MACHINE LEARNING CLASSIFICATION OF OBJECTS FROM RGB IMAGES AND POINT CLOUDS OBTAINED FROM A MLS SYSTEM IN RAILROAD ENVIRONMENT

ATHITHYA SEETHALAKSHMI LOGANATHAN

June, 2020

SUPERVISORS:

Dr. Ville. V. Lehtola
Dr. ir. S.J. Oude Elberink

MACHINE LEARNING CLASSIFICATION OF OBJECTS FROM RGB IMAGES AND POINT CLOUDS OBTAINED FROM A MLS SYSTEM IN RAILROAD ENVIRONMENT

ATHITHYA SEETHALAKSHMI LOGANATHAN

Enschede, The Netherlands, June, 2020

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. Ville. V. Lehtola
Dr. Ir. S.J. Oude Elberink

THESIS ASSESSMENT BOARD:
Prof. Dr. Ir. M.G. Vosselman (chair)
Dr. Eetu Puttonen, Geospatial Research Institute (FGI /NLS), Finland
Dr. Ville. V. Lehtola
Dr. Ir. S.J. Oude Elberink

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

ABSTRACT

The data fusion of RGB images and 3d point clouds captured from a Mobile Laser Scanning(MLS) platform is gaining more research interest recently due to its application in various fields such as railway infrastructure management, road traffic infrastructure maintenance, and autonomous vehicle navigation. Especially in the railway industry, periodical surveys are undertaken by mounting the MLS platform to the rails. And the data captured from the surveys are further post-processed to obtain a railway infrastructure model. However, the creation of such infrastructure models is time-consuming and involves a lot of manual labor, as understanding and identifying objects in the scene is quite difficult. The overall time taken to create such models can be drastically reduced by introducing a machine-learning algorithm to perform the classification task. This research proposes a novel framework that leverages both RGB images and 3D point clouds for efficient inventory mapping. The research segments are – object detection and classification, Character recognition from detected objects, and Positioning the detected objects in the point clouds. For this study, two objects of interest are selected.; they are - kilometer markers and Signals. Training samples for these objects are created from the RGB images, and a machine learning classifier is trained using these samples for object detection. As a result, the object's location in the RGB images is identified. Since the kilometer marker contains digit values in it, the detected kilometer marker images are further processed for character recognition. The task of character recognition is performed using a deep learning model that is being trained to recognize digits from a natural environment. The recognized kilometer values attribute to the semantic information of the objects.

Furthermore, to obtain the 3d geometry, the objects detected in the image are reprojected to 3d point clouds using internal spatial parameters as they are captured from the same platform. With some processing, the geometry and its subsequent point cloud structure of the objects are obtained and thus locating the object in the real world accurately. The different classifiers used in this study are tested for their performance in classifying the objects, and their accuracy is assessed. Among them, the multi-class SVM with RBF kernel is selected for further object detection. It produces an overall accuracy of 96%. Similarly, the character recognition model is evaluated for its performance, and the accuracy is assessed. The algorithms are implemented in python, and the efficiency of the framework is evaluated.

Keywords: Railway infrastructure, Mobile mapping, 3D point clouds, Machine learning, Deep learning, Object detection, Character recognition

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my first supervisor, Dr. Ville. V. Lehtola, for his expert guidance, motivation, and encouragement throughout my research. His thoughts, insightful discussion and valuable comments throughout the thesis journey helped me to shape this research to completion

I would like to thank my second supervisor, Dr. Ir. S.J. Oude Elberink for his support and kindness at every stage of my research work.

I am indebted to my chair, Prof. Dr. Ir. M.G. Vosselman for his suggestions and critical evaluation during the research phase of the thesis.

I would like to thank drs. J.P.G Wan Bakx for his support and guidance in my academic journey at ITC.

I would also like to extend my thanks to Mr. Pete Apostle, Mrs. Sudan Xu, Mrs. Drozdz, Magdalena from Fugro, Utrecht, for their technical guidance.

I am also thankful to ITC for awarding me the ITC Excellence scholarship, which helped me with resources to finish the research successfully. And I would like to thank the entire teaching faculty of ITC, who helped me in various ways throughout the study programme.

Most of all, I would like to thank my friends Sai, Akhil, Ani and Mohan for their endless support and Manushi for helping me stay positive during the thesis period.

Finally, I would like to thank my mother, aunt and sister for their unconditional love, support and trust on me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and Motivation

Public railway infrastructure requires proper maintenance of the tracks and traffic control system for efficient asset management. The top priority of railway track management is ensuring public safety and maintaining the uninterrupted flow of railway traffic without any interruptions. According to the **EIM-EFRTC-CER Working Group (2012)**, the European countries are estimated to spend around 15-25 billion dollars annually for the proper maintenance of the railway infrastructure consisting of 300,000 km of track. However, the cost of maintenance depends upon the traffic density of the region and the train operation path. With the use of Mobile Laser Scanning (MLS) platform mounted on the trains, periodical surveys are conducted to model the railway corridor to plan maintenance. Since the deterioration in the path is commonly identified using change detection, all the tracks are covered during the survey **(Lidén, 2015)**. Later on, the acquired MLS data is used to build a three dimensional model of the railway environment, which can be used to analyze the track positioning and asset inventories. The major companies involved in the railway infrastructure management rely on such techniques to produce data about the railway environment, which includes the location signal towers, railway tracks, sign markers, wires, and other objects of interest. So, it is necessary to maintain a database of such information, where it is easier to monitor the railway path and identify any potential deformity that could affect the flow of traffic

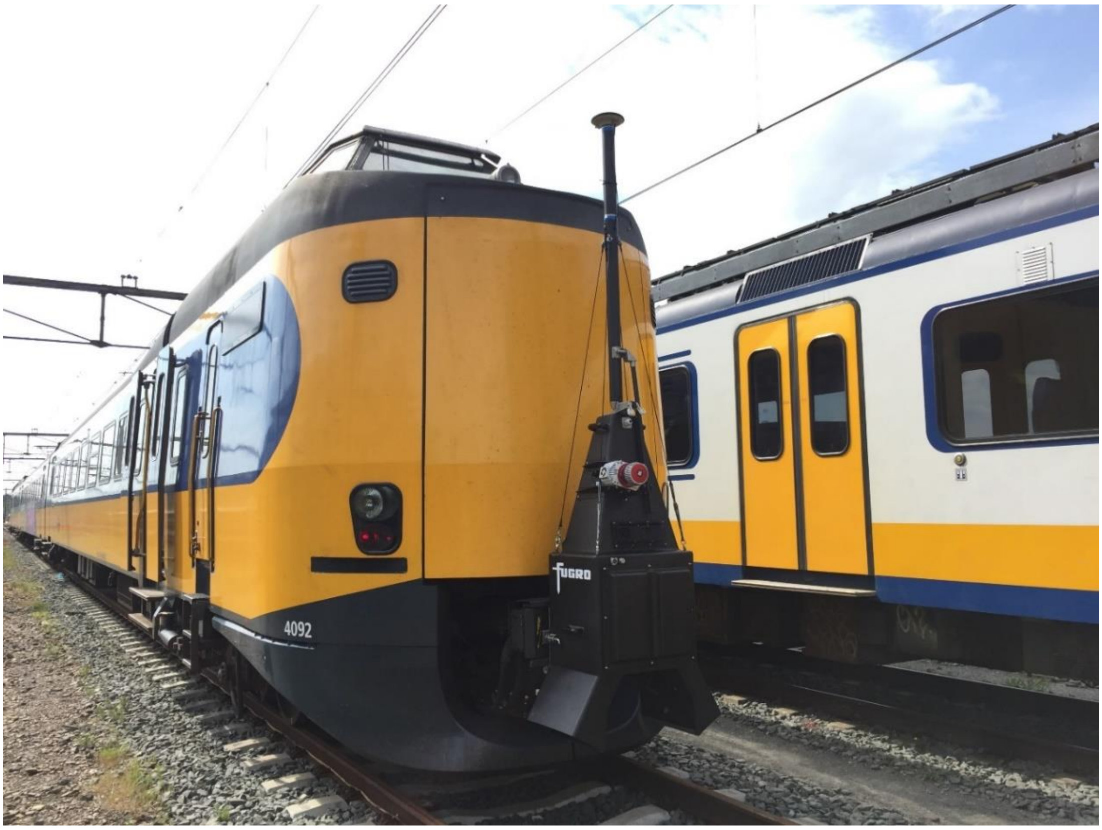


Figure 1-1 - Mobile Laser Scanner mounted on the Rail (Source -Fugro)

Similarly, monitoring the growth of vegetation and tree cover surrounding the railway corridor is an essential criterion for proper maintenance. For a smooth operation, the signals and sign markers along the track should be visible without any obstacles on sight **(Morgan, 2009)**. As the data gathered from the MLS unit contain all elements captured by the sensor, we can extract the features such as ground, building and labeled objects based on their geometric shape **(Babahajiani, Fan, & Gabbouj, 2015)**. However, the classification of these objects of interest requires a separate methodology based on their object characteristics. Linear track elements, like railway track and catenary wires, are extracted using parameter estimation **(Oude Elberink, Khoshelham, Arastounia, & Díaz Benito, 2013)**. Furthermore, in some cases, the signal intensity is used to delineate the linear elements from the entire data using its geometric characteristics. Likewise, the non-linear track objects like electric poles, signals, and ground surfaces are extracted by segmenting the ground surface separately and comparing different height passes of the overlap **(Zhu, Jaakkola, & Hyyppä, 2013)**.

The modern MLS system consists of a laser scanner, camera, GNSS system, and IMU system **(Y. Wang et al., 2019)**. The laser scanner, along with the GNSS and IMU unit, produces a high-density point cloud. The camera captures the RGB information of the terrain, which are, for example, used to assign color values to the point clouds**(Che, Jung, & Olsen, 2019; H. Wang et al., 2012)**. Most of the time, the point clouds tend to be more accurate than images or videos captured by the cameras. However, the first detection of objects of interest that are designed to be seen by the human eye, such as sign markers and signals, is more conveniently done by image data. And, the sign markers/signals are best distinguished from the environment, not by their geometry, but by their color pattern.

The primary motivation of this research is to develop a data fusion method that utilizes both imagery and point clouds. First, the RGB information is used to color pattern detection. Then the high-density MLS point cloud is used for the extraction of the exact geometric shape for the detection and classification of objects of interest in the railway corridor. Here, the objects of interest are sign markers and signals. Currently, the location of the sign markers and signals are labeled manually, and the point clouds are classified with much post-processing. So, creating a 3D environment of the railway corridor with these objects of interest becomes an expensive and time-consuming project. Thus developing a methodology to automate this process helps in decreasing the cost and speeds up the process of modeling the railway corridor, which helps in producing such models more frequently. A good starting point is to develop a primary classification algorithm that can recognize the objects of interest(OOIs), i.e., sign markers and signals from the images. By understanding the geometric characteristics of these objects, a combination of kernel function (e.g., gaussian, sigmoid, RBF) can be built that works along with a machine learning classifier**(Kari, Gao, Tuluhong, Yaermaimaiti, & Zhang, 2018)**.

One of the essential criteria while training a model to classify OOIs is the availability of the labeled dataset for building a robust classifier **(Ben-Shabat, Lindenbaum, & Fischer, 2018)**. As our OOIs are smaller in size and their occurrence in the railway path is not continuous, generating a set of labeled data is quite a challenge. We can minimize the dependency of the labeled dataset to some extent by dimensionality reduction and data augmentation methods. Still, it does not account for the planarity and geometric structure of the objects of interest **(Babahajiani et al., 2015)**. Hence, an optimal algorithm is necessary for extracting such features that consider all these elements.

The data fusion of both point clouds and RGB information for object detection is still considered complex**(Krispel, Opitz, Waltner, Possegger, & Bischof, 2019)**. Due to its complexity, many methods use single modal input **(Lang et al., 2019; Wu, Wan, Yue, & Keutzer, 2018; Wu, Zhou, Zhao, Yue, & Keutzer, 2019)** or just use the benefits of the multi modalities after single modal approach **(Qi, Liu, Wu,**

Su, & Guibas, 2018; Shi, Wang, & Li, 2019). Alongside the strategies mentioned above, identifying the exact real-world location of the objects of interest is a crucial factor. For example, using the camera calibration information and GNSS/IMU data, it is possible to locate the objects in the point cloud after preliminary object detection but, the bounding box region of the objects of interest depends upon the classification output **(Hosang, Benenson, & Schiele, 2017)** and has to be very accurate to match with the point clouds. So, a secondary algorithm is required that uses image pixels to draw epipolar lines in the 3D space forming a bounding box in the point clouds. By doing so, the object separated from the point clouds that are close to this bounding box would yield the final extracted object. Considering the above factors, a two-step process of object detection seems a good fit, which includes initial detection and localization of the objects from RGB images followed by positioning in point clouds. This approach not only leverages data fusion but also helps in the resolve of identifying OOIs.

The purpose of this research is to develop a methodology that allows the detection and classification of sign markers and signals in a railway environment. The proposed algorithm/ methodology works by recognizing the semantic information of the objects from the RGB images taken from the camera sensor and the geometric structure of the objects from the MLS point clouds. By iterative process, the classification accuracy is increased by interchanging the parameter values. This algorithm will help in reducing the time taken for modeling the railway corridor with all these OOIs. The classified output of point cloud data can be used to understand the level of maintenance action required for the specific path of railway tracks, thus attributing to railway infrastructure maintenance.

1.2. Problem Statement

The problem is dual-fold; that is, different signs cannot be distinguished from each other using just geometry of the sign markers, while the exact shape and location of the OOI cannot be reliably obtained from the imagery. Therefore, it is necessary to use both the RGB data and the MLS point cloud to solve the problem. Furthermore, this research contributes to solving,

- The problem of obtaining a reasonable size of the training sample for classification, which is difficult due to the sparse occurrence of OOIs in the dataset.
- The implications of image orientation and range differences of the detected objects while performing character recognition.
- And finding an optimal solution to get the location of the OOIs from RGB images and their corresponding point clouds for 3d modeling.

Table 1-1 Objects of Interest (Image Source - Fugro.)

S. No	Objects of Interest	Information	Image
1.	Kilometer Markers	The markers that indicate kilometers	
2.	Signals	The traffic signal poles planted near the rail track	

1.3. Research Identification

1.3.1. Research Objective

The primary objective of the proposed research is to develop an algorithm that can classify the OOIs using the RGB information, recognize the characters present in OOIs and locate them in the 3d point clouds obtained from an MLS platform. The objects of interest chosen for this study are listed above in **Table** 1-1. The present study attempts to analyze the pixel structure of these objects and their subsequent point cloud clusters. Thus, obtaining reliable results in locating the objects of interest, ensuring accuracy, and robustness of the model. The sub-objectives are listed with their specific research questions.

1.3.2. Research Questions

The research objective is further divided into the following sub-objectives with its corresponding research questions:
Sub-objective 1: To develop an algorithm that utilizes the information from RGB Images for object detection and classification

1. How to cope with the low quantity of training data, and what are the mitigation methods?
2. What are the parameters to be considered while performing feature extraction?
3. Which kernel function of SVM yields better results in the classification of objects?

Sub-objective 2: To develop an algorithm that recognizes characters from the detected objects

1. How can characters on sign markers be extracted from images of varying orientations and range?
2. How to estimate the performance accuracy of the recognition models?

Sub-objective 3: To extract the accurate geometric shape of the object and position it.

1. How to perform a reliable mapping for the OOI from the image plane into the point cloud?

1.4. Innovation aimed at

Previous studies have used either point clouds**(Niina et al., 2018; H. Wang et al., 2012)** or RGB images**(Loussaief & Abdelkrim, 2018; Zhao et al., 2019)** for object detection, which performs well with linear objects. Similarly, by using the combination of point cloud and RGB images, the road traffic signs are extracted by **Soilán et al., (2016)**. But they detected the objects in the point clouds first and then reprojected on the images, and their approach did not account for character recognition of the objects. Also, such an approach is computationally expensive as large point cloud data are processed first.

The proposed study aims to leverage the multi modalities of different data sources. It presents a novel approach that involves object detection and classification in RGB images, character recognition from the detected objects, and positioning the detected objects in point clouds, thus extracting both the semantic and geometric information of the objects of interest. A similar approach is seen to be modeled in the field of advanced computer vision and pattern recognition, making them intricate and expensive. On the contrary, this methodology can be reproduced in a different dataset captured from the different regions, thus making it more suitable for commercial purposes. At the moment, solutions like this one are not available for the intended field of study.

1.5. Thesis Structure

This thesis consists of seven-chapter. Chapter 1 discuss the motivation and problem statement of this research. Chapter 2 reviews the literature that describes the theoretical principles and studies related to this research. Chapter 3 describes the data and environments used in this research. Chapter 4 thoroughly explains the research design and methodology of the thesis. Chapter 5 presents the results, which is continued by Chapter 6 for further discussion and critical findings. Chapter 7 concludes the thesis with conclusions and future recommendations.

2. LITERATURE REVIEW

This section provides a comprehensive review of the principles of MLS and various methodologies used to extract objects from MLS point clouds specifically focused on the railway environment. Moreover, literature related to object detection methods using images and point clouds are studied as well. This section is concluded with a brief discussion about the existing research on Support Vector Machines(SVM) for classification.

2.1. Mobile Laser Scanning

A mobile laser scanning platform (MLS) consists of a 2d laser scanning unit, IMU (Inertial Measurement Unit), GNSS (global navigation satellite system), and optionally a camera. The laser scanner produces point cloud information that includes various aspects of the surrounding environment, so it needs to be further processed for modeling. They also have the Intensity values of the laser and the RGB color information, which can be further used during classification**(Diaz et al., 2013)**. These point clouds are the basis of elevation models and 3D cad modeling**(Hemmes, 2018)**. Usually, the MLS system is mounted on Vehicles, Rails, boats, etc., for mapping the 3d surface information while moving. Due to its short-range when compared to airborne platforms, the MLS platform provides dense point clouds**(Y. Wang et al., 2019)**. Thus, they have a wide array of usages, such as building information modeling, Infrastructure mapping, utility mapping, and so on. Among these, one of the notable use cases of MLS is in autonomous driving where they are coupled with high precision RGB camera sensor and RADAR for real-time pedestrian detection**(Lamon, Stachniss, & Triebel, 2006; Y. Wang et al., 2019)**.

2.2. Extracting objects of interest from 3d MLS point clouds

Linear structures such as roads, tracks, and wires are derived using the Lidar point cloud data. For instance, **Pu et al., (2011)** developed an automated method that works by recognizing and extracting the point cloud structures of road inventories. Similarly, **Himmelsbach et al. (2009)** performed a real-time object detection inside road boundaries by segmenting the data into an occupancy grid and used a supervised learning classifier for object detection.

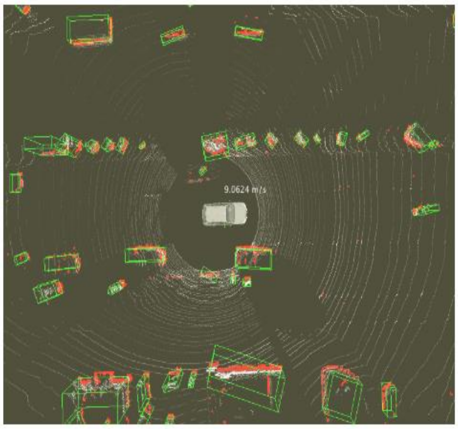


Figure 2-1 - Occupancy grids with detected objects in the green bounding box (Source - Himmelsbach et al., (2009))

The point cloud data collected from the MLS platform contain detailed geometric information of the ground objects captured from the real world. With suitable object recognition methods, these data can be classified to produce a 3d representation of the real world. The authors **Che, Jung, & Olsen, (2019)**, conducted an exploratory study in understanding different classification methods using the dataset acquired from both mobile and aerial platforms. In general, they list out three essential criteria for organizing the point cloud data (MLS) for detecting ground objects; they are rasterization of the MLS data, segmenting into scanline, and 3d point-based method. By organizing the MLS data using such methods, the computational burden is drastically reduced. According to **Babahajiani et al. (2015)**, the voxelization of point clouds helps in obtaining better results with feature extraction and labeling. They proposed a two-stage method, which includes a mix of both supervised and unsupervised classification techniques for extracting the objects of interest from the MLS dataset. **Babahajiani et al., (2015)** furthermore emphasized on the robustness of the resultant classifier as it could produce far accurate classification results with less training time.

In addition to the above studies, **Li et al. (2019)** proposed a novel approach for the semantic segmentation of the pole like road furniture, which includes labeling street lights, traffic lights, traffics signs attached to the pole structures. After a series of interpretation and decomposition, the features are classified using a knowledge-based approach and different machine learning techniques. And it is found that the Random Forest classifier performed better comparatively even when the elements are not distinctive enough.

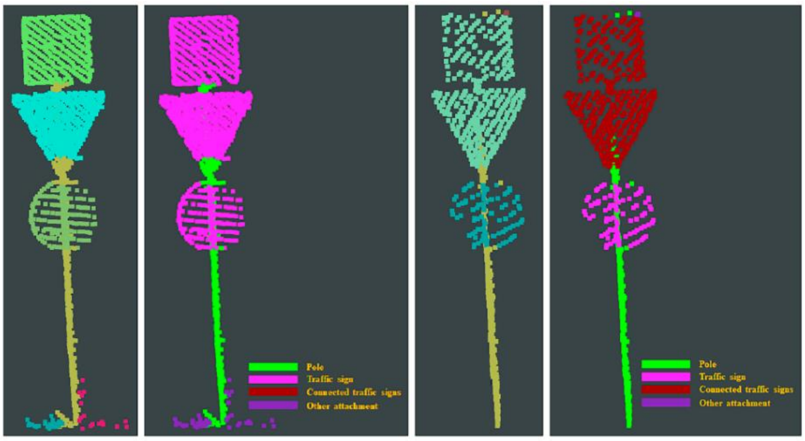


Figure 2-2 The decomposition and interpretation of road furniture

2.3. Extracting objects of interest in a railway environment from a 3d point cloud data

There are many studies in the past which performed object extraction from a Mobile Laser Scanning platform in a railway environment**(Gézero & Antunes, 2019; Hackel et al., 2015; Leslar et al., 2010)** In most cases, the objects of interest are Railway tracks, cable wires, etc. **(Table 2-1).**

Authors	Objects of Interest	Method
Elberink & Khoshelham (2015)	Centrelines of the railway track	a. Fine rail detection using geometric properties such as shape b. Coarse rail detection using height histogram. c. Piecewise 3D modeling
Beger et al. (2011)	Centrelines of the railway track	a. Data fusion of High-resolution ortho imagery and Lidar point cloud data. b. Object-oriented analysis and quadtree segmentation
Arastounia, (2012, 2015)	Railway track, catenary cable, and contact cable	Region growing, Template matching, k-dtree, and PCA
Gézero & Antunes (2019)	Railway track, Ballast top, and bottom brake lines	a. Points with the same scan angle are linear b. Douglas-Peucker algorithm for smoothening the resultant trajectory line
Leslar, Perry, & McNease (2010)	Power Line and railway platform	a. Semi-automatic tracking features, classification, and vectorization b. Clearance envelops in both horizontal and vertical directions.

Hackel et al. (2015)	Railway tracks and railway turnout	a. Shape matching SVM classifier with a piecewise linear element model. b. Fine-tuning and evaluation for height(longitudinal) consistency

Table 2-1 - List of related works in extracting objects of interest from 3d Point clouds in a railway environment

The above studies display the usage of MLS data in railway asset inventory mapping with rich spatial information and accuracy. Still, there exist some parts of the process, which involves manual interpretation of the parameters **(Oude Elberink et al., 2013)**. Especially when it comes to classifying objects of interest, with a very sparse occurrence ratio, it is highly complex to train a model to identify such features, and there exists no specific go-to approach. So, an image-based approach coupled with point clouds is required to classify these objects of interest.

2.4. Object Detection and Classification

In general, the process of classification consists of three different steps; they are - the selection of an informative region, feature extraction, and classification. Methods such as region proposal networks and candidate region selection methods are used to identify the informative region from the images**(Zhao et al., 2019)**. Once the informative regions are identified, the features are extracted using techniques such as SIFT, SURF, or using deep learning models such as R-CNN. As a final process, the feature obtained is further scored and classified using a pre-trained model such as Support vector machines to get positive and negative regions in the images**(He et al., 2015).**

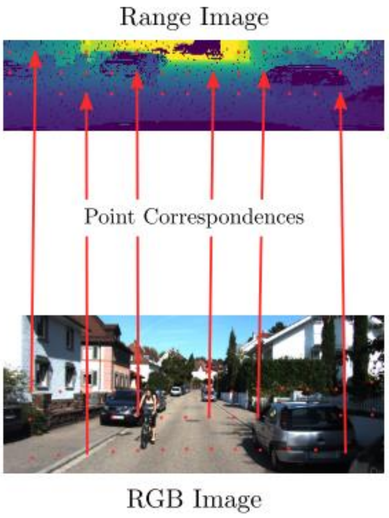


Figure 2-3 RGB/LiDAR calibration to establish point correspondences (Source- Krispel et al., (2019))

Furthermore, in a method proposed by **Chen et al. (2015)**, the images are used along with the point cloud information for feature selection and detection where depth perception is obtained by introducing 3d proposal networks. Whereas, these methods do not account for localization of the bounding boxes if not for stereo images. Likewise, while dealing with multi-modal data of point cloud segmentation, **Krispel et al., (2019)** propose an approach that utilizes both point clouds and RGB information effectively. The model uses the data from different sources with the linkage of setup calibration using range image and point cloud information. As a result, the proposed model FuseSeg yielded 18% better results in point cloud segmentation on the Kitti dataset. Even though the results of these models are quite promising, the model performance for segmenting objects of interest with a low quantity of labeled dataset is uncertain and not comparative. Likewise, in the study conducted by **Xu et al., (2019)**, fusion algorithms are used to fuse the features from the LIDAR point clouds and its corresponding RGB images. The proposed algorithm works by producing multi-level feature maps from which non-linear fusion takes place.

For classifying smaller objects of interest, kernel methods are used. It works by identifying those objects based on their characteristics. In the past, different kernel-based algorithms are implemented for processing point cloud data like the study by **Huhle et al., (2008)**, used Normal Distribution Transform (NDT) to perform point cloud registration along with color information. Similarly, for classification, **Zhang, Lin, & Ning (2013)** used the Gaussian radial basis function inside support vector machine classifier, which helped in transforming non-linear input features into higher dimensions where the linear methods could be implemented. And, the classification is continued with a component analysis for optimization. However, this method yielded some misclassification while separating the building and vegetation. With this, we can understand the class separability should be of high priority while training the classifier. With the availability of a large amount of labeled 3d Point cloud data, we can opt for a deep learning approach. **Azizmalayeri et al., (2018)** constructed a kernel correlation-based CNN on Pointnet architecture for classifying the 3d point cloud data. In their approach, they built a correspondence kernel layer to calculate the geometric distance between the points and their corresponding points in the kernel. With this technique, they were able to obtain an accuracy of 91% in the ModelNet40 dataset. This study explains to us the potential of a deep learning approach for classifying 3d point cloud data when the labeled data is available.

Similarly, **Thomas et al. (2019)** propose a novel method known as Kernel Point Convolution (KPConv), which acts as an alternative to conventional fixed grid convolutions. The deformable convolutions of the architecture enable the model to adapt kernel points concerning the local geometry. The model architecture, when trained with ModelNet40, produced an effective receptive field and better scene segmentation scores compared to other architecture without KPConv. These methods showcase the robustness of the deep learning frameworks for processing the point clouds. However, these methods require a set of point cloud structures per object for training, which itself turns to be quite an effort.

2.5. Support Vector Classifiers

Support vector machines (SVM) are mathematical tools that were based on a structured risk minimization principle of statistical learning theory**(Anthony, Gregg, & Tshilidzi, 2007)**. In machine learning, the SVM principle is used to solve the classification problem with a supervised approach. It handles the class separability function using a hyperplane in higher dimensions**(Huanrui, 2016)**. By doing so, each of the individual classes can be identified without over-fitting the model. The application of kernel functions and parameter selection while building contributes to the robustness of the SVM model **(Zhang et al., 2013)**. From the study made by **Mantero et al. (2004)**, it is evident that SVM can produce better results with a small quantity of training datasets. When dealing with data of multi-class where each class has its unique object characteristics, the one-versus-rest SVM classifier is preferred**(Anthony et al., 2007)**. Even though there is some empirical risk of classification errors, the one-versus-rest Multi-class SVM yield better results in multiclassification.

Furthermore, to minimize the misclassification, cross-validation methods are performed. Similarly, the one-versus-one Multi-class SVM produces higher accuracy if the object of interest is of the same class and follows similar geometry **(Mashao, 2005)**. As discussed earlier, one of the primary advantages of SVM is, they support kernel methods, i.e., for classes of different object characteristics, different kernels can be utilized, and parameters can be changed. This is important while handling two different data sources. In the study conducted by **Gevaert et., (2016)**, multiple kernels are used to classify a set of heterogeneous features from the UAV data. And it is evident that

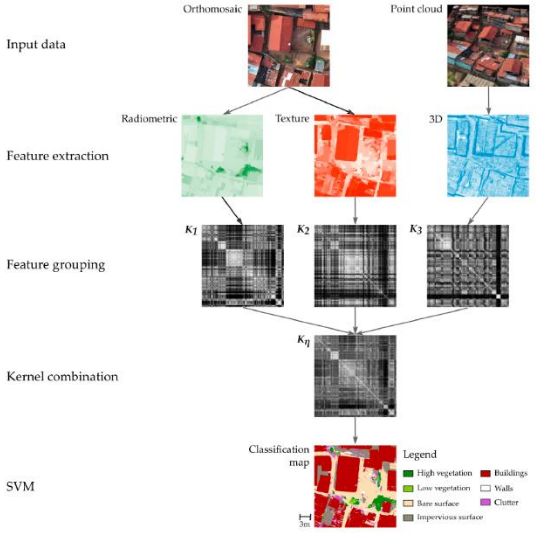


Figure 2-4 An illustrative example of the multiple kernel learning workflows (Source - Gevaert et., (2016))

by using feature grouping strategies along with multiple kernels, higher accuracy rates can be attained. Yet another feature of SVM is, a single classifier could be built using different kernel functions. For instance, in a study conducted by Huanrui (2016), two different kernels, such as Gaussian and Polynomial, are mixed to form a kernel function, which is then used for classification. And it is approved to increase the time efficiency over accuracy while training large datasets. Similar to the above, a mixed kernel function of gaussian and wavelet function increased the generalization stability of the model. While training the classifier using the point cloud data, the combination of polynomial and gaussian proves to yield high accuracy of classification while constructed as one versus the rest model **(C. Chen et al., 2019)**.

3. DATA AND SOFTWARE

For this research, three different datasets are used; they are MLS point cloud dataset, RGB images from the MLS platform and the SVHN dataset. Each of them is discussed in detail in the below sections.

3.1. Mobile laser-scanned dataset

The dataset used in this research is 3d point clouds obtained from the MLS platform mounted on the rear end of the rails. The dataset is captured in Northern Europe and covers the railway corridor along the path of the railroad. The figure represented below is a colored representation of the point clouds based on the elevation differences. The objects such as tracks, wires, cables, buildings, and ground segments are available for segmentation. The point clouds are of several laser strips with coordinates assigned to each of them with millimeters precision.

Table 3-1 Metadata of the 3d point cloud data

Data Type	Laser strip count	Data format	Data Size	Additional Info
Point Cloud Data	2864	laz	545 GB	Camera Calibration

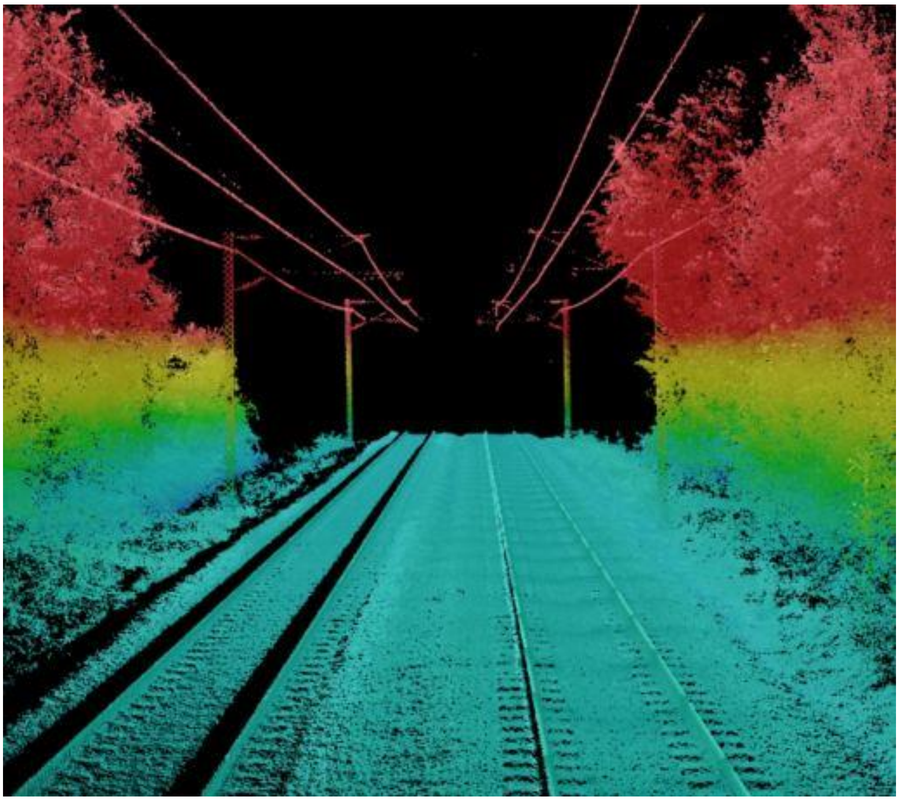


Figure 3-1 MLS point cloud dataset of a railway environment
(points are colored based on their elevation)

3.2. RGB Image dataset

The second dataset is an RGB images dataset obtained along with the point clouds from the cameras mounted on the MLS platform. Each image is constructed by stitching four different camera angles.

One out of the four image frames in the below figure is a panoramic view. Since the research aims to classify objects of interest, the image frame, which is forward moving, is selected for initial processing. Since these images frames are obtained from an original video recorded during the entire process of data acquisition, there exist few images where the edges are blurred, which is taken into consideration during the pre-processing stages.

As the dataset is obtained from the MLS platform, the GNSS/IMU information is available, which provides the location of each of the image exposure points. Also, the position of the camera is available for each unique view of the image, using which the objects in the images can be synced with the point cloud dataset. This linkage of two datasets is crucial because the research proposes a stepwise methodology that links both the dataset.

Table 3-2 Metadata of the RGB dataset

Data Type	Count	Data format	Data Size	Additional Info
RGB Image Data	90679	jpg	65 GB	Pixel Size - 4098x2048,

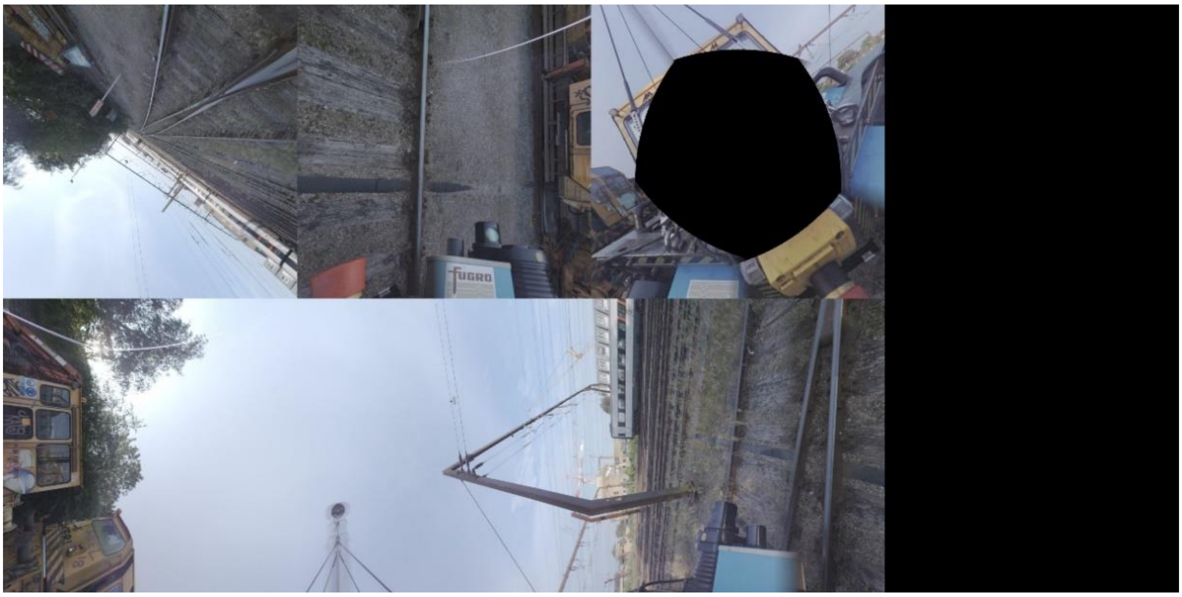


Figure 3-2 RGB images with four different camera angle

3.3. SVHN dataset

The Street View House Numbers(SVHN) Dataset is a collection of the real-world images obtained from Google street view images that are used in the development of machine learning algorithms for object recognition. The dataset is utilized in this research to train a model to extract semantic information from the objects of interest, such as the Kilometer marking numbers. Since the dataset comprises of images from natural scenes, they are useful in solving real-world problems of recognizing the digits **(Netzer et al., 2011)**. The dataset consists of 10 classes ranging from 0-9, whereas digits '1-9' has class labels 1 to 9, and digit '0' is labeled as 10. Moreover, images of different resolutions are included in the dataset, thus making it quite diverse. The images are of three color channels. Along with that, bounding box information for each character in the image is available separately for training purposes in the form of .mat file. For example, the digitStruct(10).bbox(1) gives the bounding box information of the first digit in the 10th image.

Table 3-3 Metadata of the SVHN dataset

Data Type	Count	Data format	Data Size	Additional Info
House Numbers	46,470	png	2 GB	Train - 33,402 Test - 13,068

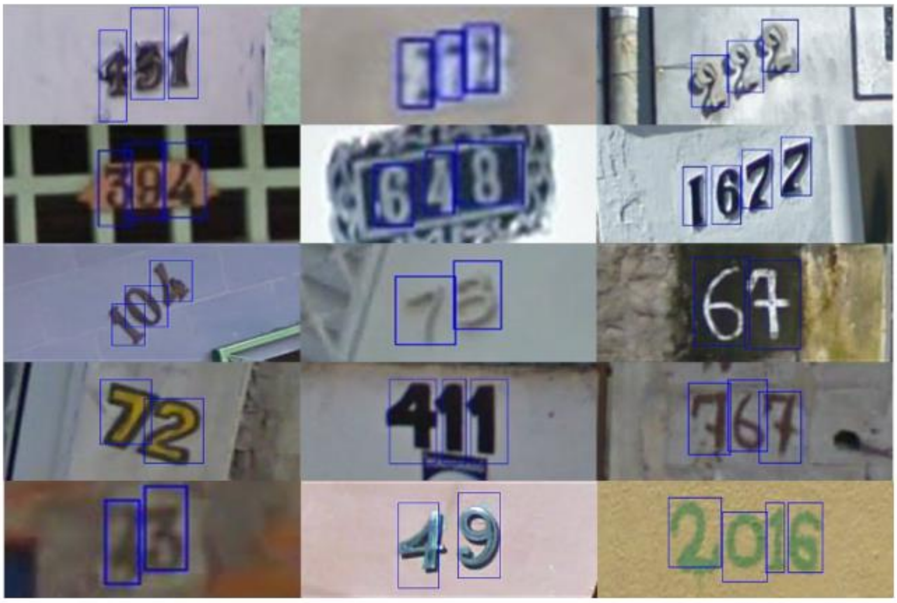


Figure 3-3 House Numbers with bounding boxes from Google Street View Images (Netzer et al., 2011)

3.4. Software

The image classification part is fully handled using Python (3.7) programming language. The annotations of the labels which are used for the training of the classifiers are performed using the Labellmg toolbox, which is an open-source toolbox written in python that produces image annotations in the PASCAL VOC format. Jupyter Notebooks are used as the preferred IDE(Integrated Development Environment). The point cloud processing is performed using Python and Erdas. The methods are prototyped in Windows 64-bit machine that runs Intel Core i5-8250U CPU at 1.60GHz with 8GB RAM. Additionally, for training the model, Google Colab is utilized, which is an open-source platform that offers 12GB RAM, Tesla K80 GPU, and TPUv2 with 180TFlops.

4. METHODOLOGY

This chapter includes the key process involved in this research. The chapter is subdivided into the following three sections:

4.1. Object Detection and Classification
4.2. Character Recognition from the detected objects.
4.3. Positioning the detected objects in the 3d point clouds.

Each section elaborates further on the methods used in the research to identify a solution for a particular problem. The key motive is to establish a complete pipeline that helps in by detecting the OOIs from the images, extracting the semantic information from the detected objects, and locating them in the point clouds to obtain the geometric information.

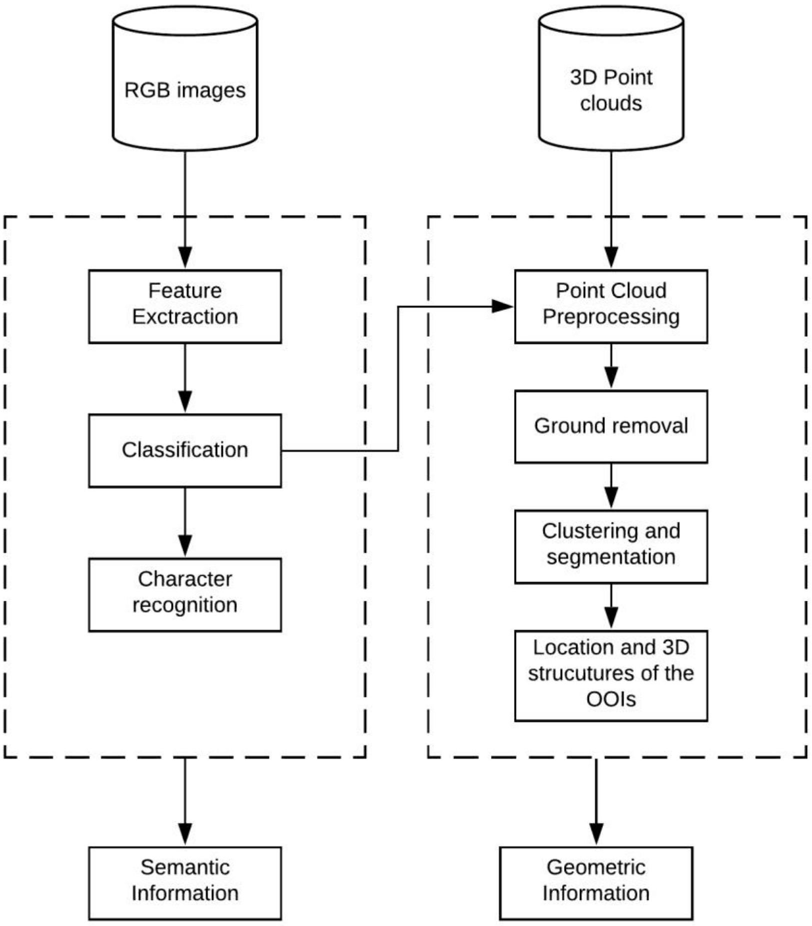


Figure 4-1 General workflow – Overview of the steps involved in the methodology

4.1. Object Detection and Classification

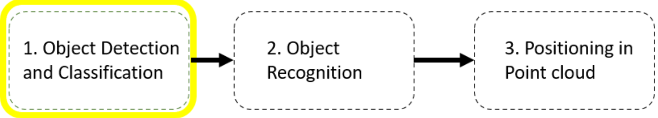

1. Object Detection and Classification
2. Object Recognition
3. Positioning in Point cloud

Figure 4-2 The three-step pipeline - This section deals with the highlighted box(in yellow)

This section covers the processes involved in classifying the objects of interest. Initially, the characteristics of the OOIs – Sign markers and Signals are studied thoroughly for understanding their occurrence in the dataset. So that the characteristics of the OOIs are well differentiated, and they can be trained based on them. The flowchart below (**Figure 4-3)** explains the proposed methodology for object detection and classification in this research.

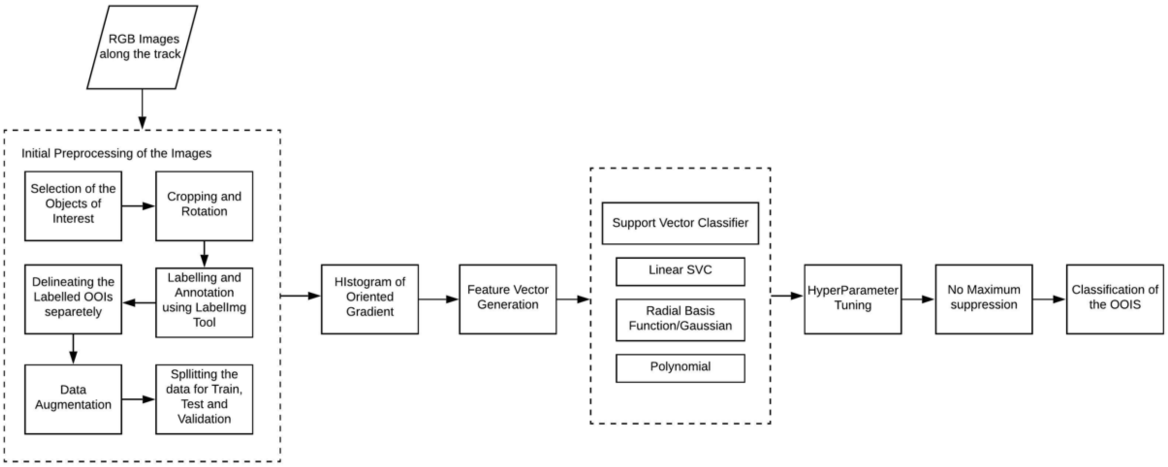

RGB Images along the track
Initial Preprocessing of the Images
Selection of the Objects of Interest
Cropping and Rotation
Labelling and Annotation using LabelImg Tool
Delineating the Labelled OOIs separetely
Data Augmentation
Splitting the data for Train, Test and Validation
HIstogram of Oriented Gradient
Feature Vector Generation
Support Vector Classifier
Linear SVC
Radial Basis Function/Gaussian
Polynomial
HyperParameter Tuning
No Maximum suppression
Classification of the OOIS

Figure 4-3 The proposed methodology for object detection and classification

4.1.1. Data Pre-Processing

After identifying the OOIs and studying them for their object characteristics and occurrence, the provided Image dataset available for this research is explored. As discussed earlier, the dataset consists of 90,000 images taken from the mobile mapping platform, with each image having four different camera angles. For this research, the forward-facing camera angles are preferred from which the OOIs are well distinguished. Those are the image data captured from an angle parallel to the track path. The selected camera angle is then cropped separately and rotated over 180 degrees for further processing **(Figure 4-4)**. Since we are to use the camera calibration information of the images to position the objects in the point clouds, no further transformations are performed to retain the projection information.

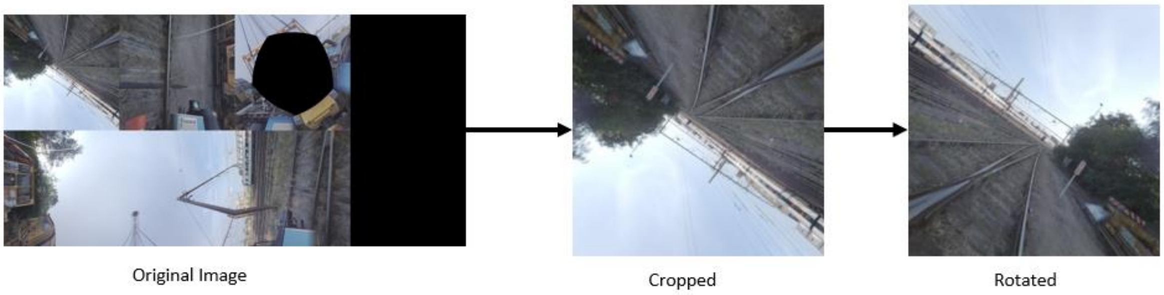


Figure 4-4 The forward-facing camera angle is cropped and rotated for further processing

4.1.2. Generating Annotations.

As we know, a training dataset is required to develop a model for object detection. Since the images are from the railway environment, other traffic signs dataset from road environments cannot be used efficiently. Similarly, the lack of standardized sign markings makes it impossible to use the dataset gathered from one region in another as they all differ. So it is necessary to produce a training dataset ourselves. Another important attribute to consider is the format of the training dataset. For this research, the PASCAL Visual Object Classes(VOC) format is used, which is a standard form for generating datasets with annotations for object detection. The annotation is created using the LabelImg toolbox[1], which is an open-source python toolbox that provides an interface to read the image folders, create/assign classes, and allocate class labels. It generates XML files for each of the annotated images with the bounding box information as below.

Bounding box format = [*xmin-top left, ymin-top left, xmax-bottom right, ymax-bottom right*]

The generated bounding box information is further processed to delineate the OOIs separately from the whole image. As a result, individual images for OOIs are obtained, which can be processed further for feature extraction. The table below(**Table 4-1**) displays the number of annotated images for each of the classes.

Table 4-1 Sample size of the objects of interest

Object	**Count**
Kilometer markers	240
Signals	130

4.1.3. Data Augmentation

Data augmentation is a process of increasing the diversity and quantity of the training data without actually collecting new data. Since the number of images of the OOIs is relatively low, augmentation methods are followed to increase the sample size. For the class sign markers(kilometer markers), the methods such as light deformation, color deformation, de-texture, decolorized, edge enhancements, flip, translate, Affine shear, contrast correction, and median blur are applied. Likewise, for the signal class, the methods mentioned above are applied except deformation and shear methods**(Figure 4-5)**. Also, some of the methods mentioned above are employed in varying degrees of intensity, providing output for each degree.

[1] https://github.com/tzutalin/labelImg

For instance, the affine shear is performed in various degrees, obtaining an output for each of them. The count of samples before and after data augmentation is listed in **Table 4-2**

Table 4-2 The sample size before and after augmentation

Object	Before augmentation	After augmentation
Kilometer markers	284	5697
Signals	64	1281

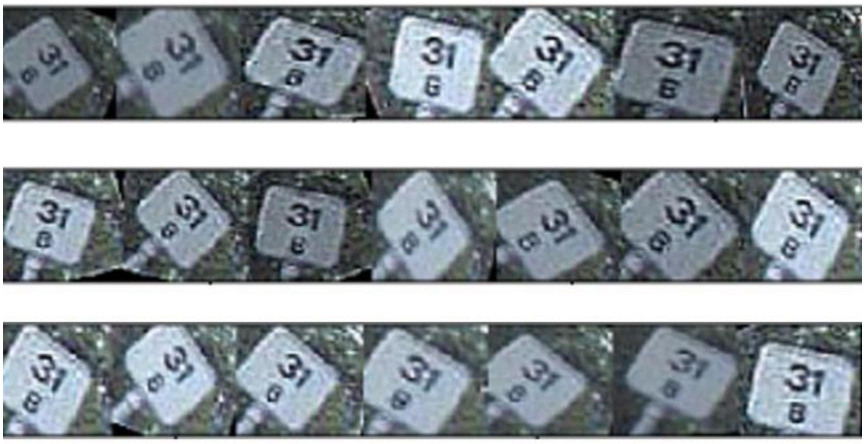


Figure 4-5 An example output of different augmentation methods performed for the Kilometer marker sign

4.1.4. Splitting of the data

As a final step of pre-processing, the augmented images are split into test, train, and validation dataset before moving forward. Since the research uses Support vector machine classifier, a combination of positive and negative samples are required for the training purpose**(Figure 4-6)**. So, a set of negative samples are produced for each class. The table below **(Table 4-3)** depicts the number of Positive and Negative sample distributions for each class.

Table 4-3 Count of positive and negative samples

Object	Positive samples	Negative samples
Kilometer markers	5697	3650
Signals	1281	1250

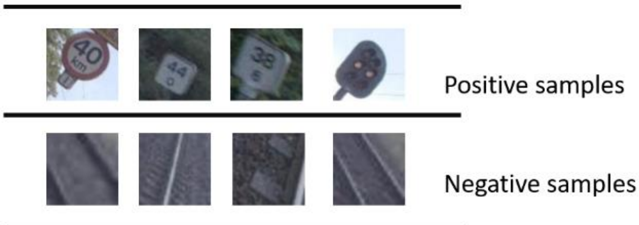


Figure 4-6 Example of Positive and Negative images

4.1.5. Histogram of Oriented Gradient (HOG)

With the positive and negative samples available for each class, the next step is to extract features from the images so that the classifier can learn to detect the OOIs. In this research, the Histogram of Oriented Gradient(HOG) feature is used as the feature descriptor. The theory of HOG is initially proposed by **Dalal & Triggs, (2005)** as a feature descriptor for human detection from the images. The idea behind this theory is to produce patches of the histogram of gradient orientations of the images using a sliding window with a defined grid and then compare the results with the know histogram of the objects with block normalization. That is, by using the magnitude and orientation of the intensity gradients, the objects can be detected in the images. Moreover, by using the HOG method, the intraclass variation in the OOIs can also be identified, and they perform better along with Support Vector Machines classifiers **(Balali & Golparvar-Fard, 2016).**

The kilometer marker samples are resized and converted into YCrCb color space before being fed into the descriptor. Likewise, the signal samples are converted into HSV color space before being fed into the descriptor. After this, feature vectors are produced for the samples using the HOG descriptor. The spatial size and histogram parameters are tested with different values to obtain reliable results. The output from three different channels of the HOG is visually inspected to see differences upon changing the parameters. **Table 4-4** displays the parameters that are chosen for this study after numerous trials. Furthermore, the performance of the HOG descriptor with regards to the class variations is also taken into consideration.

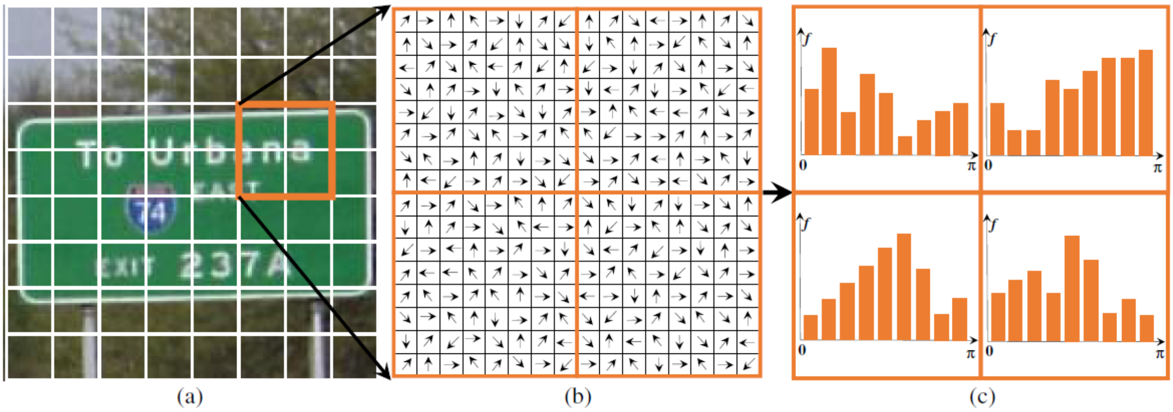


Figure 4-7 HOG performed over a sliding window (a) Pixel detection Window; (b) Pixels in each window (c) HOG for each cell (Balali & Golparvar-Fard, 2016)

Table 4-4 Parameters of HOG used in this study

Block size b	Cell size η	Orientation bins β	Pixel space p
4x4	8x8	12	32x32

As a next step, HOG features that are extracted from the images are put together to form a feature vector list which is used in training the support vector classifier. So, the classifier can match the histogram patch with the objects while scanning a detection window of the image.

4.1.6. Support Vector Classifier

As quoted in section 2.5, the SVM classifier works by finding the optimum hyperplane that separates each of the induvial classes. In this case, the optimum hyperplane is the one with the largest margin of separation between two classes.

For a given n labeled training dataset of points $\{x_i, y_i\}$;
x_i is a set of input p-dimensional HOG vector obtained from the candidate window i;
$y_i \in \{0, 1\}$ is a binary label depending upon where x_i falls **(Balali & Golparvar-Fard, 2016)**.

The optimal hyperplane derived by SVM is,
$w^T. x_i + b = 0$ …..(1)
where w^T is a normal weight vector and b is bias

Since SVM is a discriminative classifier, it works by identifying the OOIs present in the detection window. In this research, two different schemes of the support vector classifiers are employed for training a model to detect OOIs in the railway environment they are; a multi-class one-against-all SVM and two one-against-rest SVM (sign markers and signals separately). So, the performance of the individual models could be assessed. Similarly, three different kernel functions are tested for each of the classifiers. They are;
Linear kernels, $K(x, y) = x . y;$ where the K is the kernel function …...(2)
Polynomial kernels, $K(x, y) = (x . y + 1)^p$, where p is the tuneable parameter …...(3)
Radial Basis Function(RBF)/Gaussian kernels, $K(x, y) = exp\left\{-\frac{|x-y|^2}{2\sigma^2}\right\}$ **(Huanrui, 2016)** …...(4)

So, a total of seven different models are constructed for classification, which learns from the features extracted by the HOG descriptor. And the performance of each model is evaluated using a confusion matrix. After careful evaluation, a suitable model is identified for multi-class object detection, and they are implemented with a sliding window technique. It works by passing a window over an image where the objects have to be detected. The steps are as follows,

- A small window is taken as an input from the defined pixel starting point in the image
- The features inside the window are extracted using the HOG descriptor.
- The SVM classifier is then used to predict the extracted features from each window.

In this case, a fixed aspect ratio of detection window with three different scales [1, 1.5, 2] is used, and the threshold is applied to remove the false positives from the classification output. After completing the training process, the detection windows are placed in classifiers that return class labels of high classification scores.

4.1.7. Hyperparameter tuning and Non-maxima suppression

Hyperparameter tuning of SVM models widely includes three major categories; they are – kernels, Regularization, and Gamma parameters.
Kernels – In SVM classifiers, kernels are responsible for taking the low dimension data to higher dimensions. In this study, different kernel functions such as Linear, RBF, and Polynomial kernels are evaluated. And the classification results are compared to identify a suitable kernel method that performs well with the given data.
Regularization – The regularization parameter (c-parameter in python) acts as a penalty parameter, which indicates the SVM model about misclassification and its bearable amount. That is, if the regularization parameter is higher, the misclassification rate will be lower and vice versa. When the regularization

parameter is higher, the model tends to overfit. So there is a trade-off between, percentage of misclassification and decision boundary of the classifier. In this study, different c-parameters [0.1, 1, 10, 100] are tested for its performance.

Gamma – The gamma parameter determines the influence of a single training sample. When the gamma value is high, only the nearby points are considered to get the decision boundary and vice versa. Since the changes in the gamma parameter have a vast influence on the result, it is set to standard/default value.

By interchanging these parameters, the models are fine-tuned. As a final step, the process of non-maxima suppression is performed. It is a process of omitting the classification output in bounding boxes for the value below the set threshold. With this method, only the image information from local maxima is considered, and the detection with lower scores are ignored. And, as a result, the OOIs are detected in the given set of images.

4.2. Character Recognition from the detected objects

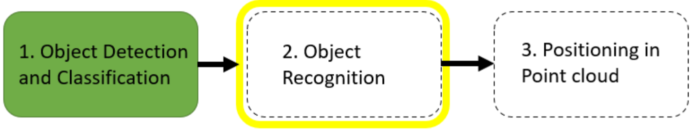

1. Object Detection and Classification
2. Object Recognition
3. Positioning in Point cloud

Figure 4-8 The three-step pipeline - This section deals with the highlighted box (in yellow and green being the completed section)

This section explains the methodology followed to recognize the characters from the images. As we know, one of the OOIs is a Kilometre marker, i.e., numerical characters(digits) exist in the detected objects. Such information is an important asset while producing a railway inventory. Traditional Optical Character Recognition(OCR) models such as Google's Tesseract does not produce reliable results as they are trained to detect characters from written text or documents **(Goodfellow, Bulatov, Ibarz, Arnoud, & Shet, 2014)**. In this case, the numbers should be detected from an image which is taken from a natural environment. So, it is necessary to build a model that is trained on recognizing the digits from natural images. Since this is a recognition problem, different aspects of the environment are also considered while choosing a model architecture, such as – lighting conditions, occlusions, resolution, and, more importantly, motion blur in the images(since the data is captured from a moving platform). In this study, a combination of RetinaNet (ResNet50) and You Only Look Once(YOLOv2) is opted for recognizing the digits from natural images.

4.2.1. ResNet50

As we know, Resnet50 is a deep residual learning network for image recognition trained on the Imagenet dataset **(He, Zhang, Ren, & Sun, 2016)**. Even though the residual network is 152 layers deep, they are comparatively easy to optimize. Also, the vanishing gradient problem is mitigated by the skip connection method in ResNet architecture. **Figure 4-9** explains the residual learning between 2 layers in ResNet architecture.

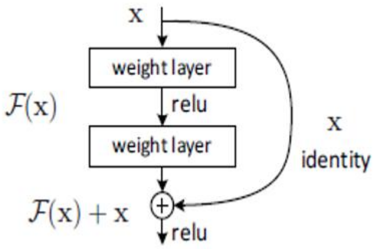


Figure 4-9 Residual Learning in ResNet (He et al., 2016)

4.2.2. YOLOv2

On the other hand, the You Only Look Once (YOLO) network is known for its extremely fast processing of images up to 45 frames per second(base model)**(Redmon, Divvala, Girshick, & Farhadi, 2016).** It uses single CNN for processing an entire image, which is divided into image grids, and scores are computed for each of those grids. YOLO consists of 24 convolutional layers and two fully connected layers, as shown in **Figure 4-10**. The YOLO network consists of two main components; a feature extractor, and a classifier. The feature extractor is inspired by the inception architecture of GoogLeNet trained on the PASCAL VOC dataset **(Redmon et al., 2016)**.

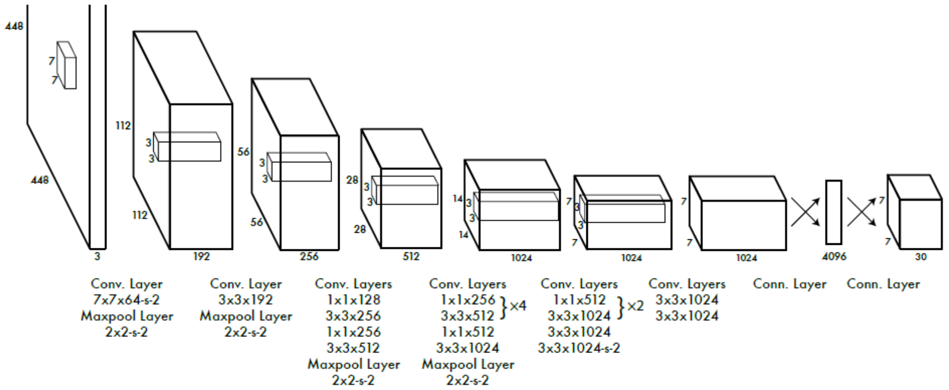


Figure 4-10 The architecture of YOLO with 24 convolutional layers and two fully connected layers (Redmon et al., 2016)

As we know, Resnet trained on the Imagenet dataset performs better that GoogLeNet according to the performance bench-marking **(He et al., 2016)**; the ResNet50 is introduced as the feature extractor in YOLO instead of GoogLeNet. And the combination of Resnet as the pre-trained model with YOLO produces better results with multi-object detection in complex natural environments **(Lu, Lu, Ge, & Zhan, 2019)**. So this study proposes to implement a similar combination for digit recognition from objects detected in the railway environment. **Figure 4-11** elaborates on the methodology used in this study.

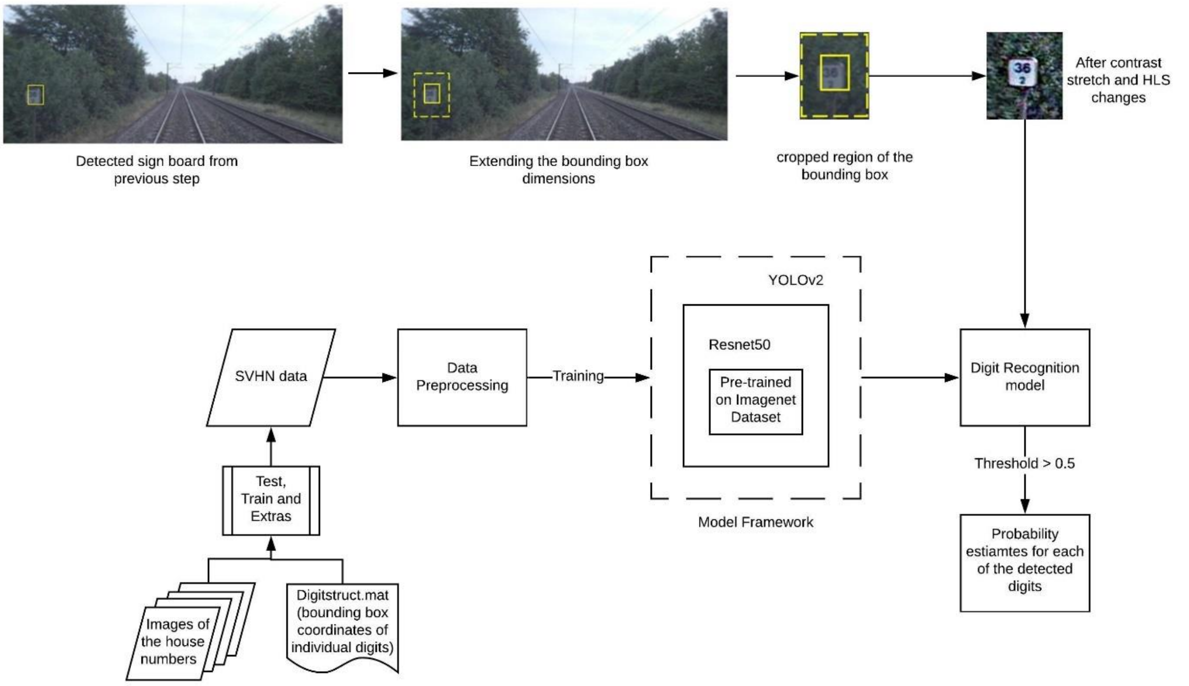

Detected sign board from previous step
Extending the bounding box dimensions
cropped region of the bounding box
36
2
After contrast stretch and HLS changes
YOLOv2
Resnet50
Pre-trained on Imagenet Dataset
Model Framework
SVHN data
Data Preprocessing
Training
Digit Recognition model
Threshold > 0.5
Probability estiamtes for each of the detected digits
Test, Train and Extras
Images of the house numbers
Digitstruct.mat (bounding box coordinates of individual digits)

Figure 4-11 The proposed methodology for recognizing digits from kilometer signs

4.2.3. Data Pre-processing and Training the model

To solve the image recognition problem in this study, the Street View House Number(SVHN) dataset is used for training the model. As discussed in **section 3.3**, it is an open-source dataset acquired from google street view images with multi-digit labels(up to 5 digits). The training sample size is reduced to tackle the computational complexity, and a set of randomized samples is taken for validation from the training dataset. **TTable 4-5** shows the count of images for train, test, and validation.

Table 4-5 The train, test and validation dataset

Category	Number of Images
Train	6600
Test	3300
Validation	1100

The next step is similar to the section - **Generating Annotations.4.1.2**, Individual images and annotations are converted into PASCAL VOC format for the training purposes. This process creates a set of XML files for each image with the bounding box coordinates. The images and annotations are stored in separate folders with common naming. Once the annotations are available for the individual images, the digits can be cropped from the whole image so that it can be further processed for image augmentation methods. During this process, instead of cropping the individual digits from the images using the bounding box, the extreme ends (Top left and bottom right), coordinates are chosen, and the images are resized to 416x416. This results in a multi-digit image. Later on, these images follow a series of image augmentation procedures.

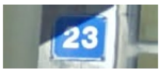
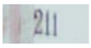
Original Image	Annotation Info *
	[1, 6, 10, 10, 10, 10]
	[2, 2, 3, 10, 10, 10]
	[2, 2, 1, 1, 10, 10]

Figure 4-12 Visuals explaining the training set [* The first digit of the array represent the number of digits in the image, the subsequent numbers represent the digit values in the image(10 being null value)]

After this step, the data is ready for training. Since we are using Resnet50 for feature extraction along with YOLOv2, all the layers of the model are trained initially[2]. Once the training is over, the last layer of the model (activation layer -49) is fine-tuned further. So that probability estimate can be obtained for each of the detected digits. The table below explains the parameters used in the training process.

[2] https://github.com/experiencor/keras-yolo2

Table 4-6 Parameters involved in the training process
(* to reduce the hardware complexity different learning rates are also tested)

	Initial training	Fine-tuning
Epoch	20	20
Learning rate	0.001*	0.001*
Activation layer	Input layer 1	Last Activation layer (49)
Batch size	64	64

The model obtained after this stage is capable of detecting the digits from natural images. Now that the model is ready, the next step is to pre-process the object detection results obtained from **section 4.1** so that the digits present in the results could be scored by the model.

4.2.4. Preparing the object detection output for digit recognition

The object detection output is in the form of bounding box information containing the OOIs. So the first step is to crop them from the whole image. To avoid cropping of pixels containing OOI information, the bounding box area is increased by an offset of 1.5 spacing. Thus making sure all the necessary pixel information for digit recognition is captured while cropping. As we know, the digit recognition model takes input images in the size 416x416 pixel; the cropped OOIs are resized initially. At this stage, the image looks nothing but a small cropped segment of the whole image without visible improvement.

For proper digit recognition, further image enhancement procedures are performed. It includes sharpening, contrast enhancements, and corrections in HLS (Hue, Luminance, and Saturation). By following these steps, the visibility of the digits in the image is drastically improved. The below **Figure 4-13**, show the improvement in details after processing.

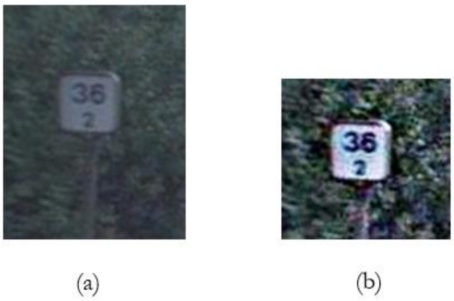


Figure 4-13 Side by side comparison of (a) Image before pre-processing; (b) Image after pre-processing

4.2.5. Digit Recognition

As a final step, the processed images of the OOIs are passed into the model for digit recognition. The output is obtained in the form of probability values of digits detected in the images. A threshold of values greater than 0.5 is set for the probability estimates so that high confidence output is obtained. Along with that, the digit count and the bounding boxes of the digits are also retrieved as an output.

4.3. Positioning the detected objects in the point cloud

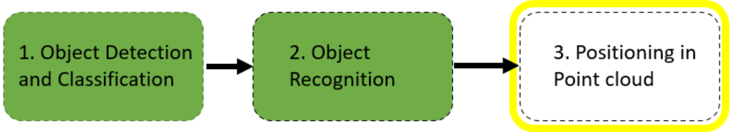

1. Object Detection and Classification
2. Object Recognition
3. Positioning in Point cloud

Figure 4-14 The three-step pipeline – (This section deals with the highlighted box in yellow and green being the completed section)

This section explains the process involved in positioning the detected objects of interest in the Point clouds. The object detection outputs from **section 4.1** contain the location of the objects in the image. The idea is, by knowing the location information of the image, GPS time, and camera orientation, the objects detected in the images are projected into 3d space in point clouds. And by using a set of procedures such as ground removal, intensity filtering, point cloud clustering, as proposed by **Li et al., (2019) and Soilán et al., (2016),** the exact point cloud structure of the objects is obtained. Thus the location of the OOIs is retrieved accurately. As shown in **Figure 4-15**, the following subsections explain the step by step process followed for positioning the detected objects in the point clouds.

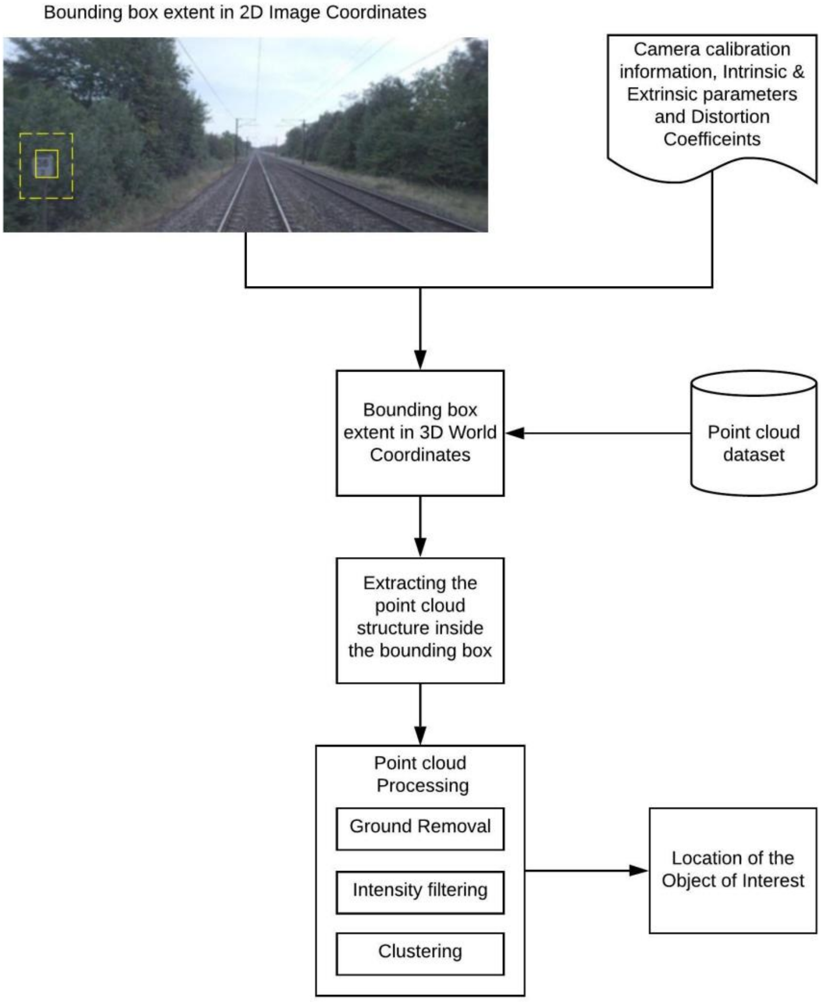

Bounding box extent in 2D Image Coordinates
Camera calibration information, Intrinsic & Extrinsic parameters and Distortion Coefficeints
Bounding box extent in 3D World Coordinates
Point cloud dataset
Extracting the point cloud structure inside the bounding box
Point cloud Processing
Ground Removal
Intensity filtering
Clustering
Location of the Object of Interest

Figure 4-15 The methodology adapted for locating the objects of interest in point clouds

4.3.1. Converting Image Coordinates to World Coordinates

The objects detected from **section 4.1** are taken as the input. To avoid cropping of pixels containing OOI information, the bounding box area is increased by an offset of 1.5 spacing. At this stage, the bounding boxes are in the image coordinate system, and they have to be converted into a world coordinate system so that they can be projected in the point clouds. In this case, the pinhole camera model is adapted to make the coordinates conversion. It is given by the formula[3],

$$s\ m' = A[R\,|\,t]\ M' \qquad \text{(or)} \qquad \dots (1)$$

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \underbrace{\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}}_{\text{Available info}}\underbrace{\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}}_{\text{To find}}$$

Where,

- u,v are the coordinates of the projection point
- A is the camera matrix,(intrinsic parameters)
- R|t is the joint rotation-translation matrix
- cx, cy is the principal point
- fx, fy are the focal length in pixels
- X, Y, Z are the 3d world coordinates

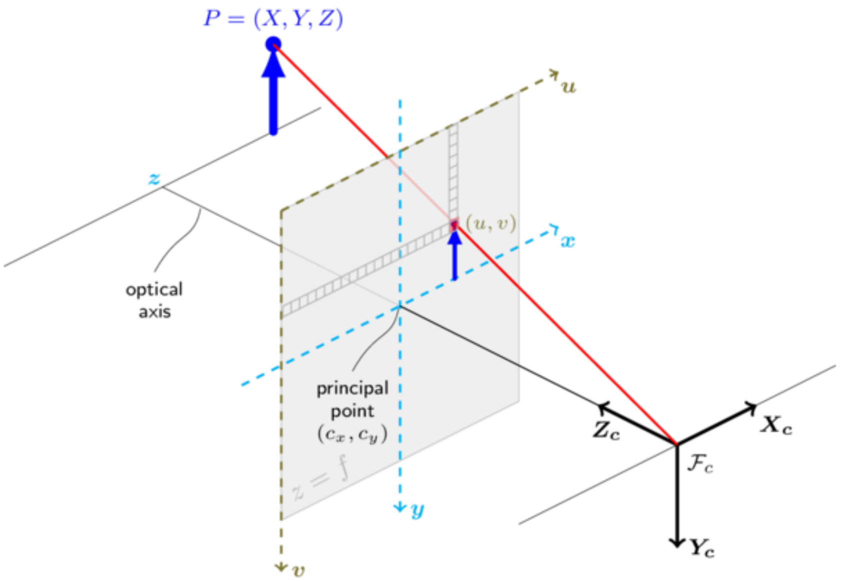


Figure 4-16 The pinhole camera model adapted for converting Image coordinates to world coordinates

Figure 4-16, illustrates the pinhole camera model where u and v are the image coordinates with origin in the top left corner and increasing in the right and down direction, respectively. And the camera coordinates are given by Z_c going towards the principal point, X_c and Y_c axis moving right and down respectively. First, the camera coordinates are calculated, which is given by (u-cx, v-cy, fc)**(Förstner &**

[3] https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Wrobel, 2016). Next, the camera coordinates are multiplied by the inverse of intrinsic camera matrix A and finally multiplied by the inverse of the Rotation and Translation matrix [R|t]. The resultant is world coordinates for the given image coordinates.

4.3.2. Extracting the point cloud structure inside the bounding box

After obtaining the world coordinates of the bounding box information, they are projected in the subsequent point cloud stretch. As a secondary validation, the GPS time recorded during the image capture and GPS time recorded in the extracted point cloud structures is compared up to two decimals for its relevance. This adds to the confidence of the extracted point cloud structure. At this stage, there exist different elements of the environment in the extracted subset of point clouds. So further processing is required to delineate the actual structures of the objects of interest and their location.

4.3.3. Point cloud processing

The extracted point cloud subset contains points that attribute to the objects of interest, terrain points, and other information. Before moving further, it is necessary to understand the characteristics of the selected OOIs for this study. They are;

- The sign markers, which are square-shaped board mounted on a cylindrical pole.
- Signals with and without signboards which are taller than the sign markers and their poles are not cylindrical

This information is useful in identifying and correlating the geometric aspects of the OOIs in the point cloud subset. As a next step, the ground removal and intensity filtering process are undertaken. It is used to separate the ground point from non-ground points. The main motivation for this step is to reduce the number of points significantly and to remove the high-intensity terrain point subsequently **(Soilán et al., 2016)**.

Furthermore, to remove the catenary wires and cables(if any) from the point cloud subset, filter based on height is employed. With few attempts of trial and error, a suitable height is fixed. All points over the height of three meters are removed at this stage. Since the OOIs are a cluster of spatial points at the given location, the clustering process is performed. Before proceeding further, the ranges in the dataset should be normalized for better results. So, the min, max, mean values of x, y, z is identified and normalized. This step is followed by the previously mentioned clustering process. For this study, the DBSCAN clustering algorithm (provided by the sklearn library) is utilized[4]. The parameters tested are available in the below

Table 4-7 Different parameters selected for DBSCAN

Parameters	**Explanation**	**Values 1**	**Values 2**
eps	neighbourhood radius	0.5	0.2
min_samples	minimum number of point in the neighborhood	5	10
algorithm	algorithm to compute the nearest neighborhood	kd_tree	ball_tree
leaf_size	size of the leaf in kd_tree or ball_tree	30	20

A suitable combination of parameter values that yields a good result is identified after a few trials. The resultant point cloud structure obtained after all the steps mentioned above is a proper representation of the objects of interest in the point cloud. For further usages, the outputs of these point cloud subsets are stored in the text file format. And the location of the object of interest is retrieved from the subset.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

5. RESULTS

In this chapter, the results obtained in each section of the methodology are studied in detail. Based on chapter 4, the subsections in this chapter is divided into:

5.1 Results on Object Detection and Classification
5.2 Results on Character Recognition from detected objects
5.3 Results on Positioning the detected objects in the point cloud

Note that the results from **Section 5.1** are taken as input for **Section 5.2 and 5.3**.

5.1. Results on Object Detection and Classification

In this sub-section, three main components are covered; they are the results from feature extraction, classifier performance, and object detection using the classifier. After preliminary pre-processing of the data, the pixel structures and color composites are studied in depth to identify a significant difference between the positive and negative samples. For this, the positive samples are plotted in different color spaces in **Figure 5-1**.

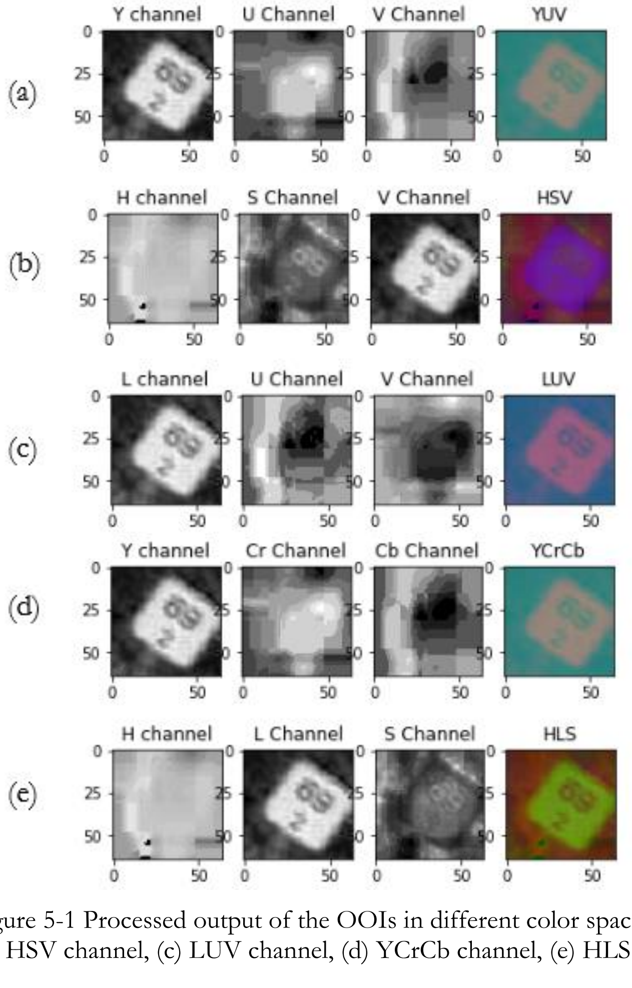


Figure 5-1 Processed output of the OOIs in different color spaces (a) YUV channel, (b) HSV channel, (c) LUV channel, (d) YCrCb channel, (e) HLS channel

After a few trial and errors with feature extraction, it is noted that, YCrCb color space yield better results for kilometer markers and HSV color space for signals. The below **Figure 5-2** represent the Histogram of Oriented Gradients(HOG) and color features extracted from the OOIs.

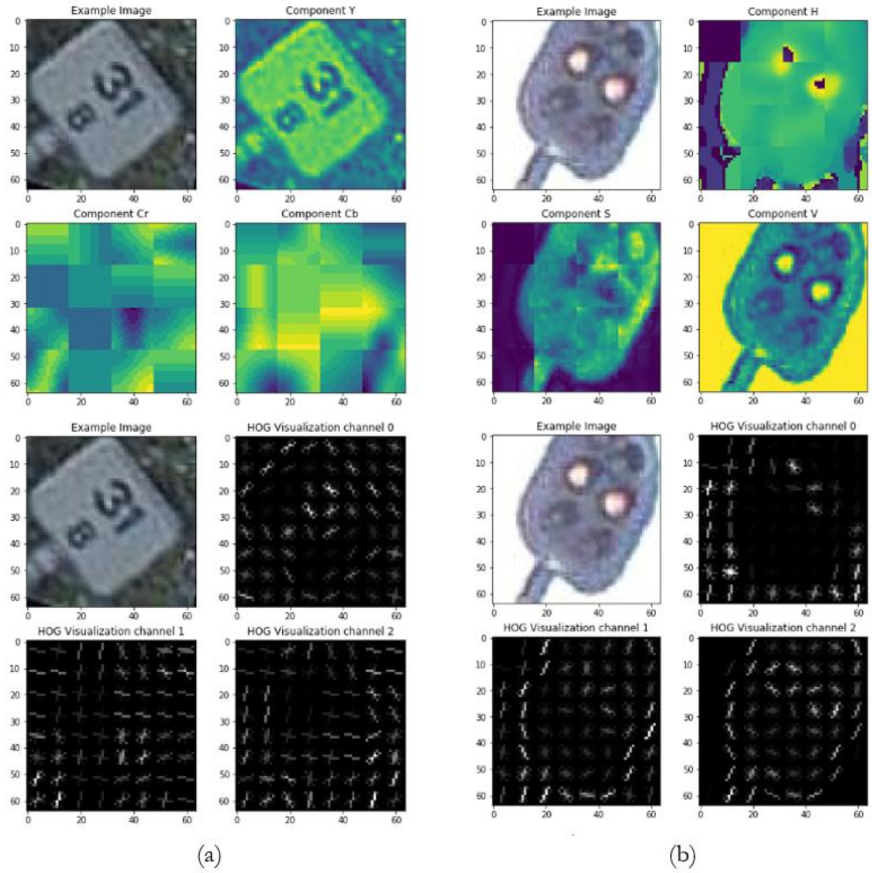


Figure 5-2 HOG and Color feature Extraction from OOIS (a) YCrCb color space for Kilometer markers and (b) HSV color space for signals

The features extracted from the positive and negative samples are used to train the SVM classifier for object detection. Three different types of kernel functions of SVM classifiers are implemented to this end, namely, Radial Basis Function(RBF)/Gaussian, Polynomial, and Linear kernels. In **Table 5-1 to** **Table** 5-6**,** the confusion matrices for actual class and predicted classes are computed for these kernels, quantifying the quality if these classifiers.

In **Table 5-1 to** **Table** 5-6**,** the User's accuracy, Producer's accuracy, and the Overall accuracy are used to assess the classifiers. For the kilometer markers, a total of 1881 samples are considered for accuracy assessment. The values in the diagonals in each table display the correctly classified samples. The overall accuracy in classifying kilometer markers is 97.71%, 95.48%, 97.71% for RBF, Polynomial, and Linear kernels, respectively, as per **Table 5-1 toTable** 5-3. The overall accuracy is obtained by dividing the total number of correctly classified samples by the total number of samples. It is evident that from the confusion matrix that all three kernels perform equally well in classifying the positive and negative samples of kilometer markers. However, the RBF and Linear kernels perform exceptionally well compared to Polynomial kernels. The good classification results are attributed to the extensive data augmentation methods followed and the color features extracted from the YCrCb color space.

Table 5-1 Confusion matrix of SVM classifier with Radial Basis Function/Gaussian kernels for kilometer markers

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		kilometer Signs	Not kilometer Signs	Total		
Predicted Class	kilometer Signs	734	23	757	3.10%	96.90%
	Not kilometer Signs	20	1104	1124	1.80%	98.20%
	Total	754	1127	1881		
Error of Commission		2.70%	2.10%			
Producer's Accuracy		97.30%	97.90%			
Overall Accuracy		97.71%				

Table 5-2 Confusion matrix of SVM classifier with Polynomial kernels for kilometer markers

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		kilometer Signs	Not kilometer Signs	Total		
Predicted Class	kilometer Signs	669	0	669	0.00%	100.00%
	Not kilometer Signs	85	1127	1212	7.02%	92.98%
	Total	754	1127	1881		
Error of Commission		11.28%	0.00%			
Producer's Accuracy		88.72%	100.00%			
Overall Accuracy		95.48%				

Table 5-3 Confusion matrix of Linear SVM classifier for kilometer markers

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		kilometer Signs	Not kilometer Signs	Total		
Predicted Class	kilometer Signs	733	22	755	2.92%	97.08%
	Not kilometer Signs	21	1105	1126	1.87%	98.13%
	Total	754	1127	1881		
Error of Commission		2.79%	1.96%			
Producer's Accuracy		97.21%	98.04%			
Overall Accuracy		97.71%				

Table 5-4 Confusion matrix of SVM classifier with Radial Basis Function/Gaussian kernels for Signals

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		Signal	Not Signal	Total		
Predicted Class	Signal	729	2	731	0.28%	99.72%
	Not Signal	4	262	266	1.51%	98.49%
	Total	733	264	997		
Error of Commission		0.55%	0.76%			
Producer's Accuracy		99.45%	99.24%			
Overall Accuracy		99.39%				

Table 5-5 Confusion matrix of SVM classifier with Polynomial kernels for Signals

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		Signal	Not Signal	Total		
Predicted Class	Signal	733	62	795	7.80%	92.20%
	Not Signal	0	202	202	0.00%	100.00%
	Total	733	264	997		
Error of Commission		0.00%	23.49%			
Producer's Accuracy		100.00%	76.51%			
Overall Accuracy		93.78%				

Table 5-6 Confusion matrix of Linear SVM classifier for Signals

Confusion Matrix		Actual Class			Error of Omission	User Accuracy
		Signal	Not Signal	Total		
Predicted Class	Signal	723	31	754	5.20%	95.80%
	Not Signal	10	233	243	5.12%	95.88%
	Total	733	264	997		
Error of Commission		1.37%	11.75%			
Producer's Accuracy		98.63%	88.25%			
Overall Accuracy		95.88%				

For signals, similarly to the kilometer markers, three SVM classifiers are built using RBF, Polynomial and Linear kernels. In **Table 5-4Table 5-6**, a set of 997 samples, including positive and negative samples, are employed to measure the performance of the classifiers. The accuracy of the individual classifiers is computed with the help of the shown confusion matrix. The overall accuracy of 99.39%, 93.78%, 95.88% is obtained for the RBF, Polynomial, Linear Kernel, respectively. The higher accuracy rate of the RBF kernels and slightly lower accuracy of the linear kernel explains the complexity of the Signals class. The higher value in the error of commission in linear kernel explains that information obtained during the process of feature extraction was not sufficient to establish a difference between the positive and negative samples. However, in future works, this can be mitigated by introducing new negative samples for clearer distinction. On the other hand, the RBF kernel produced higher accuracy with a lower error rate of omission and commission.

While looking at the kernel performances for both the classes, a conclusive decision is made to construct a multi-class SVM with RBF kernel function **(Section 2.5)**. The performance of the new classifier is evaluated with a sample size of 947 images, and the confusion matrix (**Table 5-7**) is produced. The overall accuracy of 96% is obtained with a multi-class function, which performs considerably well without overfitting the model. However, similar performance is obtained with different ratio of positive and negative samples as well

Table 5-7 Multi-class SVM with RBF kernel for class kilometer marker and signals

Confusion Matrix		Actual Class				User Accuracy
		kilometer Sign	Signals	Negative samples	Total	
Predicted Class	kilometer Sign	458	5	0	463	98.9%
	Signals	3	170	17	190	89.4%
	Negative Samples	1	11	282	294	95.9%
	Total	462	186	299	947	
Producer's Accuracy		99.1%	91.3%	94.37%		
Overall Accuracy		96%				

Figure 5-3 shows the overall accuracy of different kernels for different object classes and the performance of the multi-class SVM. Since accuracy is obtained with the multi-class SVM, they are selected for object detection, which shall be performed with a sliding window method.

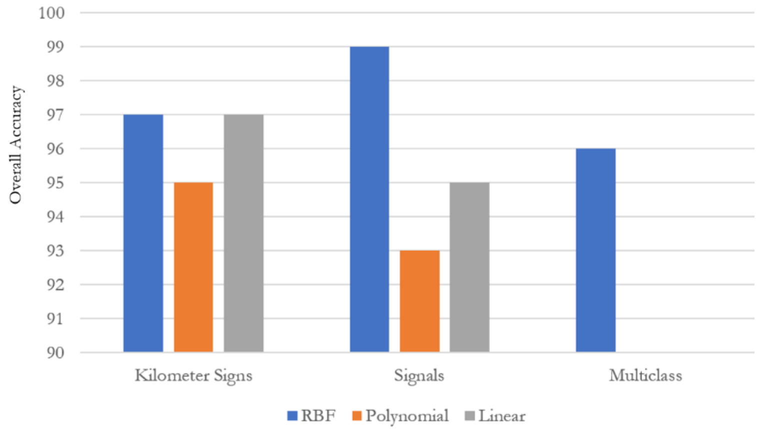


Figure 5-3 The performance of Different kernel function for individual class and Multi-class Classifiers

Accuracy assessment of the Multi-class SVM:
Using the Multi-class SVM, the objects are detected in the images using a sliding window approach. The windows are passed in different scales of the image [1,1.5,2,2.5,3] to increase the maximum detection accuracy. **Figure 5-4**, shows the detection output with a low confidence threshold of 0.5 and its subsequent heat map. By increasing the threshold values to 0.8, and restricting the bounding boxes only to the left portion of the image, a more appropriate detection output is retrieved, as shown in **Figure 5-5**.

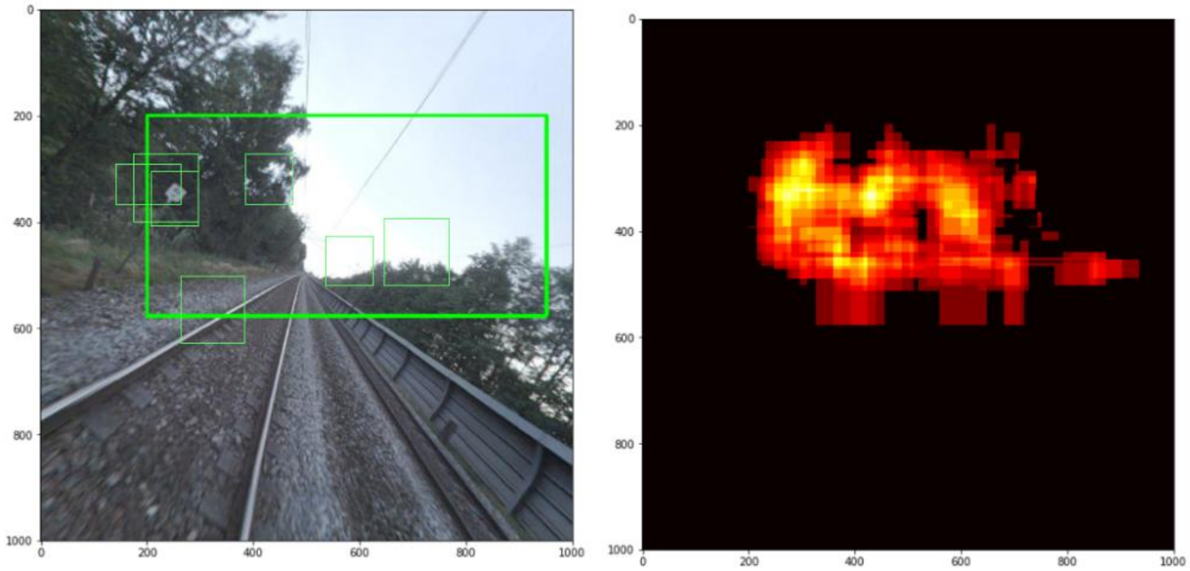


Figure 5-4 The object detection output and Heat map generated (threshold = 0.5)

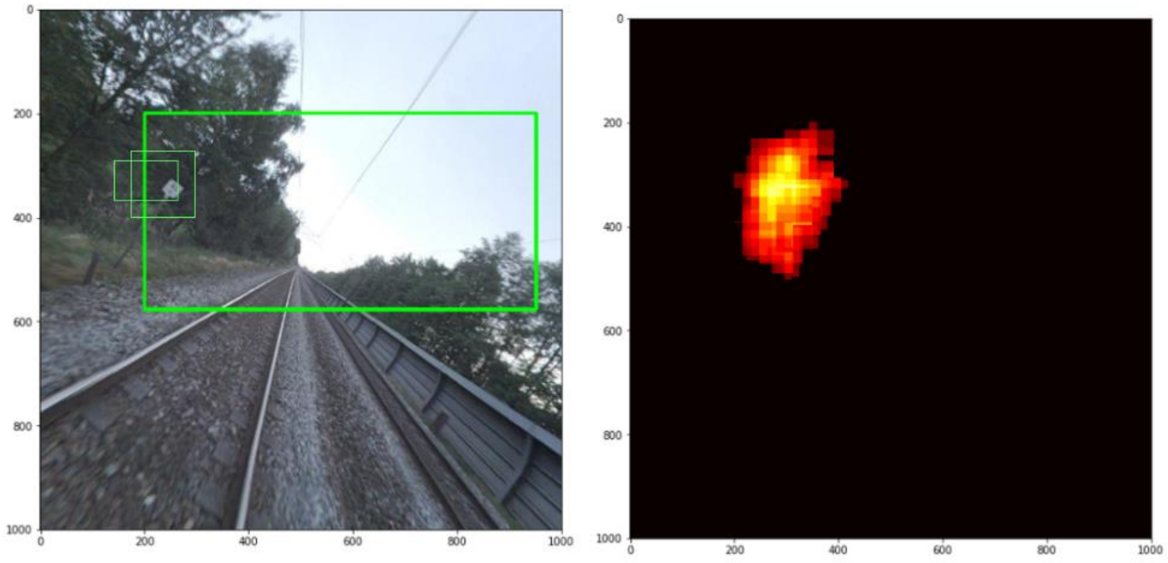


Figure 5-5 The object detection output and Heat map generated (threshold = 0.8 and choosing the bounding boxes in the left portion and above track of the image)

Furthermore, the overall detection accuracy of the system is evaluated using a different test dataset. It consists of 30 images, out of which ten images are with Kilometer markers, ten images with signals, and the remaining ten images without any objects of interest in it. The expectation is that,

- The model should be able to identify the images consisting of the OOIs(classification problem) and
- The location of the OOIs in the images(Detection/localization problem)

Table 5-8 The confusion matrix for the classification of images consisting of the OOIs

Confusion Matrix		Actual Class				User Accuracy
		Images with kilometer signs	Images with Signals	Images without both	Total	
Predicted Class	Images with kilometer Sign	**9**	**1**	**1**	**11**	**81.8%**
	Image with Signals	**0**	**8**	**2**	**10**	**80%**
	Images without both	**1**	**1**	**7**	**9**	**77.7%**
	Total	**10**	**10**	**10**	**30**	
Producer's Accuracy		**90%**	**80%**	**70%**		
Overall Accuracy		**80%**				

In **Table 5-8**, an overall accuracy of 80% is obtained when classifying the images containing the OOIs. Since there are no images that contain both the OOIs in the test dataset, it is set as a constrain during the classification process. The Misclassifications are due to the occurrence of different sign markers in the images or the wire poles classified under signals.

Table 5-9 The accuracy of the OOIs detected in the test dataset

Test Dataset	OOIs count
Visual Interpretation	20
Correctly detected OOIs	17
Total Detected OOIs	26
Correctness	65.4%
Completeness	85%

From **Table 5-9**, it is evident that the completeness of the model is acceptable for detecting the OOIs, which is 85%. However, the correctness of the system is further improvised by limiting the sliding window operation to a specific portion of the image. As we know, the kilometer markers occur in the left portion of the image, and the signals tend to exist in the top region of the images. This information is utilized while constructing the sliding window so that only those regions are concentrated instead of scanning the entire image for matching features. This not just improvise the correctness of the detection system but also require less processing power and time.

5.2. Results on Character Recognition from the detected objects

As discussed earlier in section **4.2**, one of the OOIs is the Kilometer markers from which numeric values can be extracted. For this purpose, a digit recognition model is constructed using the Resnet50 architecture as a feature extractor and YOLOv2 as the classifier, which is then trained on the SVHN dataset (**Table 4-5**) with the parameters mentioned in **Table 4-6**. The resultant model is evaluated for its accuracy and its f-score, precision and recall are computed.

Precision = true positives/(true positive + false positive) …...(1)

Recall = true positives/(true positives + false negatives) …....(2)

F-score = 2 * $[(precision * recall)/(precision + recall)]$ ……(3)

Table 5-10 The evaluation of the digit recognition model

Precision	Recall	F-score
0.754	0.783	0.768

According to **Table 5-10** The evaluation of the digit recognition model reaches an f-score of 0.768 based on its precision and recall factors. The obtained model is capable of recognizing up to five digits in a sequence. So, the performance of the model is evaluated in two aspects – Single and Multi-digit recognition. And it is quite evident from **Table 5-11**, that the model performs better classifying single digits compared to multi digits. However, the accuracy of the model drops as we move from single digits to two digits and so on.

Table 5-11 Single vs. Multi-digit accuracy

Single Digit accuracy	Multi-digit accuracy
93.2%	75.9%

- The model that is trained on house number images is used to recognize digits in kilometer marker images(object detection output). Since both are digits occurring in the natural environment, the model is expected to perform well with this new dataset. However, measures are taken to prepare the data before it is fed into the model.
- Since the model is trained using the data obtained from house numbers, there are fewer samples where the digits exist one below the other. Whereas, in this study, the kilometer markers have values in two rows - kilometer count and its subsequent meter values. This factor is also taken into consideration while preparing the data.

To properly evaluate the user accuracy, a set of 10 image samples containing 25 digits in total is derived. Out of which, only two samples are pre-processed with edge sharpening and contrast correction so that the difference in recognition between processed and pre-processed samples can be seen. Furthermore, the individual digits are manually Annotated/interpreted in the images. The expectation is that,

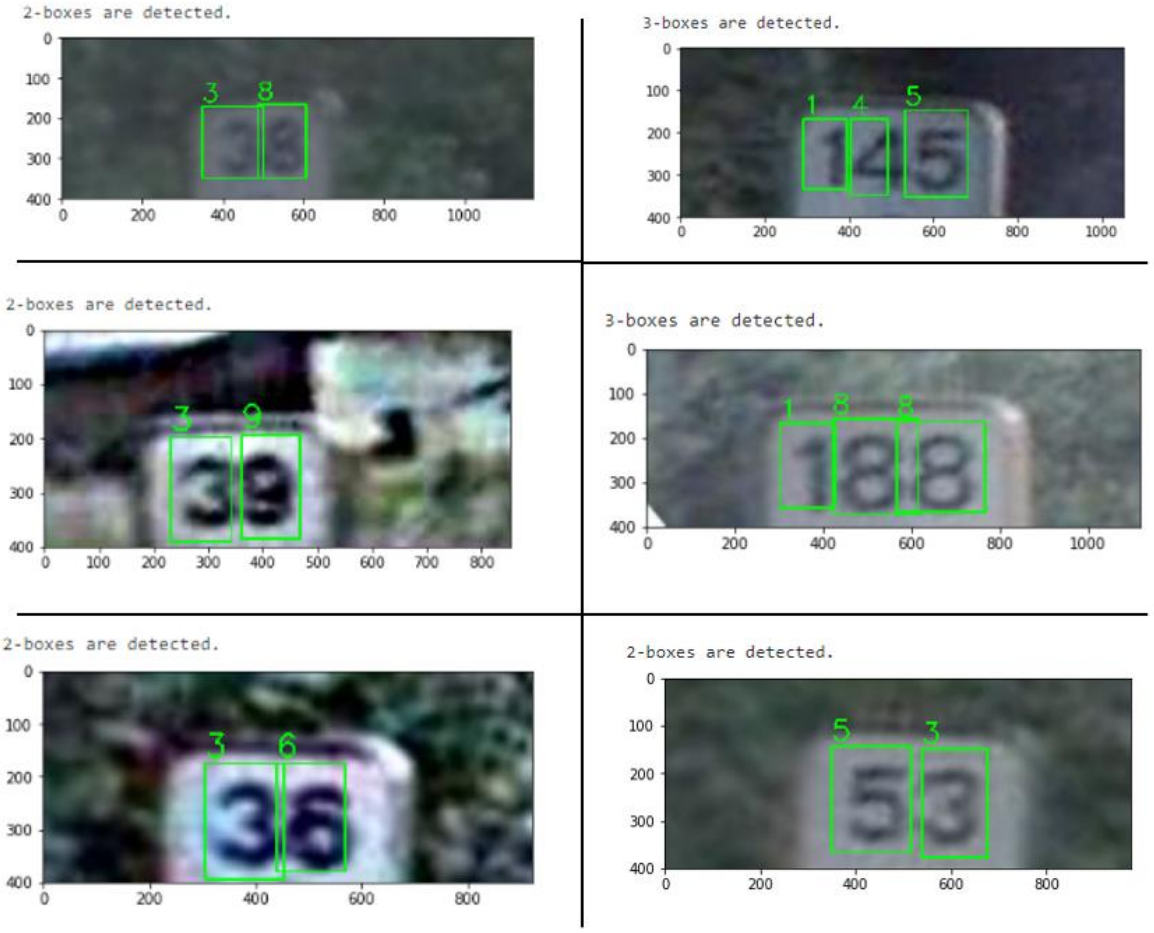
- The model should be able to delineate the digits with a bounding box and,
- The model should be able to label the digits correctly

The image samples are fed into the model, and the probability values for each digit is obtained. A threshold value of 0.5 is set to the detection results so that a low probability scores are removed.

Table 5-12 The accuracy of the Digits detected in the test dataset

Test Dataset	**Digits count**
Visual Interpretation	25
Correctly detected Digits	20
Total Detected Digits	23
Correctness	86.9%
Completeness	80%

The model achieved a correctness score of 86.9%, as shown in **Table 5-12**. The correctly predicted digit values are presented in **Figure 5-6** with digits enclosed in the bounding boxes. From the results, it is evident that the model can identify the digits in the image and label them with appropriate digit values. There exists some overlap between consecutive bounding boxes in images, and it could be avoided by adjusting the scale factor. Also, the digits in the two pre-processed samples are correctly recognized by the model (number 39 and 36).



```
2-boxes are detected.
```

```
3-boxes are detected.
```

```
2-boxes are detected.
```

```
3-boxes are detected.
```

```
2-boxes are detected.
```

```
2-boxes are detected.
```

Figure 5-6 Correctly recognized digits from the Kilometer marker using the produced model
(The probability scores for individual digits are mentioned in the appendix section)

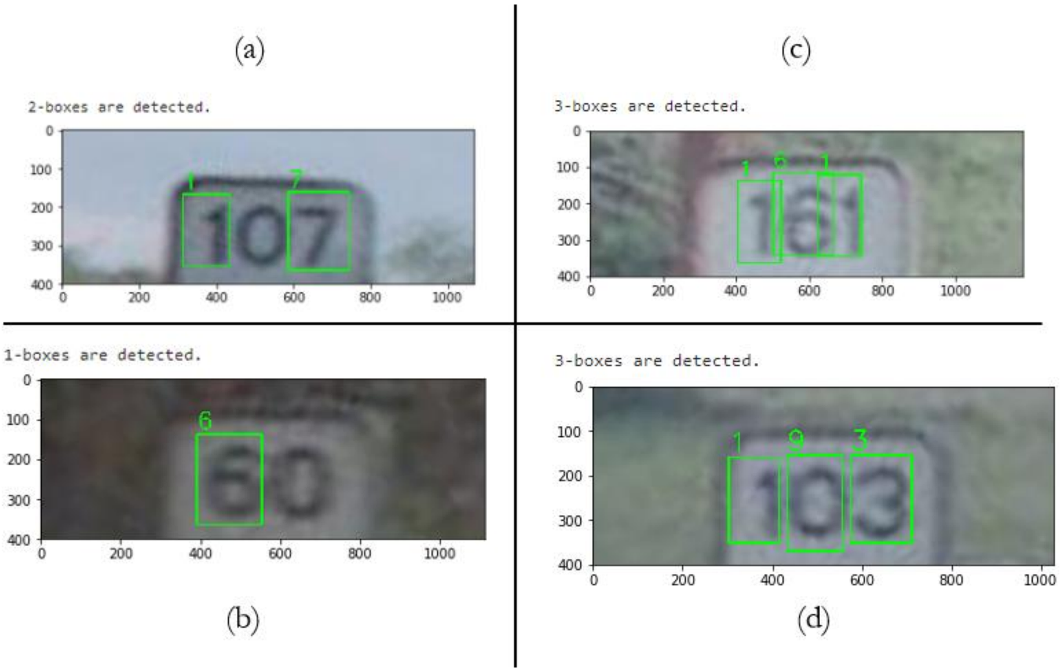


Figure 5-7 Wrongly recognized digits from the Kilometer marker using the produced model

Figure 5-7 explains four different types of errors that occurred during the recognition process. Most importantly, each of these errors is different from one another. They are discussed in detail below.

a. The input image contains three digits, out of which only two digits are detected. The model was not able to identify the sequential digits. The digit in the center is not recognized even though it is distinguishable. This situation could be mitigated by reducing the threshold value to 0.3, for instance.
b. The input image has two digits in it, but only the first digit is recognized and labeled. Moreover, the second digit is slightly distorted. This error can be minimized by applying the edge sharpening filter so that the digits are more visible.
c. The output seems acceptable with proper digit labels, but the error occurs in the positioning of the digits. Upon checking the recognition results, the digits are misplaced as '611' instead of '161'. The only possible explanation is that for some reason, the second digit is detected ahead of the sequence. However, pre-processing methods can be applied to make the digits even more distinguishable.
d. The center digit '0' is misclassified as '9'. The most suitable explanation for this error is, the model is not able to clearly distinguish the digit zero (similar to a and b). By allowing lower threshold values for digit '0', the error could be minimized.

5.3. Results on Positioning the detected objects in the point cloud

The bounding box coordinates obtained from the images for the detected objects of interest are converted into 3d world coordinates using the camera calibration information, orientation parameters, and the trajectory of the vehicle(position and orientation). The timestamp information from the detected image is used to identify the relevant point cloud stretch as both Images and Point clouds are captured at the same time from the MLS platform. With this information, the obtained 3d world coordinates are projected in the point clouds, and the bounding box region is cropped.

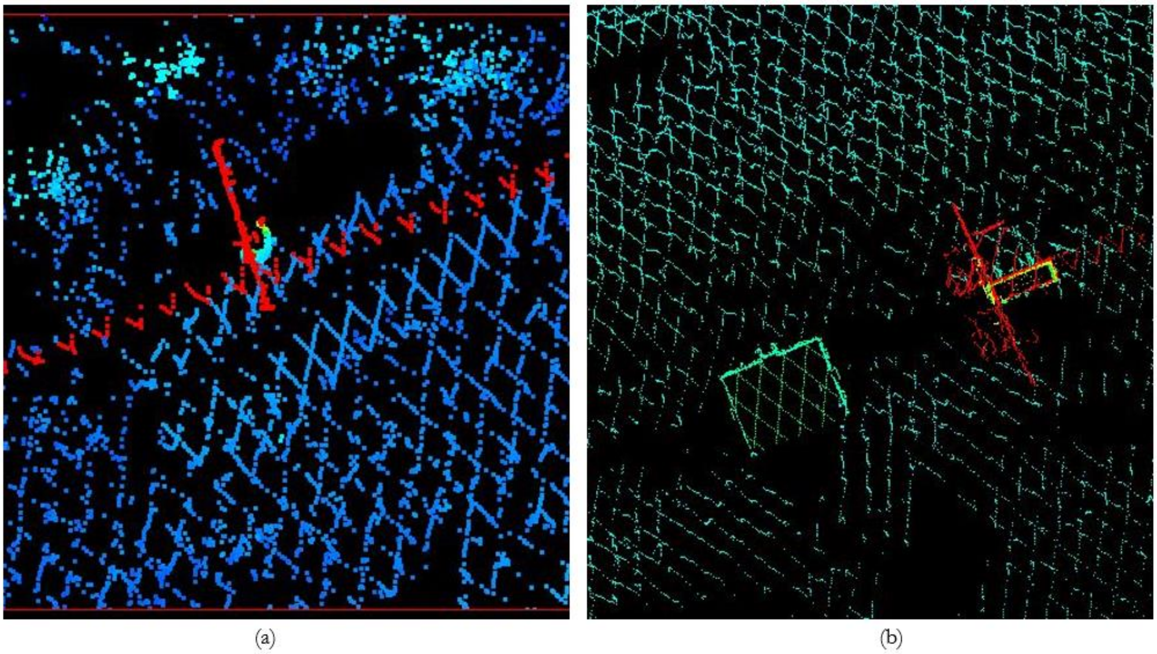


Figure 5-8 A top view of the cropped bounding box region from the MLS point cloud(points colored based on elevation) (a) Region containing a Kilometer marker. (b) Region containing the Signal

Figure 5-8 shows the cropped bounding region containing the objects – Kilometer marker and Signals. They are colored based on the elevation, and the red colored points represent the objects as they stand above the ground surface. However, in figure(a), the railway catenary lines are also visible(colored in linear red points). These out of scope points can be eliminated by introducing a height offset. To remove the out of scope points from the bounding box region containing Kilometer markers, only the points that lie within 1.7m above the ground is selected. Similarly, for Signals, only the points that lie within 3m above the ground is selected. By doing so, the points belonging to catenary wires and power transmission wires are eliminated.

Now the process of ground removal and intensity filtering is performed. Before executing these steps, the density of the point clouds representing the objects is worth noticing. The object structure can be divided into two segments for this discussion.

- "The top of the object structure" - that includes the Kilometer boards and the Signalling instrument.
- "The pole" holding the object.

As seen in **Figure 5-9 (a&b),** for signals, the point cloud density is high in both segments. The top of the object is well represented, and the pole holding the object is also distinguishable. But for the Kilometer markers, as seen in **Figure 5-10 (a&b),** the points representing the top of the object are well distributed, but the poles are not fully visible, which is due to its cylindrical nature.

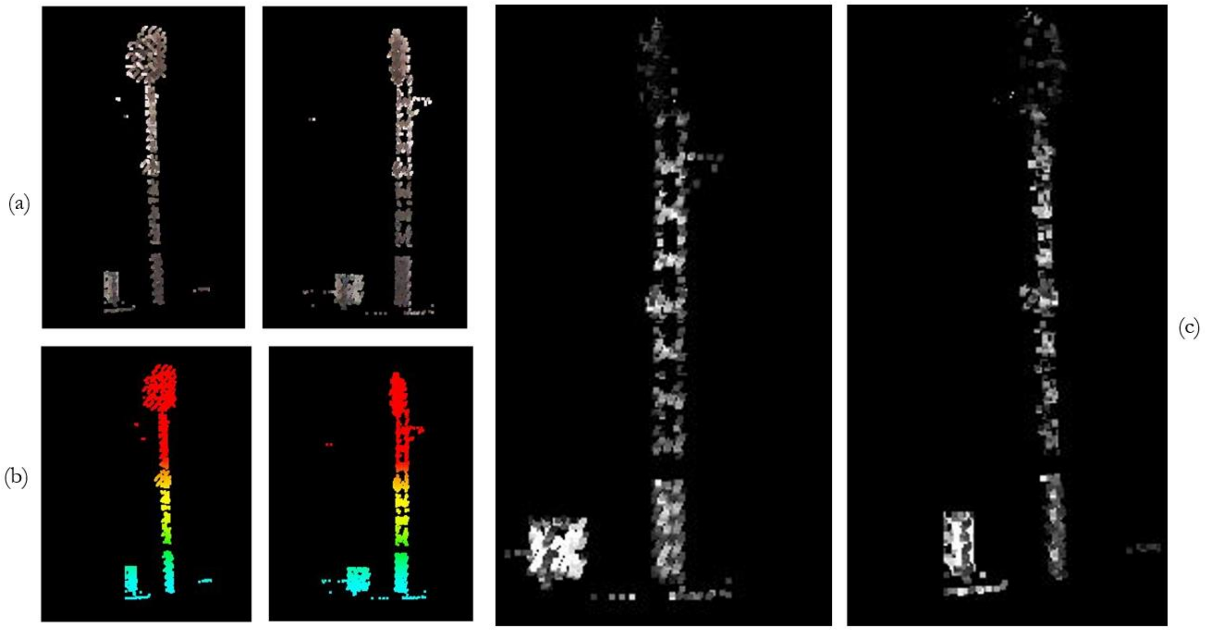


Figure 5-9 The Signal object - Front and Side view (a) Signal color coded by RGB values. (b) Signal color coded by elevation and (c) Signal color coded by Intensity

Similar to the density of the point clouds as discussed above, the difference in point cloud intensity between Kilometer markers and signals are also very much visible. As seen in **Figure 5-9 (c)**, where the points are coded with intensity values, the top of the signal object is of points with low-intensity values. However, the Kilometer markers showcase high-intensity values, as seen in **Figure 5-10 (c)** This difference in intensity can be attributed to the reflective properties of different objects of interest. In this case, the Kilometer marker surface is highly reflective, whereas the surface of the signals is not reflective enough. On the other hand, the poles holding the Signals are very much reflective compared to the poles holding the Kilometer markers.

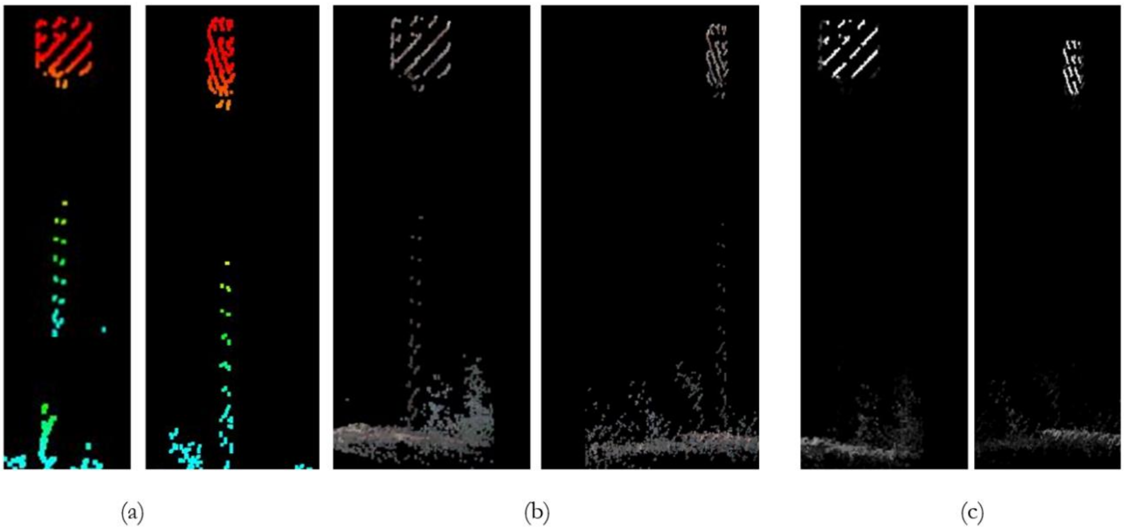


Figure 5-10 The Kilometer marker - Front and Side view (a) Kilometer marker color-coded by elevation. (b) Kilometer marker color-coded by RGB values and (c) Kilometer marker color-coded by Intensity

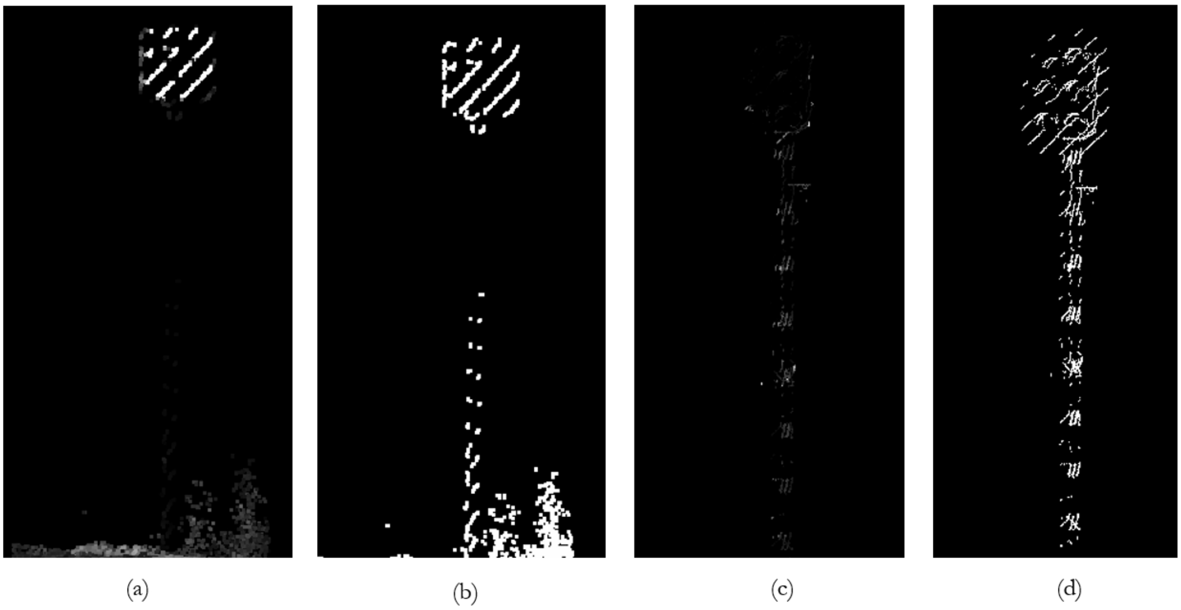


Figure 5-11 Points representing Kilometer markers and Signals before and after the radiometric corrections; (a)Kilometer markers before the radiometric adjustment ; (b) Kilometer markers after the radiometric adjustment (linear stretches); (c) Signal before the radiometric adjustment ; (d) Signal after the radiometric adjustment (linear stretches)

As seen in **Figure 5-11**, points represented by intensity values are radiometrically corrected to obtain a more even distribution. In this case, the linear stretch method is employed. After this process, the points representing the poles are distinguishable. Now, the RANSAC algorithm is applied to two reasons: to fit the object to the ground surface and obtain a planar surface on the top section of the objects(kilometer marker and signals). **Figure 5-12**, the ground plane is clearly distinguished in both objects, and the planar surface at the top of the object(which is of interest) is well developed.

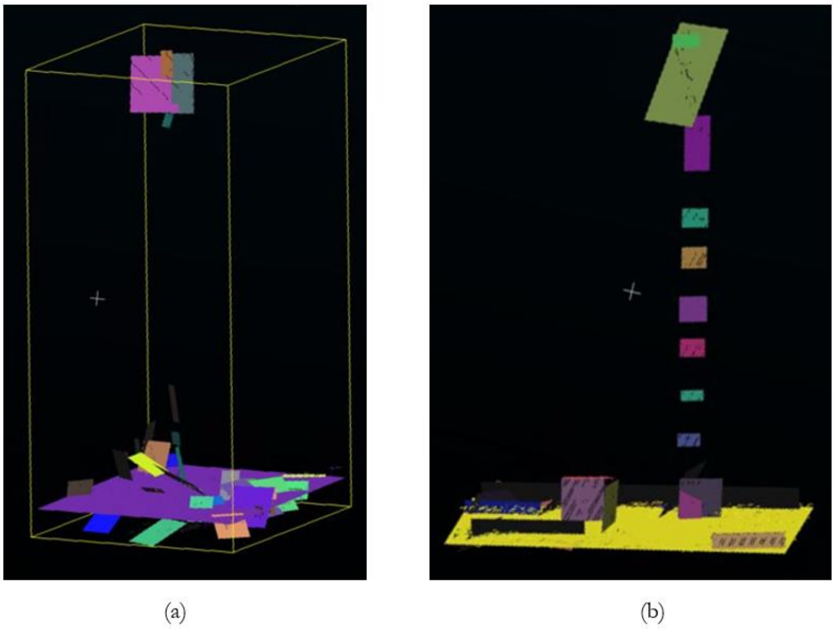
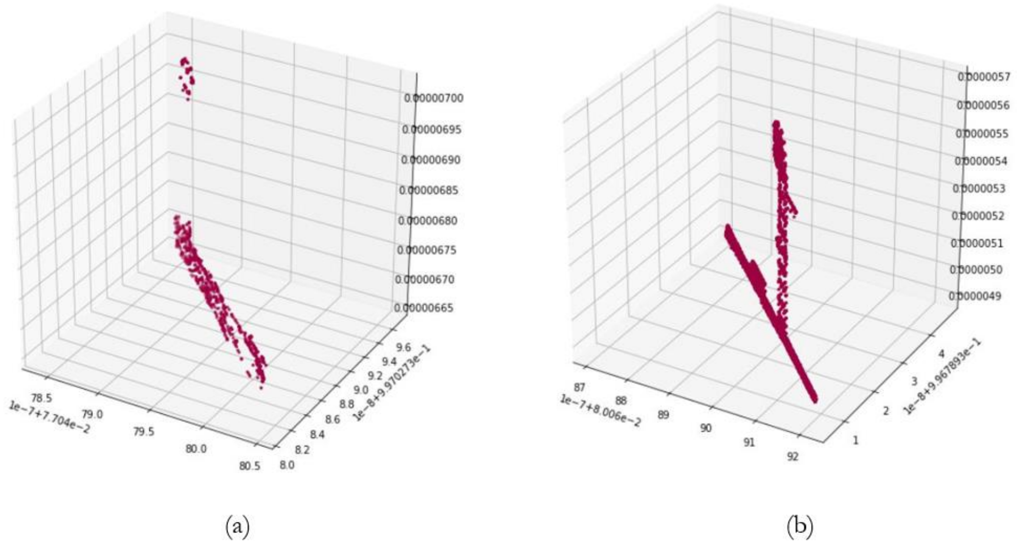


Figure 5-12 The output of Kilometer markers and signals after implementing RANSAC shape detection algorithm



(a) (b)

Figure 5-13 Results from Point cloud clustering (a) Kilometer marker (b) Signals

The points are clustered to obtain the overall cluster of points representing the object of interest. In **Figure 5-13**, the point cloud is clustered using the Density-Based Spatial Clustering of Applications with Noise[5] (DBSCAN) with the Kd-trees method to compute the pointwise distances for finding the nearest neighborhood. The point clusters are delineated separately to represent the 3d structure of the objects. However, the location of the object is derived from the centroid of the cluster in 3d world coordinates. As a result, the 3d structure and the location of the OOIs are retrieved.

Table 5-13 The geometric information of the OOIs

Height of the object	Obtained by the difference in height between the centroid and the lowest point
Location of the object	defined by the centroid of the cluster
3d object structure	Obtained from the cluster

Furthermore, the region growing approach can be applied in future works for the reconstruction of points belonging to the OOIs**(Soilán et al., 2016)**. Since this study focuses on positioning the object in the 3d point cloud, further improvement in the point cloud structure is not considered. However, if these 3d representations of the OOIs are used to build a deep learning model, then appropriate methods can be followed in improvising the object structure.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

6. DISCUSSIONS

This section discusses the advantages and limitations of the methods implemented and the critical evaluation of the outcome obtained. Similar to chapter 4 and 5, this chapter is subdivided into three sections, namely;

6.1 Discussion on Object Detection and Classification
6.2 Discussion on Character Recognition from detected objects
6.3 Discussion on Positioning the detected objects in the point cloud.

In this study, the RGB images are processed first to detect the objects, and the detected objects are projected in the 3d point cloud. The image first approach reduces the computational complexity to a greater extent as it requires low processing power due to its lower dimensions(2D).

6.1. Discussion on Object detection and classification

6.1.1. Methodology

- While generating the annotation for the kilometer markers, only the marker that occurs in the left portion of the image(left side of the track) is considered. Similarly, for the signal, only those occur in the right portion of the image(right side of the track) are considered. This is due to the track geometry and direction of the rail. The OOIs, which are present in other regions of the images, are avoided as they are poorly scaled or blurred or overexposed.
- Due to the direction of tracks, a few samples representing the backside of the signals are also collected. Since both the back and the front side of the signals have a similar pixel structure, they are placed in the same class.
- There exist few signals with circular kilometer boards mounted on them; those objects are excluded from the training samples as they are very fewer in numbers. However, the top portion of the signal is annotated and tagged under the signal class.
- The negative samples are created in such a way that includes ground cover, vegetation, transition of grass cover to the soil, transition of the green field to the sky, Catenary wires, tracks, switches, etc. And these samples are also data augmented, similar to positive samples, to maintain the difference in characteristics while training.

6.1.2. Results

- As most of the training samples are acquired from images consisting of two parallel track, the OOIs in three-track images differs in geometry. The objects present in such images are scaled out, or they lie in the edge of the images making the classification task difficult. This issue can be minimized in future studies by introducing positive samples from such images so that the classifier can learn to detect objects in such geometry.
- The results from the polynomial filters are of low accuracy compared to RBF and linear kernels. However, the error of commission and omission from those filters provides a better understanding of the input data and classification process. For kilometer markers, the polynomial kernels receive an error of commission of 11.28% (**Table 5-2**), which implies, high ratio of positive samples are misclassified as negative samples. Likewise, an error of omission of 7.80% is obtained for signals **(Table 5-5)**, which states, a high ratio of negative samples classified as positive samples. This scenario explains the lack of distinguishable features in the kilometer marker samples; and the high similarity of signal samples with the negative samples.

- While detecting the objects using a sliding window, the images are processed in five different scales. However, in some cases (see appendix), the bounding box region covers a bigger area surrounding the OOIs. This problem is reduced to some extent by applying a threshold in the detection window. But still, there exist few samples where the OOIs are surrounded by a bigger bounding box. The issue can be further mitigated by improvising the feature extraction methods since the bigger bounding box region is due to the histogram matching in that area.
- While detecting the OOIs in the samples, the top of the OOI is detected first(kilometer board, and signaling instrument), and a bigger bounding box covering the entire structure is drawn. This approach help is capturing all the information of the OOIs when projected in 3d point clouds.
- A threshold of 0.8 in feature matching produces high confidence object detection results and a relatively small region of the bounding box area. So it is computationally easier to process the 3d point clouds. However, setting a higher threshold removes objects that are true positives with low scores. And setting a lower threshold increase the region of the bounding boxes, thus making it computationally complex.

6.2. Discussion on Character Recognition from detected objects.

6.2.1. Methodology

- When a single bounding box region containing the kilometer marker is obtained as the input, the kilometer signs are cropped easily with a fixed dimension, as mentioned in **section 4.2.4**.
- In the case of more than one bounding box, image segments are created along the left portion of the image above the railway track. And the bounding box with the highest confidence in that image region is retrieved, which most probably contains the kilometer marker.
- The process of a contrast stretch and HLS changes not only makes the digits more recognizable but also removes motion blur and occlusion to some extent. However, this process is not required for all the images. So, this step is applied only to the kilometer markers that are present at the edges of the images where the motion blur is predominant.
- As the input image of the kilometer marker is of different orientation and scale, they are resized and rotated to enhance the accuracy of digit recognition.
- The model, which comprises Resnet50 and YOLOv2, already possess the learned parameters from the Imagenet dataset. Furthermore, they are trained in the SVHN dataset for digit recognition, which makes it robust enough to perform digit recognition in a new domain dataset that is not involved in training.
- One main advantage of using YOLO architecture is that it can process the recognition task at a higher frame rate making it the right choice if the model hs to be implemented in the MLS platform. The study uses version two of the YOLO framework as it works well with Resnet; however newer, and different architecture could opt for future works.

6.2.2. Results

- The SVHN dataset consists of rectangular images with house numbers placed horizontally. When the model is trained in that dataset, it recognizes the digits that are positioned horizontally. However, the model struggles to recognize the digits that are positioned vertically(in a row). This is due to the lack of vertical image samples in the training dataset that causes the model to fail while recognizing such digits. The issue can be mitigated by introducing a set of annotated images that are positioned vertically one over the other—and retraining the last layer with this new domain dataset. By doing so, the model will gain new learnable information that will help is recognizing the vertically positioned objects. However, this process is computationally expensive and also involves manual annotations of the images.

- As known, the kilometer markers extend up to three digits in the signboard. So, a model that could recognize up to five digits is redundant. The model can be retrained with images consisting of up to three digits so that the recognition accuracy for each digit increases substantially.
- Since the output of the recognition model is of probability estimates for each of the individual digits, a threshold can be held up to obtain results of high probability. However, setting up a high threshold would also eliminate true positives that are present with low probability estimates. So, a mid-range threshold of 0.5 is fixed that would allow enough estimates to be correctly recognized.
- Furthermore, in future works,

  (a) The recognition task can be coupled with the localization of digits in a bigger image using a sliding window approach. So that, the pre-processing of kilometer sign images can be reduced drastically.

  (b) The recognized kilometer marker values can be further checked against their location on the track as they display the kilometer values that match to their location along the track.

6.3. Discussion on the Positioning the detected objects in the point cloud

6.3.1. Methodology

- For mapping, the object detected image with the relevant point cloud file; the GPS time attribute is utilized similar to the approach followed by **Soilán et al., (2016)**. For instance, the GPS time of the object detected image is taken as $t_{obj,}$ which represents the time at which the photo is taken. The time t_{obj} is given an offset of $t_{obj} \pm 2$ s so that, all the points lying within this time offset is obtained and further analyzed to retrieve the point cloud structure of the OOI.
- To identify the ground elements from the point clouds enclosed in the bounding boxes, the RANSAC shape detection algorithm is employed similar to the approach followed by **X. Chen et al., (2015)**. By doing so, the planar surfaces are well distinguished. As a result, the ground elements are represented by a single flat plane.
- To understand the importance of the 4th dimension – Intensity values of the point clouds, two clustering methods are tested. Clustering with intensity values(x,y,z, intensity) and clustering without intensity values(x,y,z). And, similar results are obtained in terms of the cluster count.

6.3.2. Results

- Geometric properties of OOIs such as the 3d position of the object, height of the object, and 3d point cloud structure of the object are retrieved at the end of point cloud processing pipeline.
- In some cases, there exist two or more pole-like structure inside the bounding box region. Those similar pole structures may belong to the poles holding the catenary wires or cables. In such a scenario, the knowledge-based approach proposed by **F Li et al., (2017)** can be employed along with a height offset to differentiate the pole-like structure belonging to the OOIs.(for this study, the knowledge-based approach is not adapted, included as future works)
- The individual point cloud structures of the OOIs extracted in this study can be further used in constructing a more robust deep learning model for point cloud segmentation. For instance, a Pointnet++ model **(Qi et al., 2017)**.
- The extracted point cloud structure of the objects can be further processed with region growing/merging algorithms to reconstruct the points in the object surface for better representation.

7. CONCLUSION AND RECOMMENDATIONS

7.1. Conclusion

This study proposes a data fusion approach of RGB images and 3d point clouds to detect objects from a railway environment. The primary objective of this study is to develop a three-step framework that performs (a) object detection and classification, (b) Character recognition from the detected objects and, (c) Positioning the detected objects in 3d point clouds. For this study, kilometer markers and the signals are chosen as the objects of interest. First, the RGB images from the railway corridor are taken, and training samples containing the OOIs are annotated. The positive samples are then augmented to increase the sample size. The color features and the Histogram of Oriented Gradient(HOG) features are extracted from the samples and trained inside the SVM classifier. In this study, three different kernels of SVM classifiers are used; they are – Radial Basis Function(RBF), Polynomial, and Linear kernels. The extracted features are trained using these kernels, and their performance is evaluated. As a result, Both RBF and Linear kernel produced higher accuracy while classifying kilometer markers, and for signals, the RBF kernel produced higher accuracy compared to the other two. So a conclusive decision is made to construct a multi-class SVM with RBF kernels for object detection and classification. The overall accuracy of this classifier is computed, which is 96%. The newly constructed multi-class SVM is further selected for object detection in images. A sliding window approach is then employed to perform feature matching of windows with the trained SVM classifier on different scales. The detected objects inside the bounding boxes are featured along with the heat maps. The completeness and correctness of the model are estimated over a small sample size of ground truth, and they are 85% and 65.4%, respectively.

The object detection output obtained from this first step contain kilometer markers from which the kilometer value can be extracted. The task of character recognition is performed using a hybrid model of Resnet50 and YOLOv2 architecture. The model is trained using an open-source dataset named SVHN(Street View House Number), which contains a huge collection of house number images taken from google street view imagery. The idea here is to introduce a model that is trained on house number images to recognize digits from kilometer markers images since both of them occur in natural scenes. The feature extractor layers and the classification layers are trained separately. Due to the computational complexity, the batch size, learnable parameters, and the number of input samples are reduced. And the precision and recall are calculated, and an overall f-score of 0.768 is achieved. The model is not implemented to recognize the digit values from the kilometer marker images. For which, the model achieved a completeness score of 80% with a correctness score of 86.9%. Now that the semantic information of the objects is obtained, geometric information is to be retrieved.

The bounding box coordinates obtained after object detection is projected into the 3d point clouds. And the points are filtered based on the intensity and radiometrically adjusted. Furthermore, the RANSAC shape detection filter is implemented to delineate the ground surfaces and the planar surface that are present at the top of the object structure. Finally, the density-based clustering (DBSCAN) is performed to identify the point cloud clusters from the images, and the clusters are delineated, which in terms provides the structure of the OOIs. And the location of the OOIs is defined by taking the centroid of the cluster.

As a result, the proposed framework can derive the pixel structure of the object, character information from the object, 3d point cloud structure of the object, and the exact location of the object. However, this framework has to be further fine-tuned to make it a complete object detection pipeline.

7.2. Research Questions: Answered

The research questions from the **section 1.3.2** are answered in this section.

1. How to cope with the low quantity of training data, and what are the mitigation methods?
   Data augmentation methods are followed to increase the quantity of the training sample. As mentioned in **section 4.1.3**, filters such as edge enhancements, translate, contrast correction, etc. are performed.

2. What are the parameters to be considered while performing feature extraction?
   The color parameters of the OOIs in different color spaces are analyzed to define an optimum feature extraction method. The YCrCb color space is selected for the kilometer markers, and HSV color space is chosen for signals.

3. Which kernel function of SVM yields better results in the classification of objects?
   In binary classification, both RBF and linear kernels yielded an overall accuracy of 97.71% while classifying kilometer markers. Whereas, for signals, the RBF kernel produced an overall accuracy of 99.39%. However, the multi-class SVM with RBF kernel produced an overall accuracy of 96%, which is selected for object detection.

4. How can characters on sign markers be extracted from images of varying orientations and range?
   A model trained on naturally taken house number images is used to recognize the digits from images of different scales and orientations.

5. How to estimate the performance accuracy of the recognition models?
   The user performance of the model is estimated from a set of manually annotated digit images. And it is found, the model achieved a completeness score of 80% and a correctness score of 86.9%

6. How to perform a reliable mapping for the OOI from the image plane into the point cloud?
   A combination of bounding box coordinates along with GPS timestamp information is used to position the OOIs in the 3d point clouds accurately

7.3. Recommendation

The recommendations from this research are:

- For future studies, the task of object detection and character recognition can be combined and performed using a single model. However, one has to be keen on generating a reasonable size of the training dataset.
- The point cloud structure of each object obtained from this study can be further utilized in building a deep learning model that performs segmentation tasks in the 3d point clouds.
- The YOLO framework can be further analyzed for its suitability in performing object detection in the railway environment. Since they work at a very high framerate, it can be further researched to develop an onboard object detection system in the rails.
- The machine learning classifier, like SVM, works well with binary or two classes while performing object detection. However, deep learning approaches can be studied further for modeling the multi-class object detection problem.
- Domain adapted learning protocol can be implemented for signboard classifications. For example, the model that is pre-trained in road traffic sign detection can be further finetuned for object detection in the railway environment.

LIST OF REFERENCES

Anthony, G., Gregg, H., & Tshilidzi, M. (2007). Image classification using SVMs: One-Against-One Vs One-against-All. *28th Asian Conference on Remote Sensing 2007, ACRS 2007*, *2*, 801–806.

Arastounia, M. (2012). Automatic Classification of LiDAR Point Clouds in A Railway Environment. *University of Twente, Netherlands.*

Arastounia, M. (2015). Automated recognition of railroad infrastructure in rural areas from LIDAR data. *Remote Sensing*, *7*(11), 14916–14938. https://doi.org/10.3390/rs71114916

Azizmalayeri, F., Peyghambarzadeh, S. M. M., Khotanlou, H., & Salarpour, A. (2018). Kernel correlation based CNN for point cloud classification task. *2018 8th International Conference on Computer and Knowledge Engineering, ICCKE 2018*, (Iccke), 200–204. https://doi.org/10.1109/ICCKE.2018.8566273

Babahajiani, P., Fan, L., & Gabbouj, M. (2015). Object recognition in 3D point cloud of urban street scene. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9008, pp. 177–190). Springer, Cham. https://doi.org/10.1007/978-3-319-16628-5_13

Balali, V., & Golparvar-Fard, M. (2016). Evaluation of Multiclass Traffic Sign Detection and Classification Methods for U.S. Roadway Asset Inventory Management. *Journal of Computing in Civil Engineering*, *30*(2), 1–16. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000491

Beger, R., Gedrange, C., Hecht, R., & Neubert, M. (2011). Data fusion of extremely high resolution aerial imagery and LiDAR data for automated railroad centre line reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(6 SUPPL.), S40–S51. https://doi.org/10.1016/j.isprsjprs.2011.09.012

Ben-Shabat, Y., Lindenbaum, M., & Fischer, A. (2018). 3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, *3*(4), 3145–3152. https://doi.org/10.1109/LRA.2018.2850061

Che, E., Jung, J., & Olsen, M. J. (2019, February 16). Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors (Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/s19040810

Chen, C., Li, X., Belkacem, A. N., Qiao, Z., Dong, E., Tan, W., & Shin, D. (2019). The Mixed Kernel Function SVM-Based Point Cloud Classification. *International Journal of Precision Engineering and Manufacturing*, *20*(5), 737–747. https://doi.org/10.1007/s12541-019-00102-3

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A., Ma, H., Fidler, S., & Urtasun, R. (2015). 3D object proposals for accurate object class detection. *Advances in Neural Information Processing Systems*, *2015-Janua*, 424–432.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, *I*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Diaz, J. C. F., Carter, W. E., Shrestha, R. L., & Glennie, C. L. (2013). Lidar remote sensing. In *Handbook of Satellite Applications* (Vol. 2, pp. 757–808). Cham: Springer International Publishing. https://doi.org/10.1007/978-1-4419-7671-0_44

EIM-EFRTC-CER Working Group. (2012). *Market Strategies for Track Maintenance & Renewal*. Retrieved from http://www.cer.be/publications/brochures-studies-and-reports/report-eim-efrtc-cer-working-group-market-strategies

Elberink, S. O., & Khoshelham, K. (2015). Automatic extraction of railroad centerlines from Mobile Laser Scanning data. *Remote Sensing*, *7*(5), 5565–5583. https://doi.org/10.3390/rs70505565

Förstner, W., & Wrobel, B. P. (2016). Photogrammetric Computer Vision. Geometry and Computing. In *Photogrammetric Computer Vision. Geometry and Computing*. https://doi.org/10.1007/978-3-319-11550-4

Gevaert, C. M., Persello, C., & Vosselman, G. (2016). Optimizing multiple kernel learning for the classification of UAV data. *Remote Sensing*, *8*(12). https://doi.org/10.3390/rs8121025

Gézero, L., & Antunes, C. (2019). Automated Three-Dimensional Linear Elements Extraction from Mobile LiDAR Point Clouds in Railway Environments. *Infrastructures*, *4*(3), 46. https://doi.org/10.3390/infrastructures4030046

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2014). Multi-digit number recognition from street view imagery using deep convolutional neural networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 1–13.

Hackel, T., Stein, D., Maindorfer, I., Lauer, M., & Reiterer, A. (2015). Track detection in 3D laser scanning data of railway infrastructure. In *Conference Record - IEEE Instrumentation and Measurement Technology Conference* (Vol. 2015-July, pp. 693–698). IEEE. https://doi.org/10.1109/I2MTC.2015.7151352

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hemmes, T. (2018). *Classification of large scale outdoor point clouds using convolutional neural networks*.

Himmelsbach, M., Luettel, T., & Wuensche, H. J. (2009). Real-time object classification in 3D point clouds using point feature histograms. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 994–1000. https://doi.org/10.1109/IROS.2009.5354493

Hosang, J., Benenson, R., & Schiele, B. (2017). Learning non-maximum suppression. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6469–6477. https://doi.org/10.1109/CVPR.2017.685

Huanrui, H. (2016). New mixed kernel functions of SVM used in pattern recognition. *Cybernetics and Information Technologies*, *16*(5), 5–14. https://doi.org/10.1515/cait-2016-0047

Huhle, B., Magnusson, M., Straßer, W., & Lilienthal, A. J. (2008). Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform. In *Proceedings - IEEE International Conference on Robotics and Automation* (pp. 4025–4030). IEEE. https://doi.org/10.1109/ROBOT.2008.4543829

Kari, T., Gao, W., Tuluhong, A., Yaermaimaiti, Y., & Zhang, Z. (2018). Mixed kernel function support vector regression with genetic algorithm for forecasting dissolved gas content in power transformers. *Energies*, *11*(9). https://doi.org/10.3390/en11092437

Krispel, G., Opitz, M., Waltner, G., Possegger, H., & Bischof, H. (2019). FuseSeg: LiDAR Point Cloud Segmentation Fusing Multi-Modal Data. Retrieved from http://arxiv.org/abs/1912.08487

Lamon, P., Stachniss, C., & Triebel, R. (2006). Mapping with an autonomous car. *International Conference on Intelligent Robots and Systems (IROS)*, 11. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.7803&rep=rep1&type=pdf

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 12689–12697. https://doi.org/10.1109/CVPR.2019.01298

Leslar, M., Perry, G., & McNease, K. (2010). Using mobile lidar to survey a railway line for asset inventory. *American Society for Photogrammetry and Remote Sensing Annual Conference 2010: Opportunities for Emerging Geospatial Technologies*, *1*, 526–533.

Li, F, Elberink, S. O., & Vosselman, G. (2017). Semantic labelling of road furniture in mobile laser scanning data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* (Vol. 42, pp. 247–254). https://doi.org/10.5194/isprs-archives-XLII-2-W7-247-2017

Li, Fashuai, Lehtomäki, M., Oude Elberink, S., Vosselman, G., Kukko, A., Puttonen, E., … Hyyppä, J. (2019). Semantic segmentation of road furniture in mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *154*(May), 98–113. https://doi.org/10.1016/j.isprsjprs.2019.06.001

Lidén, T. (2015). Railway infrastructure maintenance - A survey of planning problems and conducted research. In *Transportation Research Procedia* (Vol. 10, pp. 574–583). https://doi.org/10.1016/j.trpro.2015.09.011

Loussaief, S., & Abdelkrim, A. (2018). Machine Learning framework for image classification. *Advances in Science, Technology and Engineering Systems*, *3*(1), 1–10. https://doi.org/10.25046/aj030101

Lu, Z., Lu, J., Ge, Q., & Zhan, T. (2019). Multi-object detection method based on YOLO and resnet hybrid networks. In *2019 4th IEEE International Conference on Advanced Robotics and Mechatronics, ICARM 2019* (pp. 827–832). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICARM.2019.8833671

Mantero, P., Moser, G., & Serpico, S. B. (2004). Partially supervised classification of remote sensing images using SVM-based probability density estimation. In *2003 IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data* (pp. 327–336). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/WARSD.2003.1295212

Mashao, D. J. (2005). Comparing SVM and GMM classifiers on the parametric feature-sets. In *SAIEE*

Africa Research Journal (Vol. 96, pp. 77–86).

Morgan, D. (2009). Using Mobile Lidar to Survey Railway Infrastructure. Lynx Mobile Mapper. In *Innovative Technologies for an Efficient Geospatial Management of Earth Resources, Lake Baikal, Listvyanka, Russia Federation* (p. 9). Retrieved from https://www.fig.net/resources/proceedings/2009/lakebaikal_2009_comm6/papers/06_dainamorgan.pdf

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). The Street View House Numbers (SVHN) Dataset. Retrieved June 2, 2020, from http://ufldl.stanford.edu/housenumbers/

Niina, Y., Honma, R., Honma, Y., Kondo, K., Tsuji, K., Hiramatsu, T., & Oketani, E. (2018). Automatic rail extraction and celarance check with a point cloud captured by MLS in a Railway. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2), 767–771. https://doi.org/10.5194/isprs-archives-XLII-2-767-2018

Oude Elberink, S., Khoshelham, K., Arastounia, M., & Díaz Benito, D. (2013). Rail Track Detection and Modelling in Mobile Laser Scanner Data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5W2), 223–228. https://doi.org/10.5194/isprsannals-II-5-W2-223-2013

Pu, S., Rutzinger, M., Vosselman, G., & Oude Elberink, S. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(6 SUPPL.), S28–S39. https://doi.org/10.1016/j.isprsjprs.2011.08.006

Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). Frustum PointNets for 3D Object Detection from RGB-D Data. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 918–927. https://doi.org/10.1109/CVPR.2018.00102

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (Vol. 2017-Decem, pp. 5100–5109). Retrieved from http://arxiv.org/abs/1706.02413

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Shi, S., Wang, X., & Li, H. (2019). PointRCNN: 3D object proposal generation and detection from point cloud. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 770–779. https://doi.org/10.1109/CVPR.2019.00086

Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P. (2016). Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 92–101. https://doi.org/10.1016/j.isprsjprs.2016.01.019

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., & Guibas, L. (2019). KPConv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 6410–6419. https://doi.org/10.1109/ICCV.2019.00651

Wang, H., Cai, Z., Luo, H., Wang, C., Li, P., Yang, W., … Li, J. (2012). Automatic road extraction from mobile laser scanning data. In *Proceedings of International Conference on Computer Vision in Remote Sensing, CVRS 2012* (pp. 136–139). https://doi.org/10.1109/CVRS.2012.6421248

Wang, Y., Chen, Q., Zhu, Q., Liu, L., Li, C., & Zheng, D. (2019). A survey of mobile laser scanning applications and key techniques over urban areas. *Remote Sensing*, *11*(13), 1–20. https://doi.org/10.3390/rs11131540

Wu, B., Wan, A., Yue, X., & Keutzer, K. (2018). SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. *Proceedings - IEEE International Conference on Robotics and Automation*, 1887–1893. https://doi.org/10.1109/ICRA.2018.8462926

Wu, B., Zhou, X., Zhao, S., Yue, X., & Keutzer, K. (2019). SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. *Proceedings - IEEE International Conference on Robotics and Automation*, *2019-May*, 4376–4382. https://doi.org/10.1109/ICRA.2019.8793495

Xu, K., Yang, Z., Xu, Y., & Feng, L. (2019). A Novel Interactive Fusion Method with Images and Point Clouds for 3D Object Detection Kai, 9. https://doi.org/10.3390/app9061065

Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2014). Aggregate channel features for multi-view face detection. In *IJCB 2014 - 2014 IEEE/IAPR International Joint Conference on Biometrics*. https://doi.org/10.1109/BTAS.2014.6996284

Zhang, J., Lin, X., & Ning, X. (2013). SVM-Based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sensing*, *5*(8), 3749–3775. https://doi.org/10.3390/rs5083749

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

Zhu, L., Jaakkola, A., & Hyyppä, J. (2013). THE USE OF MOBILE LASER SCANNING DATA AND UNMANNED AERIAL VEHICLE IMAGES FOR 3D MODEL RECONSTRUCTION. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-1/W2*(September), 419–423.

APPENDIX - I

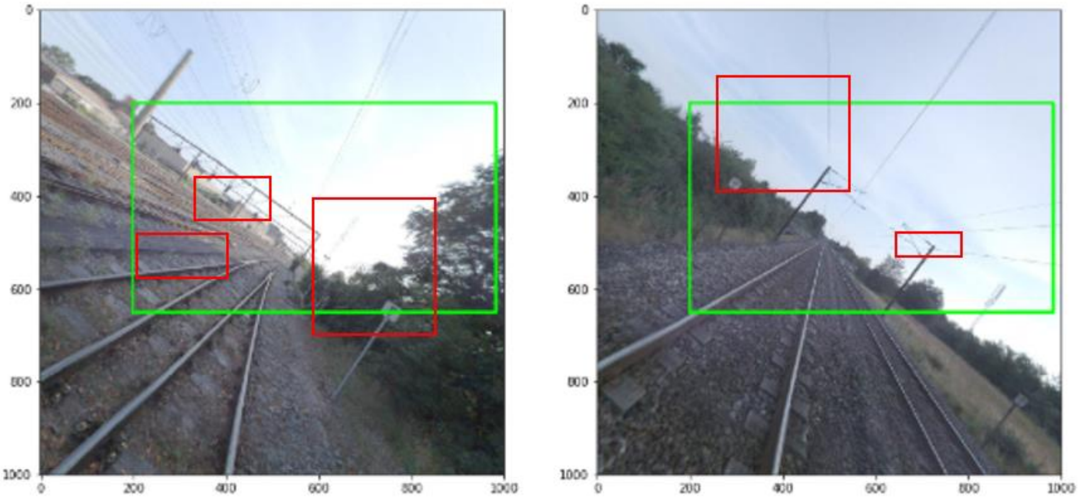


Figure 7-1 The objects covered by larger bounding box region

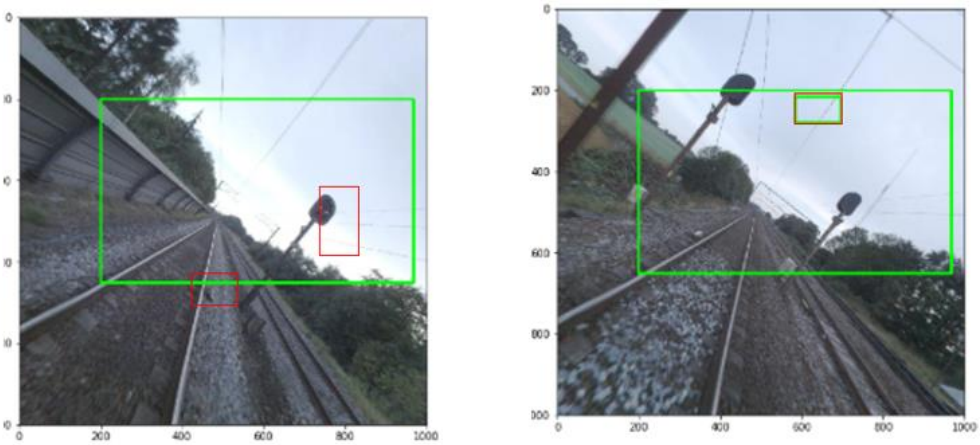
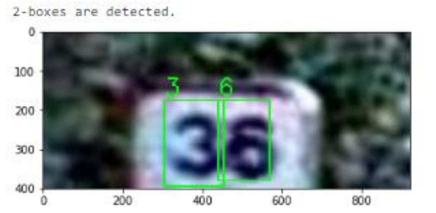
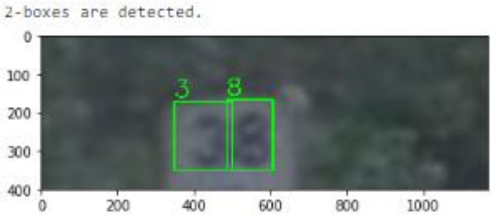
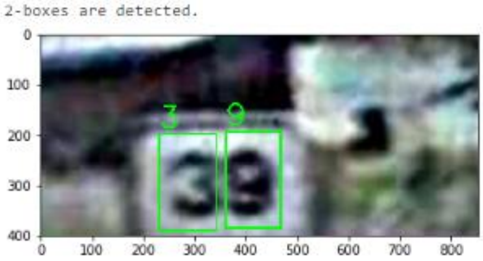
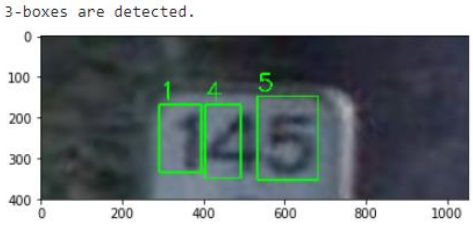
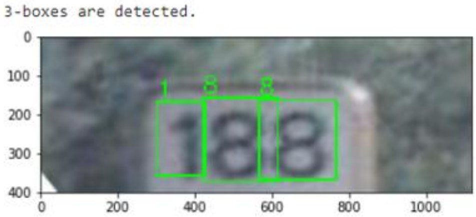


Figure 0-2 Misclassification of signal objects

APPENDIX – II

Table 0-1 The probability estimates for each of the digits

Digit Recognition output	Score for 1st digit	Score for 2nd digit	Score of 3rd digit
 2-boxes are detected. 	0.91314906	0.7855064	
 2-boxes are detected. 	0.61133546	0.9334358	
 2-boxes are detected. 	0.6910671	0.58796406	
 3-boxes are detected. 	0.60456216	0.61861753	0.855561
 3-boxes are detected. 	0.53276426	0.77921176	0.82047397

APPENDIX – III

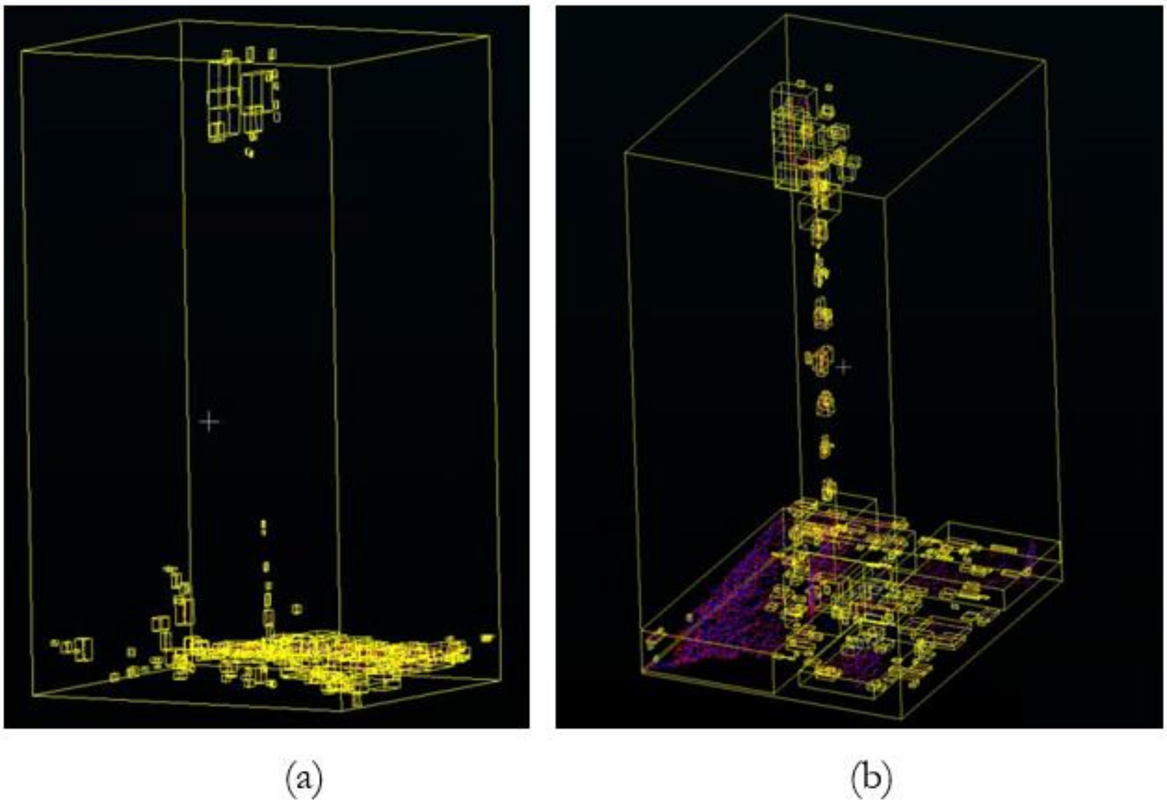


Figure 0-1 The connected components for Octree level 8 and minimum points per component is 10 (a)Kilometer marker (b) Sign markers