Autonomous Switching of Electric Locomotives in Neutral Sections

By

Christopher Thembinkosi Mcineka

(21959502)

A thesis submitted to the Faculty of Engineering and the Built Environment in fulfilment of the academic requirements for the degree of

MASTER OF ENGINEERING

in

ELECTRONIC AND COMPUTER ENGINEERING

in the

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

at the

DURBAN UNIVERSITY OF TECHNOLOGY

Supervisor: Dr Nelendran Pillay

Co-supervisor: Dr Serendra Reddy

January 2023

Preface

I, the undersigned, Mr Christopher Thembinkosi Mcineka, declare that the work embodied in this thesis, titled “Autonomous Switching of Electric Locomotives in Neutral Sections”, forms my contribution to the research work carried out under the guidance of Dr Nelendran Pillay and Dr Serendra Reddy at the Durban University of Technology. I declare that all materials used in this thesis are my original work, except in-text citations and references are used to acknowledge the works of others. There has not been any submission of work contained here, in part or whole, for a degree at any other university.

Christopher Mcineka

January 2023

Declaration 1: Supervisor

According to the contents of this thesis, as the candidates' Supervisor, I agree to the submission of this thesis.

Dr N Pillay

(Main Supervisor)

January 2022

Dr S Reddy

(Co-supervisor)

Declaration 2: Plagiarism

I, Christopher Thembinkosi Mcineka, declare in this research that:

1. The conducted research presented in this thesis is my original work, apart from where stated.
2. The research in this thesis has not been submitted to another university to obtain a master's degree or any form of academic qualification.
3. This thesis does not plagiarise other people's work, such as data, pictures, graphs, or other information unless expressly acknowledged as being sourced from other persons.
4. This research also does not contain other people's writings; however, where other sources have been used, then:
    a) When their exact words have been used, quotation marks will encapsulate their writings, and references will be included.
    b) Their writings have been re-worded, but the general information attributed to them has been referenced.
5. The thesis, where graphics, tables and text have been copied and pasted from the internet, has been acknowledged in the reference section.

Declaration 3: Publications

I, Christopher Thembinkosi Mcineka, declare that the following publications came of this thesis.

1. C. T. Mcineka and S. Reddy, "Automatic Switching of Electric Locomotives in Neutral Sections," in *Conference on Information Communications Technology and Society (ICTAS)*, 10-11 March 2021, pp. 97-102, doi: 10.1109/ICTAS50802.2021.9394969.
2. C. T. Mcineka and N. Pillay, "Machine Learning Classifiers Based on HoG Features Extracted from Locomotive Neutral Section Images," in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, 27-28 Oct. 2022 2022, pp. 1-6, doi: 10.1109/ICEET56468.2022.10007093.

Acknowledgement

I want to express my sincere gratitude and appreciation to my Supervisor, Dr Nelendran Pillay, for his support and guidance throughout this research. His friendly attitude and insightful feedback towards compiling my thesis. Furthermore, I could prepare a second conference paper through his support and guidance. He was also helpful in addressing administrative issues and would resolve them promptly.

I am also thankful to my Co-supervisor, Dr Serendra Reddy, for his invaluable guidance, knowledge, and expertise throughout this research. He provided guidance and feedback with literature and methodology sections that resulted in the publication of a conference paper.

I would also like to thank Transnet Freight Rail, Vryheid, Electrical department for allowing me to install the two markers on their infrastructure for data collection.

Family is everything; to my life partner, the mother of my kids, Nonkululeko: the support and encouragement you have showered me with have impacted me positively during my stressful times. To my kids, nephews, and nieces, who had given me happiness and joy at times when I needed stress relief.

Lastly, I thank God, the Almighty, who has given me life and has sheltered me away from any harm, evil, or illnesses that might have prevented me from completing this research.

Abstract

Electrical locomotives traversing in a neutral section must switch off as they enter a different phase voltage. The current system used to auto-switch these electric locomotives requires two pairs of induction magnets installed adjacent in-between the rails and two sensors installed underneath the locomotives. However, the return cost of investment is low, maintenance costs increase due to failures, and locomotives do not auto-switch due to the degradation of magnet strength. Additionally, damage to sensors due to animal collisions or objects also causes switching failures, and vandalism and theft are some of the challenges limiting this switching scheme. Furthermore, the latter switching method does not align with the Transnet 4.0 strategy aimed at adopting the Fourth Industrial Revolution (4IR). Therefore, to align with the Fourth Industrial Revolution, this research proposed a computer vision-based approach to switch electric locomotives automatically. The requirements are a computer, a high-definition camera, and open and close markers. While the latter gives an overview of the hardware used, creating a new dataset with training and testing images allowed for developing a machine learning classification model. Firstly, image pre-processing converts the RGB images to greyscale then the noise is removed using a bilateral filter. Secondly, segmentation and marker extraction is performed by employing the Sobel operator and Circular Hough Transform. Thirdly, features are extracted using a Histogram of Oriented Gradients and employing Linear Support Vector Machine to perform classification. However, before selecting the latter classifier, the feature extractor is tested against Quadratic Support Vector Machine, K-Nearest Neighbour and Convolutional Neural Network. The model's accuracy is then measured using the training set and ground truth dataset. The test set is used to validate the model with evaluation methods such as a confusion matrix, F1-measure and 2-fold cross-validation.

Table of Contents

List of Figures

List of Tables

Abbreviations

4IR	: Fourth Industrial Revolution
ANN	: Artificial Neural Network
ANPR	: Automatic Number Plate Recognition
BoF	: Bag of Feature
BoW	: Bag of Word
CCD	: Charged Coupled Device
CHT	: Circular Hough Transform
CMOS	: Complementary Metal Oxide Semiconductor
CNN	: Convolutional Neural Network
DET	: Detection Error Trade-off
DPE	: Department of Public Enterprises
DSP	: Digital Signal Processing
ECOC	: Error Correcting Outputs Code
FPPW	: False positive per window
HoG	: Histogram of Oriented Gradients
HSI	: Hue Saturation Intensity
K-NN	: K-Nearest-Neighbour
LoG	: Laplacian of Gaussian
MAE	: Mean Absolute Error
MSE	: Mean Square Error
MSER	: Maximally Stable External Region
NS	: Neutral Section
OHTE	: Over-Head Track Equipment
OoI	: Object of Interest
PSNR	: Peak Signal Noise Ratio
QM	: Quality Metric
QP	: Quadratic Programming
R-CNN	: Region-Based-Convolutional Neutral Network

RGB : Red, Green Blue

RoI : Region of Interest

RPN : Region proposal network

SIFT : Scale Invariant Feature Transform

SOE : State-Owned Enterprise

SURF : Speed Up Robust Feature

SVM : Support Vector Machine

TFR : Transnet Freight Rail

TNPA : Transnet National Port Authority

TPT : Transnet Port Terminal

TRE : Transnet Rail Engineering

VCB : Vacuum Circuit Breaker

YCbCr : Luminance Chroma blue Chroma red

YOLO : You Only Look Once

Chapter 1: Introduction

1.1 Introduction

Circa 1760 to 1830 was the first industrial revolution, driven by mechanisation through the invention of the steam power engine [1]. The second industrial revolution began around 1870, spurred on by a deeper understanding and consistent harnessing of electricity for use in mechanized mass production. The third industrial revolution started in the 1950s with the birth and mass production of semiconductors; this gave rise to computation, automation and the internet [2]. The Fourth Industrial Revolution (4IR) is said to have begun around 2012, bringing in the era of cyber-physical systems, the internet of things, big data, AI and more [3]. Transnet, therefore, has embarked on a strategic plan called Transnet 4.0, which sort to align its goals with 4IR.

In South Africa, the Department of Public Enterprises (DPE) is the shareholder representative of State-Owned Enterprises (SOEs). The minister of Public Enterprises with the DPE has oversight responsibility of all SOEs, either in whole or part, for six of the approximately 700 SOEs' existing national, provincial, and local governments. Transnet is one of the SOEs under the DPE, with its business objective to transport freight. There are six (6) core divisions of Transnet, namely, Transnet Port Terminal (TPT), Transnet National Port Authority (TNPA), Transnet Rail Engineering (TRE), Transnet Property and Transnet Freight Rail (TFR). The research addresses some of the challenges in TFR and does not focus on the other divisions. Six operating corridors form the TFR division, among which is the North corridor: responsible for transporting coal, ferrochrome, Chrome Ore, Pulpwood, and other commodities. Transportation of these commodities is through the use of electric locomotives on 3kVDC and 25kVAC systems. Transnet proposed a 7-year plan called the Market Demand

Strategy (MDS), which aimed at refurbishing and maintaining ageing infrastructure to ensure freight is transported from road to rail cheaply and safely. However, the MDS strategy was succeeded by Transnet 4.0 to keep with the ethos of the 4IR. The new plan aims to expand the company and compete with other big railway companies through technological innovations. This strategy envisaged improved network availability and improved productivity by employing new 4IR technologies in the railway sectors. Therefore, the researcher identified that Transnet currently uses an outdated switching scheme for electric locomotives on their railway network.

Eskom supplies Transnet with 88kVAC three-phase: a 20MVA traction transformer (single unit) steps the voltage down to a 25kVAC traction system. Two single units are feeding to the Overhead Traction Equipment (OHTE): one unit is feeding with a different phase while the other unit feeds with another phase. A phase break or neutral section is installed between two traction substations feeding different phases on the same OHTE. The purpose of a neutral section is to provide a neutral point (grounded) and separation from the two phases to prevent flashovers due to high voltages caused by locomotives shorting both phases. As an electric locomotive approaches a neutral section, induction magnets installed between the rails on opposite sides cause each locomotive to switch off as it traverses from the first set of magnets and switches on at the other induction magnets. Switching each electric locomotive off prevents flashovers or huge arcing, which may cause catastrophic damage to the OHTE as each locomotive passes from one phase to the other. The on-switching ensures that the train (consist of locomotives) can continue operating past the neutral section. If a locomotive does not switch off, the business may lose revenue due to delayed trains or cancellations from their slots. Furthermore, routine preventive maintenance activities are reduced because teams repair incidental damages.

There are commonly three automatic Neutral Section (NS) switching schemes used worldwide: ground, pole and onboard. The first two types have several disadvantages,

including a relatively significant investment in infrastructure costs [4]. The Transnet freight rail coal-line business unit uses the third type: the onboard scheme for their NSs. The disadvantage of this switching scheme is the high cost of installing induction magnets on the railway for recognition. Induction magnets are prone to be stolen, which causes a risk because the onboard sensor will not detect the induction magnets. Since electric locomotives have to switch off, there is a time delay without electricity when traversing through a neutral section; faster-switching speeds are therefore required [4]. In addition to the Onboard-switching scheme's problems, the system is nevertheless prone to inconsistencies in train switching: poor quality during maintenance, failure of the onboard sensor or deteriorated induction magnet strength. Transnet commonly employs an onboard configuration switching scheme at their neutral sections. A component of a neutral section installation, which includes two different phases, induction magnets on the opposite ends of the tracks, and a phase break, is illustrated in Fig. 1. 1.

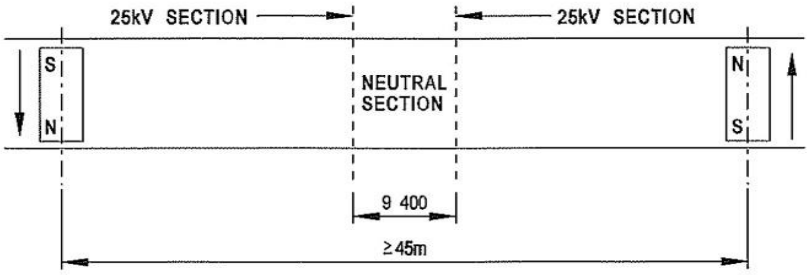


Fig. 1. 1: Current neutral section switching scheme [5]

A photograph of a typical NS onboard-switching scheme found on the Transnet railway network is shown in Fig. 1. 2; the components include (i) “N” markers between neutral sections to define the area (rectangle A), (ii) an Arthur Flury NS25 [6] phase break to separate the single-phase voltages with the centre grounded to form a neutral section (rectangle B), (iii)

induction track magnets, found on both sides of the neutral sections (rectangle C), and chevron boards used to alert the train driver when approaching a neutral section (not shown).

Fig. 1. 2: Overview of the current neutral section with "N" only

Installed at the bottom of the electric locomotives is a magnetic sensing device which forms part of the onboard-switching scheme. As illustrated in Fig. 1. 3, two sets of induction relays are activated when a magnetic flux is detected, which subsequently causes the Vacuum Circuit Breaker (VCB) to open. Similarly, when the locomotives pass the second set of magnets, the second set of relay sensors is activated, causing the VCB to close.

Fig. 1. 3: Induction magnet sensor underneath an electric locomotive

This research proposes using a branch of artificial intelligence (AI) technology to switch electric locomotives in an NS. Chen [4] proposed a method that employed computer vision to detect two Chinese markers for switching electric locomotives as they traverse through a neutral section. There is limited research on marker detection using computer vision and image processing methods and their associated use in the automatic switching of electric locomotives on railway systems. There is nevertheless a significant amount of research on number plates and traffic signs detection [7-9].

Gonzalez et al. [10] define computer vision as a branch of AI that uses computers to emulate human vision. Computer vision also includes learning and being able to make inferences and take actions based on visual inputs [10]. Image processing on the other hand is described as the process of processing digital images by employing computers [10]. The authors distinguish between image processing and computer vision in that image processing is a discipline with the input and output process using images. Additionally, digital images are defined as a two-dimensional function with spatial coordinates (x, y) and intensity (f) with finite values [10].

1.2 Research Problem Statement

The three most widely used auto-switching schemes, viz. ground, pole and onboard, suffer from significant investment costs and failures [4, 11]. Transnet is also affected by the latter shortcomings of the onboard auto-switching scheme. The costs are mainly owing to the installation of the NS system. In addition, the failures caused by trains not switching also contribute to the cost. Human negligence is also a common cause of neutral section failures [11]. The magnetic sensing device, discussed above and illustrated in Fig. 1. 3, is also affected by collision or dragging objects leading to the affected locomotive not switching. Objects such as cattle colliding with the locomotive and wood logs that fell between the rails also knock the sensor(s), causing misalignment with the induction magnets. The sensor(s) fail to detect the induction magnet, subsequently causing the locomotive to fail to automatically switch itself off and on (depending on which sensor is affected). High investment costs and failures due to the degradation of magnetic flux and stolen induction magnets are some significant issues that need to be solved. Transnet 4.0 strategy seeks to address several railway challenges by employing Fourth Industrial Revolution (4IR) technologies. Technologies such as image processing and machine learning form part of 4IR; however, these have not widely been employed in the auto-switching trains in neutral sections [4]. This research proposes a method that employs computer vision and image processing techniques to automatically switch electric trains traversing a neutral section on the Transnet railway system.

1.3 Initial Impetus

The initial impetus for conducting this research was driven by Transnet 4.0 strategy, which sought to align itself with the fourth industrial revolution (4IR). The current switching schemes used in railway industries are obsolete, and Transnet has experienced numerous failures on one of these schemes; hence this also motivated the study. The initial study showed three switching

schemes with high investment costs and failures, with none implementing 4IR technologies. Artificial intelligence (AI) is one of the 4IR technologies, and computer vision is the field of AI that was chosen. The research study of employing computer vision to switch electric locomotives was justified by implementing an approach aligned with Transnet 4.0. Also, the Faculty of Engineering and Built Environment prompted the motivation for conducting this research to fulfil the thesis for awarding a Master of Engineering degree. The research study then progressed to a literature review to determine existing methods relevant to the proposed switching scheme to address the research problems.

1.4 Research Question

This research aimed to develop a model to switch electric locomotives in Transnet railway neutral sections automatically. Subsequently, a research question could be developed; "Can computer vision and image processing techniques be employed as a viable alternative for the automatic switching of electric trains as they traverse through the neutral sections?"

1.5 Research Aim and Objectives

The research will investigate computer vision and image processing methods to automate the switching of trains as they traverse through the neutral section.

a) Conduct a comprehensive literature review on relevant publications based on image detection, segmentation, and classification.

b) Develop a model to detect, segment, and classify markers found in the neutral section.

c) Test the performance of the model based on well-known statistical methods.

d) Provide recommendations for further research in this area.

e) Publish at least one conference paper.

Initially, a problem is identified and outlined under the problem statement section; hence a comprehensive literature review is conducted to answer a research question. The literature review focused on relevant publications on computer vision and image processing methods employed in switching electric locomotives at neutral sections. The research methodology applied is quantitative; therefore, the developed model and empirical evaluation techniques align with this methodology. The research question is answered through the developed model and the experiments conducted. Two conference papers were presented at the Information Communications Technology and Society (ICTAS) conference and the International Conference on Engineering and Emerging Technologies (ICEET) and published in IEEE Xplore.

Initially, two hundred images comprising the close and open markers are captured onsite and stored in a laptop hard drive. The images are acquired at different weather conditions and distances to simulate the actual conditions of the environment. The dataset containing these images is increased by introducing random noise and rotation. The dataset is 654 and split into 422 training images and 232 test images. The impetus of using a dataset with 654 images is explained in detail in Subsection 3.3.3 (c).

The model is developed using known algorithms to recognise the predefined markers, and MATLAB is used to implement and test these algorithms. First, training images are used to train the model, and then the manually generated ground truth images from the training images are used to measure the model's performance. Testing images are introduced to the trained model as unseen data to validate the overall performance or accuracy of the model. The model's accuracy is measured by applying a confusion matrix and F1-measure or F1-score, a statistic validation method employed to determine the precision, recall and overall accuracy.

Fig. 1. 4 illustrates the proposed installation of the markers to allow the model to detect and classify the two markers bidirectionally. The two markers (open or “N”) are positioned at the opposite sides of a neutral section facing the incoming direction of a train (consist of

locomotives), allowing for the train to switch off. The two markers (close or "C") are each mounted back-to-back with the "N" markers for the train approaching the neutral section in either direction, similarly for the train to switch on.

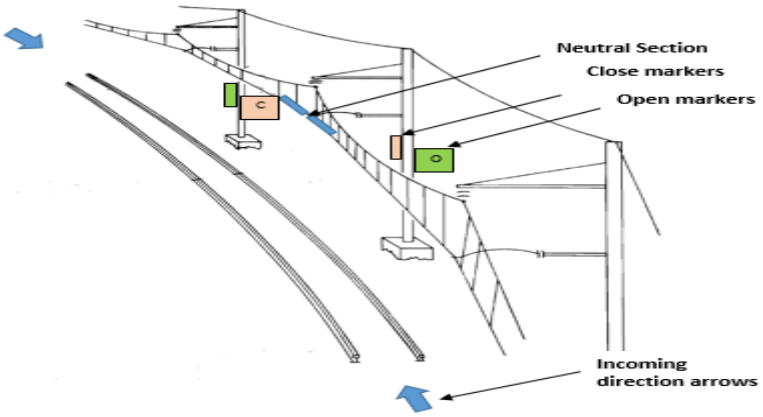


Fig. 1. 4: Proposed marker installation

1.6 Dataset Limitations

The research problem statement described in Section 1.2 proposes a computer vision system to address the problem. Therefore, this requires a new dataset to be created due to the scarcity of similar datasets. Furthermore, the limitation is that the images collected have a low resolution of 481*321 pixels because of the type of camera used. Due to financial constraints, a cheaper camera was purchased and used for this research—subsequently, Chapter 3 to Chapter 5 present images with low resolution.

1.7 Research Contributions

This research work makes the following contributions to the railway industries:

- The development of a computer vision and image process model to switch electric locomotives automatically.

- The publication of two conference papers employing computer vision and image processing techniques to switch electric locomotives.
- Creating a dataset of railway neutral section images.
- The investigation of commonly used classification and feature extraction algorithms enabled the proposition of a suitable model for the classification of neutral section markers.
- The latter also enables future work or researchers to improve on the model.

1.8 Structure of Thesis

Chapter 1 is an introduction to the background of the research and gives motivation for this research. The motivation is given through the research problem statement, objectives, and research outputs.

Chapter 2 provides a comprehensive literature review of relevant publications based on image localisation, segmentation, and classification from publications relevant to this research, such as number plate and traffic sign detection.

Chapter 3 presents detailed steps in developing the model for localising, segmenting, and classifying both markers.

Chapter 4 gives comprehensive results of the developed method, where well-known statistical methods are used to evaluate the model's accuracy.

Chapter 5 presents a concise discussion of the results obtained.

Chapter 6 concludes this thesis by highlighting the challenges, limitations and recommendations for future research.

Chapter 2: Literature Review

2.1 Introduction

This chapter gives a comprehensive review of related literature which influenced the work done in this research. The literature synthesis was carried out mainly by referring to books, web publications (google scholar), journal articles, thesis, and conference papers in common databases such as the Institute of Electrical and Electronic Engineers (IEEE) and the university library. The broad topics addressed during literature synthesis were acquisition sensor types, image pre-processing, object segmentation classification techniques and methods.

Therefore, the chapter is divided into four sections: conventional switching schemes, image pre-processing, segmentation, and classification. The first section discusses existing switching schemes and their advantages and disadvantages. Image pre-processing refers to the preferred image sensor employed in image acquisition and methods undertaken during pre-processing. The segmentation section reviews existing methods employed in segmenting Region of Interest (RoI) extrapolated from the object detection. Furthermore, it deals with different methods employed in object detection to find Objects of Interest (OoI) in RoIs. The detection stage is critical in ensuring that the RoI is accurately segmented. The classification investigates existing methods employed in classifying OoI, which are extrapolated from the RoI. The chapter concludes by giving an insight into the existing methods employed in the proposed research.

2.2 Conventional Switching Schemes

The configuration of a neutral section role is essential in ensuring the separation of two different phases. Electric locomotive, when traversing, ensures that no excessive arcing occurs, which may result in component failures, subsequently disrupting the service of trains.

Currently, three methods are employed to automatically switch trains: ground switching, pole switching and onboard-switching [12, 13].

The method of the ground-switching system is based on Vacuum Circuit Breakers (VCBs), which are installed on the ground and with sensors that detect the presence of trains. The detection of trains allows the VCB to operate, thereby switching on and off power as the train traverses through the neutral section.

Han et al. [13] mentioned that the current onboard system employed in China (Beijing and Hefei) suffers from power interruptions. These power interruptions occur on trains as they transition to a neutral section. The authors developed a ground-switching scheme employing a box ground substation. Axle counters were installed on the ground to detect the position of locomotives, enabling substations to switch breakers. They suggested that this approach will eliminate speed reduction due to power interruptions. The system uses mechanical switches VCB, which lag in switching speed and require frequent maintenance due to reduced life span.

Ran et al. [14] proposed replacing mechanical switches with power electronic switches such as Silicon Controlled Rectifier (SCR). The SCR used was a thyristor, and the traditional mechanical switches from the ground-switching scheme were replaced with the SCR switches. The authors further elaborated that mechanical switches such as VCB suffer from switching time; the switches cannot be accurately controlled and have a reduced lifespan due to over-voltages. Replacing mechanical switches for the ground-switching scheme with SCR eliminates over-voltages and slow switching times owing to the zero-crossing characteristics of the thyristor. SCRs to be controlled require a firing control circuit; this adds complexity to the neutral section control system. Xiong et al. [15] developed a Self-Supplied Gate Driver (SSGD) and pulse amplifier to control SCR. They developed the SSGD circuit with a combination of diodes, resistors, capacitors, and a current transformer to form a timing circuit which subsequently drives the driver circuits (firing control circuit). While the latter [15] presents an improved gate driver for the SCRs as opposed to [14], the system incorporates

several sub-systems, adding complexity to the neutral section control system. High-voltage semiconductor systems are an advantage in size compared to mechanical systems; however, they are expensive to procure. The high cost of high-voltage semiconductor systems is due to the manufacturing process of designing and developing semiconductor components.

Delgado et al. [16] support the use of SCR (Thyristor) with a study focused on electrical, thermal and mechanical design. A high voltage was injected to test whether the SCRs could block the maximum supply voltage on the electrical design. The thermal design looked at three modes of operation, conduction, blocking mode one and blocking mode two, with the temperature measured from the heatsinks. The mechanical design aspect was the design of a modular stack unit where the SCRs were mounted. The authors showed that mechanical switches were robust but suffered from long switching delays. They also agreed with Ran et al. [14] and Xiong et al. [15] that the delays cause loss of power and transients, reducing lifespan. The modular design proposed by these authors presents an advantage when an SCR fails; replacing them can be easily compared to the traditional mechanical installation.

The ground-switching scheme is among three internationally used switching schemes in a neutral section [4]. A pole-switching scheme is another but less common than the latter. The pole-switching scheme is similar to the ground-switching scheme, except breakers are installed on a pole. In [4], while they describe the pole-switching scheme, the authors do not discuss an in-depth implementation of such a scheme. Furthermore, literature in [12-14, 16] and several others do not discuss the implementation of pole schemes but only focus on ground and onboard-switching schemes.

The third scheme is an onboard-switching scheme installed in Transnet's neutral sections: the scheme comprises ground sensors and an onboard controller. The ground sensors range from magnets, tags and axle counters installed on the ground. The controller installed in the locomotive allows the VCBs to open and close during the detection by the ground sensors.

Several works employ an onboard-switching scheme in a neutral section to switch trains automatically.

Ning [12] designed a dedicated detector for automatically detecting and repairing onboard-switching systems on the locomotive. The GFX (while the true abbreviation is not defined in this literature, however in a web search it is defined as Graphics) system was designed with a Programming Logic Controller (PLC) on the locomotive for command execution and magnetic induction sensors for locating position. Several sub-systems made up the GFX system, such as instrument unit and test control; in total, there were eleven. In an electric locomotive, the internal space is limited; therefore, having several sub-systems add complexity and takes up most of the space. The advantage of this system is the detection of faults and dynamic and static testing of the onboard-switching system. The GFX system, however, does not address the problems defined in [4, 13] found with the onboard-switching scheme. While the GFX system is designed to detect and repair failures automatically, it adds complexity as it would clutter the locomotive as there is limited space.

In [12], much of the focus was on the automatic detection of faults and testing; subsequently, over-voltages and inrush currents were not effectively addressed. Sang et al. [17] developed a superconductor fault current limiter (SFCL) to address inrush current, which formed part of onboard-switching scheme problems. A resistor-type SFCL was simulated using a PSCAD/EMTDC software program where the authors simulated an inrush current on a modelled SFCL system. The latter only focused on one problem associated with onboard-switching schemes, adding a sub-system on the neutral section, which subsequently increases maintenance workload. The installation of components such as ground sensors on the line increased the risk of theft or vandalism; therefore, replacing such components with radio frequency systems reduced this risk.

Yi et al. [11] proposed an onboard-switching scheme that employed radio frequency to detect and operate the breakers. Pavement markers were installed on the ground and could

transmit a signal when a train is detected. An onboard receiver would receive these signals, be processed and subsequently open or close the breakers. The pavement marker's transmitter circuitry was composed of a wireless transmitter, a microcontroller for processing, and solar panels that provided power during the day and charged the battery to power the system at night. While the concept of introducing wireless communication would eliminate theft: as fewer cables would be installed onsite, the installation of solar panels and batteries also increases the risk of theft or vandalism. Solar panels are also known to be ineffective during winter seasons or cloudy days, which may put the system at risk. The system may charge the batteries less, resulting in the system not powering on due to a low voltage/low power that may affect proper operation.

Chen et al. [4] outlined the disadvantages of the three schemes mentioned above while arguing that the onboard-switching scheme is the preferred choice. The disadvantage of the onboard-switching scheme is the period without power as the train traverses through a neutral section. Magnets or sensors installed on the ground for locating locomotives can also be stolen, and installing many induction magnets is costly. Overcoming these challenges was to employ computer vision and image processing methods. The authors suggest that employing their model instead of those proposed in the literature will address most of the challenges in the three switching schemes.

The three most common neutral section switching schemes presented in the literature illustrated significant investment costs, maintenance, and failures. The latter also implies that these technologies are obsolete and do not align with the 4IR or Transnet 4.0 strategy. The introduction of a newer switching scheme such as the one proposed by Chen et al. [4], employing computer vision and image processing is preferred. The lack of literature on computer vision and image processing systems deployed in railway infrastructure motivates further research. Furthermore, deploying computer vision and image processing systems in the

railway presents a reduction in investment costs and maintenance. However, the segmentation and classification require accuracy to be higher to minimise failures. The high accuracy of the model would, therefore, be necessary during the system's deployment.

2.3 Image Pre-processing

This section is divided into two parts; the first gives an overview of the preferred image acquisition sensor, and the second describes different pre-processing methods employed in image processing. These pre-processing methods relate to common image noises and common filter types employed to remove noise artefacts from an image.

2.3.1 Image Acquisition

Mizuma et al. [18] proposed using an infrared Charged Coupled Device (CCD) camera system to monitor railway operations through computer vision. They suggested that infrared CCD through computer vision could take several images of train operations without many altercations. While the authors suggest that a CCD camera occupies minimum space, the motivation for using a CCD camera as opposed to Complementary Metal Oxide Semiconductor (CMOS) is not substantially supported. The difference between CCD and CMOS is that the CCD pixel structure is more complex [19]: the architecture of the CCD pixel structure requires a photodiode and vertical shift register for each pixel. This limitation over CMOS prevents a CCD camera from achieving smaller pixels with higher performance.

Gonzalez and Woods [20] describe three image sensors for image acquisition: single imaging, line sensors and array sensors. The predominant sensor employed in a digital camera is the CCD array [20]. In [4], the authors described the use of industrial CCD and a high-speed Digital Signal Processor (DSP) as the preferred choice in image acquisition. They used the CCD to acquire video images while the DSP was employed to process these images in real-time and recognise the markers through image matching. The authors suggest that the industrial

VC2038 CCD camera has a high image-acquiring rate. Furthermore, the implementation of DSP will reduce the computational cost since video images will be processed faster.

Suzuki [19] conducted a study on the challenges of image-sensor development. The study focused on three elements, exceeding filament quality, exceeding the human vision and future developments. The author argues that CCD is limited to its pixel structure as each pixel requires a photodiode and vertical shift, making it complex and not achieving smaller pixels with higher performance. In exceeding human vision, CMOS cameras provided less power consumption and system integration. The high speed, high functionality and less complexity in their use were one of the benefits of CMOS digital cameras. According to Suzuki, CMOS suffered from increased noise and reduced sensitivity over CCD cameras. In image processing, images with higher quality are those considered with less or no noise; therefore, increased noise in CMOS cameras would be a limitation over CCD.

Li et al. [21] argued that CCD cameras lack colour depth when portraying natural objects. They suggest evaluating image quality by measuring the modulation transfer function when selecting a camera. According to Mehta et al. [22], a CMOS camera is the cheapest compared to a CCD camera. They concluded that CMOS cameras have low image quality and fill factor while CCD cameras have improved performance.

The literature described two types of camera sensors used in computer vision, a CMOS and the prominent CCD sensor. While the authors shared their views and results on which camera sensor was best preferred, none gave an in-depth discussion on the type of noises found in images after the acquisition, as well as standard filters used for denoising these images.

2.3.2 Image Noises and Filter Types

In an ideal computer vision system, acquired images would reflect the actual object captured without invariants. However, such invariants occur due to environmental conditions, and the sensor noise is due to electronic circuitry. These invariants or abnormalities in image

processing are called image noise. Image pre-processing is therefore required to remove these noises. Firstly, it is necessary to understand common noises found in images to know which filters are best for removing each noise. Several common noises are found in images, such as Gaussian, salt and pepper, Speckle and Poisson noise [23-25]. These noises are also of relevance in the proposed research:

- **Gaussian noise:** is governed by the probability theory, where the noise has a Gaussian distribution. The sensor causes the noise due to poor illumination, increase in temperature and transmission [23-25].
- **Salt and pepper:** also called impulse valued noise. Data transmission causes the noise resulting in some of its pixel values being corrupt [23-25].
- **Speckle noise:** a multiplicative noise of unwanted random signals with a probability density function that follows a gamma distribution. This noise is caused either by acoustic, laser or radar. Other common noises are Poisson-Gaussian, and Poisson also found in digital images. The latter arises from x-rays, gamma rays and visible light and follows a Poisson distribution. The former is Poisson-Gaussian noise that arises from Magnetic Resonance Imaging (MRI) [23-25].

Boyat and Joshi [23] mathematically presented models of several standard image noises. The mathematical representation gave an in-depth understanding and behaviour of each image noise. This understanding and behaviour would then allow for an appropriate filter to be employed in denoising images effectively. The authors [23], apart from spatial domain image noises, frequency domain image noises such as white noise, Brownian, periodic and quantisation were also mentioned. White noise is a random signal from the source with the same intensity at different frequencies. The authors defined Brownian noise as a random motion of particles suspended in a liquid. The authors further alluded that this random motion is Brownian motion. They described the periodic noise as being caused by electronic

interferences. The quantisation noise [23] occurs when a signal amplitude is quantised, such as in analogue-to-digital conversion. Though the white noise occurs randomly, images are rarely affected during acquisition. Brownian noise may affect images acquired underwater due to Brownian motion. The periodic noise may affect images during acquisition and storage; however, proper shielding and filtering of electronic circuits may prevent or reduce the effect. Digital images acquired from either CCD or CMOS are affected by quantisation noise due to analogue-to-digital conversion: high quality cameras with low quantisation error would prevent or reduce this type of noise. Hambal et al. [24] also defined several common noises found in images to be similar to those discussed in [23].

Kaur [26] gave an overview of common noise types found in images and how they were acquired. Similar to [23-25], the author described the acquired noises introduced in the transmission medium and during quantisation in detail. Furthermore, the author described four noise types, Poisson, Gaussian, speckle, and salt and pepper, as common image noises. The author also defined salt and pepper noise as an impulse noise where the corrupted pixels can either be a minimum which is zero or a maximum of 255 for an 8-bit image. The insufficient number of photons recognised by the sensor causes noise, and this noise is called Poisson noise. Gaussian and speckle noise were defined identically to those in [23-25]. Studying each noise type is critical in enabling correct filters to remove image noise. In salt and pepper noise, for example, a mean or median filter can be employed to remove this type of noise. Understanding the salt and pepper noise properties justifies using such filters to remove noise. According to Santur et al. [27], images acquired from a railway vehicle are subjected to motion blur. The authors alluded that rail vibrations cause motion blur which they described as a Gaussian-blur effect. The Gaussian blur described as a spatial filter is often used to blur an image to reduce noise; however, the filter can blur the image causing prominent edges to be removed. Removing these edges would result in additive noise; hence the authors describe the noise as a

Gaussian-blur effect. Considering this type of noise during pre-processing is essential: the acquired images dataset is from a railway environment.

Understanding different types of image noises allow for specific filters to be effectively employed. Numerous researchers have employed different filters in fields relevant to the proposed research. Different filters employed in denoising images are categorised into spatial and frequency domains. Furthermore, filters are classified based on their filtering techniques which are either linear or non-linear filters. Image noises falling in the frequency domain are less common in images acquired in railways than in spatial domains [27]. The literature, therefore, focuses on several spatial domain filters employed in related work. Images containing spatial frequencies are images that vary the greyscale level in space rather than time. There are two types, low and high spatial frequencies. Spatial image filtering removes noise and smooths as well as enhances an image. Successively, the pre-processing stage allows for the image to be processed by first converting it to a greyscale or binary image, subsequently applying a filter to remove noise. Due to the data transmission of the image from the camera to a computer and adverse conditions present in the atmosphere, an image can have random noise and rotations. There are several commonly used filters in image processing for denoising an image. These standard filters are listed below:

- **Gaussian blur:** is an example of a spatial filter used to smooth an image and, in turn, reduce noise by blurring effect. Comparatively, with the frequency domain filter, the Gaussian-blur filter is faster as it does not apply a Fourier transform which is computationally expensive [28].
- **Averaging filter:** also called a mean filter since it takes the average pixel value of the neighbouring pixel and sets each pixel to its average. The filter is a disadvantage as it does not preserve edges.

- **A median filter:** is an example of a non-linear filter; it takes the median pixel value of neighbouring pixels and replaces that pixel with the median value. This approach ensures the removal of noise while preserving edges.
- **Wiener filter:** this filter falls in the frequency domain and is better suited for removing Gaussian noise since it is a frequency response filter. The disadvantage of this type of filter is that the spectra properties and noise of the original image need to be known. In addition, the Wiener filter is slow since it works on the frequency domain; the computational cost or time increases because of this [24, 26, 28].

Qadri and Asif [7] proposed a different filtering method for an Automatic Number Plate Recognition (ANPR) system. The located number plate was converted to a binary image, leaving some artefacts. They employed two filtering techniques to remove artefacts and noise: the first technique removed all white patches connected to any edges by converting them to black pixel values. The second approach used pixel count to remove the smallest regions. The proposed techniques' limitations would be best suited to remove noises such as salt and pepper or speckle noise. As described by [26], images with the Gaussian-blur effect suggest that denoising would not be effective with the techniques proposed in [7].

While the above literature focuses on filters employed in number plate recognition, other literature relevant to this research is the recognition of shaped characters. Wakabayashi et al. [29] described a suitable filter for denoising an image for the recognition of shaped characters. The authors applied a mean filter by convolving its kernel five times in an image. The filter takes the average pixel value of its neighbouring pixel and sets each pixel to its average, subsequently losing some edges.

Nguwi and Lim [30] alluded that the ANPR system implemented in over 150 cities in the United States has allowed for the monitoring of traffic light violations by employing image

processing and computer vision techniques. However, the increase in different number plates in other countries has added complexity to ANPR; for that reason, they have proposed a system that removes images with 20% noise by employing a median filter [30]. The images were first converted into greyscale to remove the hue and saturation formation while retaining the luminance. The enhancement of images was achieved by minimising luminance and reducing noise by applying a median filter. The author's research found that most models achieved accuracies above 90% compared to their proposed model which achieved 85%; however, these models were based on non-noisy images.

Vigneshwar and Kumar [31], similarly to [30], employed a median filter in a pothole detection and counting system. The median filter allowed for the removal of random noise and image smoothing while maintaining the pixel integrity of image regions and boundaries. The latter presents an advantage during pre-processing since the edges are preserved, while in [29], the method does not preserve edges. In detecting RoIs in an image by employing edge detection, the fewer discontinuous pixel edges, the better the detection. Furthermore, reducing the computational cost: more discontinuous pixel edges would require additional pre-processing methods such as morphological operations to connect the edges.

Kaur and Singh [32] presented a study of various filtering techniques. The authors focused on a bilateral filter as they argued that it was different in terms of its parameters compared to other filters. However, they also conducted a literature review on the mean, median and Gaussian filters. The authors proposed quality analysis matrixes such as peak signal-to-noise ratio (PSNR), Mean Square Error (MSE), Mean Absolute Error (MAE) and time complexity to measure the performance of each filter. They [32] argued that the median filter, which is a non-linear filter, does not preserve edges as stated by [30, 31]. The authors suggested that bilateral filters were best suited to removing Gaussian noise as they took pixel and intensity values, denoising and smoothing an image. The bilateral filter preserved the edges during the denoising process [32]. In a noisy image where a filter requires exact denoising values, such as

the Wiener filter, the authors suggested that the quality analysis matrix can also be employed to determine the type of noise embedded in an image. The bilateral filter can remove several noises such as Gaussian noise and smooth and preserve image edges, which is an advantage to the proposed research. Therefore, the filter provides a three-in-one benefit of removing noise, smoothing, and preserving edges, subsequently improving computational time.

Desai et al. [33] illustrated several image-filtering techniques and algorithms. They argued in support of [30, 31] that the median filter was the most effective non-linear filter which removed noise while preserving edges. The author's findings, therefore, disagree with those of [32] regarding the median filter. They further suggest that the median filter is best suited to removing salt and pepper noise: this supports the argument deduced from [26]. However, the authors agree with Kaur and Singh [32] that bilateral filters yield better results when removing noise.

The division of image pre-processing into three sub-sections, image acquisition, image noises, and image filters, gave an insight into the stages of image pre-processing. The current industrial sensors employed in image acquisition are the CCD and CMOS. The literature demonstrated that each has its advantages and disadvantages: while the CCD lacked simplicity, the CMOS achieved smaller pixel size, increasing the resolution quality. Furthermore, the CMOS provided low power consumption since it was the least complex; however, CCD was still the preferred choice. In [19], the author suggested that CCD cameras were better than CMOS since the latter suffered from increased noise and reduced sensitivity. While CMOS is reported to be cheaper, Mehta et al. [21] also concluded that CCDs are the preferred choice since CMOS have low image quality and fill factor. Concluding the discussion of camera sensors: common noise types were reviewed based on several pieces of literature relevant to the proposed research. The literature revealed that the most common noises to be expected in images acquired in the railway environment are Gaussian noise, quantisation noise, and salt

and pepper [23-25, 27]. The knowledge of these noise types motivates the study of filters employed to denoise images. The most common spatial domain filters, to name a few, are the Gaussian and median filters.

An example is a Wiener filter employed to remove Gaussian noise in the frequency domain. In [32, 33], the authors presented a bilateral filter promising huge benefits over other filters. The filter was able to remove noise, smooth as well as preserve edges: better computational performance presented by this three-in-one approach. Improved computational performance of a computer vision model or system would allow for real-time applications.

2.4 Segmentation

Gonzalez and Woods [20] defined segmentation as subdividing an image into its constituent RoIs. They suggested segmenting nontrivial images in image processing was the most challenging process. Therefore, considerable care required an improved segmentation accuracy to ensure that the RoIs were successfully classified. Several techniques are used to segment RoI: intensity, discontinuity, similarity, clustering, and graph-based segmentation. An example of an intensity-based technique is the thresholding segmentation method: intensity levels of pixels are set to zeros (black pixels) or one (white pixels) in a binary image based on a threshold pixel value. The implementation of threshold value can be local or global; a global approach sets a constant threshold value to distinguish objects from the background. Local thresholding subdivides an image into sub-images, and each sub-image has its threshold value. Discontinuity segmentation methods are based on detecting abrupt intensity changes in the pixel's boundaries. Edge segmentation methods are an example: Canny edge, Sobel, Roberts, Prewitt, and Laplacian of Gaussian are some common operators employed in edge detection. A region-growing method is a similarity-based segmentation that groups pixels or sub-regions

based on predefined criteria. The seed points are set based on the predefined criteria; regions are grown by appending nearby pixels with each seed.

Clustering segmentation methods such as k-means are one of the utilised methods [34]. The graph-cut segmentation method organises the image elements into a mathematical graph structure. The structure comprises a set of vertices corresponding to the image elements and edge nodes connecting specific pairs of neighbouring vertices. Literature is scarce [4] that employs computer vision in railway infrastructure for switching trains automatically. While the latter illustrates the need for more literature on railway technologies; however, relevant to the proposed research, several methods have been proposed. Segmentation methods applied to terrestrial, aerial, and aquatic acquired images have been employed in numerous research studies. These studies range from traffic signs, number plates, human object detection, animals, underwater objects, and numerous objects. Though, these studies may seem different from the proposed research; however, they are of relevance, and they give insight to different methods employed which subsequently, can be employed in this research. Therefore, this section mainly focuses on literature which may be of relevance to the proposed research. Furthermore, the literature should give an insight into which segmentation method would be best suited for the proposed research.

The fundamentals of these methods, edge detection and thresholding, are demonstrated in [20]. Image features that form part of discontinuity segmentation are edges, lines, and isolated points. Edge detection segments the RoI by detecting edges, line detection segment the RoI by detecting lines, and isolated point segments by detecting points. The line and isolated points methods are the least common compared to edge detection in segmentation [20]. The edge detection methods are classified mathematically into first and second-order derivative operators. The first-order derivative operators are Sobel, Roberts, and Prewitt; subsequently, Canny and Laplacian of Gaussian are second-order derivative operators. The second-order derivative governs the line and isolated points methods, and edge detection is the first-order

derivative. According to [20], first-order derivative operators produce thicker edges while the second derivative operators respond more strongly to thin lines, points and image noise. While the first-order and second-derivative operators could be utilized in an image with thinner edges, the former would be a better choice as less computation time will be required.

Furthermore, the second-order derivative operators having a stronger response to noise may increase the noise levels. Intensity-based segmentation method described by [20] was thresholding. This method segments RoIs by using local or global intensity levels. The thresholding method separates images into sub-images, and a local threshold value is chosen for each sub-image. In global thresholding, a global intensity value between 0 – 255 (for 8-bit images) is used to separate the foreground from the background. Otsu's method is presented as one of the optimum global thresholding methods since it maximises the between-class variance, a measure defined in statistical discriminant analysis. The authors alluded that illumination changes affect an image's histogram; subsequently, segmentation inaccuracies increase with the thresholding method. The authors suggested that the simplicity and computational speed of thresholding are favourable in image segmentation applications.

Liu et al. [35] proposed real-time object detection in dynamic scenes as they suggested that it is harder to detect these objects than those from stationary scenes. The authors further described that dynamic scenes' boundaries between the foreground and background were ambiguous. They also argued that the existing feature extraction methods must organise to generate the target object. The authors described these existing feature extraction methods as time costly. They proposed a Feature Line Section (FLS) in both vehicle and lane markings, achieving a segmentation accuracy of 96.03% for vehicles and 94.7% for lanes [35]. The limitation of the FLS model was the dependency on three constraints called FLS definitions. These definitions were experience, computational cost and scanning the image's rows and columns (x, y). The experience required human input to tune the model, therefore, not allowing the model to be adaptive—the computational cost caused by deploying FLS for feature

extraction affected real-time operation. The latter is due to the FLS scanning for the entire image pixels checking for potential features.

Khan and Ravi [36] conducted a comparative study on different types of segmentation techniques and methods. They reviewed intensity, discontinuity, similarity, clustering, graph, and Hybrid based segmentation methods. Categorically, thresholding, edge detection, region growing, k-means, graph-cut and watershed are some of the methods of each segmentation technique. The authors presented findings on each mathematical model and discussed the findings. The thresholding-based methods were found to be computationally inexpensive but were highly sensitive to noise and also susceptible to under and over-segmentation. In the edge detection methods, it was argued, viz the Canny edge operator being a second-order derivative operator gave reliable results. The authors alluded that with varying images, no single operator fits all; increasing the operators' size will increase the computational complexity. The region-growing method is said to have given accurate results compared to other methods only when the correct seed value is selected. Firstly choosing the number of clusters k is the main drawback of the k-means clustering method. The hybrid-based segmentation methods are those methods which have two or more segmentation methods, such as the watershed. The watershed has morphology and edge detection concepts in partitioning images into homogenous regions. The authors suggested that the watershed method suffered from over-segmentation, subsequently increasing computation because of the post-processing required.

D. Kaur and Y. Kaur [37] reviewed various image segmentation techniques and examined each method's benefits. The authors categorised image segmentation into two types local and global. In local segmentation, specific parts or regions of an image are segmented, while global consists of a large area with pixels or the whole image being segmented. They compared several methods such as thresholding, edge detection, region, clustering, partial differential equation and artificial neural network [38] segmentation. The authors presented their findings theoretically; however, they alluded that the results were derived from the empirical data, but

the equations were omitted for simplicity. They concluded that while some performed better, each method was suitable for a particular image type and application. While the authors do not demonstrate any quantitative or qualitative method employed to conclude on their findings, they provided an understanding of which segmentation method to apply in different applications. In the thresholding method for instance, they suggested that it is best used in images with lighter objects than background.

Wang and Liu [39] proposed segmenting traffic signs in a natural scene for intelligent transportation systems. The authors employed two segmentation techniques described in [37]: colour thresholding using colour and edge detection utilising shape features (contours). The initial RGB image was normalised to address the issue of lighting variances. The normalised RGB image was further improved using the achromatic model equation, initially segmenting the traffic sign. They then converted the image into greyscale and non-target pixels were set to zero. The greyscale image was then converted into a binary image using the highest peak in the histogram to set a threshold value. Contour extraction and chain code were employed to segment RoI shapes. The model achieved a 93.2% segmentation accuracy with a 2% false alarm. The edge contour extraction such as Sobel, Roberts, and Canny operators, while they provide fast computational speed, was unable to extract contours due to noise [39] effectively. Furthermore, contour extraction such as the commonly used active contour snake, cannot move towards objects far away. However, the authors presented a chain code algorithm that tracks and stores the extracted contours information. This approach efficiently represents binary images with shapes, improving processing time.

Balamurugan et al. [40] proposed an automatic number plate recognition (ANPR) system by enhancing the visual quality of a low-resolution image. After that, the enhanced image would be segmented for the number plate characters to be recognised. They employed a Super-Resolution (SR) technique that converts low-resolution images to high-resolution images. The authors achieved this by interpolation, regression, and post-processing. A Laplacian of

Gaussian (LoG) is applied to segment the enhanced number plate, and a bounding box is used to crop each character. The bounding box utilises the coordinates of the segmented image to crop each character for recognition. The author's research presented a quantitative approach: there was no empirical data to validate the segmentation accuracy. However, it is worth noting from the human visual perception that the SR technique enhanced the image character's resolution to be readable. This technique enhances low-resolution images, allowing for better detection and segmentation.

Like [39], Nguwi and Lim [29] proposed an object detection method for ANPR with images with 20% additive noise. The authors presented several segmentation methods: edge detection, transformation (Hough), colour segmentation and morphology. Images were first convolved with a median filter to remove noise. Then a morphological transformation was employed to fill holes in the number plate characters—these holes allowed the characters to be more prominent than the background. The prominent characters enabled the segmentation of each number plate character. The authors suggested that morphology is efficient in enhancing segmentation in number plates. They achieved an accuracy of 85%, and under suitable conditions, it was above 95%. The enhanced characters in the number plate allow for quicker segmentation as they are now prominent. Edge detection makes a better choice in images with prominent objects in colour than their background. Therefore, the edge detection method would require additional methods to segment objects in complex backgrounds effectively. Nguwi and Lim [30] suggested that Hough Transform is best used to detect lines and curves.

Similarly, an ANPR model was also proposed by Islam et al. [41]. The authors used character segments to separate each character from the number plate. A threshold was first applied on the number plate to remove any artefacts and enhance the characters. The characters were then separated using the vertical blanks (black pixels: area in-between the number plate characters without characters). These areas are vertical; hence they are called vertical blanks: blanks allow for segmentation to be fast and effective. The authors

compared similar literature that achieved 89.2%; however, their model achieved 92%. The authors' characteristics of the number plate are that segmentation can be performed effectively and faster by adopting specific features such as vertical blanks.

Soni et al. [42] employed two methods to detect objects in images. They proposed the segmentation of geospatial objects found in the ocean, such as sank ships, aeroplanes, and valuable debris. The authors aimed to minimise human efforts when shortlisting large volumes of images containing these objects. They employed colour and shape-based analysis in detecting these objects from the acquired images. The images were first segmented by colour thresholding each image from an HSV colour space. These images were further segmented by employing shape based. This second segmentation stage allowed each object to be segmented into its shape class through template matching. The authors used a template database where they iterated each image segmented against the database for shape-based segmentation. Using the colour and shape detection methods, the authors recorded an accuracy of 95%, which was high compared to [35, 38]. As described by [20], thresholding has a faster computational speed and is simple: this would be an ideal method for real-time applications. The limitation of this method is that it is affected by illumination changes that affect the image's histogram. Furthermore, images with background artefacts having similar colours and intensity levels as those in RoIs may lead to over or under-segmentation. The computation performance is also affected since each segmented region needs to be iterated through templates stored in a database for shape segmentation.

Vigneshwar and Kumar [31] described potholes as the primary cause of accidents; hence an identification and classification system employing image processing was proposed. The authors evaluated several techniques for segmenting potholes. Threshold, edge, k-means, fuzzy C-means, and manual segmentation were evaluated. The segmented RoIs obtained from each method are converted into white blobs, and segmentation accuracy and performance are evaluated. Accordingly, the threshold achieved 80.61% at 2.0467s, edge detection was 90.2%

at 0.4950s, k-means got 84.5% at 0.2766s and fuzzy C-means achieved 82.5% at 1.1028s. The authors concluded that k-means was the preferred choice if computational speed was required, while edge detection was a better choice if segmentation accuracy was required.

Aditya et al. [34] compared k-means, region growing, mean shift and watershed segmentation methods. The performance of each method was determined using quality metric (QM) parameters in RGB images. The images are acquired from entertainment, sports, and natural scenery video frames. The QM looks at five parameters grey level energy (GLE), Discrete Entropy (DE), Normalised mutual information (NMI), Information Redundancy (IR) and Mean Squared Error (MSE). The images per frame were converted from RGB to HSI colour space, where a comparison of the four segmentation methods was conducted. The result obtained by the authors was that region growing segmentation had better segmentation than the others, followed by watershed. The region-growing method is computationally expensive due to its seed points approach. While the watershed may present an advantage in preserving boundaries, it segments better with images having superpixels. The k-means compared to the mean shift requires some model assumption, such as the number of clusters (k); for example, the k parameter needs to be predefined manually. The mean shift suffers from expensive calculations: this is influenced by the general operations or time complexity $O(N^2)$; O is the big O notation, and N is the number of data points. In computer science, the big O notation approximates an algorithm's running time or space requirement as the input increases.

Huang and Hou [43] proposed an intelligent transportation system that detects traffic speed signs. They employed the Gaussian colour model to allow the segmentation of RoIs in each image. The first stage was converting RGB images to YCbCr colour space. The authors claimed that the chrominance components (Cb and Cr) were independent of luminance (Y). The chrominance components were used to model the distribution of traffic sign colours, and the Gaussian colour model was then employed to segment the RoIs. The authors then used an Otsu thresholding method to segment the RoIs further. Regions with artefacts were filtered out

by applying morphological operations while enhancing the segmented RoI. The authors tested a range of speed signs and got an average of 43.73% segmentation accuracy. The lowest speed sign with an accuracy of 0% was 30km/h, and the highest was 100km/h with an accuracy of 66.67%. The separation of chrominance components from luminance in the YCbCr colour space reduces the effects of light variations. The Gaussian colour model also prevents extra artefacts from occurring in low-resolution images: spatial and colour information allow for more features to be obtained. Furthermore, the Gaussian colour model proposed by the authors follows a Bayesian rule that reduces non-traffic sign pixels from being segmented.

Panahi and Gholampour [44] proposed a method different from the methods proposed by [40, 41]. The authors implemented an ANPR system that detected dirty number plates for high-speed applications. They presented a realistic approach in an ANPR system as most vehicles accumulate dirt on the number plate. They employed adaptive thresholding to segment the characters on the number plate. A window of sizes m and n slid through the image, and the local mean of pixels intensity was used to calculate a threshold value in each window. The system was installed in several locations and tested with a large dataset within a year. The authors recorded an overall accuracy of 91.4% on the dirty number plates. The approach proposed by the authors presented a computational problem: pre-calculations of the size of the window would be required since images may have different sizes. Furthermore, iterating the window through the entire image and calculating the local mean for each window is computationally expensive with larger images.

Contrary to this research, Raghunandan et al. [44] presented different methods for human detection in video surveillance. While detecting faces and other body parts is irrelevant to the proposed research, they proposed a target object method that may be applicable. The authors defined foreground objects as those that change from frame to frame, while background objects are those stationary objects. They employed a background subtraction method to detect an object: subtracting the current video frame from the previous frame giving the foreground

object. Accordingly, they alluded that the model could detect 95% of foreground OoI. The implementation of this method for the proposed research can also be adopted since the detection of the markers would occur in dynamic scenes. Chiu et al. [45] argued that the background subtraction method, while best suited for moving objects or dynamic scenes, suffers from several issues such as colour disturbances, dynamic background, camera vibrations and prolonged background masking.

Luo et al. [46] proposed Maximally Stable External Regions (MSERs) for detecting and segmenting traffic signs. They claimed that due to a large portion of uniform areas within the traffic signs, MSER could easily detect RoIs. A combination of different colour channels, Grey, RGB and normalised RGB employing MSER were evaluated. The different channel combinations yielded different results ranging from 65.69% with the grey channel to 98.36% with the combination of grey, RGB and normalised RGB. The authors employed F1-measure or F-score (where F can also be denoted as $F1$) to measure the accuracy of segmentation. The benefit of using an F1-measure is that it enables an effective empirical validation of a segmentation method. Joshy and Anishin [47] also evaluated MSER segmentation based on text detection by reviewing several pieces of literature. They could qualitatively outline the advantages and disadvantages of MSER segmentation based on several pieces of literature. They suggested that MSER require low computation cost but suffers from low performance with blurry images. The authors claimed that while it was one of the best region detectors, it required tuning for varying text styles. They alluded that high contrast images caused low performance, and MSER was sensitive to character sizes. The foreseeable challenge in employing MSER in the proposed model would be an increase in computation cost as more false regions would be detected. The markers are installed onsite in an environment with illumination changes, shadows, reflection, and varying focal distances; subsequently, these factors drastically affect the performance.

Li et al. [48] proposed an improved grab-cut image segmentation method based on image region. The author's segmentation method mainly focused on animals, humans, and other objects of interest. They first employed graph-based image segmentation to obtain each pixel's label and then utilised the grab-cut method to segment the image further. The authors' segmentation accuracy rate from the improved grab-cut method was 86.81%. Compared to the former, the results obtained from the latter can be considered negligible as the performance increased by 0.0259%. Additionally, the improved grab-cut method computational performance is slower (4750ms) compared to the standard method (3723ms). Arguably the grab-cut algorithm suffers from computational inefficiencies due to the iterations it does and mostly when the background is complex. The obtained results also support the computational inefficiencies: the improved algorithm slightly improved while the segmentation speed slowed down. Furthermore, the grab-cut algorithm generally requires k-means clustering, as noted in [34]. Arguably the cluster number k must be manually selected, which then affects the accuracy of the segmentation.

Othman et al. [49] implemented a real-time object detection model for detecting and measuring its size. The hardware and software implementations of the model used a Raspberry Pi 3 computer, Raspberry Pi camera and OpenCV software library. They employed a Canny edge detection to segment objects. In addition, morphological operators such as dilate and erode were employed to close gaps between edges. The model was able to achieve 98% accuracy. The mathematical models representing edge detection promise moderate computational speeds, second to thresholding [20] for real-time applications. Furthermore, edge detection methods efficiently segment objects in images where the background is less cluttered or complex. The latter is supported in [49]: images had simple backgrounds, were in a static scene and had high-contrast objects.

Deshmukh and Moh [50] developed an object detection algorithm that automates solar panel layout by detecting fine objects. The authors employed a fusion of Convolutional Neural

Network (CNN) and Canny edge detection. The CNN employed was from the TensorFlow object detection Application Programming Interface [51]. The Canny edge method showed better results in boundary detection of each obstacle; compared with ground truth, there was less than 25%-pixel count variation as opposed to the other edge detection methods. The evaluation of object detection ranged from 79% to 99% accuracy. They presented two evaluation methods, edge pixel count and variance in edge length, subsequently enabling the selection of suitable edge detection methods to be easy. The CNN can output high detection accuracy; however, CNN neural networks require a large amount of computational power and a large dataset to obtain higher accuracy.

Mane and Mangal [52] proposed an object detection and tracking algorithm for dynamic objects. Similarly, they employed CNN to detect OoI; they achieved an accuracy of 90.88%. The latter was, therefore, lower than the former [50]; the effectiveness of fusing CNN with edge detection was effective. In a model where real-time detection is not that critical, CNN evidently can output higher detection rates; moreover, CNN is developed to augment human intelligence, which would be an advantage for detection.

Saini and Biswas [53] employed the Sobel edge operator to detect geological and biological objects. The authors focused on a method of enhancing images with noise caused by the fogginess of the water by stretching the contrast. They claimed that better edges or boundaries were obtained after image enhancement using a Sobel operator. The comparison was made between other operators such as Canny and Prewitt. The average accuracy achieved for an object with a Sobel was 57.81%, Canny (54.51%) and Prewitt (55%). Evidently, in [53], this further supports that edge detection methods are best suitable in ideal environments such as those presented in [49]. Furthermore, the model presented in [49] has an advantage in deploying it in real-time applications where suitable conditions are favourable. In addition, deploying to embedded systems is advantageous since edge detection methods require less computational power.

The process of subdividing an image into its constituent RoIs was defined as segmentation in [20]. In [37], they categorised segmentation into local and global segmentation. These segmentation categories presented several segmentation techniques employed in image processing. Intensity, discontinuity, similarity, clustering, graph, and hybrid-based segmentations are some techniques used in image processing. An overview of the different types of segmentation methods was presented along with their mathematical models [20, 36]. The presentation of these mathematical models would allow for a well-informed decision when selecting a segmentation method. While numerous researchers have researched the topic of segmentation, none was an ideal choice when employed in different applications. In an instance of thresholding, it would be suitable for real-time applications in images with less complex backgrounds and high contrast in OoI. However, the same method would not be suitable in images with complex background and illumination variations; perhaps neural networks would be best applicable. Application requirements differ from user to user, and this needs to be considered in developing embedded systems. An application may, for example, require that the segmentation algorithm have less time-complexity (computational cost) and space-complexity (memory space). Therefore, neural networks may not be a suitable choice even though they may be superior in segmentation performance.

Regarding the latter, the authors concluded that none supersedes another, as each would suit different applications. The proposed research presented a challenge in that less literature was available [6]; however, several pieces of literature of relevance were reviewed. The literature presented different segmentation methods in images acquired from traffic signs and number plates with numerous objects. Much of the research focused on methods employed in traffic signs and number plates as these were more relevant. For instance, images acquired from number plates varied from edge detection to colour detection. Other methods, such as background subtraction in dynamic scenes, were employed. While research explores the best

segmentation algorithm, accuracy largely depends on detecting the correct RoI. The evaluation of each method led to a conclusion based on the results obtained from the literature. Empirical validation, such as the F1-measure, a mathematical equation, was used to calculate the segmentation accuracy of each method. Segmentation is crucial for ensuring objects are appropriately classified in computer vision applications. The segmented objects require a classification method to classify each object for the object to be interpreted by a computer—several classification methods employed in computer vision range from simple, to complex, and computationally expensive. Similarly to segmentation, researchers have proposed several classification algorithms to achieve a high accuracy rate. The section to follow reviews some literature employing classification methods.

2.5 Feature Extraction and Classification

Computer vision allows computers to identify meaningful objects in images or video frames. The segmentation of objects alone does not give meaningful information for a computer to interpret; hence classification is required. Classification is the process of imitating the human ability to analyse and classify visual objects using a computer vision system. Different classifiers have been employed in various proposed models to try and achieve the best classification accuracy. Selecting a classifier that best suits a specific dataset is essential; the categorisation of classifiers can be defined by various categories such as type of learning or data distribution [54]. The type of learning category defines supervised or unsupervised classifiers, while the parametric and non-parametric are in the data distribution category. The Artificial Neural Network (ANN) is an example of a supervised classifier, and k-means clustering illustrates an unsupervised classifier. The Maximum-likelihood is a parametric classifier, while a Support Vector Machine (SVM) is an example of a non-parametric classifier. Supervised or unsupervised, classifiers require some form of features to make sense of the

object. According to [20], segmented images can be represented by boundary-following, chain codes, polygonal approximation, signatures, boundary segments and skeleton. These representations are said to facilitate the computation of descriptors.

Descriptors can be defined as feature vectors that represent specific objects. Salau and Jain [55] categorised features into general and domain-specific. Domain-specific features are irrelevant in the proposed model, mainly focusing on fingerprints and human facial features. General features are, therefore, features of relevance as they extract colour, texture, and shape: proposed markers have these features. Colour features can be extracted from the colour histogram or colour moments; textural features can be extracted either from spatial or spectral domain and shape features from a region or contour-based extraction [55]. Similarly, with the segmentation, less research has been conducted on the automatic switching of electric locomotives in the neutral section by employing classification techniques. Therefore, several pieces of literature that discussed various classification techniques and methods were reviewed in this section.

Kim et al. [56] proposed a comparative study of two classifiers employed in image processing for classification. The authors compared K-Nearest-Neighbour (K-NN) and SVM with the Caltech-4-Cropped dataset. They created a bag of features (BoFs) or a bag of words (BoWs) for the extracted features. The BoF's primary purpose was to store features into a numerical vector. They employed Scale-Invariant Feature Transform (SIFT) for extracting features. The k-means clustering algorithm was employed on these feature vectors, creating clusters. A histogram of the training data was then compared with the histogram of testing data on both K-NN and SVM classifiers. The author's results illustrated that SVM had a higher average classification accuracy of 91.9%, while K-NN was 78.03%. The SVM outperformed K-NN by more than 13.89%; however, they argued that the variance would be reduced to 4% with cars removed from the dataset. The authors suggested that K-NN failed to distinguish cars as opposed to SVM. The K-NN employs Euclidean distance to classify objects based on their

closest feature space. On the other hand, the SVM employs a hyperplane to divide data points in space. The K-NN classification in small feature subsets leads to errors caused by using all the features in computing for similarities. While the SVM may be limited to speed, it performs better with fewer features or datasets with many attributes. The SVM classifier showed better performance also in [57].

Barstuğan and Ceylan [57] also conducted a comparison study based on two ensemble classifiers using biomedical datasets. Ensemble classifiers are defined as a system of classifiers where the evaluation of decisions on the same data is taken by more than one classifier [57]. Concisely, an ensemble classifier is more than one classifier which processes the same data. A comparison of the AdaBoost-Decision tree and AdaBoost-SVM ensemble classifiers was carried out. The authors recorded an average of 76.47% on the AdaBoost-Decision tree and 81.43% on the AdaBoost-SVM classifiers. They calculated the accuracy of each ensemble classifier by employing a 10-fold cross-validation method. The ensemble classifiers could train weak classifiers with low and high performance. The advantage of this approach is that it increases the classification rate since several classifiers are used. However, this approach presents a computational problem: more than one classifier would increase the classification time.

Ju et al. [58] proposed an enhanced object detection method in the automobile industry for dynamic scenes. The method would intelligently detect moving objects such as pedestrians and traffic signs. They proposed a robust, enhanced AdaBoost algorithm combined with a local feature method, a Histogram of Oriented Gradients (HoG) to detect the objects. They suggested that the HoG method was a better choice as a local feature extractor than other methods. The authors compared their enhanced AdaBoost with the other two proposed methods in similar literature by obtaining the accuracy and false classifications of each method. The first method achieved 90% accuracy with 59.6% false classifications, and the second method an average of 63.35% with 4.1%. Their method achieved an accuracy of 89.6% having 4.7% of false

classifications. The first method had a high performance with a high rate of false classifications, while the second method had lower false classifications and performance. The proposed method outperformed the other two since false classifications were also lower than the first method, and performance was higher than the second method. An AdaBoost algorithm, in general, is well suited for improving model predictions for learning algorithms. Subsequently, these progressive improvements to other learning algorithms would increase the computation cost. The former and the latter are also suggested in an argument presented in [57]. Furthermore, as a progressive learning algorithm, it would require high-quality features for the weak classifiers and is sensitive to outliers and noises.

Nath et al. [54] presented a research survey on various image classification methods and techniques. The research survey was based on a qualitative approach where the authors looked at six factors which suggested the selection of a suitable classifier. The main factors of relevance were the nature of the training samples used and the various parameters used in the data. The latter discussed parametric and non-parametric classifiers, while the other literature looked at supervised and unsupervised classifications. Various parameters in the parametric classifier commonly involve a mean vector or covariance matrix; a maximum likelihood is an example. Non-parametric classifiers do not use statistical parameters to determine the separation of classes; SVM and ANN are examples of non-parametric classifiers. The authors also mentioned several classifiers such as fuzzy-set, decision tree and ISODATA as some of the classifiers which can be employed based on the six factors. They reviewed SVM and Hidden Markov Model (HMM) classifiers: they claimed that SVM allowed for feature classes to be linearly separable since the data was mapped to a high-dimensional feature space. In the HMM, images were subdivided into block sizes; this caused surrounding regions to be lost. While the survey was based on a qualitative approach, the authors could give an insight into factors that can be employed when selecting a suitable classifier. They illustrated some classification methods and their applications in digital images.

Han et al. [59] proposed a robust recognition system for traffic signs that extracts features and classifies segmented images. They employed Speed Up Robust Feature (SURF) for feature extraction and the K-NN method for classification. The authors achieved a 97.54% accuracy compared to [56]; this was an improvement. The implementation of the K-NN classification shares similar effects as those suggested in [56]. However, employing SURF instead of SIFT improved the performance of the K-NN classifier.

Romdhane et al. [60] proposed an improved traffic sign recognition and tracking method for a driver assistance system. They employed the HoG method for extracting features into a feature vector. This feature vector was then used to train an SVM classifier. The SVM classifier was employed to classify segmented traffic signs. The authors achieved an accuracy of 91.32% with an F1-measure evaluation. The HoG method deals with local changes such as appearance and position: local objects are well-defined by the distribution of local gradients' intensity or edges. Therefore, calculating the orientation histogram of edge intensity in local regions makes the HoG method a suitable feature extractor. Classification of traffic signs employing an SVM classifier with HoG features presented similar results as those obtained in [56]. The advantage of the proposed method [60] is the fast computational speed of extracting features using HoG instead of SIFT [56].

Han et al. [61] researched a robust geospatial object detection system in high spatial resolution images. They employed a Region Proposal Network (RPN) for extracting features and faster Region Based-Convolutional Neural Networks (R-CNN) for detection and classification. The authors argued that other feature extractors like SIFT and HoG heavily relied on samples labelled by humans. They further argued that employing CNN can extract features and detect objects simultaneously; however, they alluded that it was computationally expensive and required a large dataset. The R-CNN object detection measured an average accuracy of 70.2% with a computational speed of 0.04s, and the CNN algorithm got 59.7% accuracy at a 5.24s computational speed.

Koskowich et al. [62] proposed a method of object detection and tracking aerial images of car parking. A YOLO (You Only Look Once) framework was employed to detect and classify objects. The efficacy of traditional tracking algorithms like the Track-Learn-Detect (TLD), Kernelized Correlation Filter (KCF) and Multiple Instance Learning (MIL) were also evaluated against the proposed Lucas-Kanade optic flow. The YOLO was trained with PASCAL VOC 2007 & 2012 data; it included people, cars, and motorbikes. The authors used an AXIS Q6044-E PTZ dome network camera: the output of the video file was 1280x720 at 20 frames per second. They [62] argued that YOLO was flexible with input size and speed compared to the R-CNN family. A detection accuracy of 99.9% was obtained with the YOLO. The efficacy of the optic-flow tracking algorithm was 85%, higher than the TLD (65%), KCF (58%) and MIL (65%). The YOLO algorithm, however, trades some accuracy over performance: it is suitable for real-time applications since it is fast but fails to accurately detect small objects compared to other neural networks.

Babu et al. [38] presented an overview of image classification methods. The authors reviewed three types, SVM, ANN and Decision Tree (DT). They suggested that SVM was more capable than the other classifiers but performed slowly. They argued that ANN was robust to noisy training datasets but had a high computational cost. In the DT, they suggested it required little hard work from the user, but it produced a high classification error rate.

Lai et al. [63] proposed a deep feature learning approach to recognise and classify traffic signs. They employed CNN for feature extraction and SVM for classification. The images were first converted into YCbCr colour space before extracting features. The conversion from YCbCr colour space showed better performance than RGB and greyscale. The evaluation was conducted by converting each dataset into one of the three colour spaces, and training error was obtained. The training error for the greyscale dataset was 0.138, RGB recorded 0.105, and YCbCr got 0.073. The error values were obtained from the ratio of the wrong samples in each batch divided by the batch size. The training error of the YCbCr noticeably was less when

compared to the other colour spaces; this proved it was a better colour space. The authors achieved a 98.6% accuracy rate when combining CNN and SVM. Implementing a CNN was that it could perform all the processes such as detection, segmentation, and classification. Furthermore, the conversion to a YCbCr colour space demonstrated better performance: as the authors suggested, colour distribution was not uniform in the RGB images. Most of the literature [38, 56, 60] that employed an SVM classifier demonstrated better classification performances.

In [41], the authors employed a template-matching technique in an automatic number plate recognition system. The segmented number plate characters were compared with a template database, and a correlation coefficient was employed to compute the classification accuracy. The overall accuracy achieved by this approach was 92%; however, the classification technique requires each template to be iterated through the entire segmented image resulting in more computation cost. Furthermore, template matching is affected by scale and rotation changes.

Serna and Ruichek [64] employed CNN to classify European traffic signs. They argued that CNN outperforms other classification methods; therefore, the authors evaluated five CNNs (LeNet-5, IDSIA, URV, CNN-asymmetric kernels and CNN 8-layers). They trained these models using the German Traffic Sign Recognition Benchmark (GTSRB) and European datasets. The LeNet-5 was the lowest, with an accuracy rate of 89.8%, while CNN-asymmetric kernels got 98.48%, followed by CNN with 8 layers at 97.88%. However, the LeNet-5 was faster than all the networks at 0.0067ms of computation time. While the CNN-asymmetric kernels outperformed the rest, they took 0.39ms, and CNN 8-layers took 0.15ms. The LeNet-5 network presents an advantage in real-time applications due to its low computational cost.

Nevertheless, these differences can be ignored mainly because the proposed research requires a 40ms switching time for electric locomotives to switch before passing a neutral section. Since the CNN-asymmetric kernels and 8-layer CNN demonstrated high accuracy, they would be a better choice even though they were slightly slower. The disadvantage of CNN

networks is that a large amount of training data is required to ensure that RoIs or OoIs are accurately classified [64]. Furthermore, the CNNs take time to train and require more computational power. Serna and Ruichek [64] used an NVIDIA GeForce GPU, 11GB of memory, an i7 core processor and 32GB of RAM.

Standard feature extraction methods and classifiers possibly adopted in the proposed research model have been reviewed in this literature. Though numerous techniques have been researched, it was essential to limit the scope of this research to relevant literature. The latter, therefore, imposed constraints in extraction methods and classifiers primarily used in signage detection. Nevertheless, these methods and classifiers could contribute to deciding which method or classifier was better suited. Furthermore, in some literature [49, 64], the computational power, complexity, memory demand and scalability of these classifiers were demonstrated through the required hardware, which also gave insight into the limitations of each approach.

2.6 Summary

The literature review chapter addresses two main objectives: reviewing related work, and techniques. Reviewing related work was to answer whether the proposed research had been done and which techniques were implemented. The literature would also reveal gaps that enable an informed approach to selecting the best method. There were several conventional switching schemes implemented and were based on mechanical components. Such schemes presented several challenges; the mechanical components suffered from switching delays and high investment and maintenance costs. Furthermore, these conventional schemes have become obsolete and do not align with the fourth industrial revolution (4IR).

The technologies applied in the 4IR present new approaches that can be implemented to switch electric locomotives. Artificial Intelligence (AI) is one of the 4IR technologies, and

computer vision is a field of AI. Though literature is scarce on the proposed research, literature of relevance presented computer vision methods which could be adopted. These related works, which were relevant to the proposed research, such as traffic signs, number plates, geospatial objects and more, employing computer vision, were reviewed. As previously mentioned, reviewing techniques and methods in the related fields was to assert which techniques and methods would best be implemented for this research.

Image processing being broad, sections were divided into image pre-processing, segmentation, and classification. A sub-division of image pre-processing; image acquisition and types of image noises and filters were discussed. Understanding the camera sensor operation types and their effects was essential in ensuring a good quality of images was captured. Therefore, the effects of each sensor would allow for an optimal selection of which camera to implement in a computer vision system. The image acquisition illustrated two standard sensor types employed in capturing images, CCD and CMOS as well as the advantages and disadvantages of each.

Studying common noises found in digital images was crucial in deciding which image filtering techniques to employ when denoising an image. The commonly found image noises described in [23-25] provided mathematical models of how each noise originates. These mathematical models gave an in-depth understanding of how each noise behaved and the types of filters that could effectively remove each noise. Santur et al. [27] further gave an insight into the type of noises found in images acquired on the railway: this was important since the proposed research acquired images in a railway environment. Image filters vary from spatial domain and frequency domain; the mathematical model of each image noise depicts the domain it belongs to. Spatial filters such as median and Gaussian-blur filters were employed to remove common image noises. In [32, 33], the authors presented a bilateral filter that could remove noise, smooth an image and preserve edges, suggesting better computational performance.

Two types of segmentation techniques are presented in the literature, local and global. The local or instance segmentation detected each distinct RoI while the global or semantic segmented the entire image objects as one and the background as another. Colour detection, morphological transformation, edge detection and transformation (Laplacian of Gaussian and Hough) are some techniques used to detect objects. These techniques had several methods used in detecting objects, each with shortfalls. In the example of colour detection, a colour thresholding method would detect an object by thresholding the image based on the colour. The disadvantage of this method was that the threshold would also be applied to the foreground pixels, thereby removing RoI. The amount of RoIs removed depended on the global or adaptive thresholding pixel value, which caused under or over-segmentation. The changes in illumination also affected the thresholding method, but it was faster than other methods. Thresholding causes incorrect segmentations, specifically when background artefacts have similar colours as RoIs. However, these methods to detect objects are still equally important in different image processing applications. Li et al. [48] segmented the images using graph-based and grab-cut image segmentation methods. These methods differ from those proposed by other authors; however, they had high computational costs. Furthermore, the accuracy was low at 86.81% compared to some commonly used methods, such as colour segmentation which achieved 93.2% [39].

The segmentation techniques and methods proposed require a computer vision system to recognise objects in an image and classify them correctly. Several classifiers were proposed, each achieving different performance and computational time. The training of the classifiers required features; hence, several feature extractors were also used. These feature extractor methods extract features like colour, shape, and texture. The methods of extracting these features range from SIFT, SURF, HoG and RPN for training classifiers. The benefits of each feature extractor were presented, with some having better performance while others being worst in computation. In [20], classifiers are grouped into recognition-based; decision-theoretic

and structural methods. Recognition-based classifiers are matching, optimum statistics and neural networks. Structural is defined as matching shape numbers and string-matching classifiers. An example of a matching classifier is template matching, and another example would be the ANN network. A structural classifier would be the Hough Transform to detect shape. Therefore, the classification methods in the literature can be grouped accordingly.

The reviewing of each piece of literature presented a choice in selecting the best possible feature extractor and classifier. Observing the author's findings, for instance, the HoG feature extractor combined with the SVM classifier promised a better combination. Arriving at these findings were based on the literature's criteria: computational cost, performance, complexity, memory utilisation and scalability.

Chapter 3: Methodology

3.1 Introduction

The research question in chapter one sought to answer whether image processing methods could be employed to automatically switch electric locomotives as they traverse through the neutral section. Key objectives were proposed, objectives such as conducting a comprehensive literature review, developing a model, evaluating the model with well-known statistical methods, and publishing at least one conference paper. Chapter 2 presented a comprehensive literature review with relevant publications on techniques and methods used in image processing to segment and classify markers. Discussing these techniques and methods highlighted important issues and the theoretical status of the identified problems. This chapter presents a methodological framework that clarifies how the research was conducted. A methodology framework sought to answer the research question presented in Chapter 1. The methodological framework discussed in this chapter gives detailed steps to answer the research question to enable replication and validation by any academic scholar or researcher. This chapter outlines the phases undertaken, such as the research's planning, design, and implementation. This chapter covers the methodological approach and methods used to allow the proposed computer vision system to detect and identify the markers. Furthermore, it justifies the methods, research instruments, and data collection and analysis techniques employed.

Therefore, this chapter intends to outline steps to design the proposed research model. In addition, a reader wishing to validate the model should replicate it based on the methodology used.

3.2 Research Design

A quantitative approach was implemented in this research study with the priority given to identify existing image processing methods employed in switching electric trains. The representation of the collected pictorial data in the database is in numeric values, and the numeric data required a statistical approach through the model to be empirically validated. Creswell [65] defined the method for testing objective theories by examining the relationship amongst variables as quantitative research. The author further alluded to those statistical methods used to analyse numeric data obtained from the variables measured by instruments. In quantitative research [65], a theory uses interrelated variables and forms propositions that specify the relationship of these variables and predict the research outcomes. Therefore, numeric data representation and empirical validation justified the implementation of the quantitative approach.

3.2.1 Research Philosophy

Williams [66], with Leedy and Ormrod [67], describes research as a process of collecting, analysing and interpreting data to understand a phenomenon.

When considering the proposed research objective and the nature of this research, it was observed that it was concerned with uncovering a computer vision model that switched electric locomotives passing through the neutral section. The research study required careful model evaluation by representing the results empirically. The researcher was able to define the research objectives by drawing conclusions from the research philosophy. Additionally, the researcher was able to collect data objectively, rather than being bias. Easterby-Smith et al. [68] cited positivism philosophy as a social world existing externally and its properties being measured through objective methods instead of inferred subjectivity through intuition. The adopted research philosophy in this research is positivism.

3.2.2 Research Approach

Defining and understanding research philosophy allowed for a suitable research approach to address the research problem or question. Creswell [65] described the research approach as plans and procedures carried out in research involving philosophic assumptions like positivism, specific methods, data collection and analysis. The latter suggests that research activities can be organised effectively, enabling research objectives to be achieved through a research approach. According to Creswell [65], three research approaches can be adapted to research; namely, quantitative, qualitative and mixed methods. Kivunja and Kuyini [69] suggested six quantitative methodologies that can be employed: experimental, quasi-experimental, correlational, causal-comparative, randomised control trial and survey research methodology.

Positivism assumptions were made while the research approach focused on collecting and analysing quantitative data and methods employed. Chapter 2, the literature review, was initially conducted to establish a rationale for the research question and ascertain techniques and methods pertinent to the proposed research. Deductively, a literature review was used to advance the research question. The research dealt with quantitative data generated through images of closing and opening markers installed in railway-neutral sections.

3.3 Research Process

An experimental research method was applied to the developed model to obtain quantitative results. Antwi and Hamza [70] described a research paradigm implies a pattern, framework or system of academic and scientific ideas, values and assumptions. Therefore, the research process comprised the instruments, data collection and methods. The following subsection describes each research process undertaken in developing the model step-by-step.

The dependent variables identified in this research are three classes: namely, open, close, and invalid. Subsequently, these classes are affected by the features extracted from the training images; therefore, the independent variables are the features.

3.3.1 Research Instruments

Research instruments are tools; research tools are defined as a strategy used to collect, manipulate and interpret data, while the methodology is defined as a general approach taken by the researcher to conduct a research project [67]. The study adopted a methodological approach that utilised experimental research and employed existing image processing methods. The utilised instruments were based on numerous literature reviews; hardware and software were employed in collecting, manipulating, and interpreting data.

Hardware: Images must be captured and stored in a storage device to be processed; therefore, the literature review suggested two standard camera sensors, Charged Coupled Device (CCD) and a Complementary Metal Oxide Semiconductor (CMOS). Based on the literature review, a CCD camera was used for image acquisition by weighing the pros and cons of each sensor type. A 640*480 outdoor waterproof HD webcam was used: this type of camera was selected based on availability and budget constraints; however, a higher resolution would have been a better choice. Nonetheless, the literature review suggested that image processing techniques could adjust the resolution of images. Additionally, a lighting stand was used to mount the camera and a measuring wheel to measure the capturing distances. Windows 10 computer with 4GB of RAM, Intel Celeron CPU N2840 @ 2.16GHz was used to store the captured images, install, and run software that enabled the development of the model.

Software: Design-Spark CAD software was used to design and develop the two markers (open and close): markers were designed according to SANS1186. The selection of Design-Spark was because it is an open-source software. Before capturing and storing the images, a graphic user interface (GUI) application was developed using Microsoft Visual Studio C#. The

GUI allowed the camera to be interfaced with the computer and conveniently label images. The GUI automated the process of labelling since images were taken at different intervals and weather conditions. Labelling images based on the conditions mentioned in the latter allowed us to deduce how each image was affected and under which condition. Pain.net software was installed to create ground truth images to validate the model's accuracy. MATLAB was implemented to develop the model by writing scripts with the proposed algorithm inferred from the literature review.

3.3.2 Data Selection

This research not only studied techniques and methods employed in computer vision through a literature review but also assessed the datasets used. Chen et al. [4] also developed a computer vision model that switched electric locomotives using special Chinese characters. The dataset of these Chinese markers revealed no standard datasets besides traffic signs and number plates. Creswell [65] suggested that important questions be answered when identifying data sources and selecting participants in quantitative research. Similarly, the selected data was based on the latter, firstly answering a question on whether the data was associated with images with markers installed in Transnet neutral sections. Secondly, looking at whether the data was appropriately labelled for training and testing the model. Thirdly, if there is recognition of diversity within the targeted data. Finally, focussing on whether improvements on the data can be done to increase the accuracy.

The data for this research concentrated on two markers based on the existing marker found in Transnet neutral sections. The selection criteria was structured from the suggested questions in [65]; the following are the main criteria:

1) Data must be quantifiable since the research approach was quantitative.
2) The shape of each marker needed to be a circle.
3) Markers required having a black background and white foreground.

4) The background and foreground be reflective.

5) Characters in each marker are clearly labelled.

6) Markers needed to conform to the SANS1186 standard.

7) The distance and weather condition when acquiring the data was considered.

The on-site markers enabled images to be captured and dataset to be created. The recognition of the diversity of the markers is reflected only in the proprietary use in Transnet neutral sections. The selection of data was considered based on the literature review where image processing techniques showed that such data could be improved. In Subsection 3.3.3, the data collection procedure is detailed for the reader to replicate the data collection process.

3.3.3 Data Collection Methods

Primary data was collected and used to train the model: as described in Subsection 3.3.2, there were no standard datasets specifically for computer vision systems that switched electric locomotives in a neutral section. The proposed research required two markers for opening and closing the Vacuum Circuit Breakers (VCBs) inside the locomotives. The design of the two markers was based on the existing marker and the two induction magnets currently installed in the Transnet neutral sections. The "N" marker was designed to improve the existing marker and replace the first set of induction magnets responsible for switching off the locomotives. Currently, the second set of induction magnets is used to switch the locomotives on by closing the VCB; therefore, the close marker was designed to replace the magnets.

(a) Designing of Markers

Before data collection, the markers must be designed first: Design-Spark, as mentioned in Subsection 3.3.1, was utilised as CAD software.

- The measurements of the existing marker were measured with measuring tape, and mounting wholes, a digital Vernier Calliper was used for accuracy.

- The measurements were manually transferred to Design-Spark by drawing the open and close markers according to the measured dimensions.
- After translating the measurements into 2D shapes, these shapes were 3D rendered to conceptualise the final expected markers.
- Once the 3D markers and the dimensions conformed to the standard measurements, the drawings were sent to a printing company which then manufactured the two markers.

(b) Design of GUI

As described in Subsection 3.3.1, a GUI was designed and developed mainly to provide the camera with an interface with the computer and conveniently label images. The computer utilised for this research did not have an application to interface with the camera to allow images to be captured. While downloadable applications were available to interface with the camera, developing a GUI gave more control, specifically when labelling each image, since this process was automated. Usage of the GUI allowed images to be labelled accordingly: the distance from where they were captured and the weather condition. The GUI was developed using Microsoft Visual Studio C#, and Fig. 3. 1 illustrates the designed application.

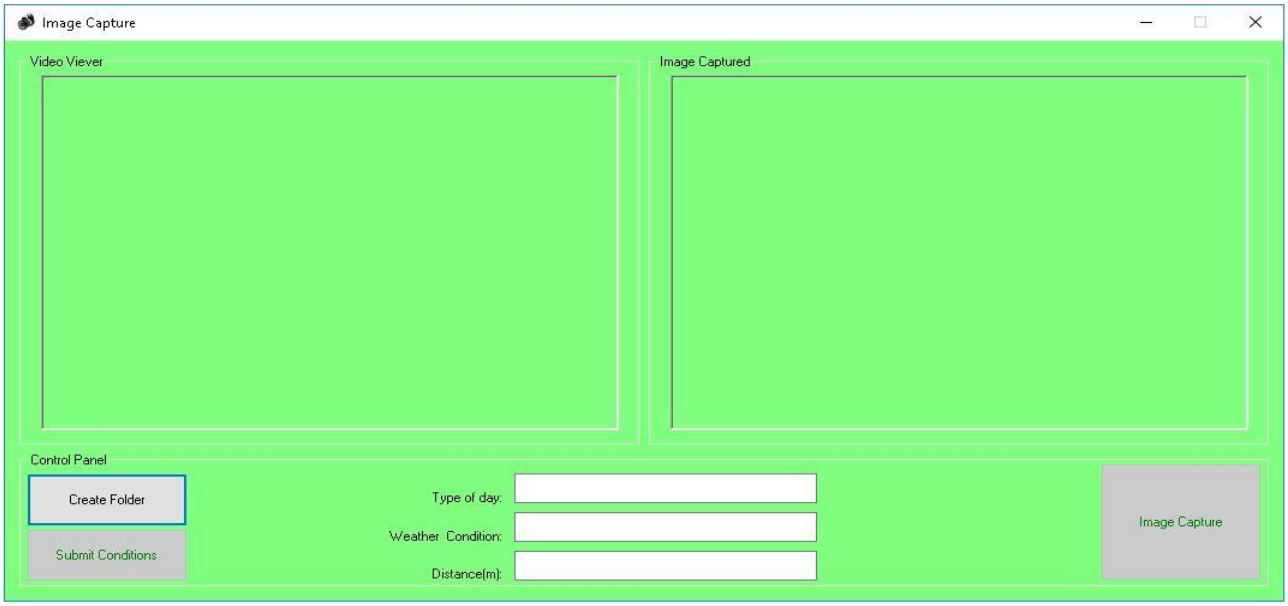


Fig. 3. 1: Image acquisition GUI

(c) Data Collection

The data collection process was based on field and lab experiments: data collected from the field experiment was acquired on-site (primary data), while the latter could be considered secondary data. Bacon-Shone [71] defines secondary data analysis as the data previously collected initially from the primary data. Likewise, the secondary data in this research was the primary data manipulated by employing image processing methods to increase the dataset. Adam [72] presented Yamane and Cochran formulas for data sampling size. These formulas could determine dichotomous and continuous variables sample sizes; however, the literature in [72] showed that they best determine survey data sample sizes. The research then investigated the literature for methods used to determine the data size for a computer vision system. The findings showed that the data size varied; however, it was observed that the training data size was, on average, above 70% larger than the testing data [41, 43, 44]. Giving an insight into the minimum expected data size for this research, but is not limited to this minimum size, as some literature had larger dataset sizes [58].

Field experiment: after manufacturing the markers, they were installed on the line where there was no movement of trains so as not to disrupt train services. The markers were installed on two mast poles at 3.85m in height from the ground and 14m apart. The justification of the distance and height was based on the existing installed markers. A measuring wheel was used to measure the distances where images were captured: 10m, 14m, 20m, 25m, 30m and 45m were the distances marked on the rail with white chalk. The 10m to 45m was based on the initial configuration of the neutral section, where the phase break was 9.4m in length, so the minimum distance was rounded off to 10m, and the magnets were 45m apart. A lighting stand was used to position the camera; magnets attached underneath it allowed the camera to be positioned on the stand since it was made of metal. The stand was placed at each distance in the middle of the railway track. The height of the stand was 1.5m, resembling the proposed camera installation height on the locomotives. The images were then acquired (in different weather conditions: sunny, cloudy and at night) by the camera using the GUI and automatically stored in a folder. Furthermore, during image acquisition, the camera was intentionally tilted and shaked randomly to simulate vibrations that would be caused by trains. The data collection setup is demonstrated in Fig. 3. 2, showing the measuring wheel, stand position, camera mounting, and computer used for running GUI and storing the images. The dataset sample had 200 images divided into training and testing images: training had 120 images while testing images were 80 for the proposed model. The training and testing dataset were then increased in a laboratory setup. Previous

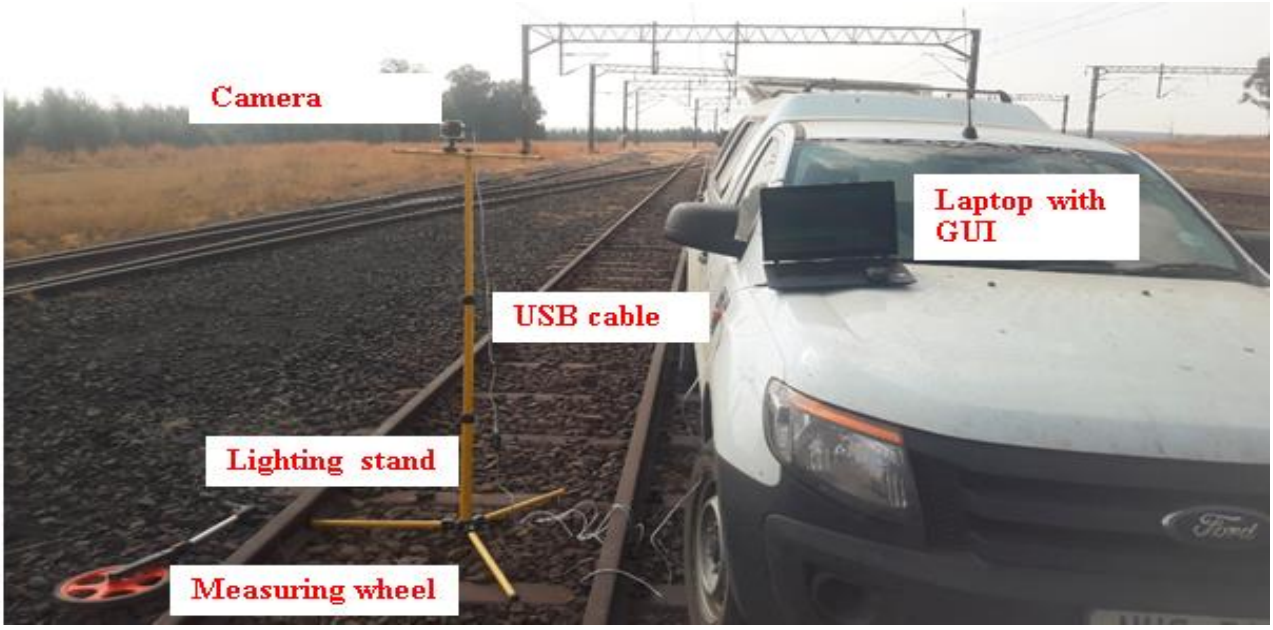


Fig. 3. 2: Data collection setup

Laboratory: the concept presented above on determining the training and data size increased the data in a laboratory. The training data was increased to 422 images, while the test data was increased to 128 images. The dataset now increased to a total of 550 images comprised of around 77% of training data and 23% of testing data. Additional data was then processed to include negative or invalid images of 104, totalling 232 test data. The dataset then increased to a total of 654 images (training and testing). Ground truth images were also manually processed from the 422 training images. The setup of the laboratory data collection was achieved as follows:

- A computer: is used for installing software to conduct research and data storage (Refer to Subsection 3.3.1 for specifications).
- MATLAB 2019a was installed: it was used to create a script which added random noise while increasing the training and testing images.
- Paint.net was installed: it was used to process the 422 images into ground truth manually.

Fig. 3. 3 illustrates some of the images in the dataset used to train the model: columns A - D show images taken while sunny, cloudy, dark, and random noise, respectively. Furthermore, descending from top to bottom rows are images captured at 45m, 30m, 25m, 20m, 14m and 10m. The challenging aspect of this research was not creating the dataset but developing the model. The techniques and methods of developing the model are detailed in Subsection 3.4.

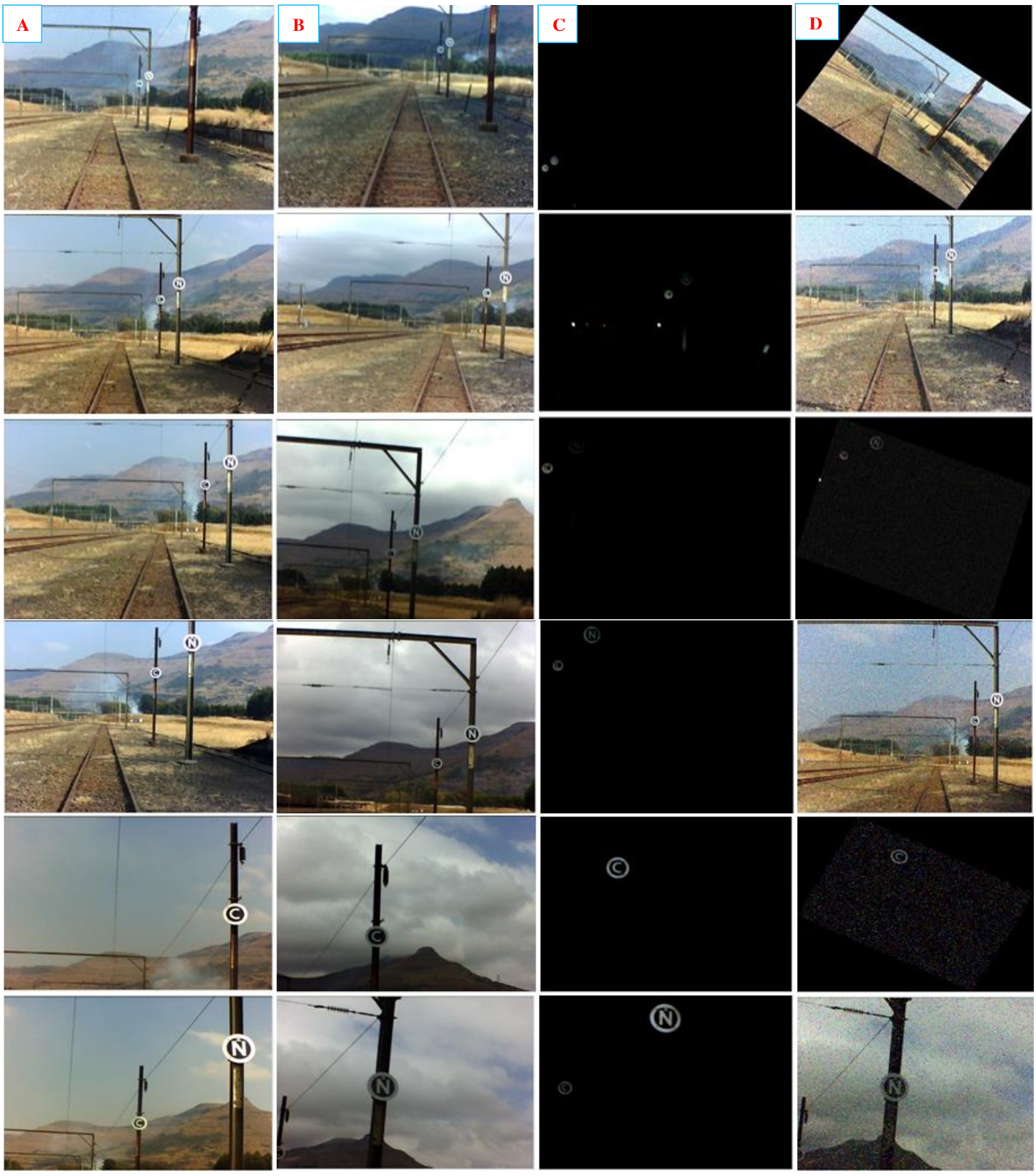


Fig. 3. 3: Dataset images captured at different weather conditions and distances. Column (A) sunny; column (B) cloudy; column (C) dark; column (D) random noise and rotation. Top to bottom row captured images: 45m, 30m, 25m, 20m, 14m and 10m, respectively

3.4 Experimental Model

This section details the techniques and methods of image processing used in developing the proposed computer vision model to switch electric locomotives. The central hypothesis of this research is the automatic switching of electric locomotives traversing through a neutral section by employing computer vision. Categorically, the structure of this section describes an overview of the model adopted, the adopted algorithms and detailed process steps. The methods undertaken in this research are reflected in the conference papers presented at the Information Communications Technology and Society (ICTAS) and International Conference on Engineering and Emerging Technologies (ICEET) conferences. Additionally, the conference papers were published in the IEEE database after they were peer-reviewed: peer review and the publication of the papers cemented the validity approach undertaken in developing the model. The conference papers summarised the methods employed; hence, the experimental model is intended to outline the algorithm and methods used in-depth.

3.4.1 Adopted Model

The structure of the adopted computer vision model followed a similar concept to those proposed in chapter two's literature review. While the proposed computer vision models employed different techniques, the structure of each model followed a similar pattern or process. Fig. 3. 4 illustrates the structure of the proposed model where image acquisition dealt with capturing images and pre-processing focused on the methods employed in denoising the images. Localisation is part of the segmentation stage since it is the first step to delineating the Regions of Interest (RoI). However, in this research, the latter was separated to give clarity on the method employed in locating RoIs. RoI, once extrapolated by the segmentation stage, the classification process allowed for the objects in each RoI to be classified.

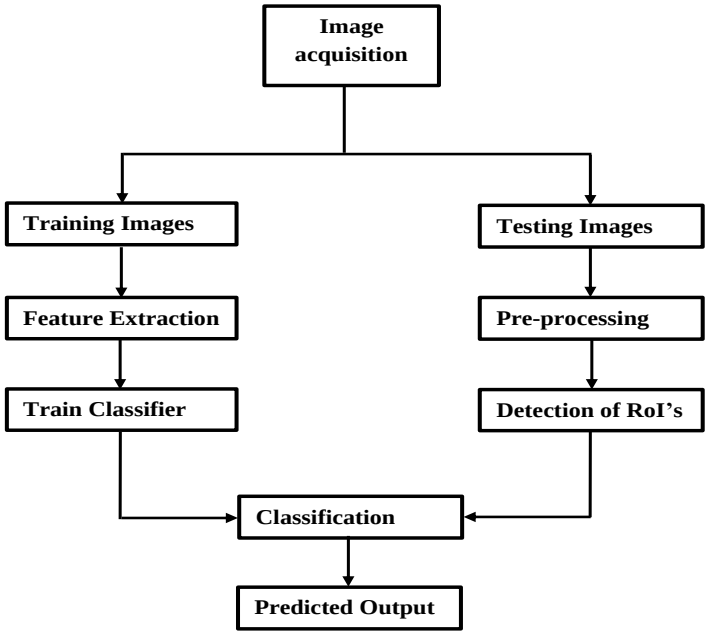


Fig. 3. 4: Proposed model block diagram

3.4.2 Pre-processing

The acquisition of images was already covered in Subsection 3.3.3(c), where the acquired images were in RGB colour space defined by **m*n*3,** which are rows, columns, and the number of channels. This sub-section focuses on Algorithm 1 represented in pseudocode; mathematical equations governing RGB colour space conversion to greyscale images and bilateral filters are discussed. Furthermore, the justification of each method employed in this stage is outlined to support the logical approach taken.

a) RGB to Greyscale Conversion:

Fig. 3. 5 illustrates the structure of an RGB and greyscale image to adopt a method of converting images to greyscale. An RGB image has three channels, red, green, and blue, respectively, with each mathematically represented as a 2D array (size defined by rows*columns). A greyscale image is a 2D array with one channel: pixel ranging from black to white depending on its luminance. Saravanan [73] proposed an improved method of

converting RGB to greyscale images using RGB values approximated from its components. The author began by employing a traditional mathematical model demonstrated in equation (3. 1) to convert the images to greyscale. Then expanded on this approach to improve the conversion accuracy, subsequently increasing the computational cost [73]. Rohrer [74] suggested that linear approximation, defined in equation (3. 1), is a better choice when computational speed is essential; while accuracy may not be 100% accurate, the conversion is within acceptable limits.

The conversion of RGB colour images to greyscale was done on MATLAB using a function called "rgb2gray". Similarly, this function was also governed by equation (3. 1), which allowed each colour image to be converted to greyscale. Rather than developing an algorithm that would convert colour images to greyscale images, the MATLAB function was adopted since it employed the same method in several pieces of literature.

$$\boldsymbol{Y = 0.299 * R + 0.587 * G + 0.114 * B} \tag{3. 1}$$

- $\boldsymbol{Y}$: is defined as the luminance of the pixels ranging from 0-255 for an 8-bit greyscale image; R, G, and B are red, green, and blue components, respectively.

The function, therefore, scans each row-by-column of each RGB channel and outputs a *Y*-pixel value. Different *Y* pixel values for each RGB pixel conversion are stored in a 2D array in the corresponding row and column. The completion of the conversion and storing of *Y* pixels in the array form a greyscale image equal to the original image's size but with one channel.

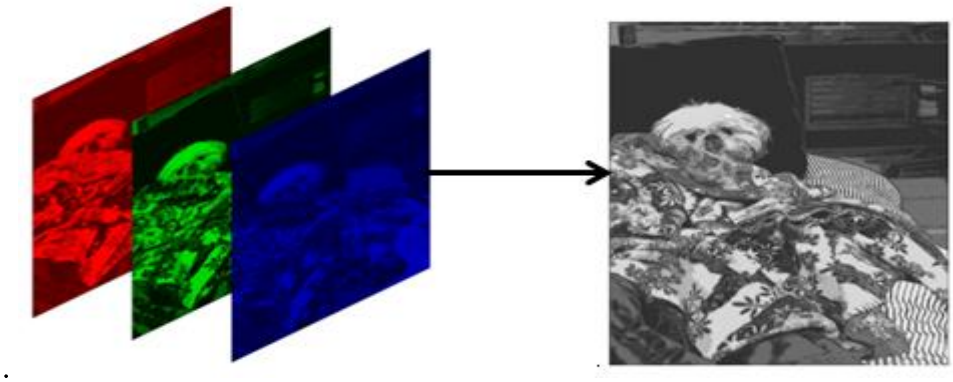


Fig. 3. 5: RGB to greyscale conversion overview [74]

b) Bilateral Filter:

Chapter 2 showed that digital images were subjected to noises from various sources such as the environment and sensors. The technique for denoising these noisy images is called image filtering: various filters depending on the images used, are employed. This research initially employed median and Gaussian filters to remove noise and smooth the images. The idea of adopting these two filters was initiated from the literature: noise was randomly added while primary images collected onsite had noises with Gaussian properties. The selection of a bilateral filter was based on combining the filtering and smoothing of a noisy image while preserving edges [75]. The function used in MATLAB was "imbilafilt" to denoise the greyscale images, and equation (3. 2) defined in [75] is the mathematical equation adopted in the function. The weight $\boldsymbol{W_p}$ is a normalisation factor that ensures the pixel weight sum does not exceed one. This weight assigned to the neighbouring pixel $\boldsymbol{p}$ to a denoise pixel located at $\boldsymbol{q}$ coordinates is defined in equation (3. 3). The algorithm below details how the pre-processing stage was achieved.

$$\boldsymbol{BF[I_p]} = \frac{\mathbf{1}}{\boldsymbol{W_p}} \sum_{\boldsymbol{q \in S}} \boldsymbol{G_{\sigma_s}}(\|\boldsymbol{p} - \boldsymbol{q}\|)\, \boldsymbol{G_{\sigma_r}}(|\boldsymbol{I_p} - \boldsymbol{I_q}|)\boldsymbol{I_q} \tag{3. 2}$$

$$\boldsymbol{W_p} = \sum_{\boldsymbol{q} \in \boldsymbol{S}} \boldsymbol{G_{\sigma_s}}(\|\boldsymbol{p} - \boldsymbol{q}\|)\, \boldsymbol{G_{\sigma_r}}(|\boldsymbol{I_p} - \boldsymbol{I_q}|) \tag{3. 3}$$

Where:

- $\boldsymbol{BF}[.]$: Denotes a bilateral filter with the final filtered image.
- $\boldsymbol{I_q}$: This is the original image value at pixel position $\boldsymbol{q}$.
- $\boldsymbol{I_p}$: This is the filtered image value at pixel position $\boldsymbol{p}$.
- $\boldsymbol{W_p}$: Defines the spatial and range weights of the neighbouring pixel $\boldsymbol{p}$.
- $\boldsymbol{p}$: This is the coordinate of the neighbouring pixel to be filtered.
- $\boldsymbol{q}$: This is the coordinate of the current pixel to be filtered.
- $\boldsymbol{S}$: This is the window centred in $\boldsymbol{q}$, so $\boldsymbol{p} \in \boldsymbol{S}$ defines another pixel.
- $\boldsymbol{G_{\sigma_s}}$: Defines the spatial Gaussian weighting ($\boldsymbol{\sigma_s}$ $\boldsymbol{for\ smoothing}$).
- $\boldsymbol{G_{\sigma_r}}$: Defines the range Gaussian weighting ($\boldsymbol{\sigma_r}$ $\boldsymbol{preserves\ contours}$).

A1) Algorithm for Image Pre-processing:

Algorithm 1, shown in Fig. 3. 6, details the steps in converting RGB to greyscale images and removing noise.

Algorithm 1: Pre-processing

Input: Greyscale marker images

Output: Greyscale noise-filtered images

1. Declare variable (*numberOfImages*)
2. Find the number of images in the dataset: store in *numberOfImages*
3. **for** each image in the dataset ≤ *numberOfImages* **do**
4. Read each image.
5. **if** an image is in RGB colour space, **do**
6. Convert the image to greyscale using equation (3. 1).
7. **else**
8. Do nothing since the image is already in greyscale.
9. **end if**
10. Apply a bilateral filter to remove noise using equation (3. 2).
11. **end for**

Fig. 3. 6: Algorithm 1

3.4.3 Segmentation and RoI Extraction

In Subsection 3.4.2, the focus was on the derivation of mathematical equations on how image conversion was achieved and noise removal along with code implementation in the form of algorithm 1. This part of the methodology chapter looks at segmentation and RoI extraction methods employed in detail.

(a) Segmentation:

Chapter two of the literature review defined segmentation as a process of subdividing an image into its constituent RoI. Therefore, for an open ("N") or close ("C") marker to be extracted and classified, the RoIs need to be located. Several segmentation methods were

proposed in the literature; however, edge detection and Circular Hough transform (CHT) were chosen. Furthermore, numerous models were developed to evaluate standard methods, such as neural networks, colour thresholding and edge detection. These methods presented several challenges [76]:

1) The Convolutional Neural Network (CNN) had a high computational cost; it worsened with increased image size.
2) The CNN obtained a high training rate of 100% but had a low classification rate of 82% on test images compared to the support vector machine (SVM).
3) The CNN also performed poorly with images 30 – 45m away.
4) Employing the colour thresholding method presented an advantage in decreased computational cost: adaptive thresholding such as Otsu was employed; however, due to the histogram of the images, most were either over-segmented or under-segmented. When equalising the image's histogram, the resulting image after thresholding yielded similar results.
5) A model which employed edge detection methods was also evaluated, and standard operators such as Sobel, Canny, Prewitt and Roberts were evaluated. The edge detection obtained faster results like thresholding but left artefacts which required further processing.

The process followed in segmenting and extracting each RoI is summarised in Fig. 3. 12: Algorithm 2. Moreover, the formulas which govern each method are also described in detail. The first stage of the algorithm is taking a denoised greyscale image as an input to the segmentation stage and performing delineation. The second step was applying a Sobel operator for edge detection, followed by CHT. The edge detection allowed the background to be removed while preserving the edges of the markers. The CHT allowed for the RoIs to be detected for extraction to be achieved.

(b) Sobel Operator:

To perform edge detection using a Sobel operator, 3-by-3 masks illustrated by (3. 4) and (3. 5) with horizontal and vertical were employed to detect the edges. Subsequently, most of the background objects were removed while preserving mostly the markers' edges.

$$\boldsymbol{G_x} = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{0} & \mathbf{2} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3. 4}$$

$$\boldsymbol{G_y} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-1} & \mathbf{-2} & \mathbf{-1} \end{bmatrix} \tag{3. 5}$$

The filtered greyscale images were convolved with a mask $|G|$, combining the horizontal (G_x) and vertical (G_y) masks. A MATLAB function "edge" was used to find the edges by calculating the absolute magnitude gradient using equation (3. 6) [77].

$$|\boldsymbol{G}| = \sqrt{\boldsymbol{G_x}^2 + \boldsymbol{G_y}^2} \tag{3. 6}$$

The selection of the Sobel operator was based on the segmentation accuracy when compared with the other operators. The Sobel, Prewitt and Roberts, as shown in Fig. 3. 7(a, b, e), yielded better results when compared to the other operators [76]. Perceptually, Canny, Log and Zero-cross have more unwanted edge artefacts presenting difficulties detecting each marker. Quantitatively employing F-measure, the Sobel operator achieved the highest detection, followed by Prewitt and Roberts, respectively [76]. The procedure followed in selecting a suitable operator is described in Fig. 3. 8 in the form of a flow chart. The flow chart is intended to give a detailed overview of the procedure or steps undertaken to reach the final decision of selecting a 3-by-3 Sobel operator. The computational complexity increases as the size of the mask increase [36]; hence a 3-by-3 was chosen as one balanced speed against

accuracy. The 422 training images were then processed using a 3-by-3 Sobel edge operator, resulting in edge-detected images. A CHT was then implemented on each edge-detected image to find the two RoIs or markers for extraction.

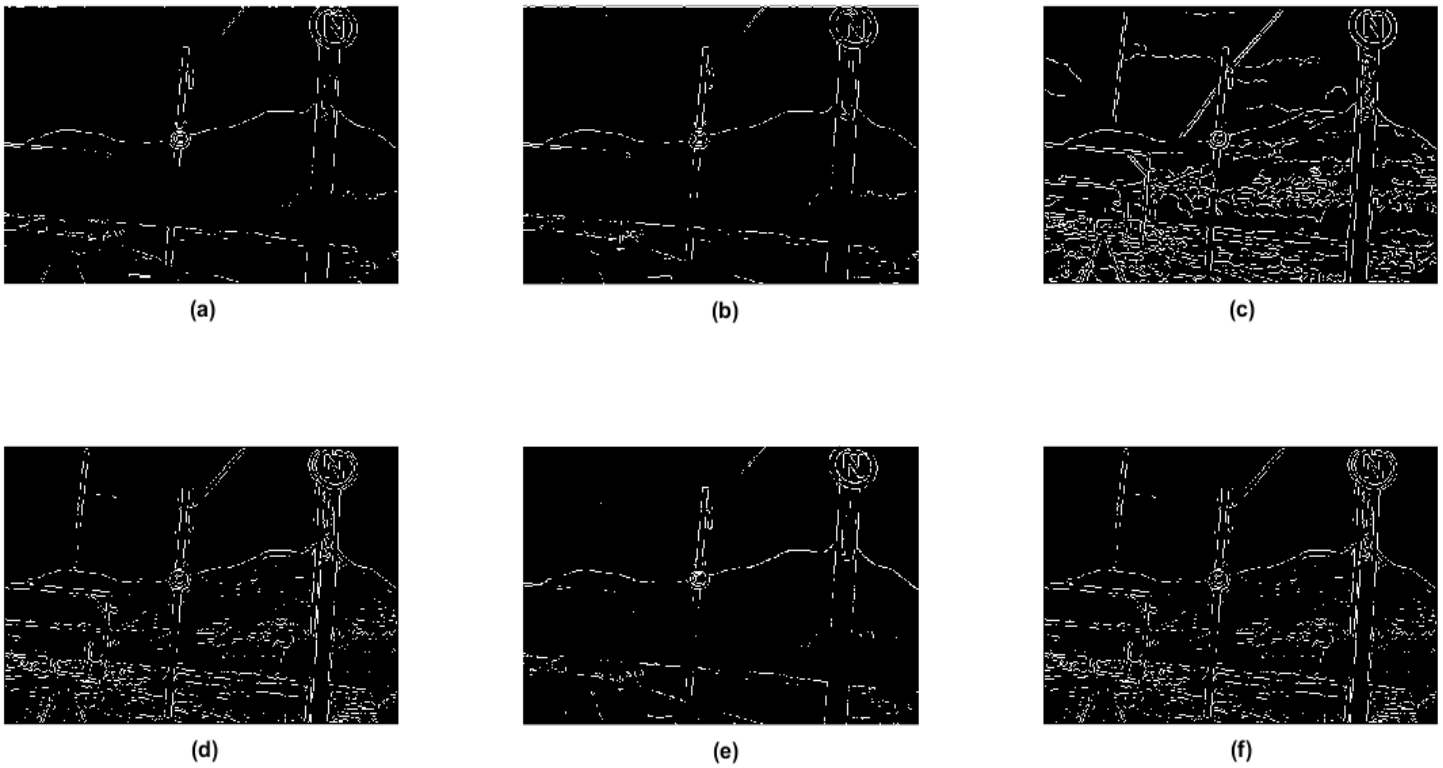


Fig. 3. 7: Comparison of edge detection operators. (a) Sobel; (b) Prewitt; (c) Canny; (d) Log; (e) Roberts and (f) Zero-cross

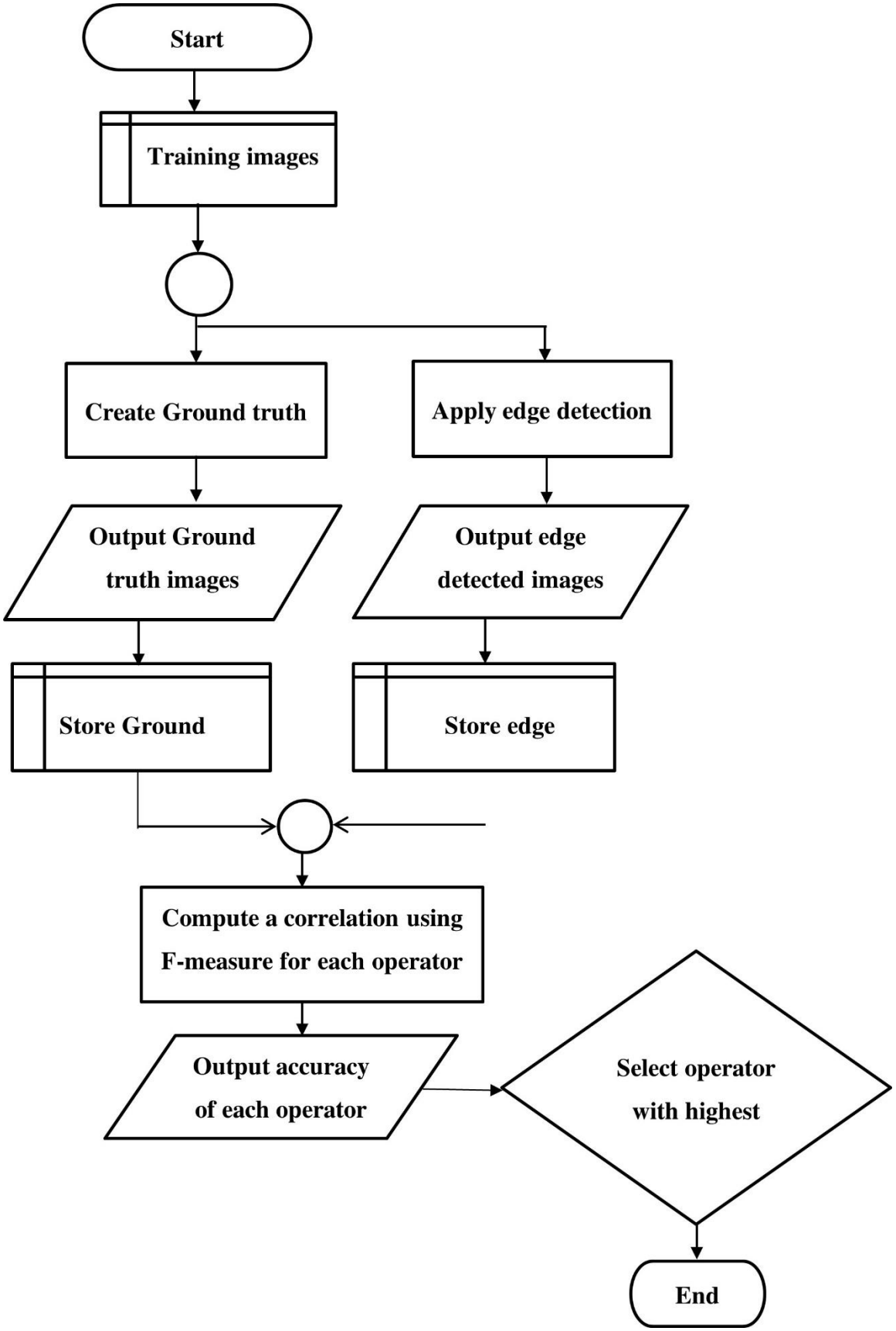


Fig. 3. 8: Edge detector selection flow chart

(c) RoI Extraction:

While the Sobel operator could detect the RoI's edges, it also detects edges of background objects, making it challenging to use this method alone in segmenting the RoI. The images from the edge detection presented some discontinuities on the edges, which required morphological transformation to be applied. The morphology was able to enhance edges with discontinuities when employing dilation, but this approach yielded unsatisfactory results. Nguwi [30] described in chapter two that Hough Transform was best suited for lines and curves: the two markers are circular by shape; therefore, this was one of the criteria used to employ CHT. Cherabit et al. [78] also suggested that CHT is invariant to gaps or discontinuities and unaffected by noise. Subsequently, CHT was employed to locate the RoIs using shape detection. The latter also performed better in finding the RoIs combined with edge detection rather than employing thresholding or edge detection alone.

The MATLAB function used to find the markers is "imfindcircles", based on the CHT principle. In [78], the authors mentioned that the transform was first introduced in 1962 by Paul Hough for detecting features such as lines and curves in digital images. Therefore, one can be confident in the robustness of this method based on its historical background. Though there are some limitations to the CHT, several pieces of literature have effectively employed this method for detecting shapes. Fig. 3. 9 displays how the CHT transforms a circle in an image from the x, y-plane to its 3D parameter space or coordinates (a, b, r). This method transforms the x and y-plane into parametric space, which contains the circle's radius (r) and centre coordinates (a, b).

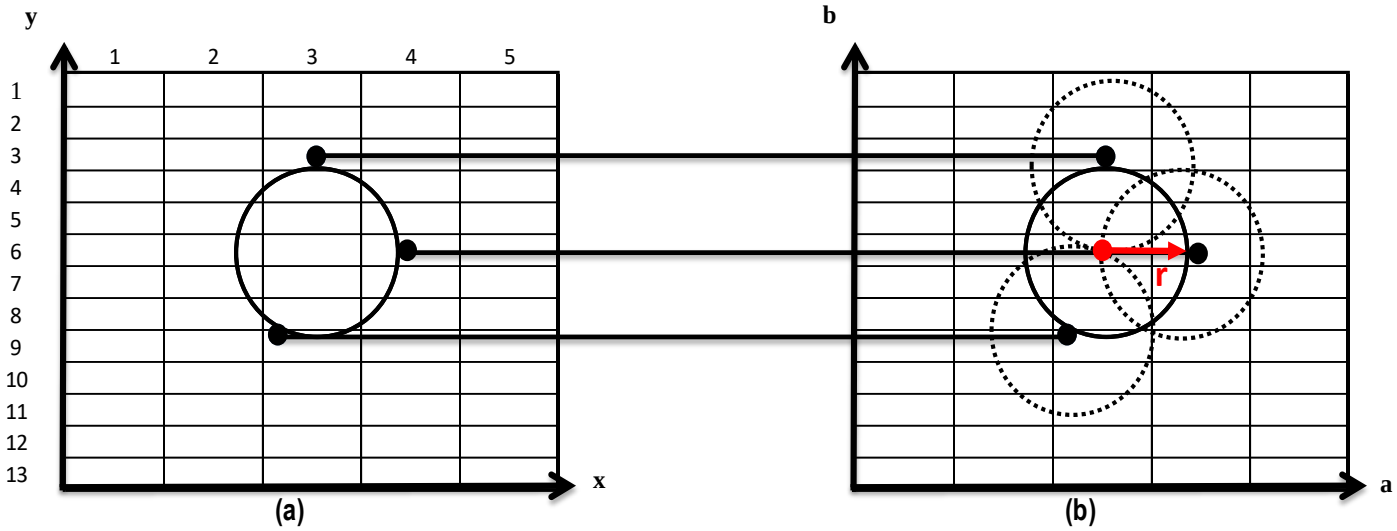


Fig. 3. 9: Transformation. (a) x, y-plane; (b) parametric space

To understand how the RoIs were located using CHT, one needs to understand the theory behind this method and its formulas. In a 2D space such as an image, circles can be described by equation (3. 7).

$$\boldsymbol{r = \sqrt{(x-a)^2 + (y-b)^2}} \tag{3. 7}$$

Where:

- r : Radius of the circle.
- a, b : Coordinates the centre of the circle.
- x, y : Coordinates of a circle on the Cartesian plane.

A minimum of 5 pixels and a maximum of 30 pixels were chosen to find the markers. The minimum and maximum pixel radii were premeasured using Image Tool in MATLAB. Fig. 3. 10 illustrates how the minimum and maximum radii were measured. The markers positioned at 10m and 45m; diameters were manually measured using the Image Tool application. The minimum diameter measured was 10 pixels, and the maximum was 60 pixels. The radii (r)

were then calculated by dividing each diameter by two. The necessity of defining minimum and maximum radii was due to the pixel radius changing because of varying distances when acquiring images.

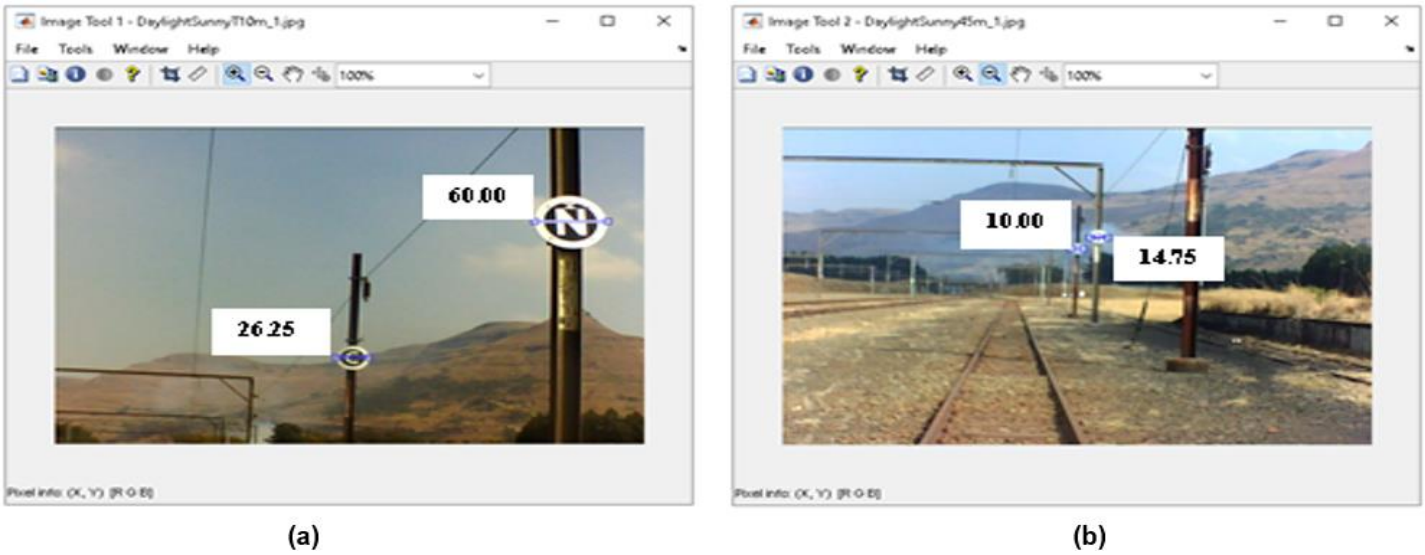


Fig. 3. 10: Image Tool measuring diameter of markers. (a) 10m and (b) 45m distance

The principal operation of a CHT method for locating the markers in images with defined radii steps is summarised as follows:

1) An accumulator array was created (To store voted candidate pixels).
2) Foreground pixels illustrated in Fig. 3. 9(b) (dotted circles) cast votes in the accumulator array. The foreground pixels with high gradients voted in a pattern that forms dotted circles from a fixed radius.
3) Computation of accumulator array with voted candidate pixels was performed: circle centre was found by detecting the peak in the accumulator array. The dotted circles coincide (solid red dot) and define the circle centre.
4) Returned coordinate (a, b, r).

The coordinates obtained from the parameter space allowed for RoI extraction by calculating x and y coordinates using equations (3. 8) and (3. 9). The bounding box approach was employed where the coordinates were obtained by applying equations from (3. 10) up to (3. 13) to crop each RoI. Fig. 3. 11 demonstrates how the bounding box was implemented in extracting RoI from the images for classification.

$$\boldsymbol{x = a + r\cos\theta} \tag{3. 8}$$

$$\boldsymbol{y = b + r\sin\theta} \tag{3. 9}$$

$$\boldsymbol{x_1 = x - r} \tag{3. 10}$$

$$\boldsymbol{x_2 = x + r} \tag{3. 11}$$

$$\boldsymbol{y_1 = y - r} \tag{3. 12}$$

$$\boldsymbol{y_2 = y + r} \tag{3. 13}$$

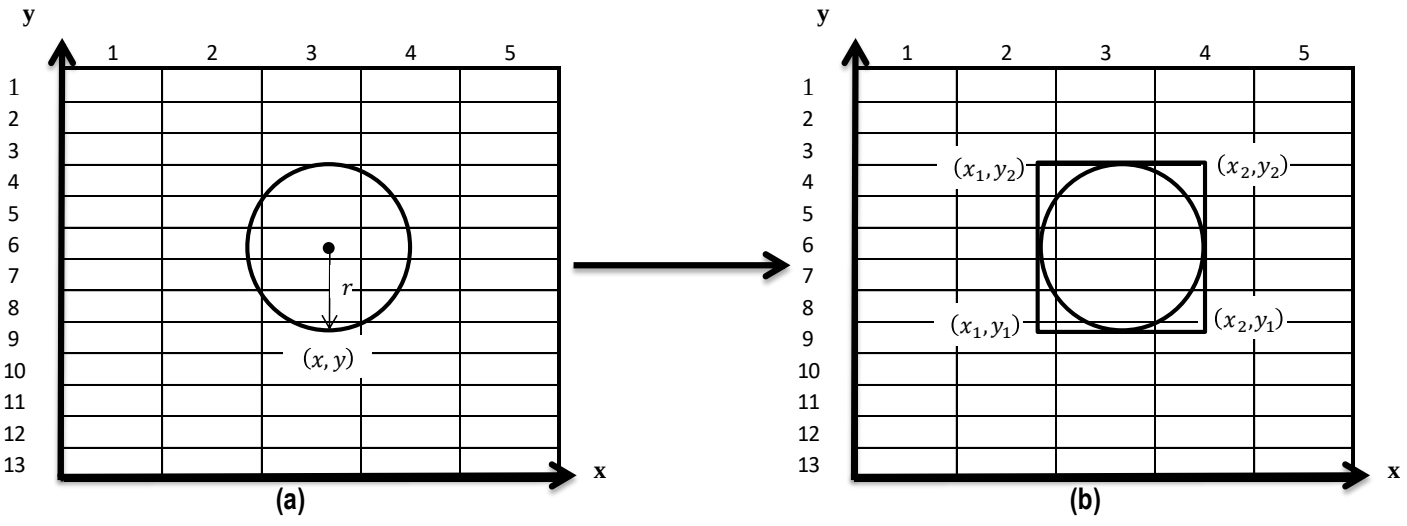


Fig. 3. 11: Overview of RoI extraction. (a) circle radius and (b) bounding box

A2) Algorithm for Segmentation and RoI Extraction:

Algorithm 2, shown in Fig. 3. 12, presents the pseudocode for the segmentation and extraction of markers in each image.

Algorithm 2: Segmentation and RoI extraction

Input: Greyscale noise-filtered images (Algorithm 1)

Output: Cropped images with RoIs (markers)

1. Declare vector variables *centres*, *radii* and *circelsFound.*
2. **for** each greyscale-filtered image, **do**
3. Apply the Sobel operator using equation (3. 6).
4. Find the centres and radii of the circles with CHT using equation (3. 7).
5. Calculate the number of circles (*circlesFound)* in each image from the radii.
6. **for** *circlesFound* ≥ 1 **do**
7. Get the radius of each circle.
8. Calculate the coordinates (x1,x2,y1,y2) using equations (3. 8) - (3. 12):
9. **if** (centre of each circle – respective radius) < 0 **do**
10. **if** x1 ≤ 1 **do**
11. x1 = 1
12. **else:** x1 = radius – centre
13. **else-if** (centre of each circle – respective radius) > 0 **do**
14. **If** x1 ≤ 1 **do**
15. x1 = 1
16. **else:** x1 = centre - radius
17. Repeat steps 9 – 16 to assign the y1 coordinate value.
18. Calculate y2 using the centre used for y1:
19. **if** (centre of each circle + respective radius) > image row size **do**
20. y2 = image row size
21. **else:** y2 = centre + radius
22. Repeat steps 19 – 21 to assign x2 using x1 centre and radius.
23. Crop image with a bounding box coordinate(x1:x2,y1:y2).
24. Resize cropped images to 60x60 (depending on the classifier input size).
25. **end for**
26. **end for**

Fig. 3. 12: Algorithm 2

3.4.4 Classification

The core of a computer vision system is enabling a computer to interpret the data and subsequently label or recognise an object of interest embedded in an RoI. Classification techniques and methods, therefore, enable a computer vision machine to classify an object or objects. The extracted RoIs are then passed through a machine learning classifier which has been trained to detect RoIs. The objectives of the classifier are to reject invalid RoIs and detect and assign object classes. Before classifying the markers, the classifier is trained using the feature extraction method.

(a) Feature Extraction:

Before training a classifier, features of both markers were extracted using HoG and stored in a Bag of Feature (BoF) vectors. Algorithm 3 defines how features were extracted by employing the HoG method. Navneet and Bill [79], original paper of HoG, suggested that the image be first processed into width-to-height ratio of 1:2, making calculations simple. The authors proposed 8*8-pixel cells and 16*16-pixel blocks with nine orientation bins from 0° - 180°. In the proposed model, the images were processed into width-to-height ratio of 1:1 with 4*4-pixel cells having 4*4-pixel blocks with nine orientation bins. The cropped images varied in size, with some having higher pixel resolution while others with less: images that were captured 45m away had less pixel resolution than images captured 10m.

During pre-processing, the width-to-height ratio was made 1:1 by resizing all the cropped images to 60*60 pixels; this seemed to be a better ratio as it preserved the images' sharpness and shape. In Fig. 3. 13, it was observed through experiments that some images that were cropped, when pre-processed by resizing them into a 1:2 ratio, the shape sharpness of some images would be affected. The selection of 4*4-pixel cells and blocks was also based on experiments conducted: a bigger size reduces features that can be extracted, while smaller cells will give increased features but also increase computation. Referring to Fig. 3. 14, it can be

noted that a 4*4 cell-size has enough feature gradients and orientations which are not too much or too small. As the cell size increases, critical pixels on the edges are missed, which may lead to false classification.

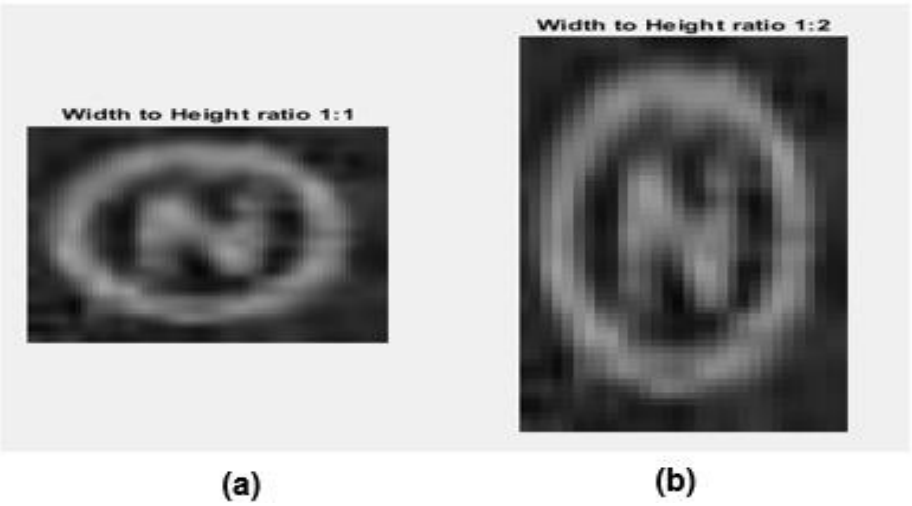


Fig. 3. 13: Width-to-height ratio overview

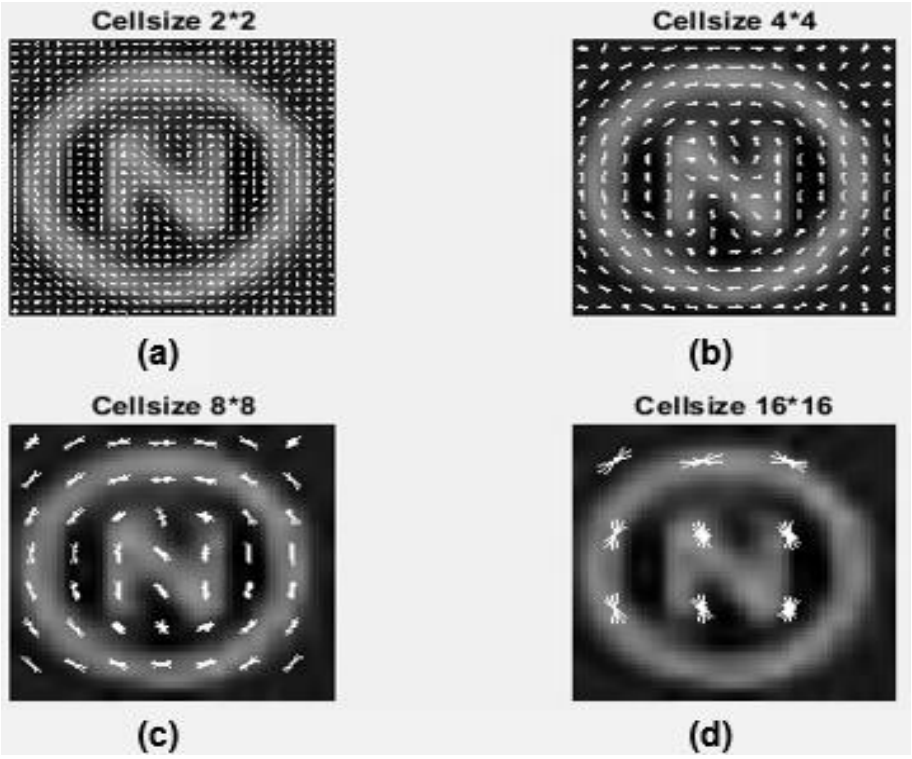


Fig. 3. 14: HoG visualisation by cell-size

Once the pre-processing was complete, the gradient change in the x and y direction for every pixel in the image was calculated. Table 3. 1 demonstrates a matrix assuming it comes from a patch taken from an image with the pixel values merely used as an example. The highlighted pixel 79 would be the selected pixel; therefore, the gradient change in both directions is referenced on this pixel value. By subtracting the pixel value on the left of the selected pixel from the pixel value on the right, the x-direction is calculated. Therefore, by subtracting the pixel value below the selected pixel from the above pixel value we obtain the y-direction.

Table 3. 1: Matrix of pixels sample

89	210	20
102	79	190
200	89	179
39	66	77

- Change in the x-direction (G_x) = 190 – 102 = 88
- Change in the y-direction (G_y) = 210 – 89 = 121

Calculating the gradient change is done for every pixel in the image (in this case, the cropped images). The magnitude and orientation of each gradient change are then calculated using the formula defined in (3. 6) and (3. 14), respectively. The gradient magnitude, for example, obtained from the above G_x and G_y is approximately 149.6162 while the orientation angle would be about 53.973°. Table 3. 2 illustrates a histogram with the contribution of each pixel gradient magnitude added to the bins on either side of the pixel gradient magnitude ($|\boldsymbol{G}|$). Equation (3. 15) was used to calculate the contribution of each pixel gradient magnitude which was then added to the absolute pixel magnitude, resulting in the histogram gradient magnitude defined in (3. 16). The higher contributing $\boldsymbol{G_h}$ is allocated to the bin value which is closer to the orientation angle. The orientation $\boldsymbol{\theta_{(x,y)}}$ of 53.973° for instance is between 40° - 60° bins

therefore, the higher value of $\boldsymbol{G_h}$ would be allocated to 60° bin since 53.973° is closer to it while lower value would be allocated to 40° bin.

$$\boldsymbol{\theta_{(x,y)}} = \boldsymbol{tan^{-1}} \frac{\boldsymbol{G_y}}{\boldsymbol{G_x}} \tag{3. 14}$$

$$\boldsymbol{G_b} = \frac{\left(\boldsymbol{\theta_{(x,y)}} - \boldsymbol{Bin}\right)}{\boldsymbol{BinSize}} * |\boldsymbol{G}| \tag{3. 15}$$

$$\boldsymbol{G_h} = |\boldsymbol{G}| + \boldsymbol{G_b} \tag{3. 16}$$

Where:

- $\boldsymbol{\theta_{(x,y)}}$: Gradient direction or orientation angle of (x, y) pixel coordinate.
- $\boldsymbol{G_b}$: Contribution pixel gradient magnitude.
- $\boldsymbol{G_h}$: Histogram gradient magnitude.
- $\boldsymbol{Bin}$: This is the bin value next to the orientation angle defined by (x, y).
- $\boldsymbol{BinSize}$: This is the bin size which has a value of 20.

Table 3. 2: Histogram of gradient magnitude at G_x= 88, G_x = 121, $\theta_{(x,y)}$ at 53.973°

Magnitude ($\boldsymbol{G_h}$)	-	-	194,703	254,1456	-	-	-	-	-
Bin	0°	20°	40°	60°	80°	100°	120°	140°	160°

Already created HoG features get affected by the gradients of each image since gradients are sensitive to the overall lighting. Navneet and Bill [79] proposed that these features be normalised to reduce lighting variation. The Detection Error Trade-off (DET) versus false positive per window (FPPW) curve was plotted to quantify the performance of each detector [79]. The authors in [79] employed a Bhattacharya distance formula to select the normalisation scheme that outperformed the others. The selected block normalisation scheme in [79] is

defined by the formula in (3. 17); this is the scheme employed in the proposed research to normalise the HoG feature vector.

$$\boldsymbol{V_{L2-norm}} = \frac{\boldsymbol{v}}{\sqrt{\|\boldsymbol{v}\|_2^{\,2} + \boldsymbol{\epsilon^2}}} \tag{3. 17}$$

Where:

- $\boldsymbol{V}_{L2-norm}$: Normalised feature vector.
- v : Un-normalised feature vector.
- $\|v\|_2$: Length of vector also called vector norm where L2-norm is used.
- ϵ : Small normalisation constant.

The total normalised feature vectors obtained were 227*7056 or 1 601 712 features from 227 cropped images of size 60*60: HoG parameters selected was a 4*4 cell-size which makes up a 9*1 matrix with 4*4 block size. The 7056 defines the number of features extracted in one cropped image, while 227 is the number of cropped images used for extracting these features.

A3) Algorithm for Feature Extraction Using HoG:

Algorithm 3 in Fig. 3. 15 are steps taken to extract features using HoG.

Algorithm 3: Feature extraction using HoG

Input: Cropped images (manually cropped RoIs)

Output: Concatenated feature vector

1. Resize cropped images to 60x60.
2. Declare vector variables (training*Features, trainingLabels*).
3. Find the number of cropped images (*numCropImages*)
4. **for** *numCropImages* ≥ 1 **do**
5. Divided into a cell.
6. **for** each cell, **do**
7. Obtain HoG gradient $|\boldsymbol{G}|$ for every pixel using equation (3. 6).
8. Compute magnitude $\boldsymbol{G_h}$ and orientation $\boldsymbol{\theta_{(x,y)}}$ using equations (3. 14) - (3. 16).
9. Compute and normalise the histogram $\boldsymbol{V_{L2-norm}}$ using equations (3. 17).
10. **end for**
11. Form BoF (concatenated feature vector).
12. **end for**

Fig. 3. 15: Algorithm 3

(b) Classification of RoIs Using Support Vector Machine (SVM):

Algorithm 4 describes the steps in training an SVM classifier by employing BoFs obtained from the extracted HoG features. Furthermore, the classifier was tested using test features extracted from the test images to validate the accuracy of the SVM classifier. In algorithm 4, the researcher describes the procedure for classifying the RoIs by employing the trained SVM classifier.

As described in algorithm two, Fig. 3. 12, the extracted RoIs are then passed through an SVM classifier that compares the RoIs' features with those from the classifier. A class is then assigned to each RoI: classes are simply labels identifying each RoI as either an open, close, or invalid marker. The SVM classifies an RoI by selecting an optimal hyperplane which segregates classes better by computing the maximum distance between the nearest data point

or support vectors. Boswell [80] expressed SVM as a classifier that separates two classes by finding the optimal hyperplane with the maximum margin that separates support vectors. The training data is defined as $\{x_i;\ y_i\}$, where $i = 1 \cdots l$ (being training points or features) with ($x_i \in R^d$), where d represents inputs and $y_i \in \{-1, +1\}$ denote classes for open or close marker. The author describes all hyperplanes in R^d being parameterised by the weight vector **($\boldsymbol{w}$)** and a bias constant **($\boldsymbol{b}$),** expressed in the below equation with feature vectors ***($\boldsymbol{x}$)***.

$$\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b} = \mathbf{0} \tag{3. 18}$$

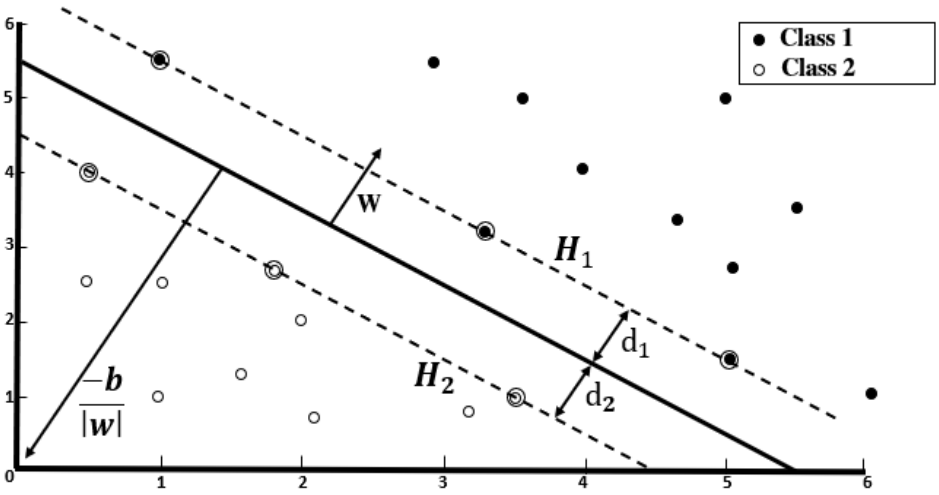


Fig. 3. 16: Hyperplane illustrating two linearly separable classes [81]

The classes closer to the hyperplane in Fig. 3. 16 are the support vectors, and the implementation of these vectors depends on the selection of variables $\boldsymbol{w}$ and $\boldsymbol{b}$ so that the training data can be defined by:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \geq +\mathbf{1} \qquad \text{for } y_i = +1 \tag{3. 19}$$

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \leq -\mathbf{1} \qquad \text{for } y_i = -1 \tag{3. 20}$$

The equations in (3. 19) and (3. 20) can be combined into one equation which equally defines the training data as:

$$\boldsymbol{y_i(w \cdot x_i + b) - 1 \geq 0 \; \forall_i} \tag{3. 21}$$

The point on the planes H_1 and H_2 shown in circles are the support vectors, and the planes are described by:

$$\boldsymbol{w \cdot x_i + b = +1} \qquad \text{for } \mathrm{H}_1 \tag{3. 22}$$

$$\boldsymbol{w \cdot x_i + b = -1} \qquad \text{for } \mathrm{H}_2 \tag{3. 23}$$

The distance which separates the hyperplane from the support vectors is the margin. Referring to Fig. 3. 17, the distance from H_1 to the hyperplane denoted by d_1 defines the margin. Similarly, d_2 describes the margin or the distance from H_2 to the hyperplane and since the planes are equidistance, this means that $d_1 = d_2$. The margin is described by the distance from a point (x_i, y_i) to a hyperplane, recall the distance from a point to a line is $A_x + B_y + C = 0$. This equation further expands to $\frac{|A_{x_i} + B_{y_i} + C|}{\sqrt{A^2 + B^2}} = \frac{\boldsymbol{w \cdot x_i + b}}{\|w\|} = \frac{1}{\|w\|}$; so the total distance between the two planes (H_1 and H_2) or the total margin is $\frac{2}{\|w\|}$. To maximise the margin $\frac{1}{\|w\|}$ subject to the constraint in (3. 20), would be equivalent to minimising $\|w\|$. Maximising the margin will ensure that the hyperplane is far away from the support vectors to accurately separate the different classes (close and open markers). The equivalent of minimising $\|w\|$ can be represented by minimising $\frac{1}{2}\|w\|^2$, which then becomes a Quadratic Programming (QP) optimisation problem. This constrained optimisation problem, therefore, can be solved by

applying the Lagrange multipliers. The primal optimisation problem is denoted by $\boldsymbol{L_p}$ while α, being the Lagrange multipliers where $\boldsymbol{a_i} \geq 0\ \forall_i$:

$$\boldsymbol{L_p} \equiv \frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{w}\|^2 - \boldsymbol{\alpha}[\boldsymbol{y_i}(\boldsymbol{w}.\boldsymbol{x_i} + \boldsymbol{b}) - \mathbf{1}\ \forall_i] \tag{3. 24}$$

$$\boldsymbol{L_p} \equiv \frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{w}\|^2 - \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i y_i}(\boldsymbol{w}.\boldsymbol{x_i} + \boldsymbol{b}) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i} \tag{3. 25}$$

The following approach is to find (w, b), which allows for minimisation and α to maximise the primal optimisation problem or Lagrange equation ($\boldsymbol{L_p}$). Therefore, this is done by partially differentiating $\boldsymbol{L_p}$ with respect to w and b.

$$\frac{\boldsymbol{\partial L_p}}{\boldsymbol{\partial w}} = \boldsymbol{w} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i y_i x_i} \tag{3. 26}$$

$$\frac{\boldsymbol{\partial L_p}}{\boldsymbol{\partial b}} = \mathbf{0} \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i y_i} = \mathbf{0} \tag{3. 27}$$

Substituting (3. 26) and (3. 27) we obtain a Dual form ($\boldsymbol{L_D}$) which requires the dot product of each input vector x_i to be calculated.

$$\begin{aligned} \boldsymbol{L_D} &\equiv \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i} + \frac{\mathbf{1}}{\mathbf{2}} \sum_{\boldsymbol{ij}} \boldsymbol{a_i a_j y_i y_j}(\boldsymbol{x_i} . \boldsymbol{x_j}) \\ \boldsymbol{s.t.} &\sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i y_i} = \mathbf{0}\ \boldsymbol{and}\ \boldsymbol{a_i} \geq \mathbf{0}\ \forall_i \end{aligned} \tag{3. 28}$$

The dual form maximises over $\boldsymbol{a_i} \geq 0$, and it is worth noting that many of the nonzero $\boldsymbol{a_i}$ are zero at their optimum. Support vectors $\boldsymbol{x_s}$ that satisfies (3. 21) will be represented by $\boldsymbol{y_s}(\boldsymbol{w} \cdot \boldsymbol{x_s} + \boldsymbol{b}) = 1$, where S denotes set of indices of the Support vectors, which is determined

by finding the indices (i). Therefore, the derivative of $\boldsymbol{L_D}$ with respect to a nonzero $\boldsymbol{a_i}$ when substituting (3. 26) gives:

$$\boldsymbol{y_s}\left(\sum_{\boldsymbol{m}\in\boldsymbol{S}} \boldsymbol{a_m y_m x_m} \cdot \boldsymbol{x_s} + \boldsymbol{b}\right) = \mathbf{1} \tag{3. 29}$$

The equation (3. 29) is then expanded by multiplying $\boldsymbol{y_s}$; referring to (3. 19) and (3. 20), we can use $\boldsymbol{y_s}^2 = 1$ to solve and simplify $\boldsymbol{b}$:

$$\begin{aligned} \boldsymbol{b} &= \frac{\mathbf{1}}{\boldsymbol{y_s}} - \sum_{\boldsymbol{m}\in\boldsymbol{S}} \boldsymbol{a_m y_m x_m} \cdot \boldsymbol{x_s} \\ &= \frac{\boldsymbol{y_s}^2}{\boldsymbol{y_s}} - \sum_{\boldsymbol{m}\in\boldsymbol{S}} \boldsymbol{a_m y_m x_m} \cdot \boldsymbol{x_s} \\ &= \boldsymbol{y_s} - \sum_{\boldsymbol{m}\in\boldsymbol{S}} \boldsymbol{a_m y_m x_m} \cdot \boldsymbol{x_s} \end{aligned} \tag{3. 30}$$

Fletcher [81] furthermore suggests averaging out all the Support Vectors in S ($\boldsymbol{N_s}$) instead of using the arbitrary Support Vector $\boldsymbol{x_s}$. Therefore, the resulting equation for $\boldsymbol{b}$ is:

$$\boldsymbol{b} = \frac{\mathbf{1}}{\boldsymbol{N_s}} \sum_{\boldsymbol{s}\in\boldsymbol{S}} \left(\boldsymbol{y_s} - \sum_{\boldsymbol{m}\in\boldsymbol{S}} \boldsymbol{a_m y_m x_m} \cdot \boldsymbol{x_s}\right) \tag{3. 31}$$

The optimal separating hyperplane is now defined since the variables $\boldsymbol{w}$ and $\boldsymbol{b}$ have been obtained; subsequently, an SVM is formed. Schölkopf et al. and Fletcher introduced a slack variable (ξ_i) [81, 82] to relax the constraints (3. 19) and (3. 20). This is to allow for data that is not entirely linearly separable to slightly have misclassification with minor errors. The substitution of this positive slack variable results in a relaxed hyperplane:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \geq +\mathbf{1} - \boldsymbol{\xi_i} \qquad \text{for } y_i = +1 \tag{3. 32}$$

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \geq -\mathbf{1} + \boldsymbol{\xi_i} \qquad \text{for } y_i = -1 \tag{3. 33}$$

Similarly to (3. 21), the above equations can be combined to form:

$$\boldsymbol{y_i}(\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b}) - \mathbf{1} + \boldsymbol{\xi_i} \geq \mathbf{0}\ \forall_{\boldsymbol{i}} \tag{3. 34}$$

Furthermore, reformulating $\frac{1}{2}\|w\|^2$ with the addition of parameter C that controls the trade-off between the margin size with the slack variable penalty, and equation (3. 35) results in the following:

$$\boldsymbol{L_p} \equiv \frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{w}\|^{\mathbf{2}} + \boldsymbol{C}\sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{\xi_i} - \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{a_i}[\boldsymbol{y_i}(\boldsymbol{w}.\boldsymbol{x_i} + \boldsymbol{b}) - \mathbf{1} + \boldsymbol{\xi_i}] - \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{\mu_i}\boldsymbol{\xi_i} \tag{3. 35}$$

The new primal Lagrangian is minimised with respect to $\boldsymbol{w}$, $\boldsymbol{b}$ and ξ_i and maximised with respect to α where $\boldsymbol{a_i} \geq 0\ and\ \boldsymbol{\mu_i} \geq 0\ \forall_i$. Partially differentiating the new primal Lagrangian, equations (3. 26) and (3. 27) remain the same; however, for $\frac{\partial \boldsymbol{L_p}}{\partial \xi_i}$ we obtain:

$$\frac{\boldsymbol{\partial L_p}}{\boldsymbol{\partial \xi_i}} = \mathbf{0} \Rightarrow \boldsymbol{C} = \boldsymbol{\alpha_i} + \boldsymbol{\mu_i} \tag{3. 36}$$

The new dual Lagrangian ($\boldsymbol{L_D}$) has the same form as the one described in (3. 28) but with new constraints, where $0 \leq \boldsymbol{a_i} \leq C\ \forall_i$. The training data that is non-linear, a Kernel Trick is then employed. The transformation from a non-linear space to a higher-dimension feature space is illustrated in Fig. 3. 17 by $\boldsymbol{x} \mapsto \Phi(\boldsymbol{x})$.

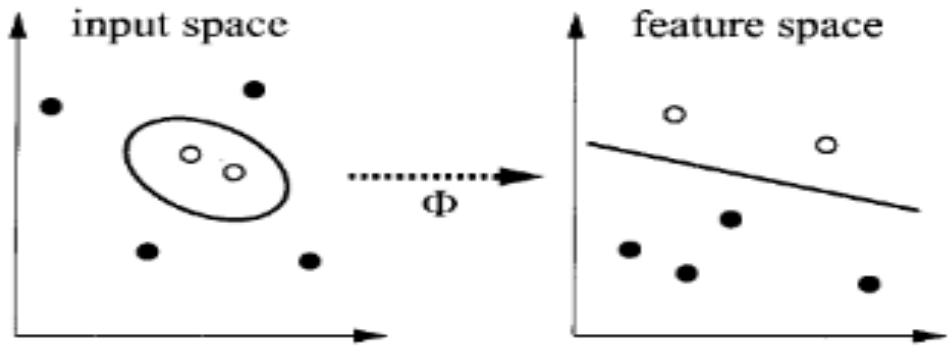


Fig. 3. 17: Non-linear data points mapped to linear separable space [82]

The Kernel function in (3. 37) is then substituted into (3. 32) and (3. 33) then forms:

$$\boldsymbol{K(x \cdot x_i) = \left(\Phi(x).\Phi(x_i)\right)} \tag{3. 37}$$

$$\boldsymbol{w \cdot K(x \cdot x_i) + b \geq +1 - \xi_i} \qquad \text{for } y_i = +1 \tag{3. 38}$$

$$\boldsymbol{w \cdot K(x \cdot x_i) + b \geq -1 + \xi_i} \qquad \text{for } y_i = -1 \tag{3. 39}$$

$$\boldsymbol{y_i(w \cdot K(x \cdot x_i) + b) - 1 + \xi_i \geq 0 \; \forall_i} \tag{3. 40}$$

In [81, 82], the authors describe several kernel functions that perform differently from non-linear SVMs. The main kernels are Linear (3. 41), Polynomial (3. 42), Radial Basis Function (RBF) (3. 43) and Sigmoid (3. 44):

$$\boldsymbol{K(x \cdot x_i) = (x \cdot x_i)} \tag{3. 41}$$

$$\boldsymbol{K(x \cdot x_i) = (x \cdot x_i)} \tag{3. 42}$$

$$\boldsymbol{K(x \cdot x_i) = e^{-(\|x \cdot x_i\|^2 / 2.\sigma^2)}} \tag{3. 43}$$

$$K(x \cdot x_i) = tanh(kx \cdot x_i - \delta) \tag{3. 44}$$

Kim et al. [56] indicated that SVM classification includes limitations in speed and size during training as well as the testing phase of the algorithm. The authors further suggested that selecting the kernel and its function parameters is a disadvantage. Therefore, considering the time and space complexity of the model, a linear SVM Kernel was selected [76]. The kernel in equation (3. 41) results in the SVM classifier:

$$f(x) = sign\left[\sum_{i=1}^{L} a_i y_i\big(K(x \cdot x_i)\big) + b\right] \tag{3. 45}$$

SVM classification is best suited for two classes; therefore, error-correcting outputs code (ECOC) is employed to reduce the problem of a multiclass classification to a binary classification problem. The implementation of ECOC seems unnecessary since there are two classes (open and close markers); however, considering an introduction of a third class (invalid marker), it becomes necessary. The invalid data is introduced as a third class that ensures that the classifier can distinguish between valid and invalid markers once trained with the training data. The ECOC adopts a one-versus-one coding design since the main classes are open or close {-1; +1} it ignores the rest while it exhausts all the class pair assignments.

Table 3. 3: ECOC coding design

Class	**Learner 1**	**Learner 2**	**Learner 3**
Open	1	1	0
Close	-1	0	1
Invalid	0	-1	-1

For the above classification model to be developed, the ECOC follows a few steps:

1. Learner 1 trains on observations in Open or Close classes: it treats Open classes as positives while Close classes as negatives.
2. The other class(s) are also trained similarly.
3. The decoding scheme utilises loss g, M is the coding design matrix with $\boldsymbol{m_{kl}}$ and s_l elements being the predicted classification score for the positive learner l.
4. The result of this is a new observation for the class ($\hat{k}$) that reduces the aggregation losses of the *L* binary learners (Refer to equation (3. 46)).

The SVM classifier then incorporates the ECOC model to address the multiclass problem and improve classification accuracy [76].

$$\widehat{\boldsymbol{k}} = \boldsymbol{argmin(k)} \sum_{\boldsymbol{l=1}}^{\boldsymbol{L}} |\boldsymbol{m_{kl}}| \boldsymbol{g(m_{kl}, s_l)} / \sum_{\boldsymbol{l=1}}^{\boldsymbol{L}} |\boldsymbol{m_{kl}}| \tag{3. 46}$$

A4) Algorithm for Feature Classification Using Linear SVM (LSVM):

Algorithm 4 below are the steps taken to train the LSVM model by employing BoF obtained from the HoG features.

Algorithm 4: Feature classification using SVM

Input: Training and Validation BoF

Output: Class label in each BoF

1. Train one vs one Linear SVM classifier.
2. **for** any classes {-1, + 1} **do**
3. Using a kernel function, use equation (3. 41) to map training BoF to higher space.
4. Use equation (3. 35) to obtain the optimal hyperplane.
5. **end for**
6. **for** each feature vector in the validation dataset **do**
7. With the majority votes, assign to the class label.
8. **end for**
9. **Measure the accuracy of the trained model:**
10. **for** each classified/predicted image (*I*) in the dataset, **do**
11. Compare image (*I*) with ground truth image (*Ig*)
12. Compute the similarity score and concatenate it into one variable.
13. **end for**
14. Calculate the overall accuracy of the trained model by finding the mean score.
15. Repeat steps 10 - 14 for test validation.

Fig. 3. 18: Algorithm 4

(c) Classification of RoIs Using Decision Tree (DT):

The Iterative Dichotomiser 3 (ID3) and the Classification And Regression Tree (CART) are the two main DT learning algorithms [83]. In the ID3 an entropy criterion and information gain are employed to grow the tree. Entropy measures the amount of uncertainty in the (data) set ***R***. Information gain measures the difference in entropy from before to after the set. CART employs a Gini impurity method: it measures a randomly chosen element from the set ***R*** on how often would it be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset. Fig. 3. 19 illustrates the basic structure of a DT classifier with root, internal and leaf nodes.

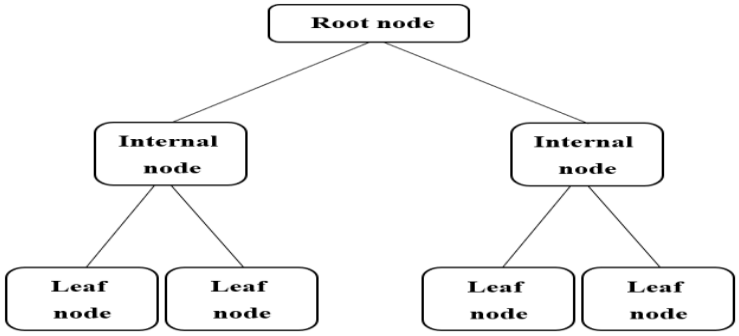


Fig. 3. 19: DT classifier structure [83]

When an ID3 is employed in a DT classifier, an entropy criterion is used: equation (3. 47) defines the entropy criterion. Where $\boldsymbol{H(R)}$ denotes the entropy criterion with $\boldsymbol{p_k}$ being the proportion of objects of the class ($\boldsymbol{k}$) in set/node $\boldsymbol{R}$, and $\boldsymbol{K}$ describes the number of classes. The information gain can therefore be obtained from equation (3. 47) as the difference between the entropy calculated before and the entropy calculated after.

$$H(R) = -\sum_{k=1}^{K} p_k \, log \, p_k \tag{3. 47}$$

A DT classifier that employs a CART algorithm also mentioned in the literature, uses a Gini impurity also denoted as $\boldsymbol{H(R)}$**.** Equation (3. 48) describes the mathematical expression that governs a DT classifier that employs a CART algorithm.

$$H(R) = -\sum_{k=1}^{K} p_k \, (1 - p_k) \tag{3. 48}$$

$$= 1 - \sum_{k=1}^{K} {p^2}_k$$

A5) Algorithm for a DT Classifier:

Algorithm 5 illustrated in Fig. 3. 20 summarises steps taken in the DT classifier employing ID3 or CART algorithm.

Algorithm 5: DT classifier
1. Set the data set ***R*** as the *initial root node*. 2. **for** every data set ***R*:** 3. Calculate $H(R)$ in unused attribute (***A***) [equ. (3. 47) or (3. 48)] 4. **end** 5. **if** $A < H(R)$: *(new root node)***:** 6. Split data set ***R*** into subsets *(internal nodes)* 7. Make a decision tree node *(leaf nodes)* 8. Recurse on each unused subset 9. **end**

Fig. 3. 20: Algorithm 5

(d) Classification of RoIs Using Convolutional Neural Network (CNN):

Fig. 3. 21 depicts the basic construction layers involved in a CNN network. The structure is divided into an input image, convolutional layer, ReLU layer, pooling layer and fully connected layer.

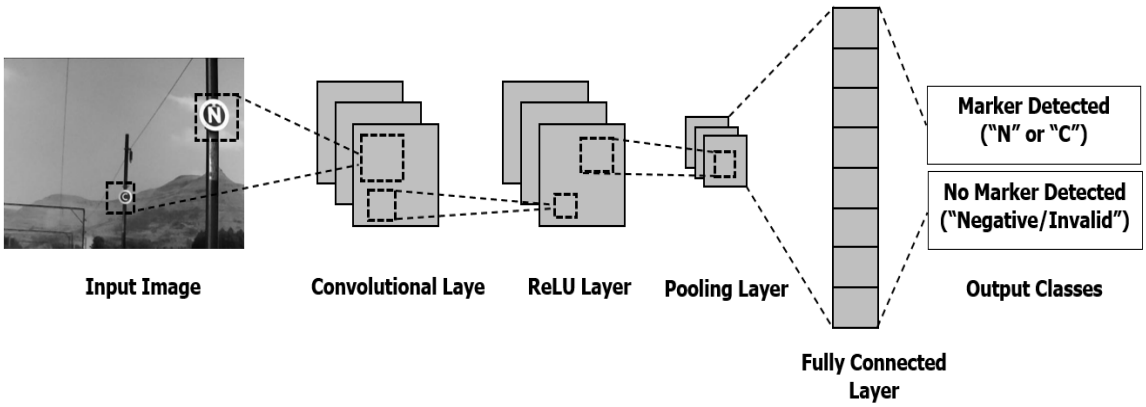


Fig. 3. 21: CNN classifier structure

The input images (RoIs) are firstly resized and then convolved with kernel/filters ($\boldsymbol{k}$) resulting in feature map $\boldsymbol{F(map)}$ defined in equation (3. 49). Where the image input size is $\boldsymbol{m}$, kernel size is $\boldsymbol{n}$, the padding is $\boldsymbol{p}$, and the stride is denoted by $\boldsymbol{S}$ in the equation. The ReLU activation function ($\boldsymbol{f}$) in equation (3. 50) models a neuron's output as a function of the input ($\boldsymbol{x}$) with an activation threshold at zero. Equation (3. 51) then outputs new feature maps after pooling which are then used to create a fully connected layer. In between the fully connected layer and output classes, a loss function is employed to calculate the predicted error. Commonly, a Softmax function is used [84].

$$\boldsymbol{F(map)_{conv}} = \frac{\boldsymbol{m} - \boldsymbol{n} + \boldsymbol{2p}}{\boldsymbol{S}} + \boldsymbol{1} \tag{3. 49}$$

$$\boldsymbol{f(x)} = \boldsymbol{max(0, x)} \tag{3. 50}$$

$$\boldsymbol{F(map)_{pool}} = \frac{\boldsymbol{m} - \boldsymbol{n}}{\boldsymbol{S}} + \boldsymbol{1} \tag{3. 51}$$

A6) Algorithm for a CNN Classifier:

Algorithm 6 in Fig. 3. 22 is a concise algorithm of the CNN network implementation which is mainly governed by equations (3. 49) - (3. 51).

Algorithm 6: CNN classifier
1. Resize RoIs to **m*m*r**. 2. **for** every resized RoI **:** 3. Compute $F(map)_{conv}$ using [equ. (3. 49)]. 4. Compute $f(x)$ using [equ. (3. 50)]. 5. Compute $F(map)_{pool}$ using [equ. (3. 51)]. 6. **end** 7. Compute a fully connected layer. 8. Classify RoIs.

Fig. 3. 22: Algorithm 6

(e) Classification of RoIs Using Discriminant Analysis (DA):

In [85], discuss the structure of a DA classifier which is illustrated in Fig. 3. 23. The latter comprises the input layer, discriminant functions and the maximum selector. The input has a vector or a set of *m* features that represent one point in m-dimensional space called pattern space (R^m). The Discimanant Functions {f1, f2,...,fc} where c is the number of classes is used to determine the decision boundaries (Sij) and decision regions for each class label. Therefore, different regions (wi, i=1, 2, ...,c) are formed by Sij discriminating between different classes. The maximum selector can be considered as the discriminating process of a DA classifier where the decision boundaries and decision regions are used to selectively output a class label.

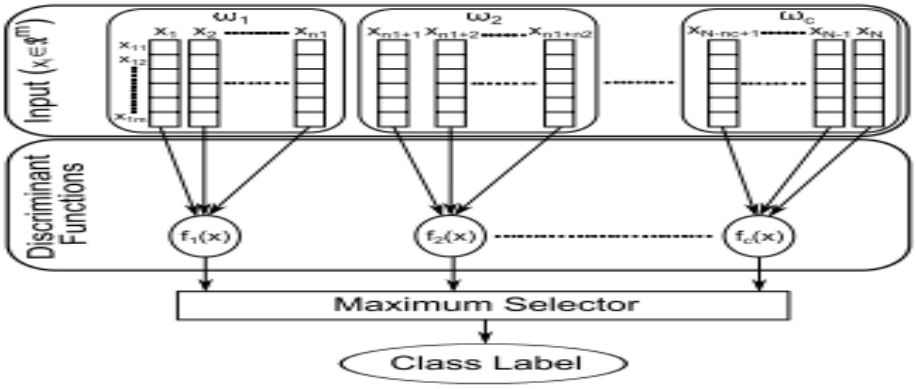


Fig. 3. 23: DA classifier structure [85]

The predicted classification output of a multi-class dataset is described in equation (3. 52) below [86]. The posterior probability denoted $\widehat{\boldsymbol{P}}(\boldsymbol{k}|\boldsymbol{x})$ is defined as the product of prior probability $\widehat{\boldsymbol{P}}(\boldsymbol{k})$ and multivariant normal density $\widehat{\boldsymbol{P}}(\boldsymbol{x}|\boldsymbol{k})$. While $\boldsymbol{C}(\boldsymbol{y}|\boldsymbol{k})$ is the cost of classifying an observation as $\boldsymbol{y}$ when class $\boldsymbol{k}$ is its true class. The number of classes ($\boldsymbol{K}$) gives range of $\boldsymbol{y}$, where {$\boldsymbol{y=1,\ldots,\ K}$}. Equation (3. 53) defines the multivariant normal density or $\widehat{\boldsymbol{P}}(\boldsymbol{x}|\boldsymbol{k})$ with mean $\boldsymbol{\mu_k}$ of ***1-by-d*** and convariance $\sum_{\boldsymbol{k}}$ of ***d-by-d*** at ***1-by-d*** point **x** [85, 86].

$$\hat{\boldsymbol{y}} = \boldsymbol{arg(min)} \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \widehat{\boldsymbol{P}}(\boldsymbol{k}|\boldsymbol{x})\, \boldsymbol{C}(\boldsymbol{y}|\boldsymbol{k}) \tag{3. 52}$$

$$\boldsymbol{P}(\boldsymbol{x}|\boldsymbol{k}) = \frac{\mathbf{1}}{((\mathbf{2\pi})^{\boldsymbol{d}}|\sum_{\boldsymbol{k}}|)^{\mathbf{1/2}}}\, \boldsymbol{e}^{\left(-\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{x}-\boldsymbol{\mu_k})\,\Sigma_{\boldsymbol{k}}^{-\mathbf{1}}(\boldsymbol{x}-\boldsymbol{\mu_k})^{\boldsymbol{T}}\right)} \tag{3. 53}$$

A7) Algorithm for a DA Classifier:

Algorithm 7 presents a generic DA classifier employing an LDA and QDA algorithm [85].

Algorithm 7: DA classifier
1. Spilt dataset into training and testing. 2. **if** a dataset has high dimensions: 3. Employ Regularised LDA: 4. Regularisation parameter $(1 > \eta > 0)$ or, 5. Employ Subspace method: 6. Principal Component Analysis (PCA). 7. **end** 8. Train model using the training data set [equ. (3. 53)]. 9. Predict the class of the test data [equ. (3. 52)].

Fig. 3. 24: Algorithm 7

(f) Classification of RoIs Using Naïve Bayes:

The Naive Bayes algorithm is mainly based on Bayes' Theorem which results in a posterior probability{P($\alpha|\beta$)}. The theorem defines P($\alpha|\beta$) as the inner product of prior probability{P(α)} and prior probability of tuple {P(β)} with product of P(α) divisable by P(β). A Gaussian distribution with a mean, standard and Gaussian Density Function is applied to obtain a Prediction [87].

$$\boldsymbol{\mu = \frac{1}{K}\sum_{k=1}^{K} \alpha_k} \tag{3. 54}$$

$$\boldsymbol{\sigma = \frac{1}{(K-1)}\sum_{k=1}^{K} \sqrt{(\alpha_k - \mu)}} \tag{3. 55}$$

$$\boldsymbol{g(a, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}} \tag{3. 56}$$

$$\boldsymbol{P(a_K|\gamma_k) = g(\alpha_k, \mu_{\gamma_k}, \sigma_{\gamma_k})} \tag{3. 57}$$

Equations (3. 54) - (3. 57) are the mean ($\boldsymbol{\mu}$), standard deviation ($\boldsymbol{\sigma}$), Gaussian density function $\boldsymbol{g(a, \mu, \sigma)}$ and a prediction $\boldsymbol{P(a_K|\gamma_k)}$ respectively.

A8) Algorithm for a Naïve Bayes Classifier:

Algorithm 8 shown in Fig. 3. 25 presents a Naïve Bayes classifier, summarised into seven concise steps

Algorithm 8: Naïve Bayes classifier
1. Spilt dataset into training and testing.
2. Separate data set by class.
3. Calculate mean [equ. (3. 54)].
4. Calculate standard deviation [equ. (3. 55)].
5. Calculate Gaussian density function [equ. (3. 56)].
6. Calculate class probability.
7. Predict [equ. (3. 57)].

Fig. 3. 25: Algorithm 8

(g) Classification of RoIs Using K-NN:

The K-NN computes the distance of an unlabelled object to all the labelled objects within the training set. Furthermore, class labels are then determined with the majority votes by considering the weightings of the distances. However, K-NN can be defined by the selection of its K-value and by computing the Euclidean distance [88].

$$\boldsymbol{d(x,y) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \quad (3.\ 58)$$

Since the K-value of the K-NN algorithm is selected; the research discusses the Euclidean distance $(\boldsymbol{d})$ in equation (3. 58) as it is commonly used [88]. Considering $(\boldsymbol{x_i},\ \boldsymbol{x_j}, \boldsymbol{y_i}, \boldsymbol{y_j}) \in \boldsymbol{D}$ as the training set where $\boldsymbol{i, j=1\ldots., l}$: and $(\boldsymbol{x}, \boldsymbol{y})$ are the feature points.

A9) Algorithm for a KNN Classifier:

Algorithm 9 in Fig. 3. 26 is also a concise algorithm of the K-NN classifier implementation.

Algorithm: K-NN classifier
1. Select K-value. 2. Calculate Euclidean distance for ***K*** number of neighbours [equ. (3. 58)]. 3. Calculate K-NN as per the calculated Euclidean distance. 4. Count the number of similar data points. 5. Predict.

Fig. 3. 26: Algorithm 9

(h) Classification of RoIs Using AdaBoost:

The AdaBoost algorithm is an ensembled classifier which trains multiple classifiers to improve classification accuracy. The literature [57] covered the DT and SVM, furthermore, the latter classifiers are covered in Subsections 3.4.4(b) and (c); therefore, since these classifiers have already been covered, only the performance of AdaBoost will be discussed in the following chapters.

3.4.5 Evaluation Methods

The preceding sections of this chapter have provided the mathematical approach and algorithms used for image pre-processing, feature extraction, segmentation, and classification. Evaluation techniques measure the model's performance based on these algorithms.

(a) Confusion Matrix:

In Table 3. 4, a confusion matrix is employed to visualise the predicted classes against the actual classes. The number of correctly recognised positive predictions is defined as **True positives,** and the number of correctly recognised negative predictions is defined as **True negatives.** Those not recognised as positive predictions are **False negatives**, while the number of those not recognised as negative predictions are **False positives.** The F1-measure then

computes the overall accuracy of the model using equation (3. 61). Equations (3. 59)(3. 47) and (3. 60) are used to derive the equation for F1-measure.

Table 3. 4: Confusion matrix

	Predicted positive	**Predicted negative**
Actual positive	True positives ($\boldsymbol{T_{pi}}$)	False negatives ($\boldsymbol{F_{ni}}$)
Actual negative	False positives ($\boldsymbol{F_{pi}}$)	True negatives ($\boldsymbol{T_{ni}}$)

$$\boldsymbol{rc} = \left(\frac{\sum_{i=1}^{L} \boldsymbol{T_{pi}}}{\sum_{i=1}^{L} \boldsymbol{T_{pi}} + \boldsymbol{F_{ni}}}\right) \tag{3. 59}$$

$$\boldsymbol{p} = \left(\frac{\sum_{i=1}^{L} \boldsymbol{T_{pi}}}{\sum_{i=1}^{L} \boldsymbol{T_{pi}} + \boldsymbol{F_{pi}}}\right) \tag{3. 60}$$

$$\boldsymbol{F_1} = \boldsymbol{2} * \left(\frac{\boldsymbol{p} * \boldsymbol{rc}}{\boldsymbol{p} + \boldsymbol{rc}}\right) \tag{3. 61}$$

Where:

- $\boldsymbol{rc}$: Recall
- $\boldsymbol{p}$: Precision
- $\boldsymbol{F_1}$: F1-measure

(b) K-Fold Cross-Validation:

In [76], a holdout method was used to perform cross-validation: data is separated into two datasets, namely the training and test data set. The training data set is used to train the model, while the test data set is used to evaluate the model. However, this method increases the possibility of business with limited data. The *K*-fold cross-validation technique is then used to improve the holdout method. Similarly, the dataset is split into training and testing sets;

however, the dataset is randomly divided into *K* number of folds. Hence, the holdout method is improved, and biasness is reduced. When the model is evaluated, the first *K*-fold is removed and trained on the remaining *K*-1 folds. Then the second *K*-fold is removed from the dataset; the model is evaluated with the first and last *K*-2 folds. This process is repeated *K* number of times with each result recorded. In this thesis, 2-fold cross-validation is used to evaluate the performance of the neutral section marker classification model.

3.5 Summary

This methodology chapter focused on outlining the details of the research design and processes undertaken to achieve results. The research design highlighted and justified the adoption of a positivism research philosophy. The methodology employed was quantitative, where the data was represented empirically. The research approach focused on the data collection and analysis of methods employed, which gave an in-depth understanding of the research. The fundamental aspect of the research process detailed the research instruments, data collection and methods employed. For instance, in the data collection, it is outlined to the reader the methods employed in collecting the data: training and testing data being crucial in the development of the model.

Furthermore, this allows the reader to replicate the proposed model if one may require validating the results obtained. For the model to achieve a high accuracy classification rate, effective, dependent, and independent variables were also listed. Differentiating between the two allows the researcher to effectively adjust the independent variables, subsequently adjusting the dependent variables that affect the model's accuracy. The chapter concludes by giving a detailed process of the methods employed in a step-by-step approach (algorithms and mathematical equations) on localising, segmenting, and classifying both markers. Chapter 4

presents the results of the developed model, where well-known statistical methods were used to evaluate the model's accuracy.

Chapter 4: Results

4.1 Introduction

This chapter aims to provide a detailed presentation of the results obtained from conducting the experiments based on the methods presented in Chapter 3. This chapter is structured as follows: First, the dataset used during the experiments is described in Section 4.2. This section follows the results obtained by the image pre-processing and marker extraction in Section 4.3 and 4.4, respectively. Then, the classification results obtained from the Histogram Oriented Gradient (HoG) descriptor are presented in Section 4.5. Furthermore, machine learning classification techniques such as Support Vector Machines (SVM), K-Nearest Neighbour (K-NN), Convolutional Neural Network (CNN), Decision Tree (DT), Naïve Bayes, Discriminant Analysis (DA) and AdaBoost classifiers are employed to compare the performance of HoG at different parameters. Section 4.6 empirically compares the obtained results to select the optimal HoG parameter and classifier. Section 4.8 concludes the chapter.

4.2 Dataset Description

Chapter 3 described the methods and instruments used to collect the dataset: images used to conduct the experiments were taken in one of Transnet Freight Rails sites. A total of 550 neutral section marker images grouped into two classes, namely open and close markers, were collected. The split consists of 422 training images and 128 testing images; however, the addition of 104 negative images adds up to 232 test images. Fig. 4. 1 illustrates the initial design of the makers using computer-aided drawing (CAD). Fig. 4. 2 and Fig. 4. 3 depict some sample images from each class.

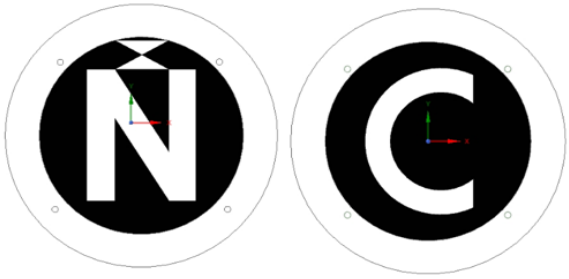


Fig. 4. 1: CAD designed. (a) Open and (b) Close markers

Fig. 4. 2: Actual onsite. (a) Open and (b) Close positive markers

Fig. 4. 3: Negative images

4.3 Image Pre-processing

The Neutral Section (NS) marker images acquired are influenced by climate conditions such as heat intensity from the sun and vibrations from the rail, which adds noise. Furthermore, the format in which the images were captured or RGB: converting each image into greyscale was employed to reduce the computational cost and average some noise. Fig. 4. 4 shows the sample result of each RGB to greyscale image conversion as defined in equation (3. 1). Once each image has been converted, filters are employed to denoise each image before further processing. The filter employed to denoise the images in the training and testing dataset was a

bilateral filter. The bilateral filter preserves edges while simultaneously removing noise and smoothing each image.

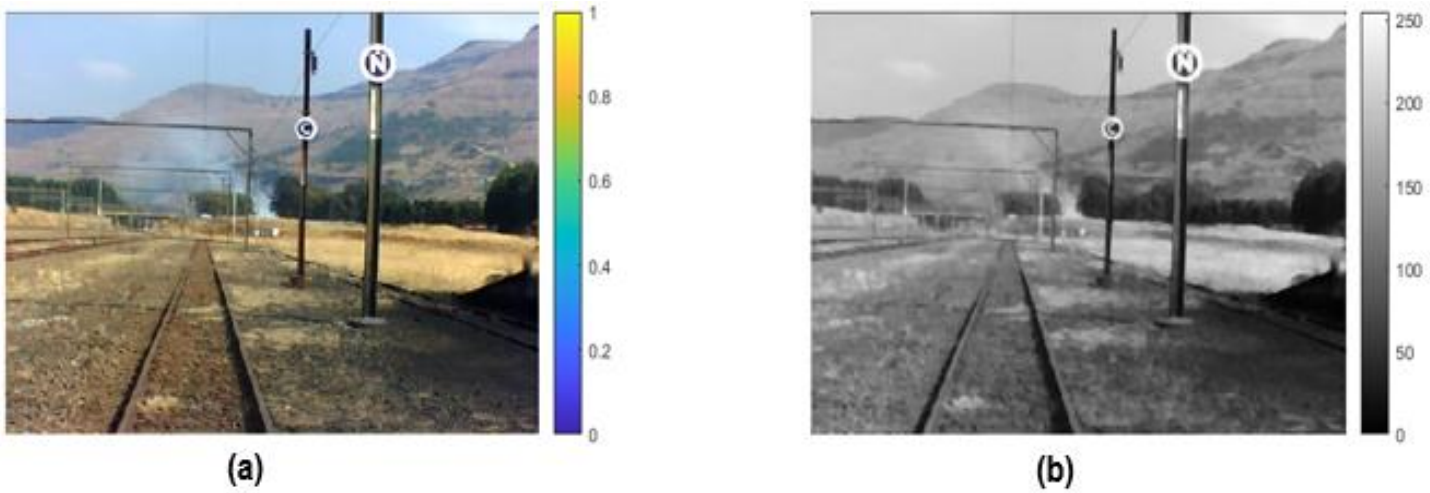


Fig. 4. 4: Image conversion. (a) RGB and (b) Greyscale images

4.3.1 Parameter Evaluation

Parameters define the effectiveness of a bilateral filter, σ_s and σ_r in equations (3. 2) and (3. 3). These parameters define the spatial and the range weight, respectively, which control an image's denoising, smoothing and contour preservation. Fig. 4. 5 shows a correlation between the original greyscale image and its copy image with additive noise. This approach sets a baseline to compare the results of bilateral filtered images. Before employing a bilateral filter on each image, the correlation between Fig. 4. 5(a) and Fig. 4. 5(b) was 99.3% as our baseline. The 99.3% baseline performance is important in that it gives an overview of how accurately the original greyscale image correlates with the same image but with noise added. The benefit of which is to select the best-performing noise filtering algorithm (one that reaches accuracy above or close to the baseline percentage after noise removal).

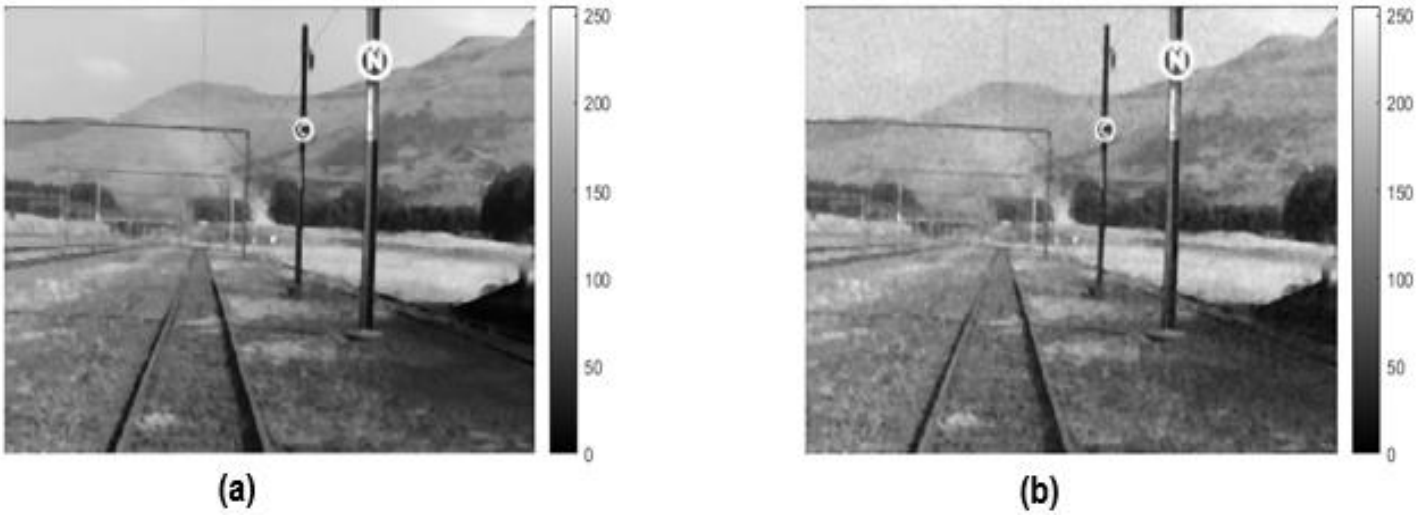


Fig. 4. 5: Correlation of greyscale images. (a) Original and (b) Noisy

The evaluation of a bilateral filter is done by selecting different σ_s and σ_r parameters [89], illustrated in Fig. 4. 6. Tomasi and Manduchi [89] employed a 3-by-4 ($\sigma_s = [1\ 3\ 10]$; $\sigma_r = [10\ 30\ 100\ 300]$) parameter comparison, while the 5th element $\sigma_r = 650.25$ was motivated by default value of MATLAB "imbilatfilt" function.

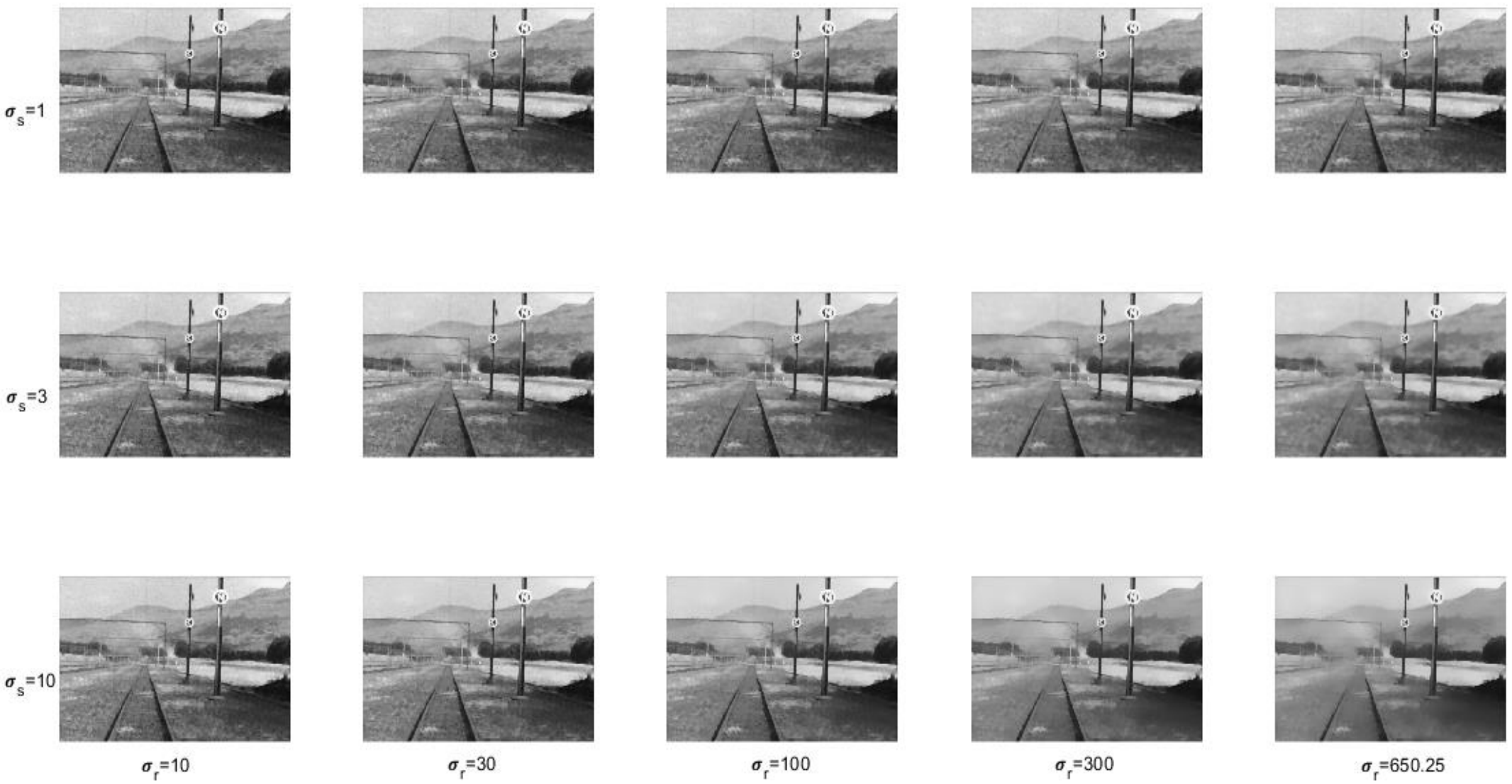
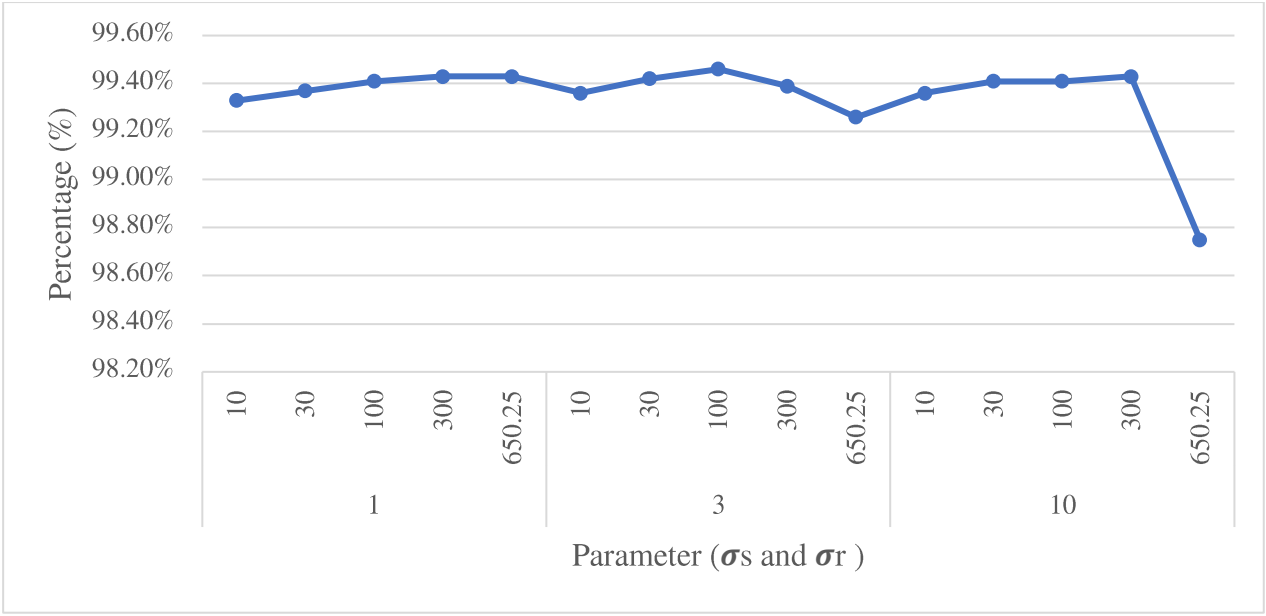


Fig. 4. 6: Parameters σ_s and σ_r

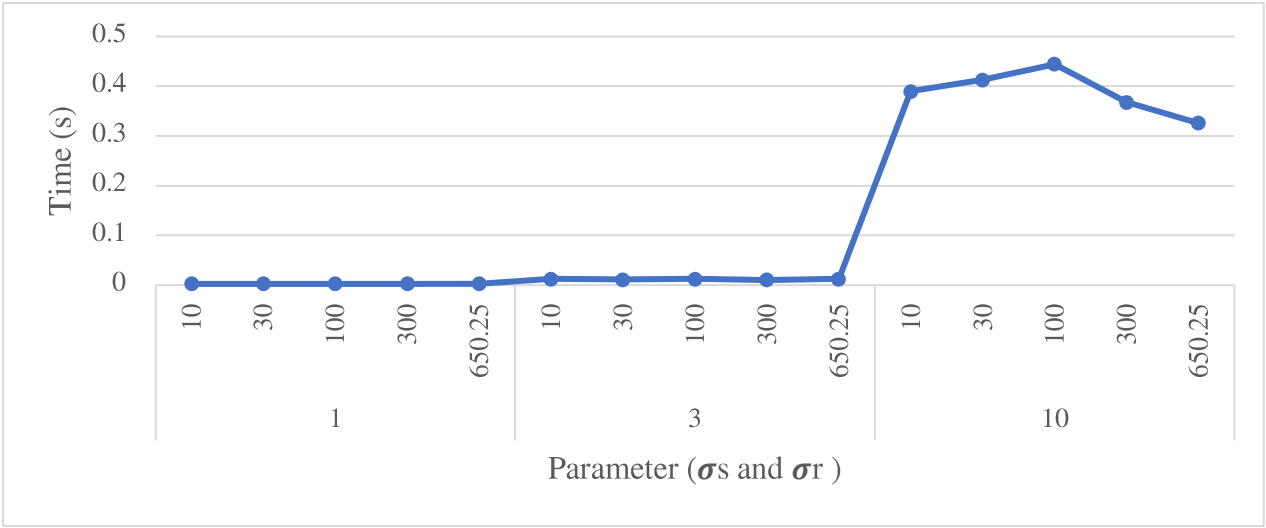
Table 4.1 illustrates the results of varying σ_s and σ_r parameters: there are two criteria's that were considered in selecting the optimal parameters for effectively filtering each image. The first is the correlation in percentage, and the second is time in seconds, while Fig. 4. 7 displays the results graphically.

Table 4.1: Results: Bilateral filtered image

Parameters		**Original vs Filtered Image Correlation (%)**	**Time (s)**
σ_s	σ_r		
1	10	99.33%	0.0027
	30	99.37%	0.00288
	100	99.41%	0.00275
	300	99.43%	0.00275
	650.25	99.43%	0.00262
3	10	99.36%	0.01237
	30	99.42%	0.01137
	100	99.46%	0.01276
	300	99.39%	0.01073
	650.25	99.26%	0.01276
10	10	99.36%	0.38952
	30	99.41%	0.41296
	100	99.41%	0.44407
	300	99.43%	0.36809
	650.25	98.75%	0.32618



(a)



(b)

Fig. 4. 7: Correlation of original greyscale and filtered images at different parameters. (a) Performance in % and (b) Computation cost

4.4 Marker Extraction

After the image pre-processing, the next step is to detect and extract the markers as the RoIs from the NS image background. The segmentation method employed in this research is the Circular Hough Transform (CHT) which detects the image regions by finding the circular shapes of each marker. The detection of markers is carried out by using algorithm 2 detailed in Subsection (A2) of the previous Chapter 3. The CHT is applied to each image in the dataset to obtain a parametric space for each RoI. First, the filtered greyscale image is segmented using the edge detection technique. The CHT employs a Sobel operator to remove most background artefacts, leaving foreground objects. Each RoI is extracted by calculating its contour boundaries from the parametric space. The contour boundaries form a rectangular bounding box superimposed on the original image, and this area is then cropped. The cropped image, therefore, represents either an open or closed marker as the RoI. Fig. 4. 8 depicts the steps implemented in segmenting RoIs; a sample image is shown in Fig. 4. 8(a) with no markers. Fig. 4. 8(b) demonstrates the Sobel edge detection as well as the CHT being employed, while Fig. 4. 8(c) and Fig. 4. 8(d) illustrate the results obtained without any RoI detected or segmented: both figures are similar and show no image which shows the algorithm was able to detect the absence of the RoIs within the image.

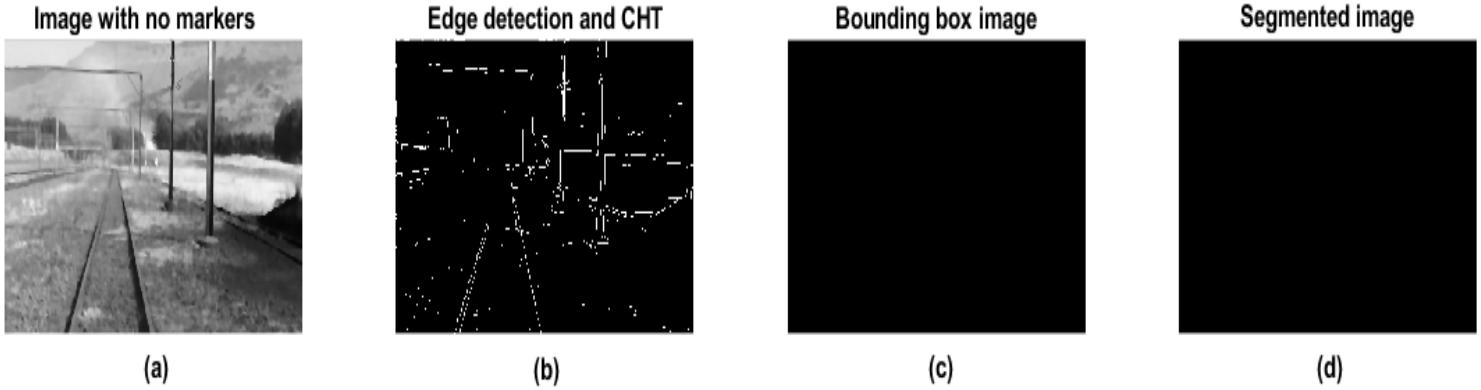


Fig. 4. 8: Marker extraction on an image with no markers

Fig. 4. 9 presents the results of marker extraction employing a Sobel operator, CHT and bounding box techniques. Fig. 4. 9(a, e, i) are sample filtered images captured in weather conditions such as sunny, cloudy, and dark, respectively. The remaining figures in the corresponding rows illustrate the output results of each extraction process. It is worth noting that the steps in Fig. 4. 9(c, g, k) undergo verification of each RoI detected before cropping each image. The verification of each RoI is done by employing a classification technique, details of which are described in Section 4.5. The former enables segmented images by filling each RoI with white pixel background (each pixel is 255 for 8-bit or 1 for binary image), as depicted in Fig. 4. 9(d, h, l). The output images from the latter allow the model's accuracy to be measured during training against the ground truth dataset.

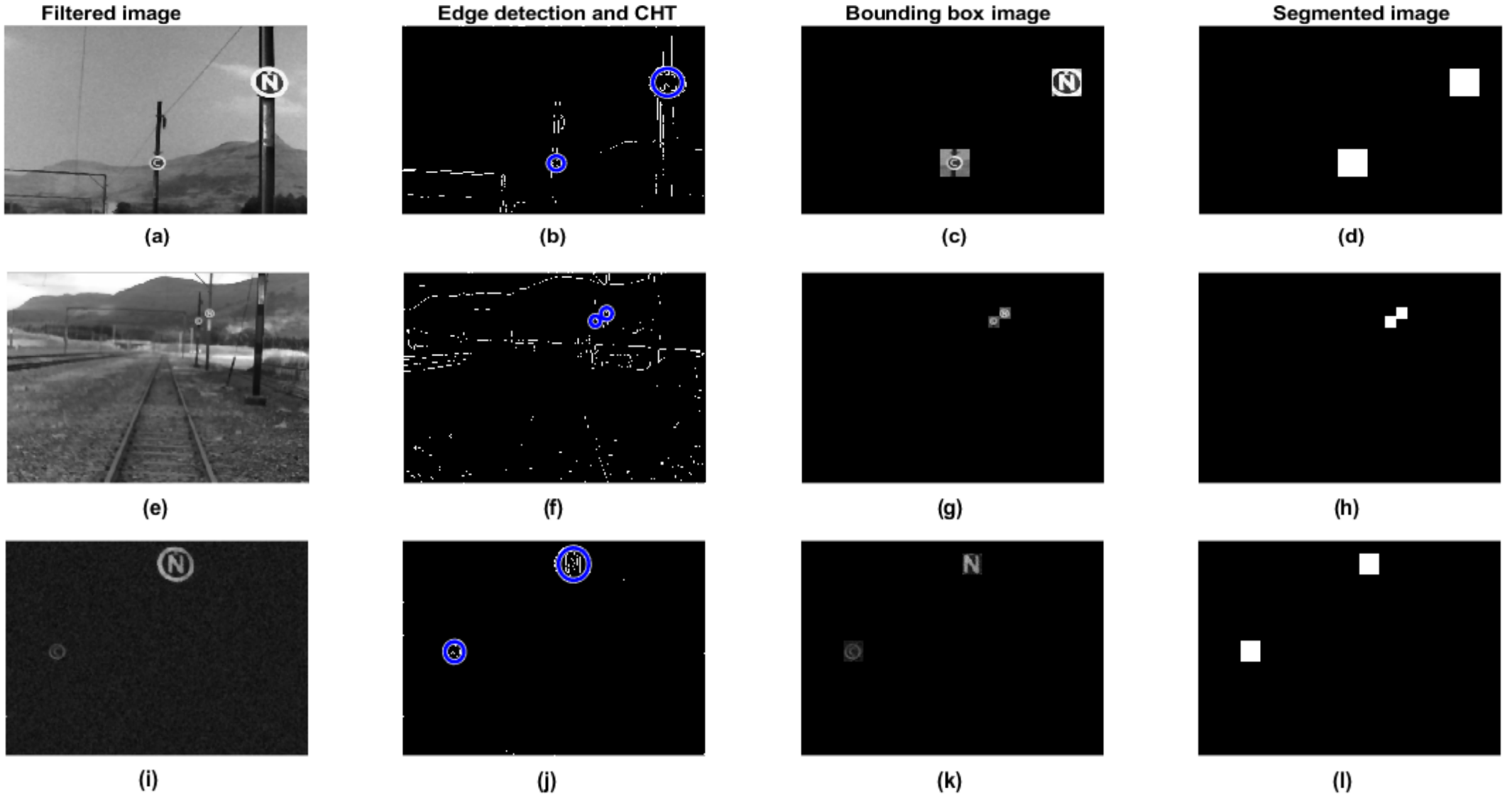


Fig. 4. 9: Marker extraction in different weather conditions

4.5 Feature Classification

The extracted markers or RoIs from the image background, HoG descriptor was employed as a feature extraction method. The extracted features were stored as a bag of features (BoF), a feature vector containing each image's features. As discussed in Subsection 3.43.4.4, the features were extracted from cropped images of 60*60 pixels.

These cropped images were extracted from 227 images (140 open, 49 close and 38 negative) taken from the training dataset. The training normalised feature vectors obtained were 227*7056 or 1 601 712 features based on a 4*4 cell-size that makes up a 9*1 matrix with a 4*4 block size. The 7056 defines the number of features extracted in one cropped image, while 227 is the number of cropped images used for extracting these features. The performance of the HoG feature extraction method was compared using seven machine learning classification algorithms, namely SVM (Linear, Quadratic and Cubic), K-NN, CNN, DT (ID3 and CART), DA (Linear and Quadratic), Naïve Bayes and AdaBoost. These classifiers are some of the state-of-the-art classifiers used in computer vision [90]. A cross-validation method was employed to validate each classification algorithm. This section is structured as follows; classification results from Linear SVM with [2 2] to [16 16] cell-sizes are presented in Subsection 4.5.1. Subsection 4.5.2 presents the results obtained from employing a Quadratic SVM, while Subsections 4.5.3 and 4.5.4 are the results obtained from K-NN and CNN, respectively, at a [4 4] cell-size. Subsections 4.5.5 - 4.5.11 summarises the results obtained from the study [90]. Subsections 4.5.1 - 4.5.4, used 84 test images [76]. While in Subsections 4.5.5 - 4.5.11 the test images were 83 and the training images 228: MATLAB Classification Learner application was employed and this slightly varied the initial data sets size [90].

4.5.1 Classification Using The Linear Support Vector Machine (LSVM)

The results in Table 4.2 - 4.9 are obtained from different cell-sizes that generate HoG features, while the SVM parameters chosen are σ_s=1 and σ_r= 650.25 (obtained from Table 4.1). The choice of selecting the LSVM as the primary classifier and its parameters are discussed in Chapter 5. In each table, the training set results are shown in a confusion matrix obtained from a 2-fold and 5-fold cross-validation (CV) at varying cell-sizes. The BoF size obtained from a [2 2] cell-size is 43MB, [4 4] is 10MB, [8 8] is 2MB, and a [16 16] cell-size is 211KB. Table 4.2, classifier results achieved a training accuracy of 83.7% (37 validation costs) and a test accuracy of 92.9% (6 test costs). Compared to the latter, Table 4.3 shows a training accuracy of 94.4% (13 validation costs) and a test accuracy of 91.7% (7 test costs). Table 4.4 achieved a training accuracy of 95.2% (11 validation costs) and a test accuracy of 97.6% (7 test costs) at a [4 4] cell-size. While Table 4.5 achieved a 96% training accuracy (9 validation costs) and 86.9% on test accuracy (11 test costs). In a [8 8] cell-size in Table 4.6, a training accuracy of 92.9% (16 validation costs) and a test accuracy of 97.2% (2 test costs) are achieved. Similarly, Table 4.7 resulted in a 96% training accuracy (9 validation costs) with a test accuracy of 90.5% (8 test costs). Table 4.8, with a [16 16] cell-size employing Linear SVM achieved 87.7% in training accuracy (28 validation costs) and 97.6% in test accuracy (2 test costs). Table 4.9 shows a training accuracy of 90.3% (22 validation costs) and a test accuracy of 94.1% (5 test costs).

Table 4.2: Confusion matrix from 2-fold CV, HoG at [2 2] cell-size and LSVM

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**36**	0	13
Negative (‘I’)	0	**16**	22
Open (‘N’)	1	1	**138**

Table 4.3: Confusion matrix from 5-fold CV, HoG at [2 2] cell-size and LSVM

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**40**	1	8
Negative (‘I’)	0	**34**	4
Open (‘N’)	0	0	**140**

Table 4.4: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and LSVM

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**41**	1	7
Negative (‘I’)	0	**35**	3
Open (‘N’)	0	0	**140**

Table 4.5: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and LSVM

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**44**	0	5
Negative (‘I’)	0	**34**	4
Open (‘N’)	0	0	**140**

Table 4.6: Confusion matrix from 2-fold CV, HoG at [8 8] cell-size and LSVM

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**42**	0	7
Negative (‘I’)	0	**32**	6
Open (‘N’)	2	1	**137**

Table 4.7: Confusion matrix from 5-fold CV, HoG at [8 8] cell-size and LSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**47**	0	2
Negative ('I')	0	**31**	7
Open ('N')	0	0	**140**

Table 4.8: Confusion matrix from 2-fold CV, HoG at [16 16] cell-size and LSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**40**	0	9
Negative ('I')	2	**23**	13
Open ('N')	3	1	**136**

Table 4.9: Confusion matrix from 5-fold CV, HoG at [16 16] cell-size and LSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**43**	0	6
Negative ('I')	1	**28**	9
Open ('N')	6	0	**134**

4.5.2 Classification Using The Quadratic Support Vector Machine (QSVM)

Table 4.10 and Table 4.11 are the confusion matrix of the training set obtained from a 2-fold and 5-fold CV at [4 4] cell-size. In Table 4.10, a 2-fold CV obtained a training accuracy of 94.7% (14 validation costs) and a test accuracy of 96.4% (3 test costs). In a 5-fold CV results are 96% (9 validation costs) for training accuracy and 86.9% for test accuracy (11 test costs), which is presented in Table 4.11.

Table 4.10: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and QSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**41**	1	8
Negative ('I')	0	**34**	4
Open ('N')	0	1	**140**

Table 4.11: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and QSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**44**	0	5
Negative ('I')	0	**34**	4
Open ('N')	0	0	**140**

4.5.3 Classification Using The K-Nearest Neighbour (K-NN)

A K-NN defines the K-value as the chosen number of neighbours. K-NN classifier performance is discussed in this subsection, where K= {1,3,5}, 2-fold and 5-fold CV are used with the HoG features at a [4 4] cell-size. On a 2-fold, K=1, in

Table **4.12**, a training accuracy of 84.6% (35 validation costs) and a test accuracy of 86.9% (11 test costs) are achieved. Table 4.13 at K=3, the K-NN achieved a training accuracy of 88.9% (25 validation costs) and a test accuracy of 84.5% (13 test costs). At K=5, Table 4.14 demonstrates 88.1% (26 validation costs) in training accuracy and 80.9% (16 test costs) in a test accuracy. On a 5-fold, with K=1, Table 4.15 shows an 88.9% (25 validation costs) training accuracy and test accuracy of 85.7% (12 test costs). At K=3, Table 4.16 illustrates an achieved training accuracy of 85% (35 validation costs) with a test accuracy of 85.7% (12 test costs). While at K=5 in Table 4.17, the K-NN achieved a training accuracy of 83.7% (37 validation costs) with a test accuracy of 77.1% (19 test costs).

Table 4.12: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and 1-NN

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**38**	10	1
Negative ('I')	0	**36**	2
Open ('N')	6	16	**118**

Table 4.13: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and 3-NN

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**44**	3	2
Negative ('I')	0	**37**	1
Open ('N')	6	13	**121**

Table 4.14: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and 5-NN

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**42**	5	2
Negative ('I')	0	**37**	1
Open ('N')	3	16	**121**

Table 4.15: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and 1-NN

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**40**	8	1
Negative ('I')	0	**35**	3
Open ('N')	0	13	**127**

Table 4.16: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and 3-NN

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**39**	9	1
Negative (‘I’)	0	**35**	4
Open (‘N’)	4	17	**119**

Table 4.17: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and 5-NN

	Close (‘C’)	Negative (‘I’)	Open (‘N’)
Close (‘C’)	**42**	7	0
Negative (‘I’)	0	**31**	7
Open (‘N’)	4	19	**117**

4.5.4 Classification Using The Convolutional Neural Network (CNN)

The results of a CNN performance with layers 1 and 2 set at 60 and 100, respectively, are presented in the below tables. The below tables are confusion matrix obtained from a 2-fold and a 5-fold CV with a [4 4] cell-size. Table 4.18 shows the results obtained from a 2-fold CV with 1-fully connected layer: the training set achieved a 97.4% and test a 91.6% accuracy. While in Table 4.19, a CNN with 2-fully connected layer obtained 90.3% accuracy during training and a 95.2% during testing. Table 4.20 and Table 4.21 are the results obtained from a 5-fold CV when employing a 1-fully connected layer and 2-fully connected layer, respectively. CNN classifier achieved 96.9% training accuracy with 92.8% for a test with a 1-full connected layer, while on a 2-fully connected layer, a 95.6% training accuracy and 96.4% for the test were obtained. In the tables below, the prefix *L* denotes the number of layers used in CNN.

Table 4.18: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and CNN (L=1)

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**47**	1	1
Negative ('I')	1	**37**	0
Open ('N')	0	3	**137**

Table 4.19: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and CNN (L=2)

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**43**	4	2
Negative ('I')	0	**35**	3
Open ('N')	3	10	**127**

Table 4.20: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and CNN (L=1)

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**47**	1	1
Negative ('I')	0	**36**	2
Open ('N')	0	3	**137**

Table 4.21: Confusion matrix from 5-fold CV, HoG at [4 4] cell-size and CNN (L=2)

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**47**	0	2
Negative ('I')	0	**37**	1
Open ('N')	2	5	**133**

4.5.5 Classification Using The Cubic Support Vector Machine (CSVM)

Table 4.22 is a confusion matrix of a CSVM classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 93% during training with 16 misclassifications. While 92.8% was achieved when the model was tested with new data and 6 misclassified.

Table 4.22: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and CSVM

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**51**	0	9
Negative ('I')	1	**30**	6
Open ('N')	0	0	**131**

4.5.6 Classification Using The Iterative Dichotomiser 3 (ID3)

Table 4.23 is a confusion matrix of an ID3 DT classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 74.1% during training with 59 misclassifications. While 77.1% was achieved when the model was tested with new data and 19 were misclassified.

Table 4.23: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and ID3 DT

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**42**	10	8
Negative ('I')	6	**25**	6
Open ('N')	11	18	**102**

4.5.7 Classification Using The Classification And Regression Tree (CART)

Table 4.24 is a confusion matrix of a CART DT classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 75.4% during training with 56 misclassifications. While 80.7% was achieved when the model was tested with new data, and 16 were misclassified.

Table 4.24: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and CART DT

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**43**	9	8
Negative ('I')	5	**24**	8
Open ('N')	11	15	**105**

4.5.8 Classification Using The Linear Discriminant Analysis (LDA)

Table 4.25 is a confusion matrix of an LDA classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 94.7% during training with 12 misclassifications. While 92.8% was achieved when the model was tested with new data, and 6 were misclassified.

Table 4.25: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and LDA

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**52**	3	5
Negative ('I')	1	**34**	2
Open ('N')	0	1	**130**

4.5.9 Classification Using The Quadratic Discriminant Analysis (QDA)

Table 4.26 is a confusion matrix of a QDA classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 85.1% during training with 34 misclassifications. While 81.9% was achieved when the model was tested with new data, and 15 were misclassified.

Table 4.26: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and QDA

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**41**	6	13
Negative ('I')	1	**36**	0
Open ('N')	0	14	**117**

4.5.10 Classification Using The Naïve Bayes

Table 4.27 is a confusion matrix of a Naïve Bayes classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 85.1% during training with 34 misclassifications. While 81.9% was achieved when the model was tested with new data, and 15 were misclassified.

Table 4.27: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and Naïve Bayes

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**41**	6	13
Negative ('I')	1	**36**	0
Open ('N')	0	14	**117**

4.5.11 Classification Using The AdaBoos Decision Tree (AdaBoost DT)

Table 4.28 is a confusion matrix of an AdaBoost DT classifier: a 2-fold CV with HoG at [4 4] cell-size was employed [90]. The classifier obtained an accuracy of 57.5% during training with 97 misclassifications. While 57.8% was achieved when the model was tested with new data, and 35 was misclassified.

Table 4.28: Confusion matrix from 2-fold CV, HoG at [4 4] cell-size and AdaBoost DT

	Close ('C')	Negative ('I')	Open ('N')
Close ('C')	**0**	0	60
Negative ('I')	0	**0**	37
Open ('N')	0	0	**131**

4.6 Overview Performance of each Classifier

Table 4.29 and Table 4.30 summarises the results obtained in Section 4.5 for each classifier, giving a performance overview at different parameters. Fig. 4. 10 and Fig. 4. 11 is a graphical representations of the summarised results, and a linear trendline is used to illustrate the performance trend over all the classifiers. It is worth noting that in Fig. 4. 10, on the x-axis, the K-NN and CNN, the cell-size used is a [4 4], but the K-values and layers are shown instead. The latter is done intentionally because each classifier's cell size is known and is standard, while the K-values and layers vary. Fig. 4. 11 used CNN with a 1-fully connected layer and the K-NN with K-values (2, 5, 10 and 20), but will HoG cell-size of [4 4] for all the classifiers. Each classifier is measured by comparing its performance at different parameters and then compared with the other classifiers.

Table 4.29: Performance overview of each classifier

Classifier	HoG Cell-size	σ_s	σ_r	K-Value	Layers_1	Layer_2	Training Accuracy	Test Accuracy	BoF size
LSVM(2-fold)	[2 2]	1	650.25	–	–	–	83.70%	92.90%	43MB
	[4 4]			–	–	–	**95.20%**	**97.60%**	**10MB**
	[8 8]			–	–	–	92.90%	97.20%	2MB
	[16 16]			–	–	–	87.70%	97.60%	211KB
LSVM(5-fold)	[2 2]	1	650.25	–	–	–	94.40%	91.70%	43MB
	[4 4]			–	–	–	96.00%	86.90%	10MB
	[8 8]			–	–	–	96.00%	90.50%	2MB
	[16 16]			–	–	–	90.30%	94.10%	211KB
QSVM(2-fold)	[4 4]	–	–	–	–	–	94.70%	96.40%	10MB
QSVM(5-fold)		–	–	–	–	–	96.00%	86.90%	
K-NN(2-fold)	[4 4]	–	–	1	–	–	84.60%	86.90%	10MB
		–	–	3	–	–	88.90%	84.50%	
		–	–	5	–	–	88.10%	80.90%	
K-NN(5-fold)		–	–	1	–	–	88.90%	85.70%	
		–	–	3	–	–	85.00%	85.70%	
		–	–	5	–	–	83.30%	77.10%	
CNN(2-fold)	[4 4]	–	–	–	60	–	97.40%	91.60%	10MB
		–	–	–	60	100	90.30%	95.20%	
CNN(5-fold)		–	–	–	60		96.90%	92.80%	
		–	–	–	60	100	95.60%	96.40%	

Table 4.30: Performance overview of each classifier: published research [90]

NO.	MIL CLASSIFIERS	TRAINING	TESTING	PREDICTION SPEED		NO.	MIL CLASSIFIERS	TRAINING	TESTING	PREDICTION SPEED
1	DT (CART)	75.40%	80.70%	72 obs/sec	6	**8**	CSVM	93.00%	92.80%	74 obs/sec
2	DT (ID3)	74.10%	77.10%	73 obs/sec	5	**9**	AdaBoost	57.50%	57.80%	68 obs/sec
3	LDA	94.70%	92.80%	78 obs/sec	2	**10**	CNN	90.80%	90.40%	82 obs/sec
4	QDA	85.10%	81.90%	71 obs/sec	7	**11**	K-NN(2)	82.00%	80.70%	13 obs/sec
5	Naïve Bayes	85.10%	81.90%	26 obs/sec	9	**12**	K-NN(5)	82.90%	86.70%	12 obs/sec
6	LSVM	93.40%	94.00%	75 obs/sec	3	**13**	K-NN(10)	84.60%	80.70%	14 obs/sec
7	QSVM	93.90%	94.00%	68 obs/sec	8	**14**	K-NN(20)	83.30%	90.40%	12 obs/sec

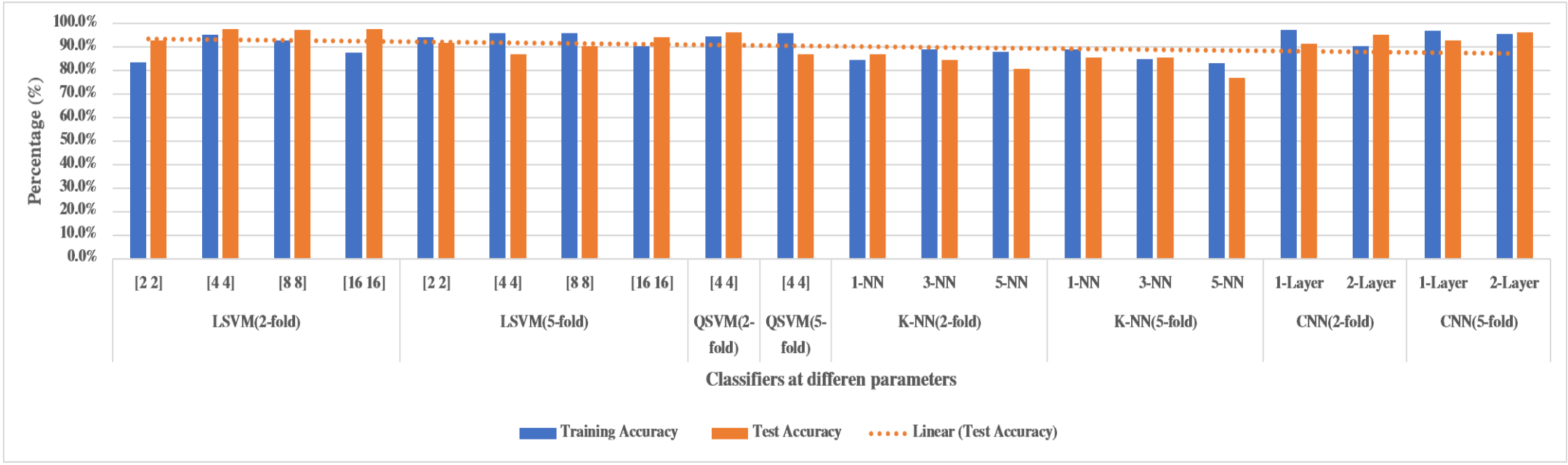


Fig. 4. 10: Graphical performance overview of each classifier

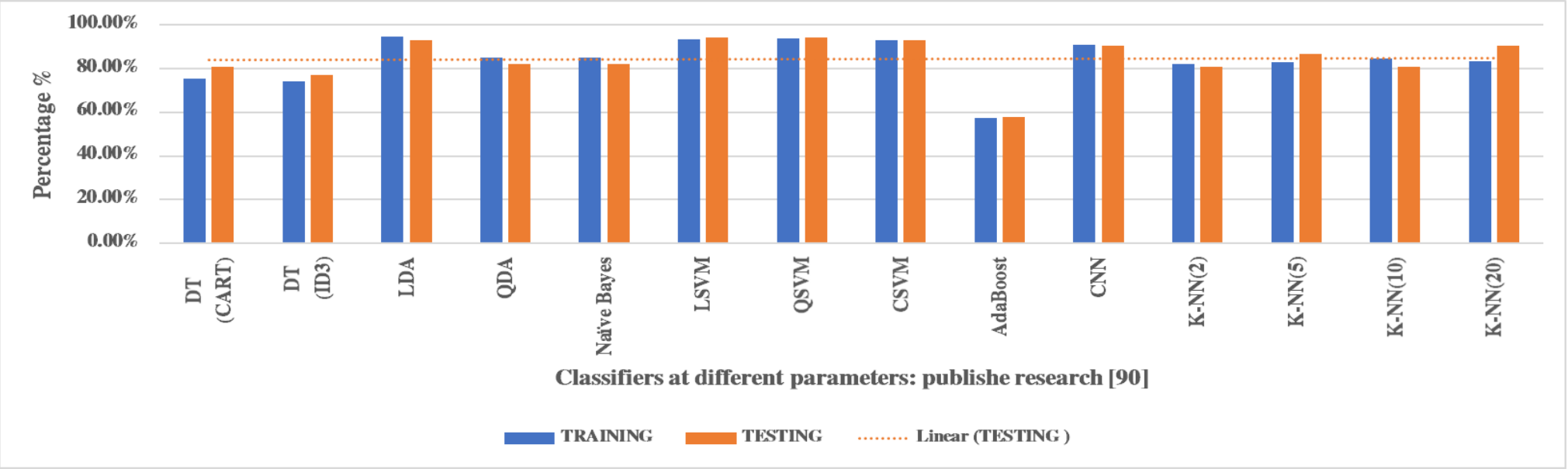


Fig. 4. 11: Graphical performance overview of each classifier: published research [90]

4.7 Measurement of Segmentation Accuracy

Once a classifier is selected based on the results in Fig. 4. 10 and Fig. 4. 11, the model's accuracy is obtained by employing equation (3. 49), where each segmented image is compared to the corresponding ground truth image. The ground truth images, as mentioned in Section 1.5 and Subsection 3.3.3(c), are manually created from the training images, and an F1-measure is employed to measure the segmentation accuracy of the model. Fig. 4. 9(d, h, l) are samples of images with segmented RoIs obtained from the model during training. These are the predicted RoIs, which are then compared to the ground truth images, and an F1-measure then calculates the segmentation accuracy of each image against its ground truth image. Fig. 4. 12 are sample images where the original images (a, d, g) are segmented to (b, e, h) and compared to their ground truth images (c, f, i). The overall accuracy obtained by the model is 72%, as shown in [76] when employing F1-measure.

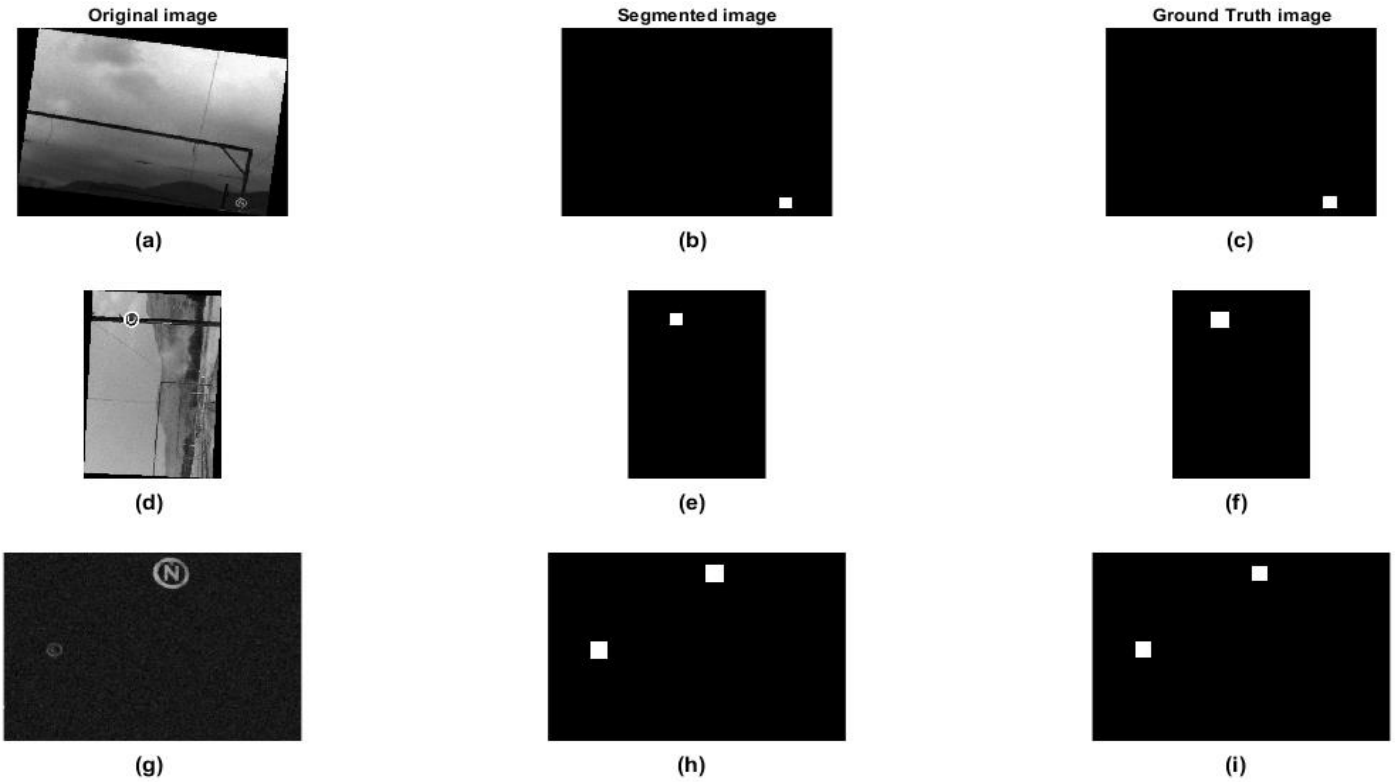


Fig. 4. 12: Segmented images versus Ground truth

4.8 Summary

The chapter set out to provide a detailed presentation of the results obtained from the research experiments in Chapter 3. This chapter discussed the dataset, pre-processing, RoI extraction and classification results. In the dataset used, sample images of open ('N') and close ('C') markers were presented in Fig. 4. 2. Subsequently, image pre-processing covered how each image in the dataset was processed in conversion from RGB to greyscale images and removal of noise using a bilateral filter. Furthermore, parameter evaluation of the bilateral filter was dealt with in detail, highlighting the choices that led to employing this filter described in Subsection 4.3.1. The results of RoI extraction employing Sobel and CHT methods were also covered. The chapter further presented the performance results of employing the HoG feature extraction method at different cell-size in conjunction with different classifiers. Section 4.7 presented the model's overall performance by measuring the segmentation accuracy by employing the F1-measure; a statistical technique for evaluating segmentation. In Chapter 5, we discuss the results obtained.

Chapter 5: Discussion of results

5.1 Introduction

In this chapter, we reflect on the results obtained in Chapter 4, where we look at the optimal performance achieved by the model. The chapter is organised in sections as follows: First, the dataset used is discussed in Section 5.2. The results from image pre-processing are discussed in Section 5.3. Thirdly, Section 5.4 discusses the results obtained from the marker extraction methods. Fourthly, feature classification results obtained are discussed in Section 5.5. Section 5.6 and 5.7 discuss each classifier's performance and the model's accuracy, respectively. Lastly, we end the chapter with a summary that reflects on the discussion of results obtained in each section.

5.2 Dataset Results

The dataset comprised 200 images which were acquired and were then increased to 654 with the addition of random noise with rotation and negative images. Fig. 3. 3 illustrates the sample images from the dataset, where a split of 422 training images and 128 for testing. Additionally, 104 random negative images (Refer to Fig. 4. 3) were added, totalling 232 test images. The dataset suffered from low-quality resolution and worsened with images captured at 30m – 45m away. The limitation is the camera's resolution of 640*480; however, upon investigating, the actual resolution used for the images is 481*321; the remainder is used for the cameras' header information. The lower resolution resulted in low-quality images, affecting the model's overall accuracy.

5.3 Image Pre-processing Results

In the dataset described in Section 5.2, images were converted from RGB to greyscale and noise was removed by employing a bilateral filter. The conversion reduces the computational cost since RGB colour images are 3-channels and greyscale images are only 1-channel. Using equation (3. 1), when converting from RGB to a greyscale colour space, there is no fragmentation of the quality of the original image, as illustrated in Fig. 3. 5. While some images in the dataset may have no noise embedded in them, an assumption that all acquired images contain noise is made. A bilateral filter is employed to remove random noise: parameters σ_s and σ_r were chosen. Prior to selecting these parameters, a baseline was established to compare the effectiveness of these parameters. A baseline of 99.3% is obtained from computing the correlation of the original greyscale image against the same image but embedded with random noise. The baseline allows a fair comparison of how effective the selected parameters are in removing noise compared to the original greyscale image. A value above or closer to the baseline would indicate a bilateral filter with optimal parameters. Fig. 4. 6 demonstrates the effects of different σ_s and σ_r parameters used and Table 4.1 is a performance summary of each these parameters. At parameter $\sigma_s = 3$ and $\sigma_r = 100$ the filter performs better in removing noise, achieving an accuracy of 99.46%; however, slower by 0.01276s. While at $\sigma_s = 1$ and $\sigma_r = 650.25$ an accuracy of 99.43% at 0.00262s was achieved. In both cases, noise was removed effectively as each accuracy is above the baseline: a trade-off is, therefore made in selecting the optimal parameters. Referring to Fig. 4. 7(b) shows that an increase in σ_s also increases the computational cost, since the variance is 0.03% (99.46% - 99.43%) therefore, the selected parameters are $\sigma_s = 1$ and $\sigma_r = 650.25$. The results obtained in Fig. 4. 6 also support the literature in Chapter 2 in that a bilateral filter is a multi-purpose filter that removes noise by smoothing and preserving edges.

5.4 Marker Extraction Results

The markers are extracted by employing a Circular Hough Transform (CHT) as described in Section 4.4. The results are illustrated in Fig. 4. 8 and Fig. 4. 9, where the former shows an image with no markers while the latter shows images with markers. The CHT employs a Sobel operator that removes most background artefacts while leaving the RoI in the foreground. Several techniques have been proposed for removing background artefacts, yet these fared poorly compared to the edge detection method. The selection of an edge detection method presented an advantage regarding computation and effectiveness in removing background pixels.

Furthermore, Fig. 3. 7 described the process followed in selecting a suitable operator: a Sobel operator achieved a 57.81%, Canny a 54.51% and Prewitt a 55%. As a result of this comparison, the Sobel operator was adopted into the CHT. The range of radii values is 5 – 30 pixels, obtained from the diameter of marker pixel sizes captured at 10m and 45m away. The lowest pixel diameter at 45m is 10 pixels which calculate as 5 pixels of radius, and at 10m, the highest pixel diameter is 60 pixels resulting in 30 pixels of radius. The advantage of having radii allows the algorithm to detect markers even when they are far or closer since it has more than one radius. This approach reduces the error rate of undetected markers due to image acquisition with varying distances, which can affect the pixel diameter of each marker. Assigning a constant radius would limit the range where markers can mainly be detected.

The disadvantage of employing a CHT in finding RoIs is that false positives are detected when there are circular-shaped objects that are not markers. However, the advantage of CHT being a shape detection method, it detects markers in images with high resolution effectively and fairs well in the dataset; however, the limiting factor is that images acquired at a distance CHT fails to detect correctly. Furthermore, extracting RoIs with CHT is simple: illustrated in Fig. 3. 11, a bounding box obtained from the parametric space (a, b, r) is used to crop each RoI.

This approach is simple and accurate in extracting each RoI for classification purposes. The results of this approach are shown in Fig. 4. 9(c, g, k), where the background pixels are removed and are in black while the extracted RoIs remain in the foreground. It is worth noting that before Fig. 4. 9(c, g, k), each RoI is validated in the classification stage Section 5.5 to reduce any false positives during the detection stage.

Table 3. 5: CHT Advantages and Disadvantages [76, 78]

Advantages	**Disadvantages**
• Detects shapes accurately in images with high resolution.	• Employes brute force approach.
• Employs edge detection method to reduce foreground artefacts.	• Complex in computation and requires large memory.
• Compared to other methods, it performs better with the dataset used in this research.	

5.5 Feature Classification Results

In Chapter 2, several approaches were proposed for extracting features, while in [76], the authors proposed using an HoG feature extractor. Implementing other feature extraction methods such as SIFT or SURF presented challenges in the dataset, such as increased computation and fewer features extracted [76]. Section 4.5 describes the implementation of the HoG feature extraction method where BoF is created from an optimal cell-size. Manually extracting markers from training images ensures that when features are extracted, the correct RoIs are used, as this restricts extracting incorrect features during training. Fig. 3. 14 visualises the HoG features at different cell-sizes: a [2 2] cell-size has more features extracted than a [16

16] cell-size, but the former suffers from an increased computational cost and storage space. Section 4.5 presents the results of each cell-size, and Table 4.29 is a performance overview of each cell-size. It is worth noting that at a [2 2] cell-size, BoF size is 45MB while at [16 16] is 211KB; which saves about 99.51% of storage space when compared to a [2 2] cell-size. A [16 16] cell-size is, therefore, an ideal choice in applications where memory is limited; however, at this cell-size with fewer features extracted for training, this leads to misclassification during training. As shown in Table 4.22, at [16 16] cell-size with cross-validation of 2-fold, an accuracy of 87.60% and 90.30% at 5-fold are achieved. When compared to a [2 2] cell-size, an accuracy of 83.70% and 94.40% is obtained with the training data set. Surprisingly, the misclassification is less than what was envisaged with a [16 16] cell-size. Introducing the test set as depicted in Fig. 4. 10, a [4 4] cell-size is chosen with 2-fold cross-validation: trade-off between storage, the accuracy of the trained model, computational cost and data size seemed a better choice. In 2-fold cross-validation with a [4 4] cell-size, the storage of BoF is 10MB, with the model achieving an accuracy of 95.20% during training and 97.60% during test validation. The results obtained from a [2 2] cell-size suggest that, while more features are extracted due to image resolution, its effectiveness can be neglected when considering memory space as performance varies less than a larger cell-size.

A [4 4] cell-size is standardised across all machine learning classification algorithms, from Table 4.10 to Table 4.30. The purpose of employing different machine learning classifiers is to compare the effectiveness of implementing HoG features by measuring the accuracy of each classifier. Subsection 4.5.1 mentioned the selection of an LSVM as the preferred classifier, and the choice is based on the comparison done on all four classifiers.

5.6 Performance of each Classifier Results

All four machine learning classifiers mentioned in Section 4.5 or detailed in Subsections 4.5.1 - 4.5.11 illustrate each classifier's performance in a confusion matrix. Fig. 4. 10 and Fig. 4. 11 is a graphical representations of each classifier's performance at different parameter settings.

a) Performance of LSVM Classifier:

The LSVM classifier is tested against a [2 2], [4 4], [8 8] and [16 16] cell-size with a 2-fold and 5-fold cross-validation. The classifiers' accuracy during training with 2-fold cross-validation is highest (95.20%) at a [4 4] cell-size. When new images (test data) are introduced, the accuracy of classifying each image is 97.60%. At varying cell-size, with 2-fold cross-validation, the test accuracy is highest, while at 5-fold cross-validation, the training accuracy is highest. Interestingly, a trained model with high accuracy is desirable, but at lower test accuracy, this presents an increased number of misclassification when a new data set is introduced into the trained model. In selecting a preferred model, a compromise is made where the test accuracy is above the training accuracy of the model, however, while maintaining a 90% or greater accuracy. Therefore, a [4 4] cell-size with 2-fold cross-validation seems a better choice than a 5-fold cross-validation. In [76], a holdout method was employed in a confusion matrix with a test performance that achieved a 95.2% accuracy. Compared with 2-fold cross-validation, 97.60% is achieved: splitting the dataset without being subjective improves the model's performance.

b) Performance of QSVM Classifier:

The QVSM classifier is tested only with a [4 4] cell-size at 2-fold and 5-fold cross-validation, comparing how each performs. The models' results on a 2-fold performed better with the test images, while on a 5-fold cross-validation, it is the opposite. On a 2-fold, the training accuracy achieved is 94.70%, with a test accuracy of 96.40%, while on a 5-fold cross-

validation, it is 96.0% and 86.9%, respectively. Comparatively, in Fig. 4. 10, the results obtained on the QSVM with those from the LSVM yield similar results in training and testing but not in each performance. It suggests that the dataset used with a 2-fold yields better results than a 5-fold cross-validation.

c) Performance of K-NN Classifier:

In Fig. 4. 10, the performance of a K-NN is shown at K= {1,3,5} with a 2-fold and 5-fold cross-validation. With 2-fold cross-validation at 1-NN, a test accuracy of 86.90% against a training accuracy of 84.60% is achieved. Compared to a 3-NN and 5-NN, 84.50% and 80.90% test accuracies are achieved, respectively. Implying that at 1-NN, the classifier performs best with test data, while the 3-NN and 5-NN perform better with the training data. The training accuracy across the K-values (1,3 and 5) in ascending order at 2-fold cross-correlation is 84.60%, 88.90 and 88.10%, respectively. At a 5-fold cross-validation during training, the accuracy decreases from 88.90% to 83.30%, achieving 85.70% to 77.1% during testing. The latter and the former prove that 2-fold cross-validation is a suitable choice. This 2-fold cross-validation supports what has already been stated in LSVM and QSVM results.

d) Performance of CNN Classifier:

The CNN classifier parameters used are Layers = {1,2} at a [4 4] cell-size with a 2-fold and 5-fold cross-validation. When the CNN has a 1-fully connected layer with 2-fold cross-validation, as shown in Fig. 4. 10, the classifier performs better with training data than with test data. A 97.40% accuracy on training and 91.60% on test data is achieved while applying a 5-fold cross-validation yields similar results with 96.90% on training and 92.80% on test accuracy. Adding layers, viz 2-fully connected layer on the CNN improves the accuracy of the test data: on 2-fold cross-validation, the training accuracy is at 90.30%, while the test is at 95.20%. Furthermore, on a 5-fold cross-validation, the training accuracy is at 95.60% and 96.40% for the test. Though both cross-validations yield similar results, noticeably, a 2-fold is

more favourable as the accuracy of the test data is more when compared to its training data. Also, in 5-fold cross-validation, the difference between the test and training is minimal, implying that increasing the *K*-fold value has less effect on the model. In [76], a CNN with 2-full connected layers achieved 100% accuracy during training but reached an 82% test accuracy. The performance of this model was obtained by employing a holdout method. Compared to *K*-fold cross-validation, where a 2-fold is employed, the model has a training accuracy of 90.30% with an increased test accuracy of 95.20%. The authors also extracted features from CNN using a convolutional layer; however, in this research, HoG features were used, drastically improving the test performance of the CNN classifier.

e) Performance Summary of Classifiers (published research [90]):

Firstly, considering the performance of each classifier in Fig. 4. 11: LSVM and QSVM take first place with an accuracy of 94%, second place is LDA and CSVM at 92.80%, and third is CNN plus K-NN(20) achieving 90.40%. The AdaBoost results show that it is the worst-performing classifier at 57.80%; even during training it still performed last at an accuracy of 57.50% when compared to the other classifiers. Secondly, considering the prediction speed/computation time summarised in Table 4.30 the CNN takes the lead at 82 objects per second (obs/sec), followed by LDA at 78 obs/sec and in third place is LSVM at 75 obs/sec. The K-NN is the worst performing classifier in terms of prediction speed even at different K-values (2, 5, 10, and 20) reaching between 12 – 14 obs/sec.

5.7 Accuracy of the Model Results

The performances presented in Section 5.6 only detail how each classifier performs against other machine learning classifiers at different parameter settings. The LSVM at a [4 4] cell-size was chosen at 2-fold cross-validation. In this section, we discuss the overall performance of the model measured by the F1-measure in equation (3. 61). The segmented images illustrated

in Fig. 4. 9(d, h, l) and Fig. 4. 12(b, e, h) are compared with the ground truth images, illustrated in Fig. 4. 12(c, f, i). The overall accuracy obtained is 72%, similar to the results in [76]. The model obtained the same outcome results from employing the same BoF when training the classifier. The 90% below target accuracy is mainly caused by the quality of the images used during training. Subsequently, several segmented RoIs were detected as either false positives or misclassifications. However, considering the overall dataset of 422 training images, a resolution of 481*321 pixels, and the manually created ground truth images, an accuracy of 72% can be considered high. A camera that produces higher resolution images at distances above 25 envisages increasing accuracy.

Furthermore, ground truth images that are precisely or well segmented would increase the model's accuracy. The disadvantage of manually creating the ground truth images is that it introduces RoI that are sometimes not the same size as the segmented RoIs as this process is subjective. It can be observed, for instance, in Fig. 4. 12(e), that the segmented RoI is slightly smaller than its ground truth image in Fig. 4. 12(f). The observation indicates that when the correlation is computed between the two images, the linear relationship's strength is reduced, reducing the overall model's accuracy. Therefore, ground truth images that are precisely segmented to be identical to the segmented RoI images have strong linear relationships, increasing the overall accuracy.

5.8 Summary

Chapter 5 intended to discuss the results obtained in the previous chapter from the dataset used on the model's overall performance. The discussions around the dataset used were concisely presented in Section 5.2, with a mix of images containing random noise and rotation and acquired under different weather conditions. Section 5.3 discussed the results obtained during RGB to greyscale conversion and noise filtering. Implementing equation (3. 1) for

conversion of RGB image to greyscale colour space as discussed, proved to be effective as this was also demonstrated in Fig. 3. 5. The selection of parameters $\sigma_s = 1$ and $\sigma_r = 650.25$ in the bilateral filter was also discussed. In Sections 5.4 - 5.6, an in-depth discussion on the results obtained was discussed. A discussion on the CHT method employed to extract the markers, the process of extracting HoG features, to the four classifiers employed were discussed. Section 5.7 concisely outlined the overall performance of the model. Therefore, the aim and objectives set out in Chapter 1 have been achieved in this research. Chapter 6 concludes this research.

Chapter 6: Conclusion

6.1 Thesis Conclusion

This research aimed to develop an image processing and computer vision model to automatically detect and classify railway neutral section markers, resulting in auto-switching electric locomotives as they traverse the neutral section. Conventionally, auto-switching electric locomotives are achieved with induction magnets installed on the rail, as depicted in Fig. 1. 1 and Fig. 1. 2. Underneath the locomotive, as shown in Fig. 1. 3 are sensors on both ends that detect the induction magnets which then either opens (off) or closes (on) the vacuum circuit breaker (VCB). The problems associated with this method, as stated in Section 1.2, led to a need to develop a method that will employ image processing and computer vision to auto-switch electric locomotives in a neutral section.

A model that detected and classified open and close markers was developed based on image processing and computer vision techniques. Five steps were proposed when several pieces of literature were reviewed: image acquisition, image pre-processing, image segmentation, feature extraction, and classification. The acquired dataset images were in RGB format, and some were embedded with random noise and rotation; therefore, image pre-processing techniques were employed to convert each image to greyscale colour space and remove noise. According to the literature study, converting to a greyscale format and using a bilateral filter provided better results. Therefore, each image was converted to greyscale and used a bilateral filter to improve the quality of the image by removing noise.

After the images were pre-processed, a segmentation method was developed to detect and extract the RoI from the image. The literature study and simulation of models showed that several methods, viz. edge-based and colour segmentation, such as edge detection and thresholding, are effective for various segmentation tasks. However, the combination of a

Circular Hough Transform (CHT) and Sobel edge detector on the collected images proved effective compared to these other methods. The delineation of RoIs by removing background artefacts with the Sobel operator and then applying the CHT achieved an overall segmentation accuracy of 72%.

The literature study presented two feature extraction techniques, namely local and global. The former is advantageous over the latter feature extractors as they are invariant to image transformations such as rotation and variation in illumination. A literature study was conducted to compare local feature extractors such as SIFT, SURF and MSER, which led to using a different extractor. The HoG descriptor was used as a local feature extractor to represent each image as a BoF (feature vector).

The obtained BoF from the HoG feature extractor is then used to train a machine-learning classifier. The performance of the HoG feature extractor was then compared to seven machine learning classifiers, namely the SVM (Linear and Quadratic), K-NN, CNN, DT, DA, Naïve Bayes and AdaBoost. The selection of these classifiers was based on the literature study, which inferred they are one of the widely used state-of-the-art classifiers. It was found that HoG features with [4 4] cell-size combined with an LSVM achieved a classification accuracy of 95.20% during training and with a test validation of 97.60% based on 2-fold cross-validation in the first experiment. In the second experiment which is the published research work [90], the LSVM also performed better when compared to the other classifiers.

6.2 Research Challenges and Limitations

The scarcity of literature presented challenges and limitations in this research, namely:

1. Open (“N”) and close (“C”) markers needed to be designed, manufactured, and installed onsite.
2. New dataset needed to be collected.

3. Ground truth images were manually segmented; this consumed time and is subjective.
4. Pixel resolution is low, affecting the quality of images, noticeably with images captured at a distance.
5. The scarcity of literature that directly employs computer vision in neutral sections found in railways made it difficult to compare work done to the proposed research.

6.3 Recommendations for Future work

1. Increase the dataset of training images to increase the model's accuracy and better predict test images or newer images.
2. Develop a machine-based segmentation algorithm that will generate ground truth images in an objective approach rather than a subjective one.
3. Deploy a night vision high-definition camera to acquire images with a higher resolution during low visibility, such as at night or when it is misty.
4. Design and develop a Graphic User Interface (GUI) application based on this research work for real-time applications.

The results obtained in this research thus can be used as a baseline to further improve the research on the auto-switching of electric locomotives in neutral sections by employing computer vision.

References

[1] Wikipedia, "Industrial Revolution," ed: Wikipedia, The Free Encyclopedia, 2021.

[2] Trailhead. "Meet the Three Industrial Revolutions " Salesforce. https://trailhead.salesforce.com/en/content/learn/modules/learn-about-the-fourth-industrial-revolution/meet-the-three-industrial-revolutions (accessed Mar. 28, 2021).

[3] Wikipedia, "Fourth Industrial Revolution," ed: Wikipedia, The Free Encyclopedia, 2021.

[4] D. Chen, M. Pan, W. Tian, and W. Yang, "Automatic neutral section passing control device based on image recognition for electric locomotives," in *IEEE International Conference on Imaging Systems and Techniques*, 1-2 July 2010, pp. 385-388, doi: 10.1109/IST.2010.5548442.

[5] D. Hattingh and H. Van Vuuren, "Track Magnet Placement and Centre Mast Earthing Arrangement for Arthur Flury Neutral Section. 25kV AC Electrification," Transnet Freight Rail, 2017, BBC1831.

[6] A. F. AG. (2019). AF_NS25_Installation_instruction. Available: https://www.aflury.ch/en/contd/documentation/railway-technology/?tx_fodocumentation_documentation%5BfileId%5D=108833&cHash=b697e6bfd8e7dcb8f4b16a675f3d191f

[7] M. T. Qadri and M. Asif, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition," in *International Conference on Education Technology and Computer*, 2009, pp. 335-338, doi: 10.1109/icetc.2009.54.

[8] Y. Wang, M. Shi, and T. Wu, "A Method of Fast and Robust for Traffic Sign Recognition," in *Fifth International Conference on Image and Graphics*, 2009, pp. 891-895, doi: 10.1109/icig.2009.130.

[9] M. Babu and M. V. Raghunadh, "Vehicle Number Plate Detection and Recognition using Bounding Box Method," in *International Conference on Advanced Communication Control and Computing Technologies*, India, 2016, pp. 106-110.

[10] R. C. Gonzalez, R. E. Woods, and L. Eddins, *Digital Image Processing Using MATLAB*, 2 ed. United State of America: Gatesmark, 2009, p. 827.

[11] C. Yi, L. Hong, and M. Lingzhi, "Research on Electric Locomotive Auto-passing Neutral Section," in *Fourth International Conference on Intelligent Systems Design and Engineering Applications*, 2013, pp. 463-467, doi: 10.1109/ISDEA.2013.511.

[12] T. Ning, "Dedicated detector for Auto-passing of Phase Separation System on electric locomotive," in *International Conference on Electrical Machines and Systems*, 15-18 Nov. 2009, pp. 1-5, doi: 10.1109/ICEMS.2009.5382895.

[13] Z. Han, S. Liu, and S. Gao, "An Automatic System for China High-speed Multiple Unit Train Running Through Neutral Section with Electric Load," in *Asia-Pacific Power and Energy Engineering Conference*, 28-31 March 2010, pp. 1-3, doi: 10.1109/APPEEC.2010.5448394.

[14] W. Ran, T. Q. Zheng, X. Li, and B. Liu, "Research on power electronic switch system used in the auto-passing neutral section with electric load," in *International Conference on Electrical Machines and Systems*, 20-23 Aug. 2011, pp. 1-4, doi: 10.1109/ICEMS.2011.6073775.

[15] L. Xiong, L. Fei, R. Wang, and T. Q. Zheng, "Thyristor Photoelectric Firing System Applied in Auto-Passing Neutral Section System," in *Asia-Pacific Power and Energy Engineering Conference*, 27-29 March 2012, pp. 1-4, doi: 10.1109/APPEEC.2012.6307740.

[16] E. Delgado, I. Aizpuru, J. M. Canales, M. Ayarzaguena, A. Galparsoro, and T. Nieva, "Static switch based solution for improvement of Neutral Sections in HSR systems," in *Electrical Systems for Aircraft, Railway and Ship Propulsion*, 16-18 Oct. 2012, pp. 1-6, doi: 10.1109/ESARS.2012.6387398.

[17] S. Hee-Sang, C. Sung-Min, H. Jae-Sun, K. Jae-Chul, and K. Dong-Jin, "Application on of SFCL in Automatic Power Changeover Switch System of Electric Railways," *IEEE Transactions on Applied Superconductivity,* vol. 22, no. 3, pp. 5600704-5600704, 2012, doi: 10.1109/tasc.2011.2177617.

[18] T. Mizuma, J. Yoshinaga, and T. Yamaguchi, "The Development of Supervising System for Train Operation by Imaging Pictures," in *IET International Conference On Railway Condition Monitoring*, 29-30 Nov. 2006, pp. 167-171.

[19] T. Suzuki, "Challenges of image-sensor development," in *IEEE International Solid-State Circuits Conference - (ISSCC)*, 7-11 Feb. 2010, pp. 27-30, doi: 10.1109/ISSCC.2010.5434065.

[20] R. C. Gonzalez, *Digital Image Processing*. Pearson Education, 2009.

[21] L. Wenjuan, C. Fenglin, L. Chuan, F. Min, and Y. Maohua, "Evaluation of imaging quality of CCD camera by measuring MTFs at different contrasts," in *Proceedings of 2011 6th International Forum on Strategic Technology*, 22-24 Aug. 2011, vol. 2, pp. 664-668, doi: 10.1109/IFOST.2011.6021113.

[22] S. Mehta, A. Patel, and J. Mehta, "CCD or CMOS Image sensor for photography," in *International Conference on Communications and Signal Processing (ICCSP)*, 2-4 April 2015, pp. 0291-0294, doi: 10.1109/ICCSP.2015.7322890.

[23] A. K. Boyat and B. K. Joshi, "A Review Paper : Noise Models in Digital Image Processing," *Signal & Image Processing : An International Journal,* vol. 6, no. 2, pp. 63-75, 2015, doi: 10.5121/sipij.2015.6206.

[24] A. M. Hambal, Z. Pei, and F. Libent Ishabailu, "Image Noise Reduction and Filtering Techniques," *International Journal of Science and Research,* vol. 6, no. 3, pp. 2319-7064, 2015, doi: 10.21275/25031706.

[25] M. Mandar, D. Sontakke, M. Meghana, and S. Kulkarni, "Different Types of Noises in Images and Noise Removing Technique," *International Journal of Advanced Technology in Engineering and Science,* vol. 3, no. 01, pp. 102-115, 2015. [Online]. Available: www.ijates.com.

[26] S. Kaur, "Noise Types and Various Removal Techniques," *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE),* vol. 4, no. 2, pp. 226-230, 2015.

[27] Y. Santur, M. Karaköse, A. İ, and E. Akın, "IMU based adaptive blur removal approach using image processing for railway inspection," in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 23-25 May 2016, pp. 1-4, doi: 10.1109/IWSSIP.2016.7502729.

[28] D. Phillips, *Image processing in C.* R & D Publications Lawrence, 1994.

[29] T. Wakabayashi, U. Pal, F. Kimura, and Y. Miyake, "F-ratio Based Weighted Feature Extraction for Similar Shape Character Recognition," in *10th International Conference on Document Analysis and Recognition*, 2009, pp. 196-200, doi: 10.1109/icdar.2009.197.

[30] Y. Y. Nguwi and W. J. Lim, "Number plate recognition in noisy image," in *8th International Congress on Image and Signal Processing (CISP)*, 14-16 Oct. 2015, pp. 476-480, doi: 10.1109/CISP.2015.7407927.

[31] K. Vigneshwar and B. H. Kumar, "Detection and counting of pothole using image processing techniques," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 15-17 Dec. 2016, pp. 1-4, doi: 10.1109/ICCIC.2016.7919622.

[32] R. Kaur and E. R. Singh, "Image Filtering Techniques-A Review," *International Journal of Advanced Research in Science and Engineering,* vol. 6, no. 8, pp. 2066-2071, 2017. [Online]. Available: www.ijarse.com.

[33] B. Desai, U. Kushwaha, and S. Jha, "Image Filtering -Techniques , Algorithm and Applications," vol. 7, no. 11, pp. 970-975. [Online]. Available: https://www.researchgate.net/publication/346583845

[34] H. Aditya, T. Gayatri, T. Santosh, S. Ankalaki, and J. Majumdar, "Performance analysis of video segmentation," in *4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 6-7 Jan. 2017, pp. 1-6, doi: 10.1109/ICACCS.2017.8014567.

[35] X. Liu, B. Dai, and H. He, "Real-time object segmentation for visual object detection in dynamic scenes," in *International Conference of Soft Computing and Pattern*

Recognition (SoCPaR), 14-16 Oct. 2011, pp. 423-428, doi: 10.1109/SoCPaR.2011.6089281.

[36] A. M. Khan and S. Ravi, "Image Segmentation Methods: A Comparative Study," *International Journal of Soft Computing and Engineering (IJSCE),* vol. 3, no. 4, pp. 84-92, 2013.

[37] D. Kaur and Y. Kaur, "Various Image Segmentation Techniques: A Review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809-814. [Online]. Available: www.ijcsmc.com

[38] K. G. Babu, D. Harith Reddy, and P. Teja, "An Overview on Image Classification Methods in Image Processing," *International Journal of Current Engineering and Scientific Research (IJCESR),* vol. 5, no. 1, pp. 27-29, 2018.

[39] Q. Wang and X. Liu, "Traffic sign segmentation in natural scenes based on color and shape features," in *IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)*, 29-30 Sept. 2014, pp. 374-377, doi: 10.1109/WARTIA.2014.6976273.

[40] G. Balamurugan, S. Punniakodi, K. Rajeswari, and V. Arulalan, "Automatic number plate recognition system using super-resolution technique," in *International Conference on Computing and Communications Technologies (ICCCT)*, 26-27 Feb. 2015, pp. 273-277, doi: 10.1109/ICCCT2.2015.7292759.

[41] R. Islam, K. F. Sharif, and S. Biswas, "Automatic vehicle number plate recognition using structured elements," in *IEEE Conference on Systems, Process and Control (ICSPC)*, 18-20 Dec. 2015, pp. 44-48, doi: 10.1109/SPC.2015.7473557.

[42] G. Soni, A. Singh, and N. Sharma, "Inshore ship and hybrid object detection and recognition using context-aware color and shape model," in *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 18-19 Dec. 2015, pp. 699-703, doi: 10.1109/ICCICCT.2015.7475369.

[43] H. Huang and L.-Y. Hou, "Speed Limit Sign Detection Based on Gaussian Color Model and Template Matching," in *International Conference on Vision, Image and Signal Processing (ICVISP)*, 2017, pp. 118-122, doi: 10.1109/icvisp.2017.30.

[44] R. Panahi and I. Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 4, pp. 767-779, 2017, doi: 10.1109/tits.2016.2586520.

[45] S.-Y. Chiu, C.-C. Chiu, and S. Xu, "A Background Subtraction Algorithm in Complex Environments Based on Category Entropy Analysis," *Applied Sciences,* vol. 8, p. 885, May 2018, doi: 10.3390/app8060885.

[46] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network," *IEEE Transactions on Intelligent*

Transportation Systems, vol. 19, no. 4, pp. 1100-1111, 2018, doi: 10.1109/tits.2017.2714691.

[47] J. Deepthy and R. M. M. Anishin, "A Survey on MSER Based Scene Text Detection," *International Research Journal of Engineering and Technology*, vol. 5, no. 3, pp. 2685-2687, 2685. [Online]. Available: www.irjet.net

[48] Y. Li, J. Zhang, P. Gao, L. Jiang, and M. Chen, "Grab Cut Image Segmentation Based on Image Region," in *IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 27-29 June 2018, pp. 311-315, doi: 10.1109/ICIVC.2018.8492818.

[49] N. A. Othman, M. U. Salur, M. Karakose, and I. Aydin, "An Embedded Real-Time Object Detection and Measurement of its Size," in *International Conference on Artificial Intelligence and Data Processing (IDAP)*, 28-30 Sept. 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620812.

[50] S. Deshmukh and T. Moh, "Fine Object Detection in Automated Solar Panel Layout Generation," in *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 17-20 Dec. 2018, pp. 1402-1407, doi: 10.1109/ICMLA.2018.00228.

[51] L. Yaping, Z. Jinfang, X. Fanjiang, and S. Xv, "The Recognition and Enhancement of Traffic Sign for the Computer-Generated Image," in *Fourth International Conference on Digital Home*, 2012, pp. 405-410, doi: 10.1109/icdh.2012.8.

[52] S. Mane and S. Mangale, "Moving Object Detection and Tracking Using Convolutional Neural Networks," in *Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 14-15 June 2018, pp. 1809-1813, doi: 10.1109/ICCONS.2018.8662921.

[53] A. Saini and M. Biswas, "Object Detection in Underwater Image by Detecting Edges using Adaptive Thresholding," in *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 23-25 April 2019, pp. 628-632, doi: 10.1109/ICOEI.2019.8862794.

[54] S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, "A survey of image classification methods and techniques," in *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 10-11 July 2014, pp. 554-557, doi: 10.1109/ICCICCT.2014.6993023.

[55] A. O. Salau and S. Jain, "Feature Extraction: A Survey of the Types, Techniques, Applications," in *International Conference on Signal Processing and Communication (ICSC)*, 7-9 March 2019, pp. 158-164, doi: 10.1109/ICSC45622.2019.8938371.

[56] J. Kim[1], B.-S. Kim[2], and S. Savarese[3], "Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines," in *Applied Mathematics in Electrical and Computer Engineering*, 2012, pp. 133-138.

[57] M. Barstuğan and R. Ceylan, "Comparison of Decision Tree and SVM Based AdaBoost Algorithms on Biomedical Benchmark Datasets," presented at the 6th European Conference of the International Federation for Medical and Biological Engineering, Dubrovnik, Croatia, 2014. [Online]. Available: https://www.researchgate.net/publication/311415719.

[58] T. F. Ju, W. M. Lu, K. H. Chen, and J. I. Guo, "Vision-based moving objects detection for intelligent automobiles and a robustness enhancing method," in *IEEE International Conference on Consumer Electronics - Taiwan*, 26-28 May 2014, pp. 75-76, doi: 10.1109/ICCE-TW.2014.6904109.

[59] Y. Han, K. Virupakshappa, and E. Oruklu, "Robust traffic sign recognition with feature extraction and K-NN classification methods," in *IEEE International Conference on Electro/Information Technology (EIT)*, 21-23 May 2015, pp. 484-488, doi: 10.1109/EIT.2015.7293386.

[60] N. B. Romdhane, H. Mliki, and M. Hammami, "An improved traffic signs recognition and tracking method for driver assistance system," in *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 26-29 June 2016, pp. 1-6, doi: 10.1109/ICIS.2016.7550772.

[61] X. Han, Y. Zhong, R. Feng, and L. Zhang, "Robust geospatial object detection based on pre-trained faster R-CNN framework for high spatial resolution imagery," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 23-28 July 2017, pp. 3353-3356, doi: 10.1109/IGARSS.2017.8127716.

[62] B. J. Koskowich, M. Rahnemoonfai, and M. Starek, "Virtualot — A Framework Enabling Real-Time Coordinate Transformation & Occlusion Sensitive Tracking Using UAS Products, Deep Learning Object Detection & Traditional Object Tracking Techniques," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 22-27 July 2018, pp. 6416-6419, doi: 10.1109/IGARSS.2018.8518124.

[63] Y. Lai, N. Wang, Y. Yang, and L. Lin, "Traffic Signs Recognition and Classification based on Deep Feature Learning," in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, 2018, pp. 622-629, doi: 10.5220/0006718806220629.

[64] C. G. Serna and Y. Ruichek, "Classification of Traffic Signs: The European Dataset," *IEEE Access,* vol. 6, pp. 78136-78148, 2018, doi: 10.1109/ACCESS.2018.2884826.

[65] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed.* United States of America: SAGE Publications, 2014, p. 342.

[66] C. Williams, "Research Methods," *Journal of Business & Economics Research,* vol. 5, no. 3, pp. 65-72, 2007. [Online]. Available: https://clutejournals.com/index.php/JBER/article/view/2532/2578.

[67] P. D. Leedy and J. E. Ormrod, *Practical research: Planning and design* 11, ed.: Pearson, 2015, p. 408. [Online]. Available: https://pce-fet.com/common/library/books/51/2590_%5BPaul_D._Leedy,_Jeanne_Ellis_Ormrod%5D_Practical_Res(b-ok.org).pdf.

[68] M. Easterby-Smith, R. Thorpe, P. R. Jackson, and L. J. Jaspersen, "Management & Business Research," 6 Ed.: SAGE Publications Ltd, 2018, pp. 60-86.

[69] C. Kivunja and A. B. Kuyini, "Understanding and Applying Research Paradigms in Educational Contexts," *International Journal of Higher Education,* vol. 6, no. 5, 2017, doi: 10.5430/ijhe.v6n5p26.

[70] S. K. Antwi and K. Hamza, "Qualitative and Quantitative Research Paradigms in Business Research: A Philosophical Reflection," *European Journal of Business and Management,* vol. 7, no. 3, pp. 217-225, 2015. [Online]. Available: www.iiste.org.

[71] J. Bacon-Shone, *Introduction to quantitative research methods.* Graduate School, The University of Hong Kong, 2013.

[72] A. M. Adam, "Sample Size Determination in Survey Research," *Journal of Scientific Research and Reports,* pp. 90-97, 2020, doi: 10.9734/jsrr/2020/v26i530263.

[73] C. Saravanan, "Color Image to Grayscale Image Conversion," in *Second International Conference on Computer Engineering and Applications*, 2010, pp. 196-199, doi: 10.1109/iccea.2010.192.

[74] B. Rohrer. (2019). *How to Convert an RGB Image to Grayscale* [Online]. Available: https://www.kdnuggets.com/2019/12/convert-rgb-image-grayscale.html.

[75] P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral Filtering: Theory and Applications," *Foundations and Trends in Computer Graphics and Vision,* vol. 4, pp. 1-74, 01/01 2009, doi: 10.1561/0600000020.

[76] C. T. Mcineka and S. Reddy, "Automatic Switching of Electric Locomotives in Neutral Sections," in *Conference on Information Communications Technology and Society (ICTAS)*, 10-11 March 2021, pp. 97-102, doi: 10.1109/ICTAS50802.2021.9394969.

[77] S. Israni and S. Jain, "Edge detection of license plate using Sobel operator," in *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 3-5 March 2016, pp. 3561-3563, doi: 10.1109/ICEEOT.2016.7755367.

[78] N. Cherabit, F. Z. Chelali, and A. Djeradi, "Circular hough transform for iris localization," *Science and Technology,* vol. 2, no. 5, pp. 114-121, 2012.

[79] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, vol. 1: IEEE, pp. 886-893.

[80] D. Boswell, "Introduction to support vector machines," *Departement of Computer Science and Engineering University of California San Diego,* 2002.

[81] T. Fletcher, "Support Vector Machines Explained," 2008. [Online]. Available: https://www.academia.edu/22709227/Support_Vector_Machines_Explained.

[82] B. Schölkopf *et al.*, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing,* vol. 45, no. 11, pp. 2758-2765, 1997, doi: 10.1109/78.650102.

[83] S. Mitrofanov and E. Semenkin, "An Approach to Training Decision Trees with the Relearning of Nodes," in *International Conference on Information Technologies (InfoTech)*, 16-17 Sept. 2021, pp. 1-5, doi: 10.1109/InfoTech52438.2021.9548520.

[84] L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data,* vol. 8, no. 1, pp. 1-74, 2021.

[85] A. Tharwat, "Linear vs. quadratic discriminant analysis classifier: a tutorial," *International Journal of Applied Pattern Recognition,* vol. 3, no. 2, pp. 145-180, 2016.

[86] MathWorks, "Prediction Using Discriminant Analysis Models," in *Help Center*, ed. [Online]: https://www.mathworks.com/help/stats/prediction-using-discriminant-analysis-models.html.

[87] H. R. Seth and H. Banka, "Hardware implementation of Naïve Bayes classifier: A cost effective technique," in *3rd International Conference on Recent Advances in Information Technology (RAIT)*, 3-5 March 2016, pp. 264-267, doi: 10.1109/RAIT.2016.7507913.

[88] H. Yigit, "A weighting approach for KNN classifier," in *International Conference on Electronics, Computer and Computation (ICECCO)*, 7-9 Nov. 2013, pp. 228-231, doi: 10.1109/ICECCO.2013.6718270.

[89] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 839-846, doi: 10.1109/ICCV.1998.710815.

[90] C. T. Mcineka and N. Pillay, "Machine Learning Classifiers Based on HoG Features Extracted from Locomotive Neutral Section Images," in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, 27-28 Oct. 2022 2022, pp. 1-6, doi: 10.1109/ICEET56468.2022.10007093.