.922	0.922	0.897	0.947
50	200	0.816	0.823	0.823	0.816	0.877
50	200	0.909	0.893	0.893	0.91	0.941
100	50	0.879	0.802	0.804	0.879	0.896
100	50	0.871	0.91	0.909	0.871	0.934
100	100	0.923	0.845	0.846	0.924	0.930
100	100	0.898	0.921	0.92	0.898	0.947
100	150	0.809	0.886	0.884	0.809	0.902
100	150	0.834	0.809	0.809	0.834	0.879
100	200	0.909	0.8764	0.877	0.909	0.935
100	200	0.844	0.818	0.819	0.844	0.887
150	50	0.646	0.924	0.918	0.646	0.866
150	50	0.822	0.897	0.895	0.822	0.911
150	100	0.903	0.721	0.725	0.904	0.879
150	100	0.928	0.878	0.879	0.928	0.943
150	150	0.841	0.785	0.787	0.842	0.872
150	150	0.827	0.885	0.884	0.827	0.908

Table 5.4: Performance of AL using V-ELM ensemble classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of ELM components (NELM), the number of hidden units at each ELM (NHELM), the number of samples added in each iteration of the active learning algorithm (NS).

Table 5.4: Performance of AL using V-ELM ensemble classifiers ensuring that set of samples added at each

test	SEN	SP	PPV	AUC
Wt	$< 1e-8$	$< 1e-8$	$< 1e-10$	$p < 1e-7$
WR	$< 1e-10$	$< 1e-9$	$< 1e-11$	$< 1e-9$

Table 5.5: Assessment of impact of sample selection strategy (Balanced unbalanced) over performance results shown in Tables 5.2 and 5.3, we give the p values of Wt=Welch's t-test, WR=non-parametric Wilcoxon rank test.

test	SEN	SP	PPV	AUC
Wt	$< 1e-8$	$< 1e-8$	$< 1e-10$	$p < 1e-7$
WR	$< 1e-10$	$< 1e-9$	$< 1e-11$	$< 1e-9$

Table 5.6: Assessment of impact of the size of training data sampling increments over performance results shown in Tables 5.2 and 5.4, we give the p values of Wt=Welch's t-test, WR=non-parametric Wilcoxon rank test.

i.e. often there is a decrease in performance when NT goes from 150 to 200 all other parameters unchanged. Computing in Table 5.2 the mean values of PPV for NT values 50, 100, 150, 200 we obtain 0.71, 0.72, 0.60, and 0.16, respectively.

Regarding the effect of the parameters of V-ELM, namely the number of hidden units and the number of ELMs in the ensemble, their effect is minor. An analysis of variance on the data of Table 5.4 reports a minor effect of NHELM on PPV and SP ($p < 0.6$). If we look in detail at the performance for varying NHELM values 25, 50, 100, and 150 we obtain average PPV values 0.68, 0.81, 0.87, and 0.82, respectively. Therefore, we may have a big performance increase going from 25 to 100 hidden units, but afterwards there is a degradation of the classifier performance. Although the number of ELMs in the ensemble seems to be non significant, it must be noted that the average PPV for NELM values 50, 100, 150, and 200 is 0.74, 0.83, 0.83, and 0.81, respectively. Therefore, there is significant improvement increasing the number of ELMs until saturation around 100 ELMs, which leads to performance degradation at 200 ELMs but not as dramatic as in the case of RF number of trees.

Effect of the classifier ensemble When comparing the V-ELM against the RF, we find a highly significant difference in Welch's two sample t-test ($p < 1e-4$ for PPV and $p < 1e-15$ for SP) and in non-parametric Wilcoxon rank sum test ($p < 1e-3$ for PPV, $p < 1e-6$ for SP). The mean vales of the TPR and SP for V-ELM over Table 5.4 are 0.80 and 0.86, respectively, while for the RF in Table 5.2 are 0.55 and 0.20, respectively. The best PPV and SP values achieved by V-ELM classifiers are 0.92 and 0.95, while RF achieves 0.89 and 0.50, respectively. Adding up the insensitivity to the parameter settings and the improved results, V-ELM appears to be a most promising approach for the full development of the large scale road images segmentation.

5.2 Results on the intensity images from LiDAR

5.2.1 Model parameter exploration.

In order to assess robustness of the approach and to find out which parameter is more influential we have studied some model parameter settings.Regarding RF computational experiments, we have explored the influence of the number of trees (NT), the number of variables taken into account at node split (NVS), and the size of the sample (NS) that is added to the current training dataset. We have also make the comparative between ensuring that the dataset subsample added to the current training dataset is class balanced and not ensuring class balance. We show in Table 5.7 a summary of the configuration of selected parameters.

5.2.2 Experimental Results

For a qualitative appraisal of the results we show in Figure 5.2 two images of the detections achieved by RF classifiers trained on the final training dataset of Active Learning processes. On the other hand, for a quantitative assessment, we also compare the performance achieved by the RF classifiers when the training dataset is increased by candidate samples with or

RF	
Number of images	10
Balanced dataset	[0,1]
NS	[50,100]
NVS	[5,10,15,20]
NT	[50,100,150,200]
TPR training threshold	0,8
Number of iterations	10

Table 5.7: Summary of the experimental design for RF.

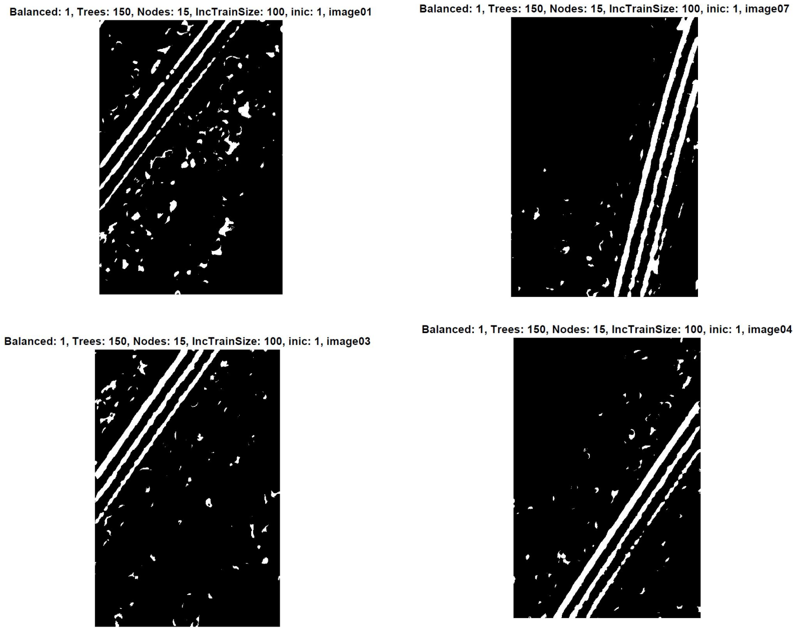

Balanced: 1, Trees: 150, Nodes: 15, IncTrainSize: 100, inic: 1, image01
Balanced: 1, Trees: 150, Nodes: 15, IncTrainSize: 100, inic: 1, image07
Balanced: 1, Trees: 150, Nodes: 15, IncTrainSize: 100, inic: 1, image03
Balanced: 1, Trees: 150, Nodes: 15, IncTrainSize: 100, inic: 1, image04

Figure 5.2: Some visual results of the trained RF ensemble classifiers using balanced training sample increments.

without ensuring class balancing. As in 3.2we are not reporting results after post-processing and noise removal of the classification results. Removing noisy detections may be easily achieved applying morphological operators and some standard computer vision tools. We think that these processes correspond to the industrial implementation of the approach, and including them here would somehow obscure the intended contribution of the paper. Regarding the quantitative analysis of the results, we have organized it into several research questions, to which answers have been obtained by carrying out a hypothesis test.

About the statistical analysis of the results Before starting, the type of distribution that our results follow has been studied. For this, we have studied the distribution of the sample data applying Lilliefors test, which is a test of normality based on the Kolmogorov-Smirnov test. It is used to test the null hypothesis that the data come from a population with a normal distribution, when the null hypothesis does not specify which distribution is normal. According to the test, the sample data do not follow a normal distribution tendency, except PPV sample data. Based on the result obtained in the normality test, it is decided to use the following statistic tests.

- Welch's t-test: Used for non normal distribution sample data.

$$t = \frac{m_a - m_b}{\sqrt{\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}}} \tag{5.1}$$

- Student's t-test: Used for normal distribution sample data.

$$t = \frac{\overline{X} - \mu}{\frac{S}{\sqrt{n}}} \tag{5.2}$$

- Wilcoxon signed-rank test: Non parametric sample data test.

$$z = \frac{W - \mu_W}{\sigma_W} \tag{5.3}$$

In order to reject or not the null hypothesis we have set up the significance level as $\alpha = 0.05$. In Table 5.8 we show the results obtained after

Parameter setting	**test**	**SEN**	**SP**	**AC**	**I**
Impact of balanced data set					
Balanced vs No Balanced	Tt	$< 3.30e - 22$	$< 1.16e - 33$	$< 4.69e - 22$	$< 1.$
	WR	$< 3.23e - 12$	$< 3.24e - 12$	$< 3.24e - 12$	$< 3.$
Effect of the size of the sample (NS)					
NS=50 vs NS=100	Tt	$< 1.06e - 10$	$< 7.96e - 22$	$< 8.33e - 21$	$< 2.$
	WR	$< 1.92e - 13$	$< 8.75e - 21$	$< 9.15e - 21$	$< 6.$
Choice of number of trees (NT)					
NT=50 vs NT=100	Tt	< 0.2764	< 0.1603	< 0.1597	$<$
	WR	< 0.3401	< 0.8532	< 0.8546	$<$
NT=50 vs NT=150	Tt	$< 4.16e - 11$	$< 1.92e - 10$	$< 2.03e - 10$	$< 1.$
	WR	$< 4.58e - 08$	$< 1.51e - 08$	$< 1.51e - 08$	$< 8.$
NT=50 vs NT=200	Tt	$< 3.02e - 13$	$< 1.09e - 11$	$< 1.19e - 11$	$< 3.$
	WR	$< 1.11e - 08$	$< 1.87e - 08$	$< 1.94e - 08$	$< 6.$
Number of variables for NVS					
NVS=5 vs NVS=10	Tt	< 0.2389	< 0.6807	< 0.6798	$<$
	WR	< 0.4518	< 0.536	< 0.5374	$<$
NVS=5 vs NVS=15	Tt	< 0.0802	< 0.8492	< 0.8483	$<$
	WR	< 0.0664	< 0.9784	< 0.9788	$<$
NVS=5 vs NVS=20	Tt	< 0.334	< 0.5924	< 0.5917	$<$
	WR	< 0.1982	< 0.4925	< 0.4898	$<$

Table 5.8: Assessment of impact of the parameter setting over results shown in Tables 5.9, 5.10. We give the p values of **Tt** = T-test, **WR** = Wilcoxon signed-rank test. Significance level $\alpha = 0.05$

carrying out the hypothesis test on the different variables comparing the different statistics used, which we will organise into several research questions in order to carry out a quantitative analysis of the results obtained.

Impact of the sample selection strategy We compare the performance achieved by the random forest classifiers when the dataset is increased by candidate sample with or without ensuring class balancing, as shown in Table 5.9 and 5.10 respectively. A one-sided t-test (Welch's t-test) comparing the sen and AUC of the balanced versus unbalanced increments confirms that there is a significant increment in performance ($\rho < 3.29e - 22$ for sen, $\rho < 9.50e - 3$ for AUC). Comparing the PPV values, using Welch's t-test

($\rho < 1.41e-77$), also shows an important affluence in the choice of balanced or unbalanced values. Taking into account PPV This result is meaningful in the sense that it provides an alternative approach to the resolution of the bias induced by strong class imbalance. In this situation, if we examine the AC performance we are mislead, because ignoring the minority class lead to very high AC values, but very poor PPV values as can be seen in Table 5.10 where unbalanced training dataset growth lead to high AC and SEN but very poor SP and PPV. In fact, the mean PPV in Table 5.10 is 0.074 while in Table 5.9 it is 0.954

Effect of parameter settings The size of the sample (NS) that is added to the training dataset at each iteration is a global parameter of the Active Learning process. In the RF experiments, we find slight effect of the sample size 100 versus 50 in Welch's t-test ($p < 1.32e-19$ for AUC, $p < 7.96e-22$ for SP) on the data of Table 5.9 . However, the effect found when we increase from 50 to 100 the sample side if we take into account Wilcoxon signed-rank test is more meaningful ($\rho < 8.75e-21$ for sen, $\rho < 8.46e-12$ for AUC).For RF the mean value of PPV for NS=100 is 0.96, while for NS=50 it is 0.95 for balanced data. Due to the results obtained with unbalanced data, we do not analyse effect of parameter setting to this data.

The effect of the number of trees chosen is another global parameter of the Active Learning process. In our experiments, using only balanced data if we focus only in number of trees, we find a good response in one-sided t-test when NT=200 ($p < 1.09e-11$ for sp, $p < 3.38e-15$ for PPV and $p < 0.0011$for AUC). If we only look at the number of trees, for the balanced data sample, as we increase the number of trees, we cannot consider an overfitting effect. Nor can we speak of a significant improvement, in fact, there is hardly any improvement at all. Looking at the mean values of TPR for NT values 50, 100, 150, 200 we obtain 0.96, 0.95, 0.93, and 0.94, respectively.

Finally, regarding the effect of the number of variables considered for the division in each node (NVS), we did not find great relevance in the performance when varying the number of variables neither in one-sided t-

NS	NVS	NT	SEN	SP	AC	PPV	AUC
100	5	50	0.167	0.906	0.899	0.966	0.628
100	10	50	0.158	0.917	0.909	0.958	0.637
100	15	50	0.15	0.922	0.915	0.953	0.638
100	20	50	0.156	0.912	0.905	0.951	0.626
100	5	100	0.172	0.901	0.894	**0.981**	0.624
100	10	100	0.153	0.922	0.914	0.962	0.64
100	15	100	0.151	0.924	0.917	0.957	0.644
100	20	100	0.165	0.904	0.896	0.964	0.622
100	5	150	0.15	0.919	0.912	0.957	0.633
100	10	150	0.161	0.91	0.902	0.948	0.628
100	15	150	0.154	0.927	**0.919**	0.95	0.651
100	20	150	0.181	0.889	0.882	0.966	0.614
100	5	200	0.168	0.898	0.891	0.969	0.616
100	10	200	0.159	0.914	0.907	0.964	0.634
100	15	200	0.159	0.91	0.903	0.952	0.626
100	20	200	0.159	0.914	0.906	0.958	0.632
50	5	50	0.174	0.89	0.883	0.968	0.608
50	10	50	0.176	0.884	0.877	0.961	0.6
50	15	50	0.199	0.855	0.849	0.973	0.582
50	20	50	0.17	0.887	0.88	0.948	0.597
50	5	100	0.194	0.851	0.844	0.951	0.569
50	10	100	0.186	0.87	0.863	0.956	0.59
50	15	100	0.186	0.87	0.863	0.955	0.588
50	20	100	0.165	0.892	0.885	0.863	0.6
50	5	150	0.181	0.87	0.863	0.949	0.584
50	10	150	0.196	0.856	0.85	0.963	0.58
50	15	150	0.141	0.907	0.9	0.89	0.598
50	20	150	0.184	0.877	0.87	0.941	0.598
50	5	200	0.179	0.878	0.871	0.93	0.595
50	10	200	0.185	0.87	0.863	0.955	0.589
50	15	200	0.205	0.837	0.831	0.947	0.563
50	20	200	0.185	0.869	0.862	0.969	0.587

Table 5.9: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS). The added set of samples is class balanced.

NS	NVS	NT	SP	SEN	AC	PPV	AUC
50	5	50	0.003	0.999	0.989	0.068	0.634
100	5	50	0.001	0.999	0.989	0.06	0.581
50	10	50	0.005	0.999	0.99	0.026	0.717
100	10	50	0.002	0.999	**0.99**	0.04	0.662
50	15	50	0.005	0.996	0.987	0.119	0.56
100	15	50	**0.007**	0.998	0.988	0.106	0.67
50	20	50	0.004	0.997	0.987	0.125	0.561
100	20	50	0.003	0.998	0.988	0.078	0.584
50	5	100	0.001	0.999	0.99	0.027	0.592
100	5	100	0.002	0.999	0.99	0.041	0.66
50	10	100	0.002	0.999	0.989	0.033	0.658
100	10	100	0.002	**1**	0.99	0.02	0.721
50	15	100	0.006	0.998	0.988	0.052	0.649
100	15	100	0.002	0.998	0.988	**0.128**	0.529
50	20	100	0.003	0.997	0.987	0.108	0.5
100	20	100	0.002	0.998	0.989	0.068	0.5
50	5	150	0.001	0.999	0.989	0.076	0.5
100	5	150	0.002	0.998	0.988	0.094	0.503
50	10	150	0.002	0.998	0.988	0.082	0.545
100	10	150	0.003	0.999	0.989	0.073	0.661
50	15	150	0.005	0.999	0.989	0.06	0.675
100	15	150	0.002	0.999	0.989	0.083	0.626
50	20	150	0.002	0.998	0.989	0.088	0.577
100	20	150	0.002	0.999	0.989	0.075	0.567
50	5	200	0.002	0.999	0.989	0.063	0.636
100	5	200	0.003	0.998	0.988	0.116	0.598
50	10	200	0.001	0.999	0.989	0.075	0.501
100	10	200	0.001	0.999	0.989	0.085	0.5
50	15	200	0.005	0.998	0.988	0.107	0.63
100	15	200	0.004	0.999	0.989	0.066	0.686
50	20	200	0.005	0.998	0.988	0.105	0.629
100	20	200	0.001	1	0.99	0.023	0.701

Table 5.10: Performance of AL using RF classifiers without ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS).

Table 5.10: Performance of AL using RF classifiers without ensuring that set of samples added at each iter

RF	
Number of Point Clouds	10
Balanced dataset	[1]
NS	[50,100]
NVS	[5,10,15,20]
NT	[5,10,25,50,100,150,200]
TPR training threshold	0,8
Number of iterations	10
Number of initializations	3

Table 5.11: Summary of the experimental design for RF.

test, nor the Wilcoxon signed-rank test; $\rho < 0.0257$, $\rho < 0.0171$ for PPV respectively when NVS goes from 5 to 15 variables considered for the node´s split.

5.3 Results on the LiDAR sensor Point Cloud data

5.3.1 Model parameter exploration.

In order to assess robustness of the approach and to find out which parameter is more influential we have studied some model parameter settings.Regarding RF computational experiments, we have explored the influence of the number of trees (NT), the number of variables taken into account at node split (NVS), and the size of the sample (NS) that is added to the current training dataset. In contrast to the previous experiments in the **??** section, accordance with the results obtained, we have not carried out experiments taking into account the unbalanced dataset. We show in Table 5.11 a summary of the configuration of selected parameters.

Experimental Results

For a qualitative results, we show in Fig 5.3 some images showing the detection achieved using RF classifier with different parameters configurations. Below we show the images in top view and front-isometric view, where it can be seen that the extraction of road lane-marks is done maintaining the altimetry, unlike the images that are only extracted in planimetry.

For a quantitative assessment, we only compare the performance achieved by the RF classifiers when the training dataset is increased by candidate samples ensuring class balancing. We don't take into account possible noisy detection and removal. As we explain previously, we consider removing noisy detections may be easily achieved applying morphological operators and some standard computer vision tools. As for the quantitative analysis of the results, we divided them into several research questions, the answers to which were obtained by conducting a hypothesis test. The results obtained are shown in Tables 5.12 to 5.16.

Before starting, we have considered the nature of the distribution followed by our classification results. For this end, we examined the distribution of the sample data using the Lilliefors test, a normality test based on the Kolmogorov-Smirnov test. It is used to test the null hypothesis that the data come from a population with a normal distribution when the null hypothesis does not specify which distribution is normal. According to the test, the sample data follows a normal distribution tendency. Based on the result obtained in the normality test, it is decided to use the following statistic tests; Student's t-test and Non parametric Wilcoxon signed-rank test.

To reject the null hypothesis or not, we set the significance level at $\alpha = 0.05$. In the table 5.17 we sumarize p values after performing the hypothesis test for the different variables, comparing the different statistics used, which we will divide into several research questions to perform a quantitative analysis of the results obtained.

Effect of parameter settings The size of the sample (NS) that is added to the training dataset at each iteration is a global parameter of the Active Learning process. In the RF experiments, we find moderate but significant effect of the sample size 100 versus 50 in one-sided t-test ($p < 1.66e - 29$, for sp, $p < 6.08e - 07$ for PPV and $p < 2.47e - 47$ for AUC values) and Wilcoxon rank sum test ($p < 7.77e - 21$, for sp, $p < 3.80e - 03$ for PPV and$p < 1.22e - 46$ for AUC values) on the data of Table 5.17. However, we

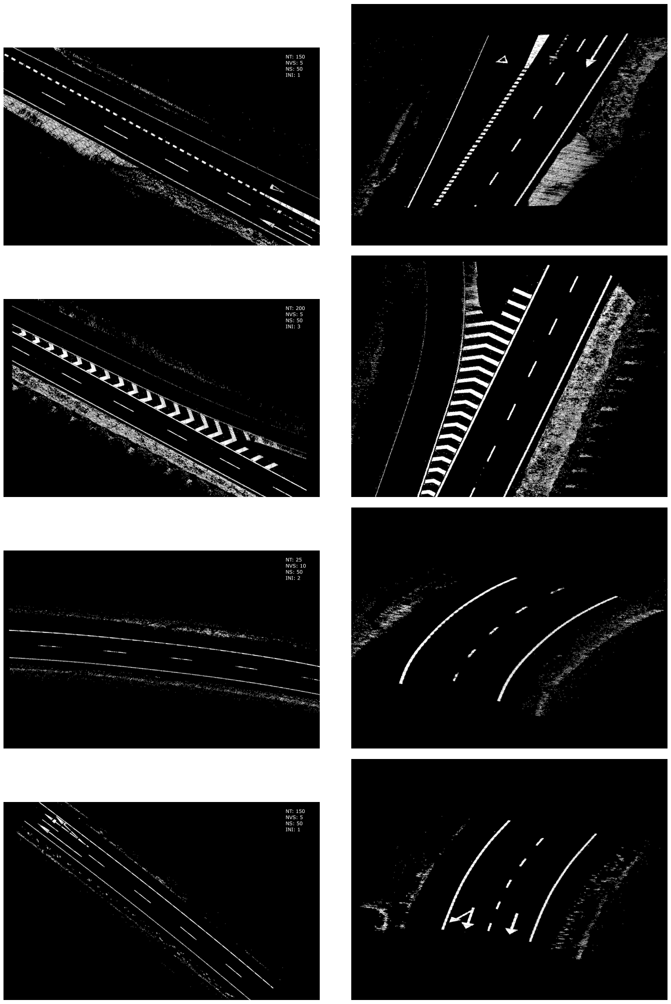


Figure 5.3: Some visual results of the trained RF ensemble classifiers using balanced training sample different parameters settings. Left column top-view, right column front-isometry view

Table 5.12: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI).

INI	NS	NVS	NT	SEN	SP	AC	PPV	AUC
1	50	5	5	0.935	0.913	0.935	0.914	0.956
2	50	5	5	0.960	0.924	0.960	0.925	0.968
3	50	5	5	0.796	0.974	0.796	0.968	0.937
1	100	5	5	0.968	0.947	0.968	0.948	0.977
2	100	5	5	0.959	0.928	0.959	0.929	0.969
3	100	5	5	0.949	0.914	0.949	0.916	0.962
1	50	10	5	0.848	0.962	0.848	0.958	0.947
2	50	10	5	0.814	0.952	0.814	0.947	0.932
3	50	10	5	0.848	0.882	0.848	0.882	0.915
1	100	10	5	0.919	0.956	0.919	0.955	0.965
2	100	10	5	0.894	0.940	0.894	0.938	0.952
3	100	10	5	0.966	0.927	0.966	0.929	0.971
1	50	15	5	0.898	0.978	0.898	0.975	0.967
2	50	15	5	0.979	0.865	0.979	0.869	0.958
3	50	15	5	0.853	0.928	0.853	0.925	0.935
1	100	15	5	0.919	0.964	0.919	0.962	0.968
2	100	15	5	0.956	0.960	0.956	0.960	0.977
3	100	15	5	0.979	0.890	0.979	0.893	0.965
1	50	20	5	0.896	0.953	0.896	0.951	0.957
2	50	20	5	0.964	0.857	0.964	0.861	0.95
3	50	20	5	0.961	0.942	0.961	0.942	0.973
1	100	20	5	0.970	0.923	0.970	0.924	0.971
2	100	20	5	0.944	0.944	0.944	0.944	0.969
3	100	20	5	0.836	0.944	0.836	0.940	0.936
1	50	5	10	0.931	0.891	0.931	0.892	0.948
2	50	5	10	0.875	0.921	0.875	0.919	0.939
3	50	5	10	0.882	0.839	0.882	0.842	0.911
1	100	5	10	0.964	0.913	0.964	0.915	0.966
2	100	5	10	0.944	0.928	0.944	0.929	0.964
3	100	5	10	0.957	0.942	0.957	0.943	0.973
1	50	10	10	0.975	0.872	0.975	0.876	0.958
2	50	10	10	0.910	0.930	0.910	0.929	0.954
3	50	10	10	0.964	0.935	0.964	0.936	0.972
1	100	10	10	0.959	0.933	0.959	0.933	0.97
2	100	10	10	0.916	0.979	0.916	0.977	0.972
3	100	10	10	0.971	0.906	0.971	0.908	0.967
1	50	15	10	0.932	0.964	0.932	0.963	0.972
2	50	15	10	0.972	0.890	0.972	0.893	0.962
3	50	15	10	0.949	0.922	0.949	0.924	0.964
1	100	15	10	0.891	0.977	0.891	0.974	0.964

Table 5.13: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI).

INI	NS	NVS	NT	SEN	SP	AC	PPV	AUC
2	100	15	10	0.958	0.958	0.958	0.958	0.977
3	100	15	10	0.968	0.937	0.968	0.939	0.974
1	50	20	10	0.955	0.852	0.955	0.855	0.945
2	50	20	10	0.913	0.904	0.913	0.905	0.946
3	50	20	10	0.902	0.975	0.902	0.973	0.967
1	100	20	10	0.898	0.983	0.898	0.980	0.969
2	100	20	10	0.963	0.919	0.963	0.921	0.968
3	100	20	10	0.917	0.962	0.917	0.960	0.967
1	50	5	25	0.881	0.852	0.881	0.853	0.916
2	50	5	25	0.863	0.965	0.863	0.962	0.952
3	50	5	25	0.934	0.976	0.934	0.975	0.976
1	100	5	25	0.921	0.970	0.921	0.968	0.97
2	100	5	25	0.880	0.967	0.880	0.964	0.958
3	100	5	25	0.917	0.965	0.917	0.963	0.967
1	50	10	25	0.878	0.980	0.878	0.976	0.962
2	50	10	25	0.969	0.883	0.969	0.886	0.959
3	50	10	25	0.969	0.853	0.969	0.857	0.951
1	100	10	25	0.855	0.947	0.855	0.944	0.943
2	100	10	25	0.905	0.978	0.905	0.975	0.969
3	100	10	25	0.966	0.955	0.966	0.955	0.979
1	50	15	25	0.922	0.950	0.922	0.949	0.964
2	50	15	25	0.918	0.955	0.918	0.953	0.964
3	50	15	25	0.967	0.875	0.967	0.879	0.956
1	100	15	25	0.902	0.962	0.902	0.960	0.962
2	100	15	25	0.978	0.832	0.978	0.838	0.949
3	100	15	25	0.874	0.969	0.874	0.966	0.957
1	50	20	25	0.951	0.932	0.951	0.932	0.967
2	50	20	25	0.904	0.927	0.904	0.926	0.951
3	50	20	25	0.832	0.978	0.832	0.973	0.949
1	100	20	25	0.929	0.974	0.929	0.973	0.974
2	100	20	25	0.908	0.958	0.908	0.957	0.963
3	100	20	25	0.878	0.952	0.878	0.949	0.952
1	50	5	50	0.963	0.936	0.963	0.937	0.972
2	50	5	50	0.968	0.937	0.968	0.938	0.975
3	50	5	50	0.809	0.971	0.809	0.965	0.939
1	100	5	50	0.912	0.975	0.912	0.973	0.969
2	100	5	50	0.967	0.942	0.967	0.943	0.975
3	100	5	50	0.878	0.983	0.878	0.980	0.963
1	50	10	50	0.962	0.906	0.962	0.908	0.963
2	50	10	50	0.891	0.970	0.891	0.968	0.962
3	50	10	50	0.922	0.898	0.922	0.899	0.947

Table 5.14: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI).

INI	NS	NVS	NT	SEN	SP	AC	PPV	AUC
3	50	10	50	0.922	0.898	0.922	0.899	0.947
1	100	10	50	0.972	0.921	0.972	0.923	0.971
2	100	10	50	0.918	0.970	0.918	0.969	0.97
3	100	10	50	0.910	0.960	0.910	0.958	0.964
1	50	15	50	0.847	0.987	0.847	0.982	0.956
2	50	15	50	0.902	0.961	0.902	0.959	0.962
3	50	15	50	0.885	0.863	0.885	0.865	0.921
1	100	15	50	0.917	0.938	0.917	0.937	0.959
2	100	15	50	0.885	0.978	0.885	0.974	0.963
3	100	15	50	0.985	0.878	0.985	0.882	0.964
1	50	20	50	0.967	0.946	0.967	0.947	0.976
2	50	20	50	0.922	0.924	0.922	0.924	0.956
3	50	20	50	0.909	0.951	0.909	0.949	0.96
1	100	20	50	0.896	0.966	0.896	0.963	0.962
2	100	20	50	0.943	0.967	0.943	0.966	0.976
3	100	20	50	0.909	0.982	0.909	0.979	0.971
1	50	5	100	0.903	0.959	0.903	0.957	0.961
2	50	5	100	0.956	0.858	0.956	0.861	0.947
3	50	5	100	0.961	0.939	0.961	0.939	0.973
1	100	5	100	0.957	0.962	0.957	0.961	0.978
2	100	5	100	0.956	0.961	0.956	0.961	0.978
3	100	5	100	0.883	0.986	0.883	0.982	0.966
1	50	10	100	0.927	0.971	0.927	0.970	0.972
2	50	10	100	0.936	0.869	0.936	0.872	0.943
3	50	10	100	0.964	0.879	0.964	0.883	0.956
1	100	10	100	0.874	0.960	0.874	0.957	0.953
2	100	10	100	0.833	0.969	0.833	0.965	0.945
3	100	10	100	0.953	0.967	0.953	0.967	0.979
1	50	15	100	0.942	0.970	0.942	0.969	0.976
2	50	15	100	0.952	0.955	0.952	0.955	0.975
3	50	15	100	0.867	0.984	0.867	0.980	0.961
1	100	15	100	0.966	0.942	0.966	0.943	0.975
2	100	15	100	0.960	0.961	0.960	0.961	0.979
3	100	15	100	0.956	0.947	0.956	0.947	0.973
1	50	20	100	0.866	0.957	0.866	0.954	0.95
2	50	20	100	0.878	0.977	0.878	0.973	0.961
3	50	20	100	0.909	0.965	0.909	0.963	0.965
1	100	20	100	0.968	0.946	0.968	0.947	0.977
2	100	20	100	0.970	0.948	0.970	0.949	0.978
3	100	20	100	0.937	0.931	0.937	0.931	0.963

Table 5.15: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI).

INI	NS	NVS	NT	SEN	SP	AC	PPV	AUC
3	100	20	100	0.937	0.931	0.937	0.931	0.963
1	50	5	150	0.832	0.987	0.832	0.982	0.953
2	50	5	150	0.866	0.962	0.866	0.959	0.952
3	50	5	150	0.909	0.955	0.909	0.953	0.962
1	100	5	150	0.961	0.947	0.961	0.948	0.975
2	100	5	150	0.925	0.953	0.925	0.952	0.966
3	100	5	150	0.965	0.945	0.965	0.945	0.976
1	50	10	150	0.986	0.871	0.986	0.875	0.963
2	50	10	150	0.932	0.976	0.932	0.974	0.975
3	50	10	150	0.903	0.943	0.903	0.942	0.956
1	100	10	150	0.973	0.926	0.973	0.928	0.973
2	100	10	150	0.965	0.953	0.965	0.954	0.978
3	100	10	150	0.983	0.902	0.983	0.905	0.97
1	50	15	150	0.858	0.987	0.858	0.982	0.959
2	50	15	150	0.867	0.984	0.867	0.980	0.961
3	50	15	150	0.954	0.957	0.954	0.957	0.976
1	100	15	150	0.960	0.962	0.960	0.962	0.979
2	100	15	150	0.974	0.936	0.974	0.937	0.976
3	100	15	150	0.980	0.903	0.980	0.906	0.969
1	50	20	150	0.879	0.976	0.879	0.973	0.961
2	50	20	150	0.932	0.963	0.932	0.962	0.971
3	50	20	150	0.790	0.972	0.790	0.966	0.935
1	100	20	150	0.921	0.972	0.921	0.970	0.971
2	100	20	150	0.888	0.972	0.888	0.969	0.962
3	100	20	150	0.898	0.978	0.898	0.976	0.967
1	50	5	200	0.965	0.930	0.965	0.931	0.971
2	50	5	200	0.975	0.913	0.975	0.916	0.97
3	50	5	200	0.858	0.969	0.858	0.965	0.952
1	100	5	200	0.913	0.978	0.913	0.976	0.971
2	100	5	200	0.974	0.928	0.974	0.929	0.974
3	100	5	200	0.957	0.943	0.957	0.943	0.972
1	50	10	200	0.825	0.976	0.825	0.971	0.946
2	50	10	200	0.881	0.962	0.881	0.959	0.956
3	50	10	200	0.952	0.892	0.952	0.895	0.956
1	100	10	200	0.885	0.968	0.885	0.965	0.96
2	100	10	200	0.919	0.976	0.919	0.974	0.972
3	100	10	200	0.918	0.981	0.918	0.979	0.973
1	50	15	200	0.983	0.849	0.983	0.854	0.955
2	50	15	200	0.958	0.929	0.958	0.930	0.969
3	50	15	200	0.927	0.934	0.927	0.934	0.96

Table 5.16: AL using RF classifiers ensuring that set of samples added at each iteration is class balanced. Performance results measured by sensitivity (SEN), specificity (SP), accuracy (AC), Positive Predictive Value (PPV), Area under the ROC (AUC) of RF varying the number of trees (NT), the number of variables considered for the split at each node (NVS), the number of samples added in each iteration of the active learning algorithm (NS) and the number of initialization (INI).

INI	NS	NVS	NT	SEN	SP	AC	PPV	AUC
3	50	15	200	0.927	0.934	0.927	0.934	0.96
1	100	15	200	0.976	0.933	0.976	0.934	0.976
2	100	15	200	0.932	0.964	0.932	0.963	0.972
3	100	15	200	0.930	0.970	0.930	0.968	0.973
1	50	20	200	0.916	0.952	0.916	0.951	0.963
2	50	20	200	0.890	0.969	0.890	0.966	0.961
3	50	20	200	0.949	0.946	0.949	0.946	0.971
1	100	20	200	0.910	0.972	0.910	0.970	0.968
2	100	20	200	0.964	0.954	0.964	0.954	0.978
3	100	20	200	0.964	0.954	0.964	0.955	0.978

think that a greater sample (NS=100) increment is preferable; for RF the mean value of SP for NS=100 is 0.95, while for NS=50 it is 0.93. Similar values if we compare AUC values (0.957 for NS=50 and 0.966 for NS=100). Regarding the effect of the RF parameters, we find that the number of trees in RF has a very strong effect, especially if we compare the effect with 5 trees or 200 trees, obtaining great p-values performance for t-student test; $p < 6.36E - 277$, for sen, and $p < 4.24e - 231$ for AUC values. Considering the increase of the trees chosen with different values (5, 10, 25, 50, 100, 150, 200) we notice that as the forest is increased the p-values improve considerably, especially for the specificity and the AUC values as can be seen in table 5.18. Turning to the p-values for sensitivity, there is a significant performance increase when we go from 150 trees to 200. On the effects of the RF parameters, if we take into account the number of variables added in each split of the active learning (NS) algorithm we observe there is a very slight performance in p-values choosing different values as denoted in table 5.18; AUC values of 0.95 and 0.96 for NVS=5 and NVS = 20 respectively.

Parameter setting	**test**	**SEN**	**SP**	**AC**	**P**
Effect of the size of the sample (NS)					
NS=50 vs NS=100	Tt	$< 6.08e-07$	$< 1.66e-29$	$< 1.06e-31$	$<$
	WR	$< 3.80e-03$	$< 7.77e-21$	$< 4.18e-23$	$<$
Choice of number of trees (NT)					
NT=5 vs NT=10	Tt	0.0511	0.4326	0.4554	0.
	WR	0.2274	0.3794	0.4270	0.
NT=5 vs NT=25	Tt	0.4204	$2.30e-03$	$1.99e-03$	0.
	WR	0.6328	$1.73e-03$	$1.05e-03$	0.
NT=5 vs NT=50	Tt	0.2960	$6.47e-05$	$4.54e-05$	0.
	WR	0.4080	$5.86e-06$	$2.26e-06$	0.
NT=5 vs NT=100	Tt	0.6973	$4.61e-08$	$3.15e-08$	0.
	WR	0.8911	$7.84e-08$	$4.22e-08$	0.
NT=5 vs NT=150	Tt	0.0836	$2.66e-09$	$9.01e-10$	0.
	WR	0.1446	$1.68e-08$	$2.54e-09$	0.
NT=5 vs NT=200	Tt	$6.36e-277$	$1.43e-11$	$4.84e-16$	1.
	WR	$3.23e-41$	$9.61e-15$	$8.04e-19$	2.
Number of variables for NVS					
NS=5 vs NS=10	Tt	< 0.1728	$< 3.35E-03$	$< 2.38E-03$	$<$
	WR	< 0.3346	$< 1.32E-03$	$< 7.68E-04$	$<$
NS=5 vs NS=15	Tt	< 0.2261	$< 1.56E-02$	$< 1.22E-02$	$<$
	WR	< 0.3554	$< 1.96E-03$	$< 1.34E-03$	$<$
NS=5 vs NS=20	Tt	$< 1.48E-02$	< 0.0760	< 0.0582	1.
	WR	< 0.0630	$< 1.42E-02$	$< 9.27E-03$	$<$

Table 5.17: Assessment of impact of the parameter setting over results shown in Tables 5.9, 5.10. We give the p values of **Tt** = T-test, **WR** = Wilcoxon signed-rank test. Significance level $\alpha = 0.05$.

Parameter settings	**SEN**	**SP**	**PPV**	**AUC**
NS=50	0.917	0.929	0.917	0.957
NS=100	0.928	0.950	0.928	0.966
NT=5	0.920	0.928	0.920	0.957
NT=10	0.926	0.928	0.926	0.958
NT=25	0.920	0.939	0.920	0.961
NT=50	0.922	0.943	0.922	0.963
NT=100	0.917	0.948	0.917	0.963
NT=150	0.926	0.950	0.926	0.966
NT=200	0.926	0.941	0.926	0.963
NVS=5	0.919	0.935	0.919	0.959
NVS=10	0.922	0.943	0.922	0.963
NVS=15	0.922	0.941	0.922	0.962
NVS=20	0.926	0.939	0.926	0.963

Table 5.18: Average performance measures according to the used parameters (SEN, SP, PPV, AUC).

Comparing the increment between NS=5 and NS=10 we obtain p-values for AUC $p < 3.53e - 4$ for the t-student test and AUC values $p < 3.03e - 3$ for Wilcoxon rank sum test, while incrementing the values between NS=5 and NS=10 we obtain p-values for AUC of $p < 3.42e - 4$ and $p < 1.01e - 3$ for the t-student and Wilcoxon rank sum test respectively.

5.4 Conclusions

This Chapter contains a large number of experimental results that have been computed in different times of th Thesis. Initial results over the panoramic images were encouraging, but have been improved by the application of the machine learning approaches over the features extracted from the point cloud data, achieving average AUC over 0.95, which is an extraordinary result over such strongly imbalanced data. Overall, the application of active learning has been quite effective in achieving such excelelnt results.

Chapter 6

Conclusions

In this final chapter we present the summary and conclusions of the Thesis, including some comments on future lines of work. The core research effort of this work is aimed to the solution of a practical problem and therefore the conclusions are considered from a pragmatical point of view. The accomplished goals and some limitations of the work are discussed in section 6.1 and section 6.2, respectively. No conclusions chapter is complete without a list of future work guidelines that see their place in section 6.3.

6.1 Achievements

The main contribution of the thesis is a collection of methods in order to extract automatically the existing road markings on a road based on a previous training by means of AL techniques, whatever their typology (lines, figures, texts), from the data collected by an MMS sensor. Experimental validation of real life dataset obtained a highly accurate classification.

In this Thesis we present an AL approach to deal with the labeling of road landmarks in images obtained by an onboard sensor that includes LiDAR and image as well as positioning sensors for the purpose of detailed road signaling inventory that will involve a large volume of images and LiDAR points taken under very diverse light and atmospheric conditions, as well as road conditions. The underlying problem is a two class classification problem with strong class imbalance. Due to the cost of data labeling (image dataset or Lidar Point Cloud), the adaptation follows an AL approach, where the training set is built incrementally with the most informative dataset samples. We have explored the performance of two ensemble classifiers, namely the RF and the V-ELM ensemble of extreme learning machines. Our computational experiments comparing methods (only reported with panoramic images due to de huge time consu,ing of V-ELM), have found that the V-ELM improves RF in terms of SEN, PPV and AUC, performance measure that are more appropriate than the Accuracy for strongly class imbalanced datasets.

Among the analysed data (panoramas, images obtained from the intensity response of the LiDAR sensor, and LiDAR points), the best result in terms of SEN, PPV and AUC has been achieved from the LiDAR point clouds data. Not only have excellent performance results with SEN, PPV and AUC values averaging over 95%, but the computational times have been relatively fast, compared to the times required for the compuational experiments carried out with panoramic images and intensity images, even running the experiments on a slower machine (32 gb of Ram compared to the 90 gb used in the other experiments) and adding more trees to the model. In the imaging experiments, the average time was in the order of 5 hours, whereas with the lidar points, the average time was no more than 3 hours.

The different algorithms used, both in the Thesis and presented in the literature, are capable of extracting road markings with some effectiveness, but they never give a direct terrain coordinate. Normally, from the image pixel coordinate, the terrain coordinate can be estimated (planimetry only). In the case of the automatic extraction of road markings using AL algorithms, we have been able to obtain a good approximation of the horizontal signs coordinates, both in planimetry and altimetry.

As for the typology of the data, by treating the point clouds, we obtain a model of road markings of the continuous type, unlike the images, which are of the discrete type. Thanks to this type of model we can vectorise elements of interest or by means of topology rules help data integrity with vector data models, network analysis, proximity operations using vector data structures.

6.2 Limitations

Although the time consumed is not high, even taking into account the point cloud data, we believe that the computation time could be reduced, even achieving to perform the calculations in real time, which would be an important step towards the new autonomous vehicle industry.

Another limitation that we have seen and that we have not been able to solve directly is the removal of vegetation in the final result when we treat. Although the aforementioned noise could be eliminated by applying morphological operators and some standard computer vision tools, it would have been interesting to be able to eliminate the noise corresponding entirely to the vegetation. As for the images, we can speak of less noise, especially in the experiments carried out with the images obtained from the LIDaR intensity. On the other hand, when we apply the AL techniques on images, whether panoramic or images obtained from LiDAR point clouds, we are limited to detect only lines, not being able to detect the rest of the elements that exist on a road, whether they are texts or symbols.

In our segmentation process, we clearly distinguish two classes; elements that belong to road markings, and those that do not belong to this class. Within the road markings, we include all of them in the same typology. It would have been very interesting to be able to make more than two groups. In other words, it would not be a problem of two classes, but of as many classes as groups of road markings are catalogued.

Another limitation is that we are only able to extract horizontal signalling when dealing with LiDAR data. This is because due to the configuration of the MMS system (capture rate, capture mask, number of returns and other configurable parameters in the data capture) the sensor is mostly focused on the horizontal plane as we can see in the image 6.1

6.3 Future work

Future work guidelines come naturally from the limitations aforementioned in the above section. To begin with, it would be really interesting —and

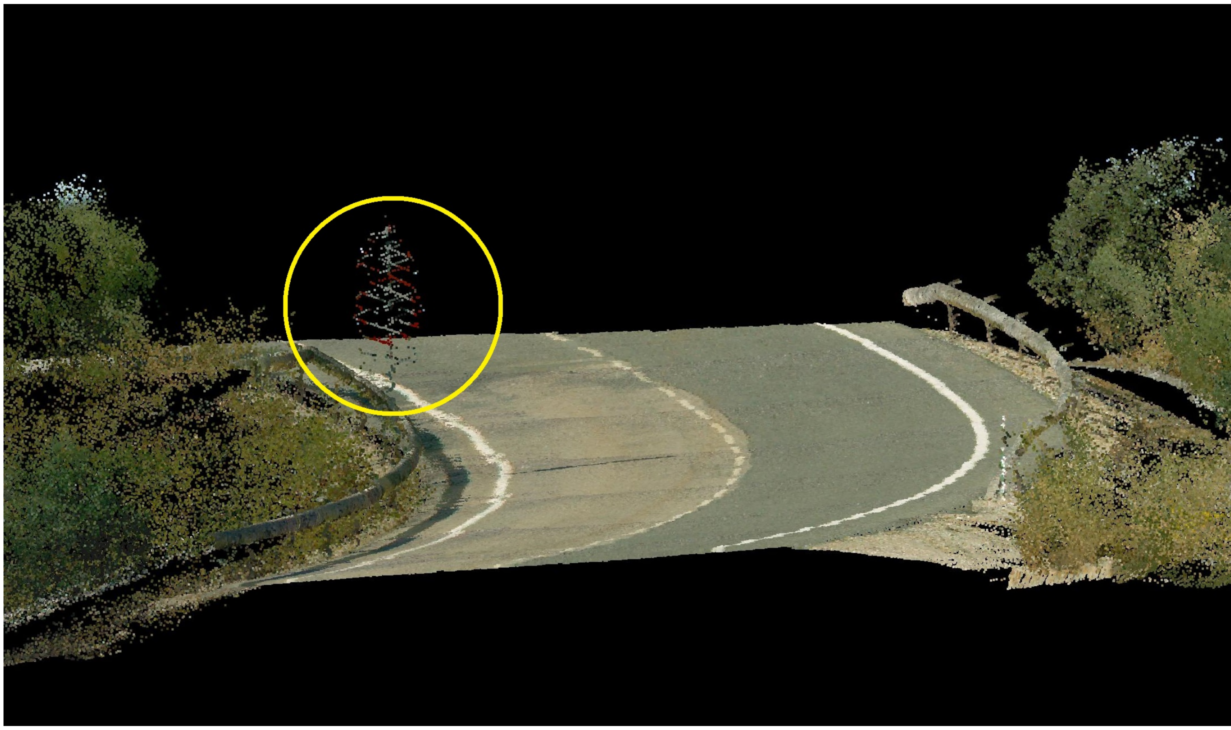


Figure 6.1: Limitations of the current MMS system configuration. The low density of points capturing the elements in the vertical plane is marked by a yellow circle.

equally challenging— to feed the learning algorithms with raw, unlabelled clouds of candidate points. Such solution would avoid the human operator the excruciating task of creating manually the input ground truth.

The thesis has worked independently with image data and LiDAR data. It would be very interesting, since both data, although obtained from different sensors, are obtained from the same system, to be able to complement them. Currently, in most cases, the image is only used to give colour to the point cloud, since, as we remember, the point cloud itself has no colour. The interesting thing about this combination of data would be to be able to detect and classify vertical signals. With the current configuration, the LiDAR sensor is only capable of capturing with a certain density the elements that are in the horizontal plane. On the other hand, the images, without changing the system configuration, only capture the elements in the vertical plane. Combining the two sensors, with the image part we could basically detect the elements, and with the LiDAR sensor we could position the elements in space.

Another common problem is noise. LiDAR point extraction generates a lot of noise, most of which is vegetation. Another challenge could be to be able to detect vegetation without the need of techniques outside the "AL" and to label it as a class of non-interest. For this purpose, remote sensing techniques could be used to add some kind of vegetation index to the features for corrective labelling. A widely used index is the NDVI (Normalized Difference Vegetation Index) which works with reflectivity indices and helps to discriminate between bare areas and the presence of water.

Bibliography

[1] Elsa Abbena, Simon Salamon, and Alfred Gray. *Modern differential geometry of curves and surfaces with Mathematica.* Chapman and Hall/CRC, 2017.

[2] Fatai Adesina Anifowose, Jane Labadin, and Abdulazeez Abdulraheem. Ensemble model of non-linear feature selection-based extreme learning machine for improved natural gas reservoir characterization. *Journal of Natural Gas Science and Engineering*, 26(0):1561–1572, 2015.

[3] José Juan Arranz Justel. *Diseño, optimización y análisis de sistemas basados en técnicas láser, para el modelado geométrico, registro y documentación, aplicados a entidades de interés patrimonial.* PhD thesis, Topografia, 2013.

[4] Arkaitz Artetxe, Borja Ayerdi, Manuel Graña, and Sebastian Rios. Using anticipative hybrid extreme rotation forest to predict emergency service readmission risk. *Journal of Computational Science*, 20:154 – 161, 2017.

[5] B. Ayerdi and M. Graña Romay. Hyperspectral image analysis by spectral–spatial processing and anticipative hybrid extreme rotation forest classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5):2627–2639, May 2016.

[6] Borja Ayerdi and Manuel Graña. Hybrid extreme rotation forest. *Neural Networks*, 52:33 – 42, 2014.

[7] Borja Ayerdi and Manuel Graña. Hyperspectral image nonlinear unmixing and reconstruction by elm regression ensemble. *Neurocomputing*, 174:299 – 309, 2016.

[8] Borja Ayerdi, Josu Maiora, Alicia d'Anjou, and Manuel Gra na. Applications of hybrid extreme rotation forests for image segmentation. *International Journal of Hybrid Intelligent Systems*, 11:13–24, 01 2014.

[9] Borja Ayerdi, Ion Marques, and Manuel Graña. Spatially regularized semisupervised ensembles of extreme learning machines for hyperspectral image segmentation. *Neurocomputing*, 149:373 – 386, 2015.

[10] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[11] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 25(3):727–745, 2014.

[12] Sebastiano Battiato, Giovanni Farinella, Antonino Furnari, and Giovanni Puglisi. *A Customized System for Vehicle Tracking and Classification*, pages 5–7. Number 2. Springer International Publishing, 2014.

[13] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[14] James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record*, 2291(1):61–71, 2012.

[15] Mathieu Brédif, Bruno Vallet, and B Ferrand. Distributed dimensonality-based rendering of lidar point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3/W3:559–564, 08 2015.

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

[17] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[19] Leo Breiman, Michael Last, and John Rice. *Random Forests: Finding Quasars*, volume 45, pages 243–254. Springer-Verlag, 2001.

[20] Roberto Brunelli. Template matching techniques in computer vision. 2008.

[21] Feilong Cao, Bo Liu, and Dong Sun Park. Image classification based on effective extreme learning machine. *Neurocomputing*, (0):–, 2012.

[22] Jingjing Cao, Sam Kwong, Ran Wang, Xiaodong Li, Ke Li, and Xiangfei Kong. Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing*, 149, Part A(0):275 – 284, 2015.

[23] Jiuwen Cao, Tao Chen, and Jiayuan Fan. Landmark recognition with compact bow histogram and ensemble elm. *Multimedia Tools and Applications*, 75(5):2839–2857, 2016.

[24] Darya Chyzhyk, Borja Ayerdi, and Josu Maiora. Active learning with bootstrapped dendritic classifier applied to medical image segmentation. *Pattern Recognition Letters*, 34(14):1602 – 1608, 2013.

[25] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 5 1994.

[26] A. Davison. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1052-1067*, 2007.

[27] Konstantinos G Derpanis. Overview of the ransac algorithm. 2010.

[28] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, Jun 1989.

[29] Xiang guo Zhao, Guoren Wang, Xin Bi, Peizhen Gong, and Yuhai Zhao. Xml document classification based on elm. *Neurocomputing*, 74(16):2444 – 2451, 2011.

[30] Werner Gurtner and Lou Estey. Rinex-the receiver independent exchange format-version 3.00. *Astronomical Institute, University of Bern and UNAVCO, Bolulder, Colorado.*, 2007.

[31] K. Hammoudi, F. Dornaika, and N. Paparoditis. Generating virtual 3d model of urban street facades by fusing terrestrial multi-source data. *2011 7th International Conference on Intelligent Environments*, pages 330–3, 2011.

[32] K. Hammoudi, F. Dornaika, B. Soheilian, and N. Paparoditis. Extracting Outlined Planar Clusters of Street Facades from 3D Point Clouds. In *Proceedings of the 2010 Seventh Canadian Conference on Computer and Robot Vision (CRV 2010)*, pages 122–9, 2010. 2010 Seventh Canadian Conference on Computer and Robot Vision (CRV 2010), 31 May-2 June 2010, Ottawa, ON, Canada.

[33] Karim Hammoudi, Fadi Dornaika, Bahman Soheilian, Bruno Vallet, and Nicolas Paparoditis. Generating occlusion-free textures for virtual 3d model of urban facades by fusing image and laser street data. *IEEE Virtual Reality Conference 2012 Proceedings*, page IEEE; IEEE Comp Soc; IEEE Visualizat & Graph Tech Comm, 2012.

[34] Min Han and Ben Liu. Ensemble of extreme learning machine for remote sensing image classification. *Neurocomputing*, 149, Part A(0):65 – 70, 2015.

[35] L Hansen and P Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 10 1990.

[36] A. Hata and D. Wolf. Road marking detection using lidar reflective intensity data and its application to vehicle localization. In *17th Interna-*

tional IEEE Conference on Intelligent Transportation Systems (ITSC), pages 584–589, 2014.

[37] Steven Hoi, Rong Jin, Jianke Zhu, and Michael Lyu. Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems*, 27(3):1–29, 5 2009.

[38] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16–18):3460 – 3468, 2008.

[39] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489 – 501, 2006.

[40] Juan Iglesias, Ender Konukoglu, Albert Montillo, Zhuowen Tu, and Antonio Criminisi. *Combining Generative and Discriminative Models for Semantic Segmentation of CT Scans via Active Learning*, pages 25–36. Springer Berlin Heidelberg, 2011.

[41] Asier Izquierdo, Jose Manuel Lopez-Guede, and Manuel Graña. Road lane landmark extraction: A state-of-the-art review. In Hilde Pérez García, Lidia Sánchez González, Manuel Castejón Limas, Héctor Quintián Pardo, and Emilio Corchado Rodríguez, editors, *Hybrid Artificial Intelligent Systems*, pages 625–635, Cham, 2019. Springer International Publishing.

[42] Luo Juan and Gwun Oubong. *SURF applied in panorama image stitching*, pages 495–499. IEEE, 7 2010.

[43] Wang Jun, Tao Wu, and Zhongyang Zheng. Lidar and vision based pedestrian detection and tracking system. *Proceedings of 2015 Ieee International Conference on Progress in Informatcs and Computing (ieee Pic)*, pages 118–122, 2015.

[44] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek,

D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, June 2011.

[45] Kwangyong Lim, Yongwon Hong, Minsong Ki, Yeongwoo Choi, and Hyeran Byun. Vision-based recognition of road regulation for intelligent vehicle. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1418–25, 2018.

[46] Nan Liu, Jiuwen Cao, Zhiping Lin, Pin Pek, Zhi Koh, and Marcus Ong. Evolutionary voting-based extreme learning machines. *Mathematical Problems in Engineering*, 2014:1–7, 2014.

[47] Xiaolong Liu and Zhidong Deng. Segmentation of drivable road using deep fully convolutional residual network with pyramid pooling. *Cognitive Computation*, 10(2):272–281, 2018.

[48] David G Lowe et al. Object recognition from local scale-invariant features. In *ICCV 1999*, number 2, pages 1150–1157, 1999.

[49] Huan Luo, Cheng Wang, Chenglu Wen, Ziyi Chen, Dawei Zai, Yongtao Yu, and Jonathan Li. Semantic labeling of mobile lidar point clouds via active learning and higher order mrf. *IEEE Transactions on Geoscience and Remote Sensing*, 56(7):3631–3644, 2018.

[50] Josu Maiora, Borja Ayerdi, and Manuel Graña. Random forest active learning for aaa thrombus segmentation in computed tomography angiography images. *Neurocomputing*, 126:71 – 77, 2014.

[51] A. Mancini, E. Frontoni, and P. Zingaretti. Automatic road object extraction from mobile mapping systems. In *Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 281–286, July 2012.

[52] Pabitra Mitra, B Uma Shankar, and Sankar Pal. Segmentation of

multispectral remote sensing images using active support vector machines. *Pattern Recognition Letters*, 25(9):1067–1074, 7 2004.

[53] H. L. Nguyen, D. Belton, and P. Helmholz. Planar surface detection for sparse and heterogeneous mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 151:141–161, 2019.

[54] Harshal Patil and S.S. Deshmukh. Homography estimation using ransac. *IJREAT International Journal of Research in Engineering & Advanced Technology*, 1(3):1–4, 2013.

[55] Cristiano Premebida, Oswaldo Ludwig, and Urbano Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9):696–711, 2009.

[56] I Puente, H González-Jorge, J Martínez-Sánchez, and P Arias. Abstract: Mobile surveying is currently one of the most popular topics in the lidar industry. the collection of highly precise point cloud data is provided by laser scanning systems on moving platforms with an integrated navigation solution. the potential of lidar based mobile surveying technology is now well proven. this article introduces an analysis on the current performance of some outstanding mobile terrestrial laser scanning systems. in this work, an overview of the positioning, scanning and imaging devices integrated into these systems is also presented. as part of this study, a systematic comparison of the navigation and lidar specifications provided by the manufacturers is provided. our review suggests that mobile laser scanning systems can mainly be divided into two categories (mapping and surveying) depending on their final purpose, accuracy, range and resolution requirements. *Measurement*, 46(7):2127–2145, 2013.

[57] S. Rogers. Creating and evaluating highly accurate maps with probe vehicles. In *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493)*, pages 125–130, 2000.

[58] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2010.

[59] Linda Shapiro and George Stockman. *Computer vision*, volume 3. Prentice Hall New Jersey, 2001.

[60] L. Smadja, J. Ninot, and T. Gavrilovic. Road extraction and environment interpretation from lidar sensors. volume 38, pages 281–286, 2010. cited By 28.

[61] Yimo Tao, Zhigang Peng, Bing Jian, Jianhua Xuan, Arun Krishnan, and Xiang Sean Zhou. *Robust Learning-Based Annotation of Medical Radiographs*, volume 5853 of *Lecture Notes in Computer Science*, pages 77–88. Springer Berlin Heidelberg, Berlin / Heidelberg, 2010.

[62] D Tuia, E Pasolli, and W Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242, 9 2011.

[63] Devis Tuia, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Munoz-Mari. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 6 2011.

[64] R. Wan, Y. Huang, R. Xie, and P. Ma. Combined lane mapping using a mobile mapping system. *Remote Sensing*, 11(3), 2019.

[65] Dianhui Wang and Monther Alhamdoosh. Evolutionary extreme learning machine ensembles with size control. *Neurocomputing*, 102(0):–, 2012.

[66] Jianwei Wu, Wei Yao, and Przemyslaw Polewski. Mapping Individual Tree Species and Vitality along Urban Road Corridors with LiDAR and Imaging Sensors: Point Density versus View Perspective. *Remote Sensing*, 10(9), 2018.

[67] William A. Young, Scott L. Nykl, Gary R. Weckman, and David M. Chelberg. Using voronoi diagrams to improve classification performances when modeling imbalanced datasets. *Neural Computing and Applications*, 26(5):1041–1054, 2015.

[68] Xin Zeng, Shunsuke Araki, and Ken'ichi Kakizaki. An improved extraction method of individual building wall points from mobile mapping system data. In *2016 Sixth International Conference on Innovative Computing Technology (intech)*. IEEE, 2016.

[69] Yong Zhang, Panpan Fu, Wenzhe Liu, and Guolong Chen. Imbalanced data classification based on scaling kernel-based support vector machine. *Neural Computing and Applications*, 25(3):927–935, 2014.

[70] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. In this paper, the relationship between the ensemble and its component neural networks is analyzed from the context of both regression and classification, which reveals that it may be better to ensemble many instead of all of the neural networks at hand. this result is interesting because at present, most approaches ensemble all the available neural networks for prediction. then, in order to show that the appropriate neural networks for composing an ensemble can be effectively selected from a set of available neural networks, an approach named gasen is presented. gasen trains a number of neural networks at first. then it assigns random weights to those networks and employs genetic algorithm to evolve the weights so that they can characterize to some extent the fitness of the neural networks in constituting an ensemble. finally it selects some neural networks based on the evolved weights to make up the ensemble. a large empirical study shows that, compared with some popular ensemble approaches such as bagging and boosting, gasen can generate neural network ensembles with far smaller sizes but stronger generalization ability. furthermore, in order to understand the working mechanism of gasen, the bias-variance decomposition of the error is. *Artificial Intelligence*, 137:239–263, 2002.

[71] Djemel Ziou and Salvatore Tabbone. Edge detection techniques - an overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559, 1998.