Safety Screening of Auckland's Harbour Bridge Movable Concrete Barrier

A thesis submitted to Auckland University of Technology in partial fulfilment of the requirement for the degree of Master of Computer and Information Science (MCIS) 2021

May 2021

By

Munish Rathee

School of Engineering, Computer and Mathematical Sciences

Auckland University of Technology

Supervisor

Dr Boris Bačić

ABSTRACT

A moveable concrete barrier on the Auckland Harbour Bridge facilitates traffic flow control and optimisation. The concrete barrier's block segments are interconnected with metal pins, which sometimes can pop out of their safe position. This thesis aims to use deep learning to assist visual metal pin inspection to improve traffic safety. The thesis proposes real-time pin status detection and alerting solutions using various types of video sources. The first part of the proposed network detects and classifies the unsafe pins. The second part actively tracks and alerts the user of unsafe pin status. Preliminary experiments on a small dataset indicated that we could detect unsafe pin status with high precision and recall.

The novel contributions presented in the thesis include: (1) A universal system globally applicable to similar traffic flow regulation and safety contexts with minimal modifications. (2) A novel technique for obtaining synthetic frames to produce different degrees of unsafe pin images obtained from the original video frames. Collectively, synthetic minority-class data boosting, adaptive, incremental, and transfer learning utilising pre-trained neural networks allow a robust approach to data analysis and modelling on initially small and unbalanced datasets for circumstances where the expected size of the dataset may or may not become available within the expected timeframes (such as during the pandemic lockdowns and added safety requirements). From the presented proof-of-concept, future work is intended to include collaborative user-centred design, where models, software upgrades and analytical platform upgrades will be under the oversight of New Zealand NZ Transport Agency and Auckland System Management.

Keywords: Deep learning, Machine learning, Transfer learning, Traffic safety, Object detection and classification, Object tracking.

PREAMBLE

ॐ सरस्वती मया दृष्ट्वा, वीणा पुस्तक धारणीम् ।
हंस वाहिनी समायुक्ता मां विद्या दान करोतु में ॐ ।।

I owe my gratitude to Dr Boris Bačić for the guidance and encouragement during my studies and sympathetic, friendly suggestions during my struggle in these challenging times. His careful analysis and attention to detail helped immensely during the research. I also wish to thank Associated Prof. Russel Pears for his help during ICONIP 2020 conference. A big thanks to Anna Matich, Bumjun Kim(BJ) and Terry Brydon for their prompt support.

I want to express my boundless love to my beautiful wife Pooja, my kids Gargi and Nimit for their selfless patience and love and my parents for their blessings and cares. I also like to thank my dear friends Vatsal Gor, Adhil Thangal and Rahul Khoond for their help during fieldwork and technical activities.

My heartfelt thanks to the Auckland University of Technology (AUT) for the summer scholarship. And for the registration for the ICONIP conference 2020, the library and video recording equipment access. Finally, I want to show my gratitude for documentation made available by various contributors to the MathWorks, the Orange Data-Mining, SqueezeNet, TensorFlow, ImageNet, the Google Cloud and the Open CV for the libraries and various tools.

My special thanks go to Gary Bonser, Angela Potae and Martin Olive from New Zealand Transport Agency(NZTA) and Auckland System Management(ASM) for the kind assistance, including on-site transport, safety briefings and supervised video recordings, helpful insights into various project requirements and their continuous enthusiasm to assist with this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMS

A.I. or AI = Artificial Intelligence

AHB = Auckland Harbour Bridge

ANN = Also known as Artificial Neural Networks are a data processing architecture (collection of interconnected nodes called artificial neurons) inspired by the biological brain.

AUT = Auckland University of Technology

Feature Map = A feature map represents a spatial-relational construct of an object.

MBT = Movable Barrier Transfer Machine

MCB = Movable Concrete Barrier

MVP = Minimum Viable Product

NZTA = New Zealand Transport Agency

Resnet = A residual neural network that is derivative of the artificial neural network (ANN).

Transfer Learning = Learning rate improvement in a neural network to perform new tasks by transferring information from a related task that its network has already learned.

YOLO = Also known as "You only look once"; single-stage real time object detection model of convolutional neural networks.
PoC = Proof-of-Concept (PoC)

Note:
The list of terms is provided for multidisciplinary readership and is applicable to the context of this thesis.

ATTESTATION OF AUTHORSHIP

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly define in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature _____ Date 30/05/2021

ETHICS APPROVAL AND REUSE OF PREVIOUSLY PRODUCED RESEARCH

No ethical approval is needed for this thesis's research work because we only observe movable concrete barriers and metal pins without interference in human work. As for the reuse of previously produced research(Bačić, Rathee, & Pears, 2020), IEEE's guidelines state that the author can reuse the previously produced work under the creative commons attribution license (CC BY).

1 INTRODUCTION

Fig. 1: Movable concrete barrier joints and metal pins.

Imagine someone walking for over a kilometre with a bent neck and tilted spine, carrying over a kilogram weight on their head. A safety surveyor on the Auckland harbour bridge endured this every day to ensure traffic safety.

- *The author's observations during 1st AHB visit*

New Zealand Transport Agency (NZTA) manually monitor the movable concrete barrier on the Auckland harbour bridge (AHB), especially the safety of metal pins holding the movable concrete blocks (MCB) together. The hardworking staff of NZTA works in dangerous surroundings, with fast-moving traffic

buzzing past them all the time. On top of this, the weather conditions and heavy load of safety gear (hard hat and safety boots) make the surveillance process more difficult.

The humane research question that this thesis tries to answer is whether we can make the working environment better for the hard-working NZTA staff?

This chapter introduces the background of the movable concrete barrier and its contribution to the Auckland harbour bridge to show the significance and motivation behind the research. Afterwards, the research gaps will be identified, using artificial intelligence and computer vision for detecting unsafe pin positions—finally, the presentation of the thesis structure.

1.1 The Background

This research applies computer vision and deep learning to detect metal pins connecting movable concrete barriers (MCB) on the AHB and detect if a pin has popped out and is in danger of disconnecting the MCB. Computer vision helps object detection and tracking on a video in natural environments—for example, computer vision used for workplace safety, traffic monitoring and security surveillance. Real-time detection and tracking techniques are developed for moving objects. These techniques enable automated surveillance and event detection. The majority of contributions use computer vision to detect objects and track them in the natural environment such as big workplaces, public places, traffic scenes with pedestrians and cars. Most of these detections often carried out using immobile or moving cameras (Ikoma, Haraguchi, & Hasegawa, 2014).

The thesis research extends computer vision to metal pin detection, classification and alerting the user of unsafe pin position endangering overall traffic safety on Auckland harbour bridge. The videos captured from different angles and views are two dimensions with 720×480 resolution illustrated in Table 3 and Fig. 2: Video recording settings. Such videos can display the metal pins to show if the

pin is about to pop out of the joint (Fig. 1: Movable concrete barrier joints and metal pins). However, the pin tops are comparatively small in size, a challenge for any computer vision application detecting them. Hence, the entire pin block (region of interest) should be detected first; then, unsafe pin positions detected from individual pin block images.

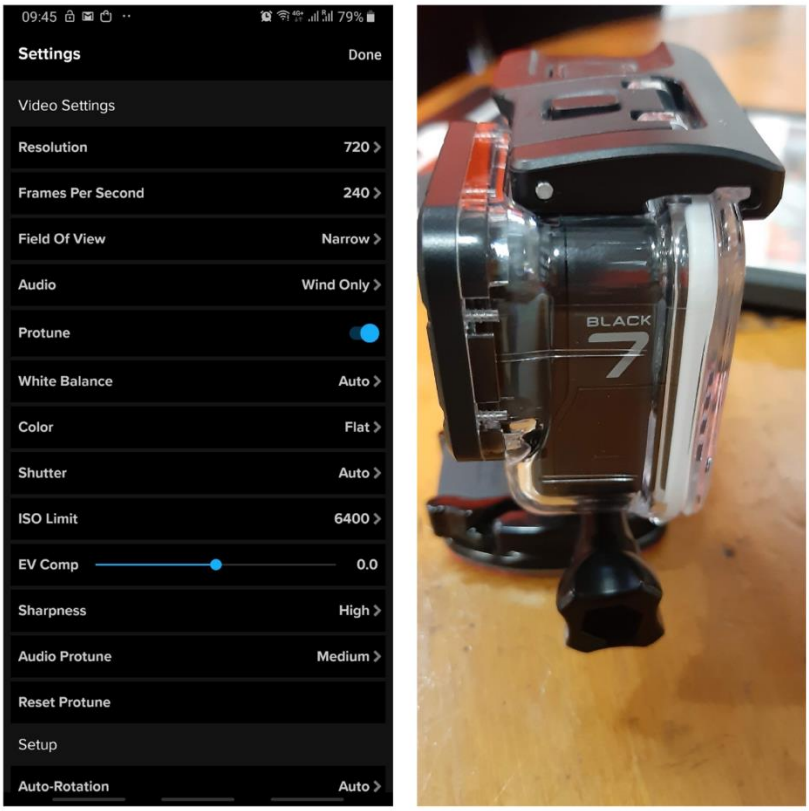

Fig. 2: Video recording settings.

The primary issue throughout the research is the shortage of minority frames (pin unsafely out of joint). NZTA restricted the author to shoot videos with unsafe pin positions in natural conditions. Moreover, all efforts to create unsafe pin positions in a controlled environment proved labour intensive. This created a unique situation where solving the problem was challenging because the problem cannot be found easily.

To boost the minority class, the author introduced a novel approach of synthetically creating additional clusters of different frames and different pin-out of position views for extra validity. Image editing software Gimp is used initially for creating synthetic frames (TheGimpTeam, 2020). With plans of automating the process at the advanced stage of research (Fig. 18). The process of creating synthetic frames depicting unsafe pin positions is time-consuming. Cloning a single frame could take 15 to 20 minutes on average.

Gaps between two movable concrete blocks can easily be detected on the video frames because of their large size and unique features. Motion detection can easily be used for detecting pin region of interest (ROI) because movable concrete barriers and the gaps between them are the predominantly visible objects in motion on the video. Conversely, Pins' bodies have a background matching grey colour and a reddish colour of rust, blending them with the background. In addition, the moving traffic in the background creates additional noise.

After ROI detection, pins detected on the individual video frames. Using image processing and analysis techniques, the pins' frames then examined to detect the structure of the ROI, such as concrete barrier joints, pin body, and unsafe pin positions. The unsafe pins have their top parts moved out of the sockets and the bottom pointy part disappearing inside the socket. A feature-based detection approach is employed to classify video frames and develop an automated alerting system. The approach used on the pin status detection was to insert resized video frames into the pretrained CNN and obtain a vector representing pin image features. Afterwards, the generated feature space from CNN is used with the traditional neural network techniques for further classification and data analysis.

1.2 The Significance and Motivation Behind the Research

The AHB is the eight laned transport link that connects the central city to the North Shore. In 2019, it was reported that 170,000-18,0000 cars, 1,000 buses and

11,000 heavy vehicles are crossing during workdays (Wilson, 2019). A moveable concrete barrier manages the middle four lanes of the bridge motorway. The MCB balances the peak hour traffic flow and minimises commuter fuel consumption (heritage, 1959; Wikipedia, 2020). The existing $1.4 million barrier transfer machines are relocating sixteen concrete blocks (weighing 750kg) at a time, four times during weekdays (NZTA, 2014). However, the barrier transfer machine do not have any automated solution for the safety screening of the metal pins that hold the movable concrete blocks together. A removable metal pin connects movable barrier blocks (Fig. 3b) (Fig. 4). Thus, a problem may occur if a metal pin is not securely connecting the contiguous concrete blocks.

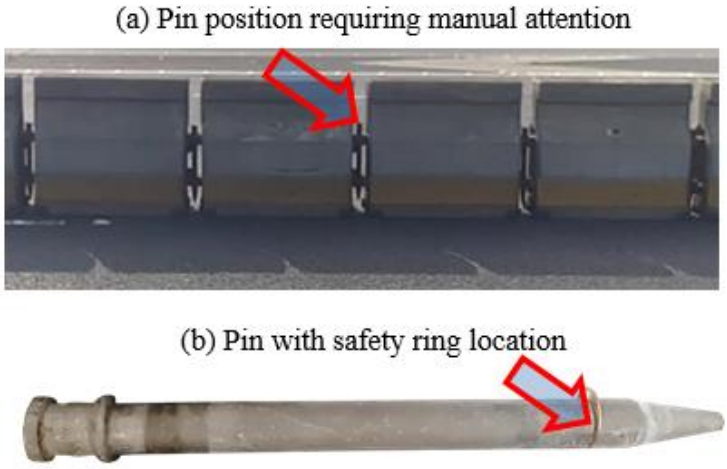

Fig. 3: A video frame highlighting the need for a manual check of a metal pin of moveable barrier block: (a) suspected Pin_Out status and (b) a metal pin safety ring. Adopted from (Bačić et al., 2020).

The movable concrete barrier is frequently inspected by an NZTA staff member who walks at the back of the barrier transfer machine (BTM). However, the manual inspection process could lead to safety hazards with "close encounters with truck mirrors" (NZTA, 2014). Manual inspections are strenuous, labour-intensive observations work. The surveyor has to walk in a physically uncomfortable body position (back bent laterally and the neck skewed with a load of safety hat), additionally hindered by bad lighting or the glare of fast-moving traffic (approx. 80 km/h).

Fig. 4: The barrier transfer machine (BTM) in action. Adopted from (Bačić et al., 2020)

The BTM operating crew consists of one operator at each end and the manual pin surveyor who moves along the empty lane in front of the machine. Apart from shifting the lanes, this crew is also responsible for ensuring that pins safely join the movable concrete barriers. If they find a Pin_Out of a safe position, they hammer it down and secure the safety latch (Fig. 5).

Fig. 5: Manual Pin inspection by NZTA staff. 1. Unsafe pin found during the manual inspection, 2. hammering down the metal pin, 3. securing the safety latch.

The safety of metal pin connecting the MCB is of global importance. Because even though many countries use the MCBs, no automated solution is introduced or even discussed to addresses the pin safety concern. The author's thesis and other research work (Bačić et al., 2020) are the first steps to solve this million-dollar problem.

1.3 Research Questions

The intention and objective of the thesis are to develop computer vision and deep learning based solutions for automated inspection of metal pins. The current pin inspection routine involves strenuous, labour-intensive observations. The intended solution would enable frequent inspections, regardless of the weather and traffic encounter risks. Aiming to improve traffic safety around the movable concrete barrier, provide support for manual pin inspection and minimise potential risks of human errors:

Can screening of unsafe pin positions be automated using AI and computer vision approaches? If so:

- What are possible solutions to reporting unsafe pin status and location(s)?
- What are the recommended considerations for improving the intended system performance, including processing costs and near-real-time anomaly detection?
- Can a system be developed that works in non-ideal lighting conditions while minimising vibration and background noise?
- What are possible considerations regarding visualisation, system adaptability?
- Can tuning the ratio of False-Positives/False-Negatives be achieved?
- What are the global implications and potential for future advancements?

The thesis focuses on developing a MATLAB app-based pin status detection and alert system as a demonstration prototype. Reported proof-of-concept (PoC) and supporting artefacts are pertinent to the minimum viable product (MVP) for our industry partner, NZTA. The final solution will help NZTA staff inspect movable concrete barrier(s) for unsafe pin positions and provide a system to perform daily safety inspections of Auckland's Haroubur Bridge moveable barrier.

1.4 Outline of The Thesis

A system for unsafe metal pin detection and tracking has never been researched or developed. This thesis will review prior contributions to traffic safety and computer vision to breach this gap in chapter 2. In addition to reviewing research work concerning object detection and tracking, it also includes a review of research in movable concrete barriers, their use, and other related research work done in traffic safety. Chapter 2 also explains the methods and theories related to this research. This chapter includes a review of popular pretrained deep neural networks, traditional artificial neural networks, the Gaussian Mixture Model for motion detection, region proposal based template matching for object tracking, colour thresholding, Hough transform, morphological calculations, Kalman filter, and transfer learning.

Chapter 3 covers the methodology of data collection, pin detection and developing a MATLAB app-based detection and alert system. The observations and interpretations of the experimental design, transitional evidence and suggestions. The comparison of different Deep CNNs in Liu with transfer learning is also discussed. Chapter 4 discuss the developed solution for the minority class boosting of unsafe pin positions and plans to automate it. The chapter also discusses the final solution developed to support an automated pin detection system, presents the developed system's design and prototype, and reflects on the produced results and advantages of the intended solution.

Finally, chapter 5 provides conclusions and future work.

1.5 Thesis Contribution

This thesis contributes to computer vision applications and traffic safety, particularly on minority class boosting and low resource object detection, ubiquitous computing and traffic safety apps. The developed algorithms, prototype, software tools for automated pin detection help in the following ways:

1. A universal system globally applicable to similar traffic flow regulation and safety contexts with minimal modifications.
2. A novel technique for obtaining synthetic frames with different degrees of unsafe pin frames obtained from the original video frames.
3. To provide MATLAB app-based pin detection and alert system that can be used with previously recorded videos or work on a real-time video feed from a camera mounted on BTM arm, or on a car body to detect unsafe pin positions in real-time;
4. To provide count based tracking that could help to locate the unsafe pin positions.

1.5.1 Authors' Publication Related to This Research Thesis

The initial work was submitted to an A-grade conference, ICONIP 2020 (http://portal.core.edu.au/conf-ranks/?search=ICONIP&by=all&source=CORE2020&sort=atitle&page=1), which has been accepted (1 Sep. 2020) with favourable reviews and entered into "best paper award" competition. Our ICONIP 2020 paper was also selected to be upgraded as a chapter in the upcoming book by Springer titled "Neural Information Processing".

Bačić, B., Rathee, M., & Pears, R. (2020). Automating Inspection of Moveable Lane Barrier for Auckland Harbour Bridge Traffic Safety Springer. Symposium conducted at the meeting of the International Conference on Neural Information Processing.

2 LITERATURE REVIEW

This literature review aims to present multidisciplinary background and applications of AI-based technologies to improve traffic safety on Auckland Harbor Bridge by automating the movable concrete barrier system's pin detection and alert process that could be monitored daily. The additional purpose is to provide high-level justification and decisions synthesis reflecting on chosen methodology, research steps, and produced artefacts pertinent to primary research objectives.

As this thesis is a snapshot in time, additional considerations are minority class boosting, object detection, computer vision, providing a state-of-the-art video and image analysis in the context of technology-mediated automated detection and alert. The main work of this thesis focuses on computer vision and artificial intelligence-based solutions to automate the screening of the metal pins connecting movable barrier segments.

2.1 Auckland Harbour Bridge

Our client, New Zealand Transport Agency (NZTA), maintains the eight-lane motorway over the Auckland Harbour Bridge (AHB) (NZTA). This 1020-meter-long bridge facilitates more than 200,000 vehicles on workdays. NZTA installed a moveable concrete lane barrier during the 1990s to prevent crashes and

optimise peak hour traffic flows. This moveable lane barrier along AHB enables two-way traffic flow control and traffic flow optimisation during peak hours.

2.1.1 Movable Concrete Barrier System

The moveable concrete barrier consists of 750kg concrete blocks joined by a metal pin (Fig. 6). This metal pin is secured in two metal joints with the help of a safety clip. Traffic accidents, movements and vibrations may cause the pins to pop out of their safe position. The disjoined heavy concrete barriers can slide in front of the upcoming traffic. Every pin position along AHB is inspected and secured manually, which is a labour-intensive task and subject to human error.

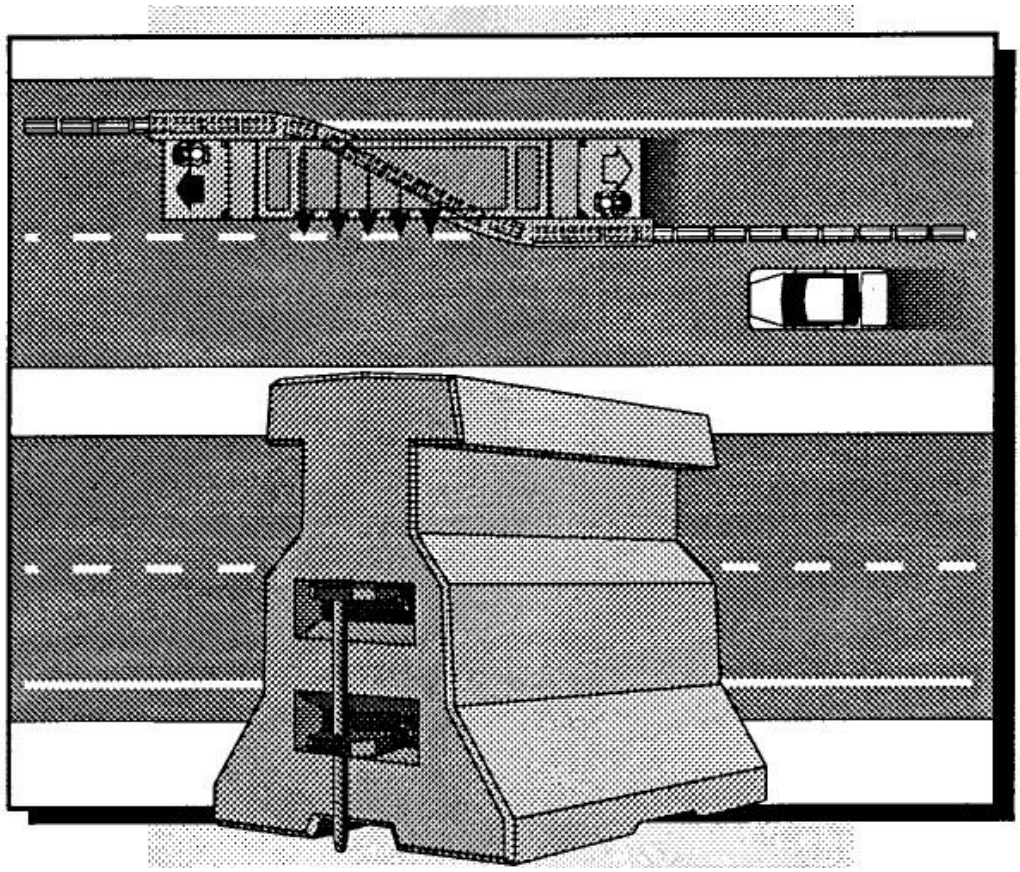

Fig. 6: Illustration of a barrier transfer machine changing active traffic lane (Cottrell, 1994).

Barrier Transfer Machines or the BTMs move sixteen concrete blocks simultaneously, usually four times during workdays (NZTA, 2014). During the lane changing process, one of the NZTA staff members walks in front of the BTM, collecting rubbish and inspecting the pins out of position. There are no existing automated pin inspection systems in the 19 metres long (US$ 1.4 million) machines. The movable concrete barrier (MCB) system consists of 1-meter long concrete block sections connected by steel pins in hinges to form a long wall moved laterally with a barrier transfer machine (BTM). Cottrell (1994); observed that the movable concrete barrier system helps with the quick changing of lanes to facilitate smooth traffic flow. Cottrell (1994) also reported that it takes roughly 10 minutes to reposition a one-kilometre long barrier wall. The movable barrier provides adequate protection against errant drivers (e.g. car exiting lane) and facilitates quick reassingment of traffic lanes, hence facilitating quick response in controlling uneven traffic flow in both directions by changing the number of available traffic lanes accordingly. However, various maintenance and environmental issues were faced (Cottrell, 1994; Poe, 1991). Multiple factors like wet weather and errant vehicle collisions have caused wear and tear in the MCB system. To address the maintenance issues and ensuring the MCB system's integrity, manual surveyors spend many hours moving along and looking at MCB pin joints (NZTA, 2014).

2.2 AI and Deep Learning for Object Detection

Detecting objects and tracking them using deep-learning and computer vision(CV) is a much-researched field (Heaton, 2018). The last decade saw the Artificial Intelligence (AI) research community solve problems and accomplish many notable feats like natural language processing, translations and interfaces between humans and machines; autonomous driving, face-recognition; human activity analysis for wellbeing or monitoring body function and individualized healthcare (Lee, Ullah, Wan, Gao, & Fang, 2019; Ma, Li, & Zhang, 2015; Mohapatra, Kar, Dash, Mohanty, & Swain, 2014). When we train a machine to

imitate the human brain with visual information to understand the 3D world around us, the processing includes segmentation, recognition, reconstruction and reorganisation(He, Zhang, Ren, & Sun, 2016). Deep learning for image processing falls under the broader machine learning sphere, with artificial learning at its core. This chapter provides a systematic review of multiple contexts and research topics to support and complement the main research work in this thesis. The main topics covered are a review of related work around traffic safety, specifically on anomaly detection, minority boosting methods, video and image processing methodologies, including popular transfer learning approaches and state-of-the-art neural networks.

Much of science derives inspiration from nature. Marvels of the ingenuity of the biological brain inspired the formation of artificial neural networks. A mathematical function representing biological neurons artificially is known as an artificial neuron. These simple, interconnected processors form artificial neural networks (ANN) (Schmidhuber, 2015). A single neuron model connecting traditional ANN models with deep-learning systems is the multilayer perceptron (MLP) (Heaton, 2018). A precursor to more extensive neural networks and considered a common mathematical function approximator, MLP maps input values to projected output values. A layered structure interconnects artificial neurons to perform MLP's internal operation. These structured artificial neuron groups pass their weighted output as input data to the following layer. Every connected neuron has the same activation function common to ANN, which is relatively pre-set and straightforward during the initialisation. Parallel execution of neural network processes is another revolutionary idea connected to deep learning approaches. One of the main problems with deep learning is representational learning or feature learning. As a functional composition, representational learning comprises other more superficial subsequent representational learning methods that are a part of deep learning with several levels of representations, attained by combining non-linear but simple modules. These modules alter the representation at one level, starting

at the raw input into higher and more abstract levels (LeCun, Bengio, & Hinton, 2015).

Similarly, the deep learning architectures have specific sub-systems that can produce the framework of processed frame pixels in specific layers producing edges, contours and other features. These can be regarded as basic processing tasks linked with representational learning concepts (LeCun et al., 2015). The main feature of deep learning is that humans do not design these feature layers: these are learned from data using a common learning method (Liu et al., 2016). Supervised learning makes use of labelled data to classify images. The machine is trained by feeding the images, which produces an output for each category in a vector of scores. The highest score is desired for the preferred category, but this requires training of the system. The function that calculates the distance (error) between the produced and desired values is computed and adjusted. The internal parameters of the system are modified to adjust the distance or error. The adjustable weights or parameters are real numbers. These parameters are also seen as 'knobs' that define the input and output function of the system. There may be millions of these flexible weights in a standard deep-learning system and millions of labelled instances used to train the system (Schmidhuber, 2015). Using a linear classifier with hand-engineered features is popular with many of the current practical machine learning applications. This approach enables a linear classifier with two classes to computes a weighted sum of the feature-vector modules.

One deep learning group of artificial neural networks employed in image and video processing is CNN, which have specific layers to perform pooling and convolution tasks (Szegedy et al., 2015). The convolution layer combines elements (like input pixels areas) into a smaller space while pooling picks the highest value elements. Together, such information processing through hidden layers constitutes machine-generated feature maps, providing classification at the end of the processing cycle (Girshick, Donahue, Darrell, & Malik, 2015). Variants of this

basic design are ubiquitous in the image classification literature. Notable examples of datasets that have yielded excellent results on classification are MNIST, CIFAR and the ImageNet (Krizhevsky, Sutskever, & Hinton, 2012; Zeiler & Fergus, 2014).

2.2.1 R-CNN, Fast and Faster R-CNN

The proposed candidate regions are input for a convolutional neural network (CNN) skewed into a square. R-CNN uses selective search to extract 2,000 proposed image regions (Girshick et al., 2015). The output is a dense layer consisting of features extracted from the image using CNN as a feature extractor. The next step is to input the extracted features into SVM to classify the object's presence within the proposed candidate region.

Even though R-CNN is no longer as comprehensive as the traditional techniques, during the first step of the R-CNN process, as many as 2,000 candidate region proposals are extracted from the original picture through selective search, each of these 2,000 candidate bounding boxes needs to extract CNN feature maps and conducts SVM classification. Thus, the amount of calculation is very large, leading to the R-CNN detection speed being very slow, around 47 seconds for each image.

Options for network architecture are AlexNet (Yuan & Zhang, 2016), Resnet 50 and SqueezeNet. After the test, the accuracy of Alexnet is 59.5 percent, and the accuracy of SqueezeNet is 98 percent. However, We selected Resnet 50 because deep residual networks (He et al., 2016) are deemed to be the newest and best in utilizing convolutional neural networks for image recognition. In addition, ResNet also won the Visual Recognition Challenge in 2015 (ILSVRC-15) with a top 5 error of 3.57 per cent. Therefore, in our study, we used ResNet-50; the structure is shown in Table 1.

2.2.2 Transfer Learning

Transfer learning aims at gathering knowledge while solving one problem and applying gathered knowledge to a different application. Transfer learning can do sample-based, feature-based, model-based, and relationship-based transfers.

The sample-based transfer learning process used calibrated samples to completes knowledge transfer. In comparison, the Feature-based transfer does it by mapping the source and target to the same space and minimizing the distance between them (Bengio, 2012). The model-based transfer learning combines the target and source models with samples to adjust parameters. On the other hand, relation-based transfer learning does it by learning the relationship between concepts in the source and then analogizing it to the target.

Table 1: Details of ResNet-50 layers. Inspired by (He et al., 2016)

Layer	Kernel Size	Stride	Padding	Output Size
Input				[224 × 224 × 3]
Conv1	7 × 7 × 3	2	3	[112 × 112 × 64]
Max pool	3 × 3	2	-	[56 × 56]
Conv2	[1×1conv,64],[3 × 3conv,64],1 × 1conv,256]	2	-	[56 × 56]
	[1×1conv,64],[3 × 3conv,64],1 × 1conv,256]	1	-	
	[1×1conv,64],[3 × 3conv,64],1 × 1conv,256]	1	-	
Conv3	[1×1conv,128],[3 × 3conv,128],[1 × 1conv,512]	2	-	[28 × 28]
	[1×1conv,128],[3 × 3conv,128],[1 × 1conv,512]	1	-	
	[1×1conv,128],[3 × 3conv,128],[1 × 1conv,512]	1	-	
	[1×1conv,128],[3 × 3conv,128],[1 × 1conv,512]	1	-	
Conv4	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	2	-	[14 × 14]
	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	1	-	
	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	1	-	
	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	1	-	
	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	1	-	
	[1×1conv,256],[3 × 3conv,256],[1 × 1conv,1024]	1	-	
Conv5	[1×1conv,512],[3 × 3conv,512],[1 × 1conv,2048]	2	-	[7 × 7]
	[1×1conv,512],[3 × 3conv,512],[1 × 1conv,2048]	1	-	
	[1×1conv,512],[3 × 3conv,512],[1 × 1conv,2048]	1	-	
Average pool	7 × 7	7	-	[1 × 1]
fc1000 softmax				1000

In most cases, pre-trained models can improve generalisation capabilities more or less than the train-from-scratch model (Mishkin, Sergievskiy, & Matas, 2017). The details on how transferable features are applied in deep neural networks also explain by Yosinski (Yosinski, Clune, Bengio, & Lipson, 2014). Deep Neural Network or DNN is a hierarchical feature illustration of data acquired through pre-training and takes advantage of high-level semantic classification. The bottom layer of the model is low-level semantic features like colour and edge information. The characteristics are constant in most classification tasks, while the distinction is the high-level features, which also explains that the new datasets sometimes exploited to update the last few layers of Resnet 50, Alexnet and GoogLeNet weights to achieve a simple transfer.

2.3 Object Detection and Classification in Image Processing

The digital image consists of digitised spatial coordinates and brightness represented as arrays (f(x,y)). These digitised arrays are picture elements or pixels, where x represents the corresponding horizontal pixel position and y represents the vertical position in a typical cartesian coordinate system. Detecting an object in a single frame or image falls under the digital image processing technique. Over the last decade, image processing has come of age with several innovative works. Several single frame object detection methods with limited areas of application have surfaced. The standard approaches make use of the detection of features, such as contour, colour and texture. The features come from pixel values like corners, edges, blobs or ridges. Pixels with similar features forms regions of interest. Image segmentation techniques employ different methods and detectors to achieve desired results. One way to find regions of interest in images is by looking for abrupt fluctuations in pixel values; this usually indicates edges that define regions. Another method divides images into regions based on texture and colour values.

2.3.1 Colour Images Segmentation

Based on colour features of image pixels, colour image segmentation takes the similar colours in the image as separate clusters and hence objects of interest in the image. Colour based segmentation is more straightforward than other feature-based segmentation techniques. However, the segmentation results vary based on the used colour space, and there is no single colour space that can provide relevant results for every type of image. So there is no one size fits all here, and different authors have first to determine the colour space that will suit their specific colour image segmentation problem (Busin, Vandenbroucke, & Macaire, 2008). No consensus emerged as a popular opinion about which is the best choice for colour space-based image segmentation so far. The typical colour space used in video and image acquisition is Red, Green and Blue (RGB). Respective channels represent the intensity for the respective colour. Different sorts of colour

spaces like LAB, CMY, XYZ, HSV, YCbCr, YIQ, YUV and DHT are used for various settings and methods. To help with research done during this thesis, the author analyses several papers to understand different approaches. A comparative study by Jurio, Pagola, Galar, Lopez-Molina, and Paternain (2010) between various colour spaces in cluster-based image segmentation using two similar clustering algorithms. This analysis involved examining four colour spaces, RGB, HSV, CMY, and YUV, to identify the best colour representation for hardware and human perception. Best results are obtained in most cases using CMY colour space where the quality of segmented images was higher, HSV also provided satisfactory outcomes. Busin et al. (2008) suggested a method based on a one-dimensional histogram and its discriminating powers to select a specific colour space among classical colour spaces automatically. This selection took place according to the evaluation criterion based on spatial connectedness properties of the pixels in the image. This principle assesses the quality of the segmentation in every space and picks the finest; this preserves its specific properties. Ananth et al. (2014) proposed a novel algorithm based on 2D histogram grouping for colour image segmentation. The intermediate features of the maximum overlap wavelet transform (IMOWT) method are used as a preprocessing step. Kumar (2014) tried a supervised method to obtain appropriate colour space for face detection using fuzzy logic. Kumar matched the HSV, YCbCr and RGB colour spaces. His research suggested that the YCbCr colour space detected face skin more precisely. Ganesan and Rajini (2014) discovered that YIQ colour space exploits the properties of the human eye, which makes it suitable for satellite image processing. The intensity elements can be collected with better precision, and the illumination changes in the image can effortlessly be solved since the illumination element is not dependent on the colour. the National Television System Committee (NTSC) also defined how by inverse transformation happens from RGB to YIQ. The authors reported how the eye of human is sensitive to intensity variations rather than variations of hue or saturation. This makes YIQ colour space more suitable because it separated the Y component luminance from the chrominance in the I

and Q elements. In the segmentation of satellite images, only histogram equalisation is performed on the Y channel. Compared to the RGB colour space, the YIQ colour space was more efficient because it does not need to be performed on three channels. T.-W. Chen, Chen, and Chien (2008) utilised natural image segmentation by using HSV colour space to different segmentation objects. They proposed an image segmentation algorithm with a quantisation method to generate grey and colour histograms in HSV colour space for K-Means clustering. This technique assisted them to cluster pixels utilizing K-means clustering. And if one colour space is not sufficient, more colour space can be helpful for detection.

An image is a collection of data or datasets where pixels have spatial locations and colour values. Post segmentation, the processed pixels should be organized into separate groups of coherent spatial connectivity and colour. Let us discuss the methods used for arranging pixels in different group regions.

Histogram Thresholding is one of the more common methods where it is assumed that images are comprised of regions with separate greyscale or colour ranges and separated into peaks, each representing a different region. For example, Raju and Neelima (2012) presented histogram thresholding following greyscale image segmentation. The objective was to get an optimum threshold that could separate object regions easily. This technique was proposed to improve the image segmentation thresholding of the last histogram. It takes to define a low pass filter and the augmentation and dilution of the peaks and valleys, correspondingly or the basic variation of the imagined Gaussian modes in the definitive thresholding. Sakthivel, Nallusamy, and Kavitha (2015) describes a new approach for colour image segmentation based on SVM and Fuzzy C-Means. They described how image segmentation achieved by clustering pixels into salient image regions. Their applied method demonstrated that segmentation could be utilised for object recognition, occlusion border estimate within motion or stereo systems, image compression, editing or database lookup. They extracted these

features utilizing the homogeneity model and Gabor Filter. Along With the extracted pixel-level features, the SVM Classifier is trained using FCM (Fuzzy C-Means). Their image segmentation method takes advantage of both the pixel level information of the image and the SVM Classifier. Sharma, Mishra, and Shrivastava (2012) presented a comparative study of the primary image segmentation techniques, i.e., EdgeBased, K-Means Clustering, Thresholding and Region-Based techniques. They are further describing colour image segmentation techniques that can be compared with these methods. The thresholding was done based on colour. The segmentation allowed eliminating a significant amount of unwanted pixels and retained only necessary pixels.

Neural Networks for colour image segmentation is another popular approach. Initially, the algorithm needs training on the object for computing features from provided colour information. After extraction, obtained features are utilised to detect objects in other images, such as distinct backgrounds. One instance of employing a neural network for colour segmentation (Littmann & Ritter, 1997). The used local linear map or LLM based approach. An image of the human hand is used as an example to train the network. They used images of the same human hand from various viewpoints. During the experimentation, the authors compared the LLM and standard statistical techniques. The LLM network works robustly under difficult image conditions. A different experiment by Hassanat, Alkasassbeh, Al-awadi, and Esra'a (2016) trained ANN using pixel colours to perform the image segmentation. The authors applied this technique to the human skin and tried on the entire face and more minor features like lip and eye, full hand and its finger, then the leaf of a tree. The colour information extracted from every pixel and its adjacent neighbours were utilised to produce the feature vectors. Later the feature vectors were classified utilizing the artificial neural network. Their experiments using ANN produced comparatively fast and effective results using their special colour information. The ANN also proved faster than the machine learning algorithms, like K nearest neighbour and its other variants.

The author investigated some unique experiments with colour information during this literature review. To improve the object detection accuracy, some researchers combined different features with colour. One example used hue and saturation elements to develop a shadow highlight invariance technique for detecting road signs (Fleyeh, 2006). Fleyeh found out that the colour information was susceptible to illumination situations such as clouds, shadows and sunlight. This HSV colour space-based method proved that hue and saturation are not affected by the effects of variation in illumination under different lighting conditions, and hence they can be used to develop shadow-invariant algorithms for colour based segmentation. Fleyeh tested the method on hundreds of images and successfully detected road signs in more than 95 percent of cases. The shape of an object is combined with colour space to execute object detection (Peng, 2015). The author utilised an automated seeded region growing technique for image segmentation. They used the colour only as complementary to shape, augmenting (Hinterstoisser, Lepetit, Ilic, Fua, & Navab, 2010) DOT method for textureless object detection. Further enhancing the DOT or dominant orientation template method, he presented a colour template similar to the DOT method combining the templates for object detection. In The End, the author examined the complexity of this technique to offer a speed-up approach.

When colour is used as a feature for image segmentation, no single method provides efficient results for all images. Histogram thresholding is the easiest and quickest in all techniques, although its capabilities are restricted in complex colour detection. Apart from histogram thresholding, most of the other colour-based segmentation techniques are complex, and if merged with other methods, the computational cost is possible to be extremely high.

2.3.2 Feature-Based Image Segmentation

Apart from colour, features like textures, intensity and contours can be merged to detect complex objects. Many researchers in different algorithms use these features to do object detection. More popular algorithms consist of thresholding, clustering, neural networks and machine learning as shown in figure below.

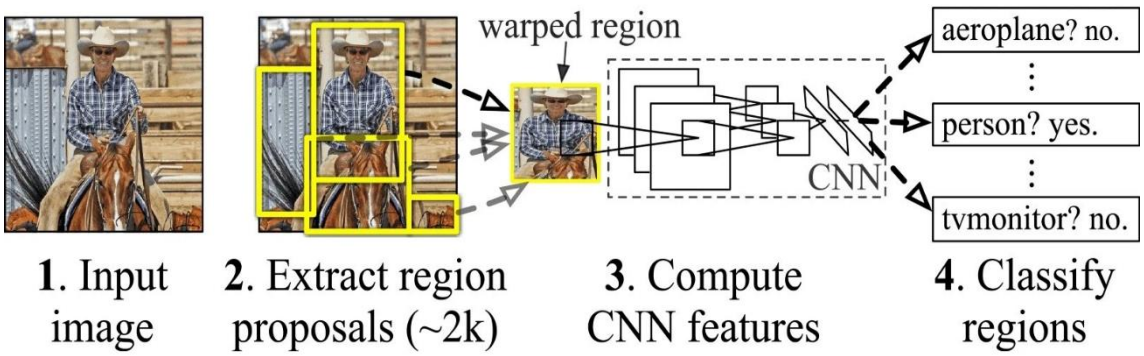

Fig 6.2: R-CNN: Region-based Convolutional Network (Girshick et al., 2015), reproduced under ?? license/permission??

The pixel intensity value is the value of the pixels in a greyscale image. The intensify feature consist of the info of brightness and illumination. Mohapatra et al. (2014) used illumination and reflection on grayscale images to detect a human face. Since they used nose shape as an identifying feature for face detection, variations in illumination conditions produced noticeable changes in facial appearance. They removed this obstacle and normalised the images by enhancing the reflection factor while reducing the illumination. Finally, they used the normalised images to train the SVM to get hold of the classifier. Thus attained classifier stores the faces' features from training and detects the human face by filtering out the subsequent images' features.

The contours are edges of objects highlighted against the background as a feature to reveal the shape. These edges split and highlight an object from various backgrounds so edge detection can discover the structure of the object. Usually, the contour method reveals the shape to edge or region-based detection methods.

Arbelaez et al., for object detection, built a gPb Contour detector by combining localised colour, brightness and texture cues to a robust globalisation framework using spectral clustering (Arbelaez, Maire, Fowlkes, & Malik, 2010). To enhance the method, the authors then connected the contour detector with a generic grouping algorithm. They started by constructing original regions from an adapted contour signal using an Oriented Watershed Transform. They proceeded by using an agglomerative clustering process to form the initial areas into a hierarchy signified by an Ultrametric Contour Map. The hierarchical region trees ultimately operated as an absolute beginning point for interactive segmentation. Another method (Rebouças Filho, da Silva Barros, Almeida, Rodrigues, & de Albuquerque, 2019) called optimum path snakes enables users to correct the inaccuracies in the automated segmentation with minimum annotation. The edge-based snake or active contour model works on the snake's edge-driven energy environment to grow. This model separates the image into object and background with maximum intensity separation. To overcome the limitations, these techniques were combined with the region-based snake technique using edge. The contour was developed using a region-based snake on grey-level C.T. scan images, while the Canny edge detector used to develop a partial-edge map of the lymph node. The contour left to let grow without restrictions until any segment comes across the partial edges of lymph nodes. The result indicated that this integration improved the segmentation significantly.

The texture feature defines visual info associated with regional variants of the image. Different filters extract it on a group of pixels since each pixel's texture cannot be described. Akbulut et al., Obtained the texture feature by using Hermite transform for image segmentation. Hermite filters use colour and texture for information independently for efficient colour texture image segmentation. The generation and selection process follows the grayscale transformation (Akbulut, Guo, Şengür, & Aslan, 2018). The selected filters are used on the grayscale image, and a Hermite texture magnitude is computed for each pixel. These filters are utilised for texture characterization, and by utilizing edge-preserving filtering,

smoothened features are acquired. Hence, by integrating the colour elements the input image can be efficiently segmented. The comparison estimates show the effectiveness of the proposed technique. However, the proposed technique has a shortcoming that we need to adjust several factors. In addition, the filter selection method needs to be reliant on a mathematical concept. Another unique texture descriptor was a factorization based active contour model for 2-phase texture segmentation (Gao, Chen, Zheng, & Fang, 2016). The feature the authors used was a local spectral histogram as the texture features and then established a novel energy function based on the theory of matrix decomposition. The spectral histograms captured local spatial patterns via filtering and global impression through histograms. When the filters were properly chosen, the spectral histogram could represent an arbitrary texture appearance.

2.3.3 Region-Based Segmentation in Images

Every image consists of regions made by pixels grouped based on colour, intensity, shape or texture. Constituent pixels or boundaries help identify these regions. Region-based segmentation builds uniform regions, satisfying given parameters or having a specific meaning in images (Z. Wang, Jensen, & Im, 2010).

Another Region-based segmentation, also known as region growing, assumes that the neighbouring pixels within one region have similar values (Tang, 2010). The typical method is comparing pixel with their neighbours and clustering them if the similarity criterion satisfied. The process falters and stops when similarities in neighbouring pixels are no longer found. The region growing algorithms are easy to use but rarely used alone and require other segmentation methods in tandem. One practical method combines the watershed algorithm and region growing algorithm for colour image segmentation (Tang, 2010).

The Seeded region growing technique takes a set of pixels as input and marks the objects for segmentation (Sharma et al., 2012). This method facilitates the iteratively growing of regions using unallocated neighbouring pixels. Different pixels are measured and assigned to the respective region. This process continues until all pixels are assigned. One innovative method based on seeded region growing is training a semantic segmentation network commencing from the discriminative regions and gradually increasing the pixel-level supervision used by seeded region growing (Huang, Wang, Wang, Liu, & Wang, 2018). The seeded region growing module is combined with a deep segmentation network and can gain from deep features.

2.3.4 Hough Transform based Object detection

Used to detect the edge of the shape of an image, the Hough transform is a feature extraction method utilised in digital image processing and computer vision. The technique aims to find objects by using shape parameters and edges. Before (Ballard, 1981) introduced the generalised Hugh transform, the classic method only detected analytically defined shapes (like line, triangle, rectangle, circle, ellipse, etc.). In these cases, we know the shapes beforehand and aim to find their location and alignment in the image. (Ballard, 1981) generalized the Hough transform by creating a map from edges to accumulator space. This modification is termed as "R-table" and contains the information to identify arbitrary non-analytic shapes. This modification enables the Hough to transform to detect an arbitrary object.

Chiu and Liaw (2005) proposed a voting technique for circle detection while reducing the calculation and storage needs of Standard Hough Transform. The process starts by selecting two edge points to verify the third point; this improved the Randomized Hough Transform (RHT) efficiency. Razavi, Gall, Kohli, and Van Gool (2012) created the Latent Hough Transform (LHT) to enforce vote consistency to support an object hypothesis. Their method augmented the Hough

space with latent variables. In turn, the system discriminatively learned the optimal latent assignments of the training data for object detection. Unlike clustering, the learning is not dependent on selecting the correct number of latent groups. Instead it shares training examples between groups.

2.4 Object Detection in Computer Vision

Object detection in videos recognises the physical movement of an object by observing object motion in multiple frames. Background subtraction, frame differencing, Temporal Differencing, and Optical Flow are traditional moving object detection methods. Gaussian Mixture Model (GMM) is the most popular background separation method.

2.4.1 Template Matching Approach for object detection

Template matching efficiently tracks object tracking in contiguous frames using *normxcorr2* and *regionprops* MATLAB function (Zuehlke1a, Henderson2a, & McMullenb, 2019). After completion of initial image processing, each video frame attempts to have the template matched. The template match location follows from the maximum peak of the normalized cross-correlation and yields the location of the centre of the template image on the main image. After further processing around the template location, a centroid is found for the object of interest, and the algorithm moves onto the following image.

Fig. 7: ROI tracking example. Sub-section is taken out of the frame in view for further processing. Centroids are detected for all contiguous regions in the subsection before calculating the object centroid.

An intensity-based centre-of-mass centroiding algorithm is then applied to the sub-window to locate the true centroid of the object found from the template image. MATLAB's built-in regionprops function utilizes the grayscale and binary sub-images to find centroid locations of contiguous regions in the image. Intensity weighted centroids are found using equation (2). Xc is the centroid location, xi is the current pixel location, and wi is the intensity of the current pixel.

$$x_c = \frac{\sum_{i=1}^{N} Xiwi}{\sum_{i=1}^{N} wi} \qquad 2\text{-}1$$

Pin ROI tracking achieved via the template matching algorithm. Each contiguous frame tries to have a template matched after initial processing is completed (Rege, Memane, Phatak, & Agarwal, 2013). The matched template locations follow the peaks of the normalized cross-correlation.

2.4.2 Foreground Background Separation

If we take computer vision as a virtual representation of human vision, then video analysis is crucial to mining information from visual data. Background foreground separation is a popular method to detect changes in image sequences (Ye et al., 2015). We divide a video clip into two parts: the background and the foreground, for video analysis to detect motion, recognise and detect objects, and video coding. The background stays continuously visible as a static object that has little or no variance distribution. The static nature of the background complements the moving object or foreground (Stauffer & Grimson, 1999). A moving object creates a new distribution variance over the existing low variance of the background. Greater pixel variance is another factor that distinguishes the foreground object from the background.

A novel background model estimates the probability of pixel intensity based on individual pixel values (Elgammal, Harwood, & Davis, 2000). This non-parametric approach made the model highly sensitive to moving objects by adapting to the background process quickly. This model can handle minor background noises and even adapt to minor movements like moving tree branches and bushes.

2.4.3 Gaussian Mixture Model

GMM is a favourite among all motion detection methods. (Friedman & Russell, 2013) introduced this model initially in 1997 as an efficient, incremental version of the EM algorithm. This unsupervised learning uses video series to eliminate the background or segment the foreground. Later, Stauffer and Grimson generalized it for real-time tracking (Stauffer & Grimson, 1999). Instead of explicitly modelling individual pixels, they modelled pixels as a mixture of Gaussians. This method can determine which pixel corresponds to the background colours using the variance and the persistence of pixels. If Pixel values do not fit the back-

ground, then they are considered as foreground values. This method can efficiently counter lighting changes caused by slow-moving shadows, minor disturbances and other problematic features and regains swiftly when the background resurfaces.

Building on this method, many researchers improved the GMM. Jain et al. introduced different methods to improve GMM, checking new pixels with present model parts to run in real-time (Jain, Reddy, & Dubey, 2014). If the pixels comparatively stay static, then they are classified as the background. A method focusing on enhancing GMM for complicated scenes, such as traffic in the background, was introduced by Ma et al. (2015). They also reported that GMM is not very accurate while detecting slow-moving or large vehicles. This paper improved vehicle detection by combining frame difference and GMM. They tested the improved GMM in different weather and lighting conditions. The result showed the GMM acquired improved adaptability, accuracy and real-time operations. Another background subtraction algorithm based on Gaussian mixture by K. Wang, Liang, Xing, and Zhang (2015) combined the discrepancy process technique for each frame to the GMM to improve the target detection. Firstly, a candidate background was obtained using GMM. Then, a variance process analysed the current frame and the background—finally, the "and" operation with moving targets is used for detection by GMM after the thresholding process.

2.4.4 Frame Difference Method

Frame difference is a standard method for motion detection. The moving object is detected by taking the last frame as a reference compared to the current frame. At times, three or more frame difference is calculated (Husein, Halim, & Leo, 2019). This method employs pixel-based differences to detects objects in motion. They employed background subtraction to improve frame differences to increasing methods effectiveness and precision.

One motion segmentation and tracking technique for visual surveillance was introduced (Ha & Lee, 2010). They were updating background images while detecting foreground objects, and this required many parameters. To counter this difficulty, they utilised an algorithm of various different images, which was more robust and only required one parameter. The results their method produced required less computation time.

2.4.5 Moving Object Detection Using Optical Flow

The optical flow-based motion assessment principle has enabled substantial research work and is still one of the most active domains in computer vision. In 1994 Barron et al. produced an empirical comparison of different optical flow techniques, including differential techniques, region-based matching, energy-based methods and phase-based techniques (Barron, Fleet, & Beauchemin, 1994). According to Schröder, Senst, Bochinski, and Sikora (2018) the history of research on optical flow shows that the accessibility of public benchmarks provided the most substantial push for significant innovation in the field (Schröder et al., 2018). This technique computes the image optical flow field and subsequently clusters pixels according to the optical flow distribution characteristics of the image. It can detect the overall movement from the background. Shafie et al. stated that optical flow could arise from the relative motion of objects and the viewer. This helps retrieve information about the objects' spatial arrangement in viewed and the rate of change. In simpler words, the optical flow field represents the three-dimensional motion of object pixels through a two-dimensional image (Shafie, Hafiz, & Ali, 2009).

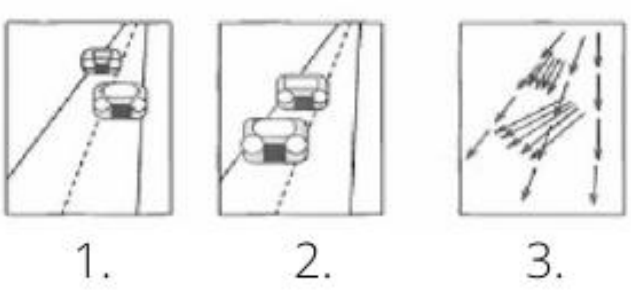

Fig. 8: Focus of Expansion 1. Time t1 2. Time t2 3. Optical flow. Modified from Shafie et al. (2009).

2.4.6 Shadow Removal

Several computer vision applications start with detection algorithms to automatically segment a video sequence from a static camera into background and foreground regions. One of the significant challenges for these algorithms comes from shadows cast by foreground objects (Varghese & Sreelekha, 2017). As a rule, shadows have the same colour information and move with foreground objects, even though it is darker than the original background. Populating the shadow model with shadow pixel values observed in real-time solved the complex problem of modelling different shadows. Their two-stage shadow detector accurately detects shadows, no matter the illumination situation, geometry or texture of the background, orientation or type of shadow.

As a shadow removal solution by Elgammal et al. (2000) separated colour information from illumination information. Suppose we take colour variables as chromaticity r, g and b coordinates.

$$r = \frac{R}{R+B+G}, g = \frac{G}{R+B+G}, b = \frac{B}{R+B+G} \; where \; r + g + b = 1 \qquad 2\text{-}2$$

However, shadow pixels caused sensitivity issues when the illumination changes were minor. Overall, this method proved effective in removing shadows from outdoor and indoor conditions.

2.5 Object Tracking in Multiple Frames

The process of identifying one or multiple objects across different video frames is called object tracking. Object tracking proceeds object detection, where already identified features like colour, texture and shape are used for tracking. Another essential feature is the object position, especially when similar objects like people, cars or metal pins are tracked. Apart from these, features as size and orientation are likewise helpful for object tracking. According to Stauffer et al., a robust object tracking system should not depend on the placement of cameras, the field of view or lighting conditions. Moreover, it should also deal with movement through cluttered areas and objects overlap (Stauffer & Grimson, 1999). The standard tracking techniques are the Kalman and particle filter. The Kalman filter efficiently tracks objects with linear motion, while the particle filter efficiently tracks objects with non-linear motion.

2.5.1 Blob Analysis Method

The analysis of a binarized image is called "blob analysis". Binary images are generated during the object detection process. The foreground object is white (1), and the background object is black (0). The white pixel or foreground object group is the blob. The object blob can be examined to determine blob features like area, height, width, orientation and position. For example, Thou-Ho et al. attempted to track vehicles utilizing object blobs to acquire the features of the perimeter, blob area, and bounding-box info such as height and width (T.-H. Chen, Lin, & Chen, 2007).

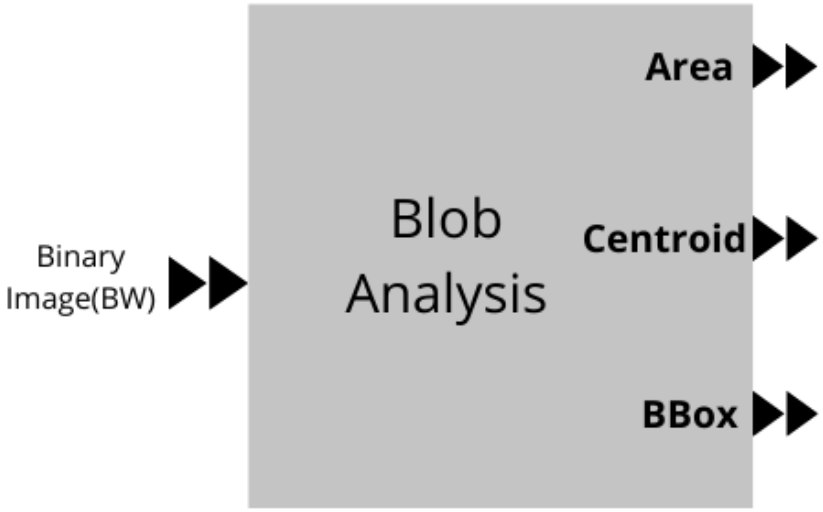

Fig. 9: The Blob Analysis used for calculating statistics for labeled regions in a binary image.

The object is detected and tracked in successive frames based on proximity and size. Distance between the centroids of objects (vehicles) measured using Euclidean distance and vehicle tracking was further enhanced using a vehicle's area. The same objects identified by measuring the least distance amongst blobs and similar sized blobs in two successive frames. This method can calculate the vehicles and also assess their velocities. Yusuf et al. propose BLOB analysis for object detection of fruits with different shapes and colours (Yusuf, Kusumanto, Oktarina, Dewi, & Risma, 2018). They extracted the blobs using 8-connectivity and template matching. The next step is to classify the different BLOBS according to size and shape.

2.5.2 Kalman Filter for Trajectory Estimation

The Kalman filter is popularly known for trajectory estimation. Because of its capability to produce comparatively accurate estimates of unknown variables, it accurately forecasts the following location of the object. Kalman filter combines predicted object location with the following detected location and calculates a refined location. The discrete Kalman filter was introduced to provide an efficient computational means solution of the least-squares technique and track different

objects using past, present and future states (Welch & Bishop, 2006). The Kalman filter includes prediction and measurement and can also implement non-linear object tracking (Extended Kalman filter) by employing the Jacobian matrix of partial derivatives of transition matrices and noise. According to Patel and Thakore, the Kalman filter equation fall into two groups: time update and measurement update equations (Patel & Thakore, 2013). The time update equations project forward the current state and error covariance estimates to obtain the next step's deductive estimate. The measurement and correction update equations are responsible for the feedback.

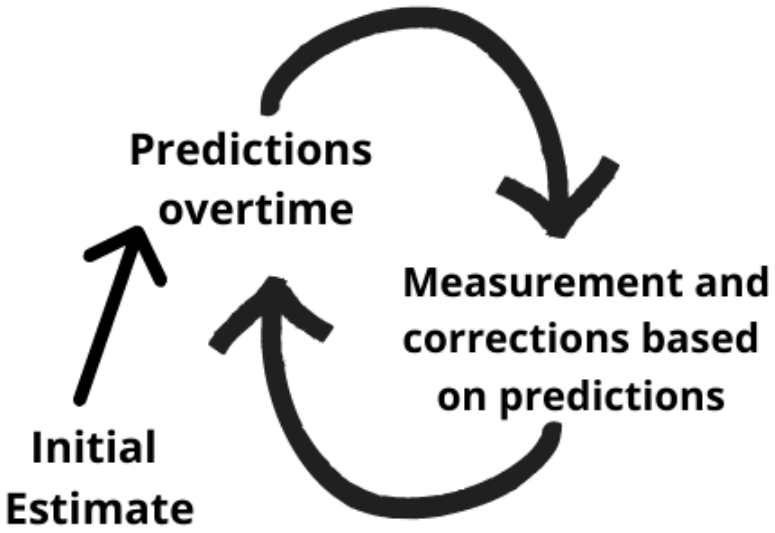

Fig. 10: Discrete Kalman filter cycle.

The second cyclic step incorporates new measurements into the apriori estimate to obtain an improved a posteriori estimate.

2.5.3 Particle Filter Based Tracking Method

The simple rule of particle filter centred tracking is to sample the object pixels and then resample pixels in the following frame. Then attempt to find similar features by comparing both samples. Three popular object motion tracking techniques (mean-shift, Kalman and particle filter) were evaluated in a study (Iqbal,

Shah, & Khan, 2014). They tracked a ball shifting linearly and a woman running non-linearly. An indoor environment was chosen for stable lighting conditions.

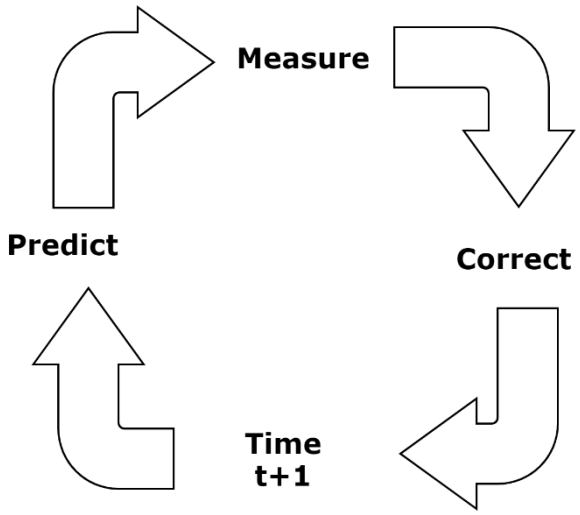

Fig. 11.2: Particle filter cycle.

All three techniques were observed efficient for linear tracking. Still, in the non-linear movement, the mean-shift and Kalman filter were less efficient than the particle filter. The mean-shift algorithm is based on a histogram, and fast change disturbed the tracking, while the Kalman filter depends upon the object's previous state to assess the current state. However, in the particle filter, the tracked object is symbolized by a potential location denoted by a set of weighted particles, making it more effective.

Particle-based tracking works better when targeted objects are undergoing significant variations. This method can efficiently cover denser samplings in comparison to other object tracking methods. In simple words, when multiple particles are sampled to construct a robust object appearance model, tracking algorithms based on particle filter are likely to perform reliably in such cluttered and noisy scenes (T. Zhang, Xu, & Yang, 2017). Zhang et al. propose a multi-tasking "correlation particle filter" that can effectively handle scale variation and exploit

interdependencies among different features to learn their correlation filters jointly.

2.5.4 Mean-Shift Algorithm

The mean-shift algorithm utilised a colour centred statistical model to estimate the next spot of the target (Iqbal et al., 2014). The central elements of the mean shift algorithm are mode pursuing, tracking and clustering density assessment. The mean-shift algorithm iteratively swings data points in its vicinity and locates the target's new position using density assessment and the colour histogram. The mean-shift algorithm is widely used in tracking clustering, however convergence of the mean-shift algorithm has not been rigorously proved. In this research, the mean-shift algorithm combined with a Gaussian profile is studied and applied to object tracking (Wen & Cai, 2006).

2.5.5 Adaptive Local Movement Model

The adaptive local movement model (ALMM) was utilised to trace a particular object (B. Zhang, Li, Perina, Del Bue, & Murino, 2015). The simple notion is that the dissemination of the movements of the regional patches was simpler to prototype instead of the entire object movement. In this approach, the regional patch centres were associated with the centre of gravity of the entire traced object. The first traced the local patches to assess the position and the validity of every image area. In the next step, the spots of patches were additionally rectified by using an outlier detection method based on GMM that pruned the patches which deviated from the current statistics. Finally, they assigned a weight to each patch, with the patch tracking, to determine whether a patch should be kept or removed in calculating the centre of gravity. The experimentation results indicated that this technique was robust for occlusion problems, fast motion and texture variations. This method can be adopted in future to find a simpler way of detecting pin ROI

and status tracked as a single object. If successfully applied, this method can simplify the process and make it less resource-intensive.

2.6 Synthesis, Justification and Methodology Rationale

In general, data collection and expert (and/or manual) labelling of images and videos can be expensive, resource-challenging and a time-consuming process. Compared to traditional machine learning, deep learning approaches typically require larger datasets and computational resources required for modelling tasks. Deep learning is also known to outperform traditional learning approaches in a number of areas of video and image processing. On the other hand, expert-driven feature extraction techniques combined with traditional ML approaches could also provide a viable solution with initially smaller datasets and often requiring less computational resources for the intended target system design. For computer scientists, pretrained deep learning models can also be used to automatically generate *feature space* (as opposed to expert-driven feature extraction engineering) and transfer learning approaches (allowing models to evolve their originally pretrained function without computationally demanding resource requirements).

While having a big picture of the intended project extending beyond minimum viable product (MVP) and related contexts of smart city and improved usability of spaces where human movement occurs, this thesis reports on a snapshot in time focused on proof of concept (PoC) to provide evidence answering a set of associated research questions. From research project methodology design and decision perspectives, main considerations include the following: (a) the intended target platform is not likely to have restricted resources such as with a low-cost IoT or consumer-grade technology such was used for PoC development; (b) intended dataset may be unbalanced and not available as planned; and that (c) traditional ML, video and image processing methods should be considered to provide data insights and potentially viable improvement options. The selected

candidate approaches include (1) transfer learning for object detection and classification, (2) generating feature space relying on deep learning approaches, (3) combining traditional ML classifiers and data analysis techniques to provide additional insights from data beyond the PoC.

3 METHODOLOGY

This chapter describes the data collection methods and equipment used, experimental setups, evaluation of existing inspection methodologies, and shortcomings. As a response to global pandemic and lockdowns, this chapter also provides additional insights and rationale that may help computer scientists to overcome similar obstacles associated with deep learning and data-driven research projects.

The methodology of this thesis focuses on solving real-world problems while producing technological solutions as a logical end. However, the Covid induced lockdowns infected the flow of the thesis by obstructing the experimental and data collection work and access to university resources. The author needed to adopted and adjust during various phases of the thesis and produced a robust pandemic proof methodology supported by a novel technique of background cloning to produce synthetic frames depicting unsafe pin positions. A brief outline of obstacles faced and problems solved can be observed in the table below.

Table 2: Obstacles during various phases of the research.

Obstacle	Problem	Solution
Heavy traffic on AHB	Recording high-quality videos at slow vehicle speed were not possible.	Two GoPro cameras mounted on BTM arms and videos recorded under the supervision of NZTA staff.
Multiple Lockdown and freak accident(as reported by media) on AHB	Limited data and unavailability of GPU machine during initial experiments	Combine deep learning with traditional ANNs to produce a prototype with limited resources ().
Restrictions on shooting videos of unsafe pin positions in a natural environment	Unavailability of minority class for training a network model for detection and classification	Produced synthetic frames using background cloning as shown in Fig. 18

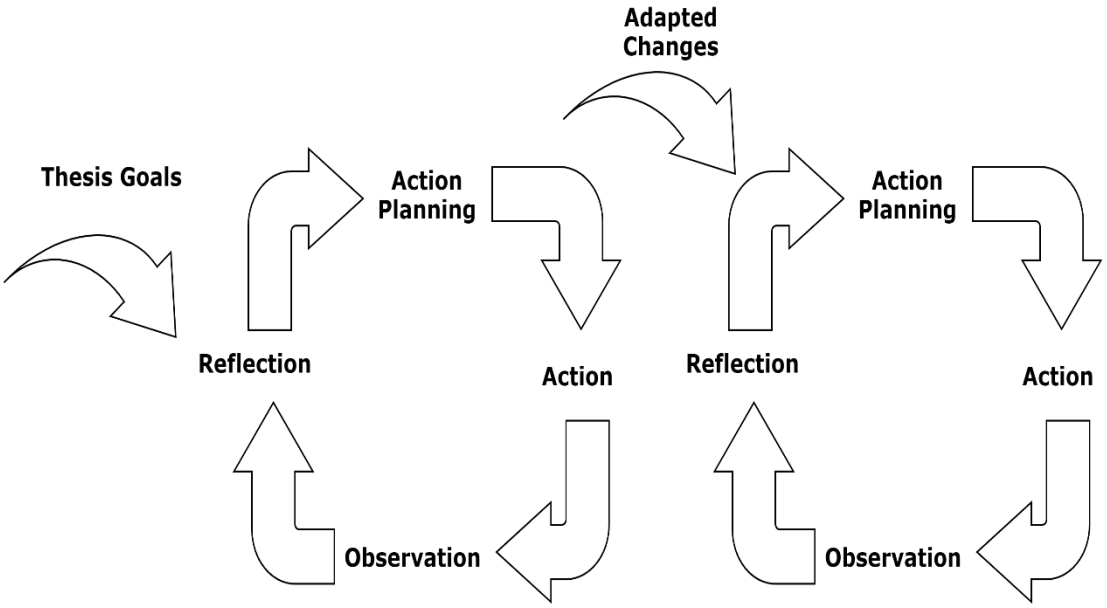

Fig. 12: Adopted action cycle inspired by (Vallenga, Grypdonck, Hoogwerf, & Tan, 2009).

Since the author needed a way to bridge the gap between theory and practice and counter problems of implementing research findings due to lack of fit or lack of motivation due to the covid circumstances in the practical setting, the author adopts the idea of action research (Vallenga et al., 2009), which is a form of research that enables practitioners to investigate and evaluate their own work. The methodology also aids the cyclic nature of the experimental approach (Carlson & Bloom, 2005), which helps evaluate the ideas and implement proofs of concepts from the initial stages towards achieving viable solutions. The multidimensional problem-solving framework adopted by this thesis has four phases: orientation, planning, executing, and rechecking on top of action research.

Multiple experiments were performed using computer vision, deep learning, and image processing to produce solutions like algorithms, architecture, and frameworks for unsafe pin detection. The ideas behind these experiments were logically developed and organised from a literature review.

3.1 Data Collection Tools and Setup

Fig. 13: Data collection setup: Samsung A7 Mobile, Apple Ipad 6, External power bank, GoPro cameras and mounting equipment, Camera mounting on BTM.

The equipment for video recording and camera mounting during data collection visit includes:

- Video recording equipment: Two GoPro Camera (GoPro 8, GoPro 5), Samsung A7, IPad 6, duct tape, strips, power bank.
- GoPro camera settings for video recording settings
  - Frame rate set to 240 frames per second.
  - Resolution of 720px with narrow field vision.
  - Barrier transfer machine speed between 6kph to 9kph for sharper frames.
  - Multiple videos captured from two different angles during sunny and overcast conditions.
- MATLAB R2020a and Orange 3.25 used for prototyping and developing viable solutions.

3.2 Data Collection Sessions

New Zealand Transport Agency (NZTA) allowed the author and the supervisor to attend a safety briefing, visit the movable concrete barrier sites, and record video data under the supervision of NZTA staff. Initially, the supervisor collected short videos from a handheld camera while walking behind the barrier transfer machine. The first visit by the supervisor paved the way for initial experiments done for multiple research papers. Later on, the author recorded six more videos from two cameras mounted at the front and rear arm of the barrier transfer machine (BTM). The NZTA staff also recorded one video on a rainy day. This video was collected from a camera mounted on the front arm of the BTM. The complete detail of video sessions are illustrated in Table 11: Details of used data

3.2.1 Safety Requirements

NZTA allowed first access to the Auckland harbour bridge BTM sites during covid restrictions at level 1. The staff on-site briefed the author and the supervisor on the phone and email about human-to-human distancing and hygiene protocols. Wearing masks was mandatory during the entire visit, and the entire team from Auckland University of Technology (AUT) was provided with hand sanitiser bottles.

During the harbour bridge site visit, high vests and hard hats made available to the visiting team from AUT. Strict social distancing and safety protocols were adhered to during all supervised data collection visits. During the second visit, the author, the supervisor and another AUT student were extensively briefed about safety protocols and possible hazards. During data collection and other fieldwork (riding the BTM) (Fig. 13), two NZTA staff members always supervised and accompanied the visiting team. The safety gear and safety compliance form can be observed in Fig. 14.

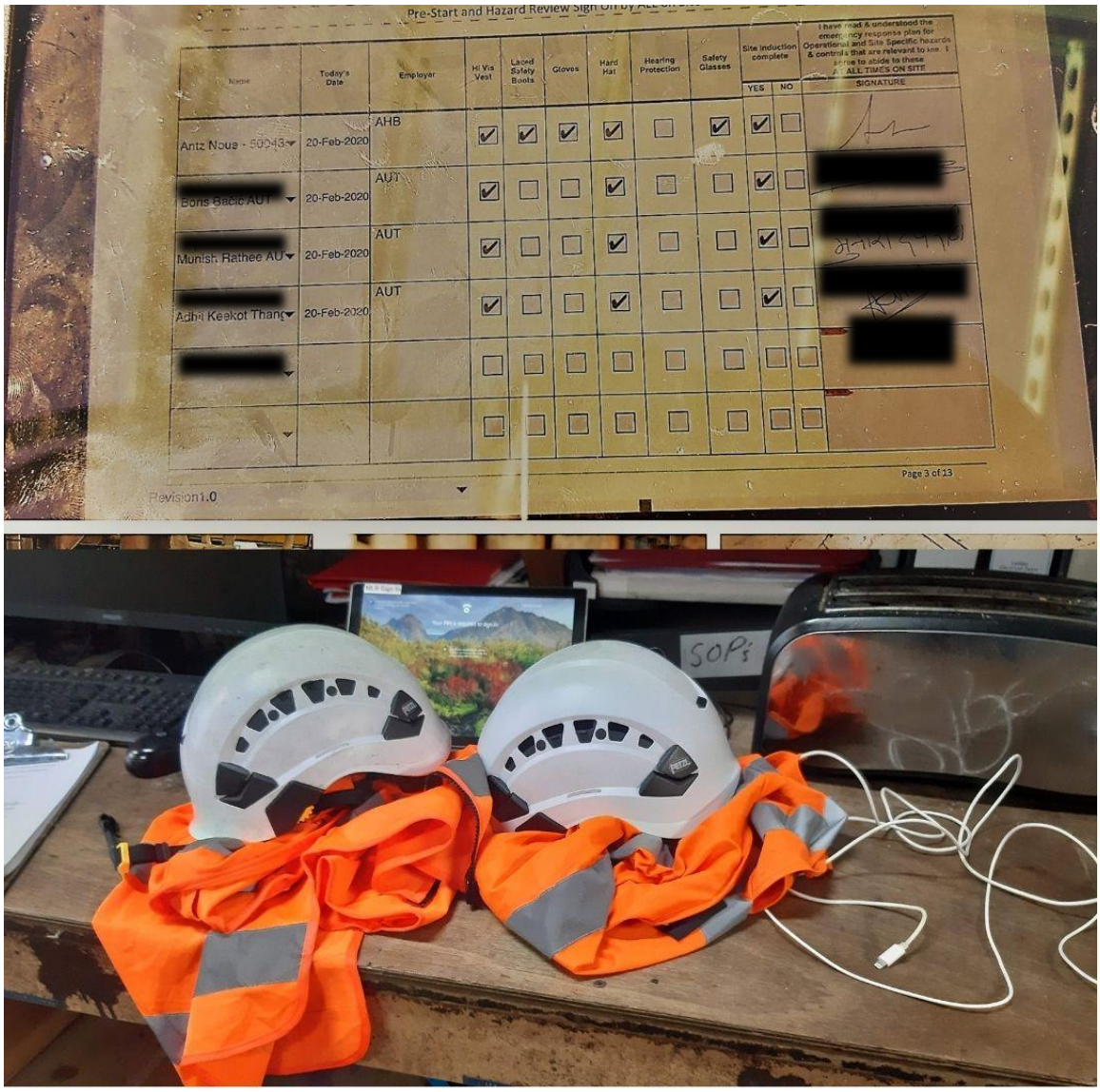

Fig. 14: Hazard review form and Safety gear provided by NZTA during the site visit.

3.2.2 Camera Mounting and Supervised Recording

The front BTM arm was the optimal location for mounting the GoPro cameras (Fig. 15). The GoPro camera mounted at the rear BTM arm had a very close pin ROI view. Only half of the pin is visible from the rear camera at some junctures when the slop is upwards. However, it provided a different field of view and camera angles for variety in datasets. Cameras mounted on outstretched BTM arms provided an ideal vantage point for high-quality video recording. During

the third and fourth visits, a waterproof GoPro 8 camera was used to counter the eventualities of rain. The fourth session used the same GoPro 8 camera provided by NZTA to record video during heavy rain.

Fig. 15: A. View from inside the moving BTM bogey, B. Mounted camera on stationary BTM.

The author used GoPro apps installed on the mobile phone to monitor the Video recordings. These apps sync the mobile device with GoPro cameras and provide a real-time update on video recording and remote control of the camera. There were two cameras in action, so the author requested another AUT student to accompany as a second camera operator. The author and his accomplice rode the barrier transfer machine's front and rear bogies. A view of the rear camera can be observed in Fig. 15A.

The speed of BTM was maintained between 6kph to 9kph to shoot high-quality videos while reducing the vibration effects. The camera settings were kept universal to collect uniform videos for pin detection model training. The universal camera setting followed during all video recordings are shown in Table 3 below.

Table 3: Universal camera setting for GoPro cameras used in data collection.

Camera setting attribute	Parameter
Resolution	720
Frames per second	240
Field of view	Narrow
Audio	Wind only
Protune	Enabled
White balance	Auto
Colour	Flat
Shutter	Auto
ISO limit	6400
Sharpness	High
Audio protune	Medium
Auto-rotation	Auto

The data collection process faced multiple challenges like vibrations, camera heating, waterproofing, battery autonomy at high frame rates. The author requested the BTM to stop multiple times to inspect the cameras to ensure mounting contraption integrity for additional safety. NZTA organised four data collection trips attended in person by either the supervisor, the author or both. On request, one 22 minutes long video is recorded by NZTA staff under overcast conditions. In total, 110 minutes long video data collected during four visits. The data volume of the videos is approximately 100 gigabytes. In order to capture the videos under different lighting and weather conditions, the video recording sessions were scheduled based on the weather forecast. Table 44 describes the video recording sessions in detail.

Table 4: Weather and lighting conditions during visits.

Visit No.	Cameras	Recording time	Weather and lighting conditions
1	One mobile camera	Approx. 2 minutes	Overcast and rainy, poor lighting
2	Two GoPros	41 minutes	Sunny with dark shadows, very bright lighting
3	Two GoPros	45 minutes	Semi overcast with patches of sunshine
4	One GoPro	22 minutes	Heavy rain, abysmal lighting

3.3 Boosting Minority Class

Many real-world machine learning models consist of learning from imbalanced data sets. However, training a classification model using datasets that contain comparatively fewer samples of the minority class typically produces biased predictions. These classifiers have greater predictive accuracy with the majority class(es) but lower minority class's predictive accuracy. The imbalanced class problem arises in different disciplines when one of the target classes has fewer instances than other classes. A typical classifier usually ignores or neglects to detect a minority class due to the small number of class instances (Bunkhumpornpat, Sinapiromsaran, & Lursinsap, 2009).

To counter biased predictions or minority class neglect, the author actively tried multiple ways of boosting the minority class.

The prevailing and frequently discussed potential solution was finding and recording the minority class (unsafe pin positions) with sharp frames while driving on the Auckland harbour bridge. However, all the attempts by the authors or the supervisor while driving at 60kph produced unsatisfactory results (Fig. 3a). Al-

ternatively, the author requested NZTA staff to push pins out of position. However, manually pushing the pin out of place proved to be labour intensive and not feasible on the scale required (as described in section 3.3.1).

3.3.1 Manual Pin Adjustment for Recording of Unsafe Pin Positions

At the author's request, the authorised NZTA staff member manually pushes out the metal pins. The idea behind this activity is to record videos with unsafe pin positions in a natural environment and various lightning conditions (Fig. 166).

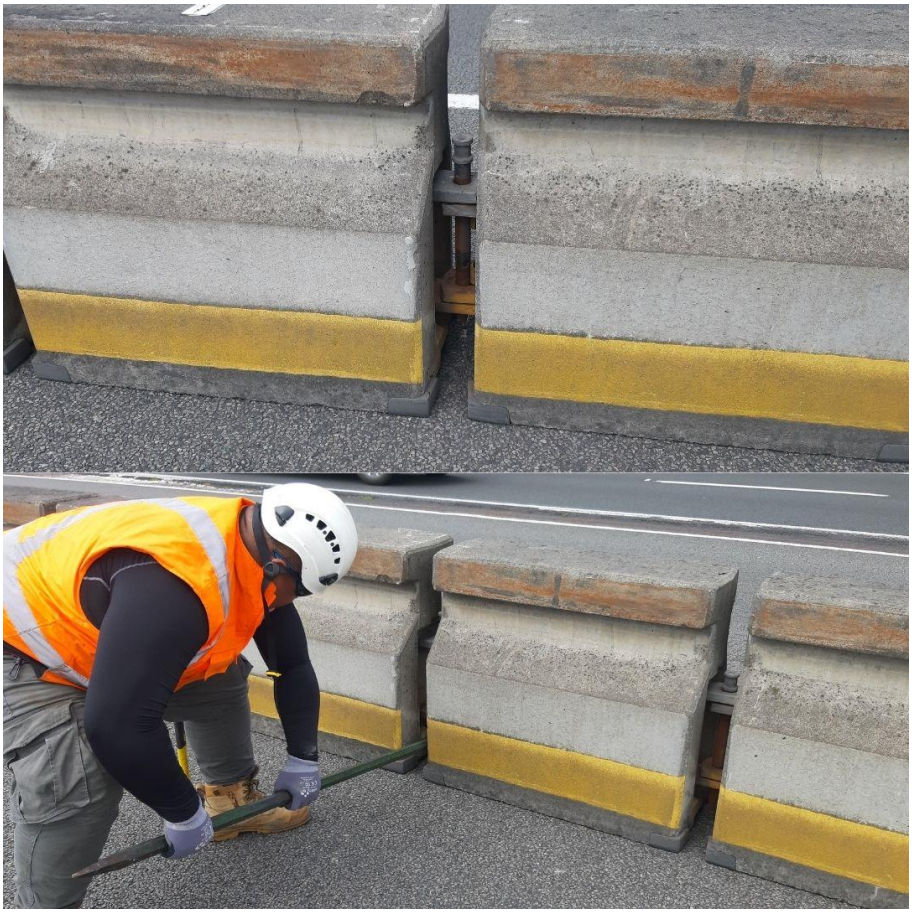

Fig. 16: Pin manually pushed out of place by NZTA staff at the author's request.

The manual altercation process needs three steps: 1) loosening the safety latch using a unique crowbar (observe from Fig. 3: A video frame highlighting the need

for a manual check of a metal pin of moveable barrier block: (a) suspected Pin_Out status and (b) a metal pin safety ring.), 2) removing the safety latch from the metal pin using pliers, 3) hammer out the pin from the bottom. The long process proved that forcing out multiple pins is complicated, time-consuming, and not feasible at the scale required for 'pin detection model' training purposes. Another safety concern was shifting concrete barriers with pins popped out. Trying to lift a semi-loose concrete barrier could result in a disjointed MCB falling in front of oncoming traffic. These potential safety risks and intensive labour tasks forced the author to abandon shooting unsafe pin position videos to boost the minority class.

After failing to produce Pin_Out class frame manually, the author decided to fabricate the minority class frames synthetically. The initial efforts produced unworkable frames with jagged pin edges, as shown below in Fig. 17: (a) shows the failed effort of background cloning to produce synthetic frames These frames were produced using Gimp and used during initial efforts of pin classification.

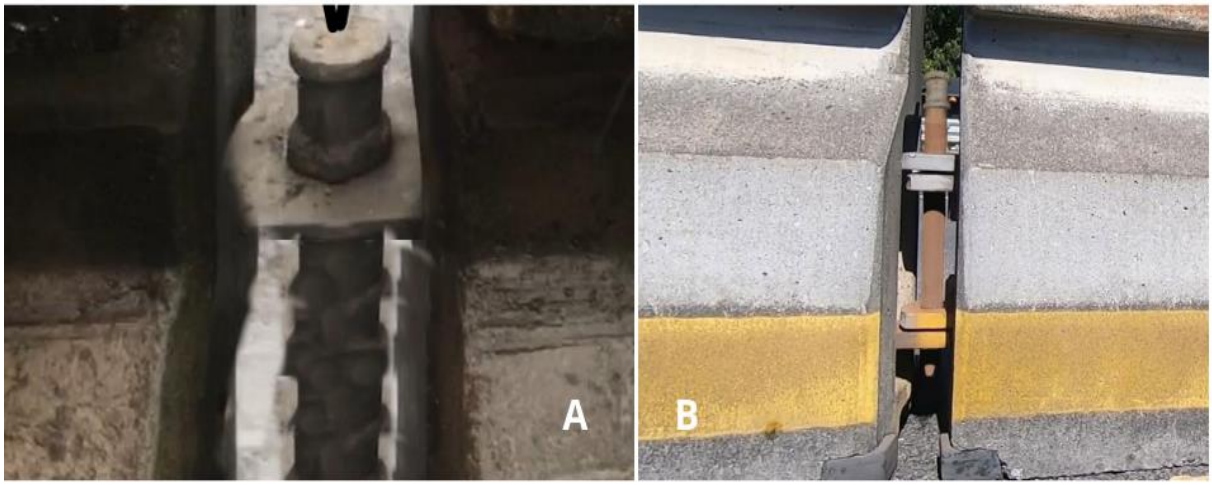

Fig. 17: (a) shows the failed effort of background cloning to produce synthetic frames and (b) synthetic frames successfully created from cloned background pixels.

The author presents a novel method to create synthetic frames to boost the minority class. First, the author used original videos frames to clone the background

and create training data. Then, a process for acquiring a synthetic frame (as described in Fig. 18) lets us generate various degrees of unsafe pin frames using the initial frames. Unlike the standard minority boosting methods, where misclassified examples given identical weights, the author creates synthetic minority class frames applying the cloning method based on a sampling vector taken from adjacent neighbourhood pixels. Thus by implication, updating weights and compensating for skewed distributions.

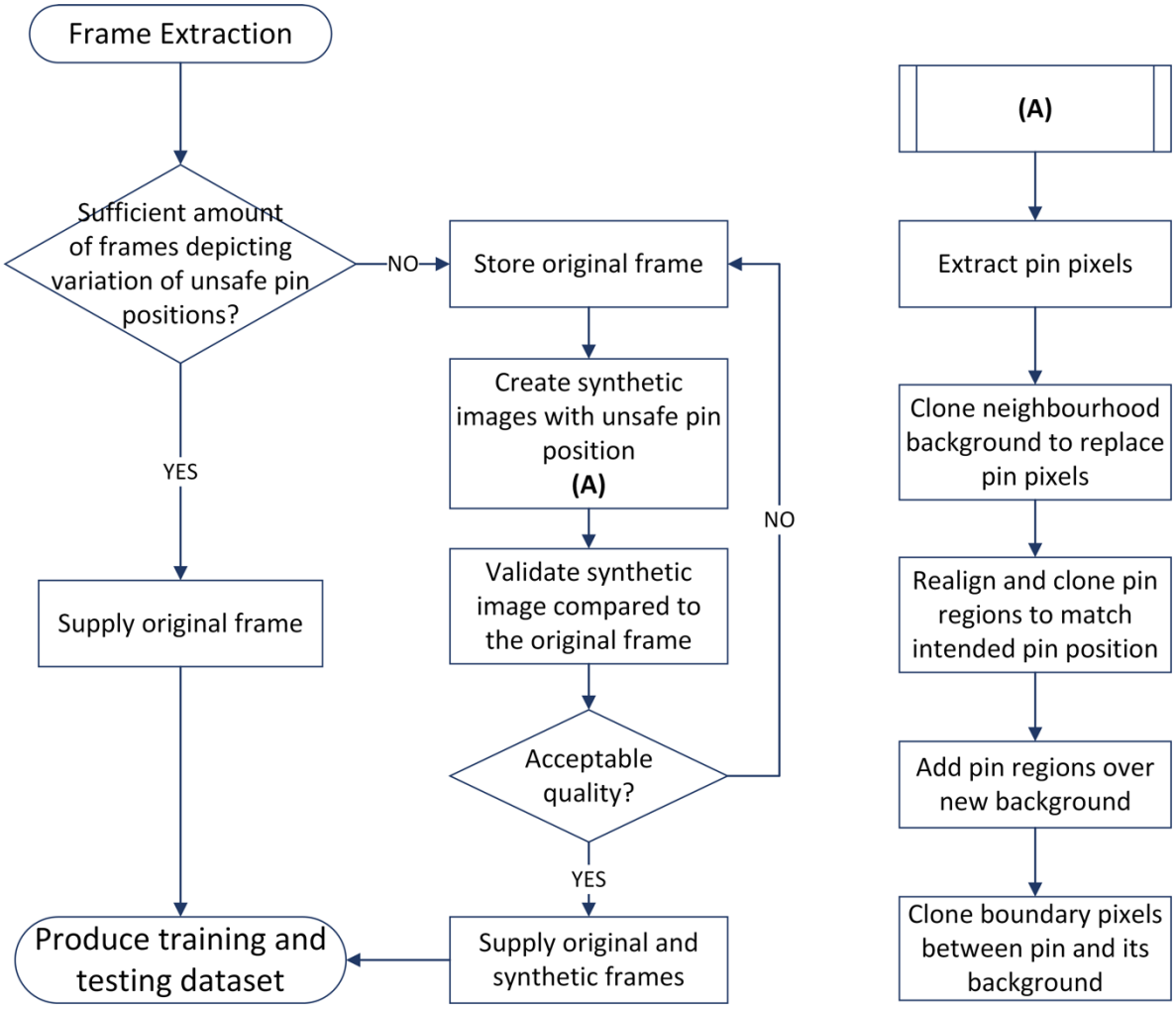

Fig. 18: The flowchart illustrating the process for synthetically creating frames with unsafe pin positions. Adopted from (Bačić et al., 2020).

3.3.2 Data Distribution Analysis Post Minority Boosting

Developing the methodology to create synthetic frames solved a critical modelling gap created due to the lack of a dataset depicting unsafe pin position (Fig. 18, Fig. 21). Thus collected minority class data helped mitigating intital problems associated with small and unbalanced datasets. Nonotheless, full automation of synthtic data processing based on semantic labelling and segmentation was not needed with more data becoming available althought semi-automated creation of synthetic data was found to be a time-consuming process. Semiatomated region clonning with manual pin selection of a single frame could take between 15 to 20 minutes. The cloning process may still prove to be an important contribution and the author plans to automate it fully for future use.

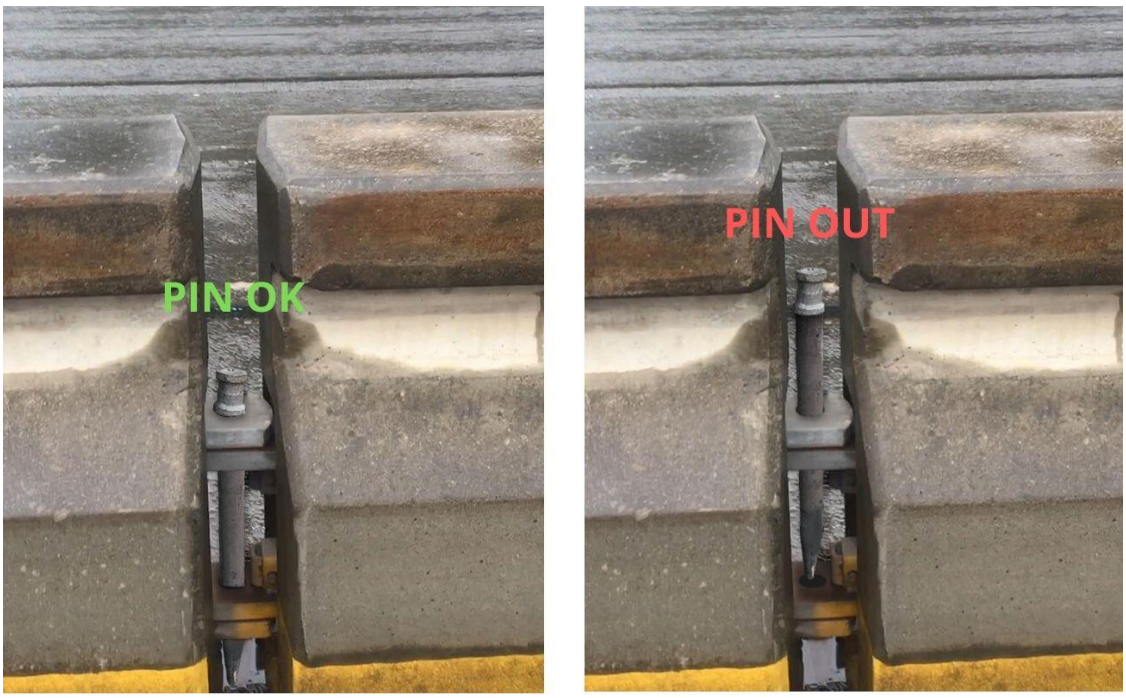

Fig. 19: An illustration of the intermediate data pre-processing results showing: (a) the original frame ('PIN OK') and (b) the synthetic frame showing Pin_Out position labelled as 'PIN OUT' for further modelling purposes. Adopted from (Bačić et al., 2020).

Balancing the dataset with synthetic frames helped achieve the best possible classification on a manually labelled and comparatively small dataset. The

labelled data for network training from the first video can be observed from the bar graph in Fig. 20.

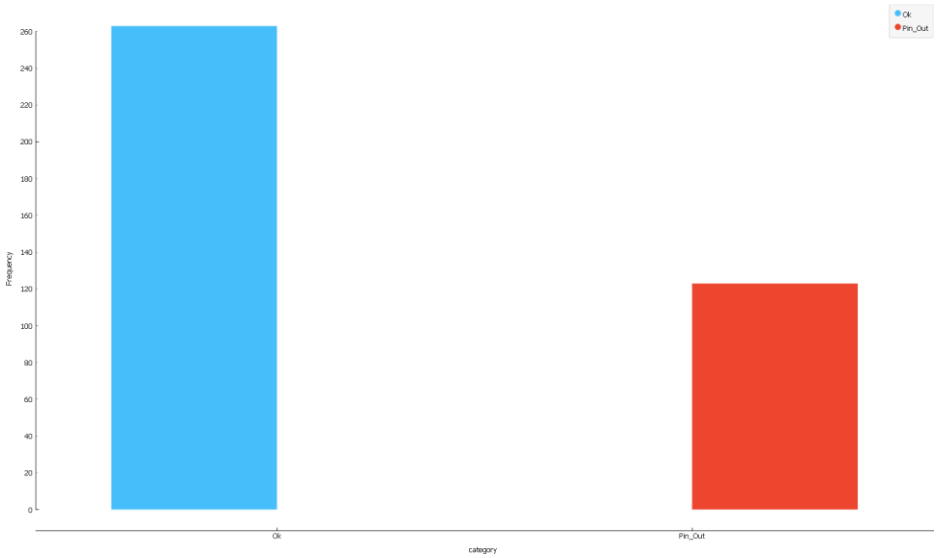

Fig. 20: Bar graph showing data distribution from video 1. Pin_OK (blue) and Pin_Out(red) categories.

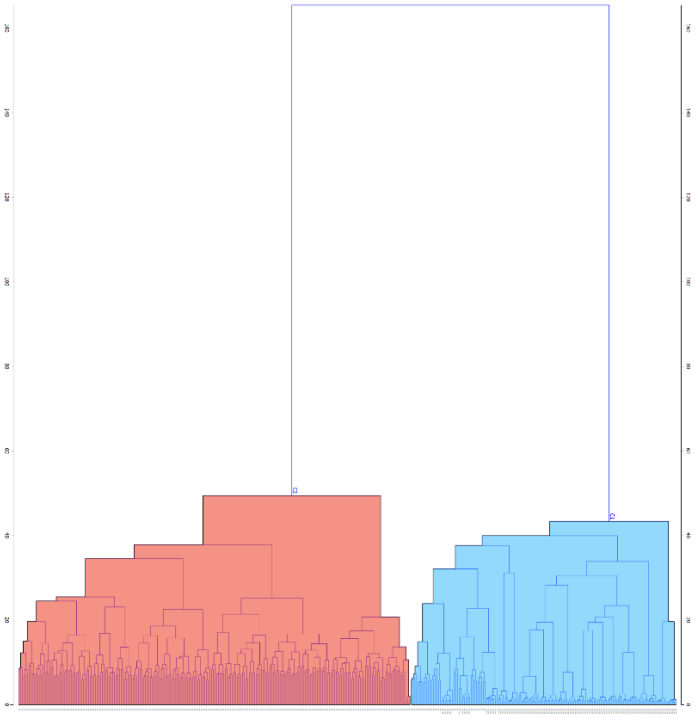

Fig. 21: The dendrogram graph derived from Fig. 19, showing two clusters (Pin_OK (red) and Pin_Out (blue) and a visual separation of generated multidimensional feature space.

Producing synthetic data was a strategic research decision to mitigate lockdown circumstances where (as initially planned) regular data collection was not possible. An additional benefit of creating synthetic data around the system decision boundaries is additional examples close to the expert's decision boundaries (allowing inter-and intra-rater expert label validity) and fine-tuning in terms of Precision and Recall's model evaluation. For automated pin inspection safety, it is important to keep in mind that false negatives are more important than false positives. Additionally, from a safety and model classification error perspective, detecting false positives or a 'false alarm' for the Pin_OK position are preferred over false negatives or Pin_Out position, where positions would be undetected (Table 6: The Cross-validation test and score results updated from initial research.).

Table 5: Initial classification results achieved from data clusters from Fig. 21. Adopted from (Bačić et al., 2020)

Model[2)]		**Confusion Table**		
			Actual	
			PIN_OK	PIN_OUT
Logistic regression	Predicted	PIN_OK	[98.5%	1.5%
		PIN_OUT	0	100%]
MLP		PIN_OK	[98.9%	1.1%
		PIN_OUT	0	100%]
SVM		PIN_OK	[98.1%	1.9%
		PIN_OUT	0	100%]

Producing synthetic frames with different degrees of Pin_Out positions also facilitated fine-tuning the ratio of False Positives (misclassified PIN _OK status) and False Negatives (undetected PIN_OUT status). Achieving False Positive (FP) > False Negative (FN) is feasible by using a labelled dataset where the pin position is borderline unsafe. The minority class is also essential for better

accuracy, which in turns produces more classification model. The accuracy is calculated as

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} = \frac{TP+TN}{All\ detections} \quad 3\text{-}1$$

Here TP is true positive or real-world Pin_Out frames. TN is the majority of frames occurring as safe pin or Pin_OK positions. Precision is the proportion of positive examples that are classified as positive; precision is defined as

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{All\ detections} \quad 3\text{-}2$$

Recall describes how robust the model is or how many of the true samples are predicted using the designed model as a positive example. The recall is described as

$$Precision = \frac{TP}{FN+TP} = \frac{TP}{All\ ground\ truths} \quad 3\text{-}3$$

The three stratified cross-validation process on naturally collected data and synthetically created data is illustrated in Table 6.

Table 6: The Cross-validation test and score results updated from initial research.

3-fold Stratified Cross-Validation		
Model	**Precision**	**Recall**
Logistic regression - parameters: Regularization: Ridge (L2), C=1	0.995	0.995

3-fold Stratified Cross-Validation		
Model	**Precision**	**Recall**
Multilayer Perceptron (MLP) - parameters:	0.995	0.995
Hidden layers: 2 Neurons: [180, 20]		
Activation function: ReLu		
Solver: Adam		
Alpha: 0.02		
Max. iterations: 200		
Backpropagation algorithm		
Support Vector Machine (SVM) - parameters: C=1.0, ε=0.1	0.985	0.984
Kernel: Linear		
Numerical tolerance: 0.001		
Max. iteration: 100		

During one of the later visits to Auckland Harbour Bridge, the author found out that borderline Pin_Out positions can be Pin_OK and vice-versa. Hence in some cases, even after auto-detection, it would still require visual assessment.

3.4 Pin Detection

The region of interest detection is achieved using region proposal plus template matching, Gaussian mixture model (GMM), and colour based segmentation methods. Region proposal or the *regionpros* based method detects ROI by computing the bounding box region. However, lots of adjustments and readjustments are needed to make the detection and counting process accurate. For example, pin ROI detection based on the colour segmentation detects the pins' colour spectrum, but it also detects other objects with the same colour, especially the parts of road and concrete blocks. Therefore, a combination of multiple methods can help improve the accuracy. Thus, the blob analysis method detects the positions of the region of interest and puts a bounding box around

every ROI blob. Upcoming sections discuss the region of interest detection methods in detail.

3.4.1 Pin Detection using Region Proposal

Regionprops function yields measurements for a set of properties for every '8-connected' element or object in the binary image. The *regionprops* function can be used on contiguous regions to detect the region of interest in video frames. Connecting regions are also called connected components, objects or blobs. For example, a frame comprising adjacent regions might look like this:

1 1 0 2 2 0 3 3

1 1 0 2 2 0 3 3

Here, elements of the image equivalent to '1' belong to the first linked component, whereas the elements of the image equivalent to 2 belongs to the second linked component, and so on. This function computes and returns the area and location of the bounding box of each region, and then the *rectangle* function is used to draw rectangular boundaries around the region of interest.

Fig. 22: Detecting Pin ROI using *regionprops* MATLAB function (illustarted in Table 7).

The *regionprops* in action can be observed from Fig. 22 and pseudo code flow can be observed from Table 7.

Table 7: Pseudocode for pin ROI detection using regionprops().

```
Algorithm 1: Pin Roi Detect
Input: video file
Output: labelled video frames, bbox data in csv format
Obj = Video Reader(video file)
grayFrame_prior = convert videoObj.frame1 to gray-
scale image
for i = video Start to video End
 rgbFrame = videoObj.read (frame(i))
 grayFrame = convert rgbFrame to grayscale image
 fill image regions and holes
 remove small objects from binary image
 Label connected components in binary 2D image
 Measure properties of image regions
 Record length of the largest array
 for i = length of largest array
  detect centroid of the ROI shape
  if the axis length correct
   Label the ROI with a bounding box
   Plot centroid for next frame
  End
End
End
```

3.4.2 Pin Detection with Adaptive Background Modeling Using GMM

Tracking the motion of the Metal pin between two concrete blocks is a facet of the thesis research. The recorded video pixels are split into two groups: one representing foreground movement while the other represents the stationary background (Stauffer & Grimson, 1999). The probability density distribution (pdf) in GMM estimates the intensity sequence of a pixel at position x $[I_{0,x}, I_{1,x}, \ldots I_{t,x}]$. Distribution of K gaussians mixture model at time t can be:

$$P(I_{t,x}) = \sum_{k=1}^{K} \omega_{t-1,x,k}\, \aleph(I_{t,x}, \mu_{t-1,x,k}, \sigma^2_{t-1,x,k}) \qquad 3\text{-}4a$$

Where ℵ represents the probability density function (pdf).

$$\aleph(I, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(1-\mu)^2}{2\sigma^2}\right) \qquad 3\text{-}4b$$

In equation 3-4a, the $\mu_{t-1,x,k}$ and $\sigma^2_{t-1,x,k}$ are the parameters of 'Gaussian mean and variance' of 'k-th' single distribution. The $\omega_{t-1,x,k}$ is the representation of mixture weight to maintain the model. Here k is usually 3 to 5 for the object detection in the video frames.

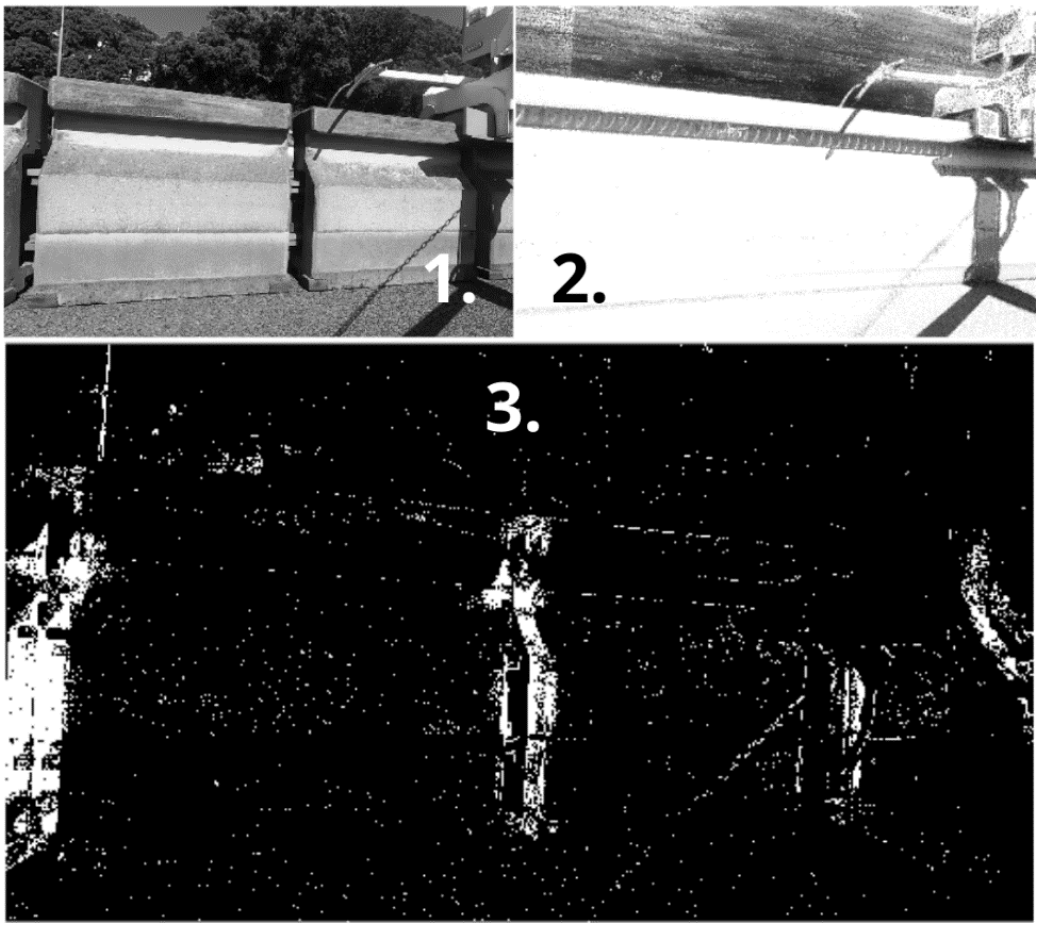

Fig. 23: GMM method of detection 1. Prospected video frame 2. Background 3. Foreground.

The Gaussian mixture model is used for foreground subtraction to detects the movement of objects with the video. The implementation of the Gaussian mixture model for pin detection includes definition, update and foreground detection. Since the model is anticipated to be stable Gaussian distribution, a lower learning-rate is sufficient. The model is employed using Matlab's foreground detector function.

Table 8: Pseudocode for Gaussians foreground detector.

```
Algorithm 2: Algorithm GMM_foreground_detector
Input: foreground image frame from videofile
Parameters:
NUM GAUSSIANS = 10
NUM TRAINING FRAMES = 500
MINIMUM BACKGROUND RATIO = 0.65
Library functions:
Gaussians_foregroundDetector()
Purpose
Detecting Foreground with GMM
1:  Foreground Detector = Gaussians_foregroundDetec-
2:  tor(NUM GAUSSIANS, NUM TRAINING FRAMES, MINIMUM
3:  BACKGROUND RATIO)
4:  videoReader = read(videofile)
5:  while video Reader hasFrame
6:  frame = readFrame(video Reader)
7:  foreground = foregroundDetector(frame)
8:  filled = fill holes in image foreground
9:  cleaned = remove the smaller objects
10: show output image
11: end
```

There are three channels of RGB present in the colour video. The colour does not affect the motion, so the image is converted to a greyscale image by this method. Once foreground pixel detection completed in a video frame, the model produces a binary image that shows the background pixels as black and the foreground as white. The white foreground pixels illustrate the motion of the blobs of moving objects. Most of these are metal pin ROI blobs.

3.4.3 Pin Detection Using Colour Based Segmentation

The pins have reddish-orange and grey colours, with the top mainly being grey and the bottom part reddish-orange to grey.

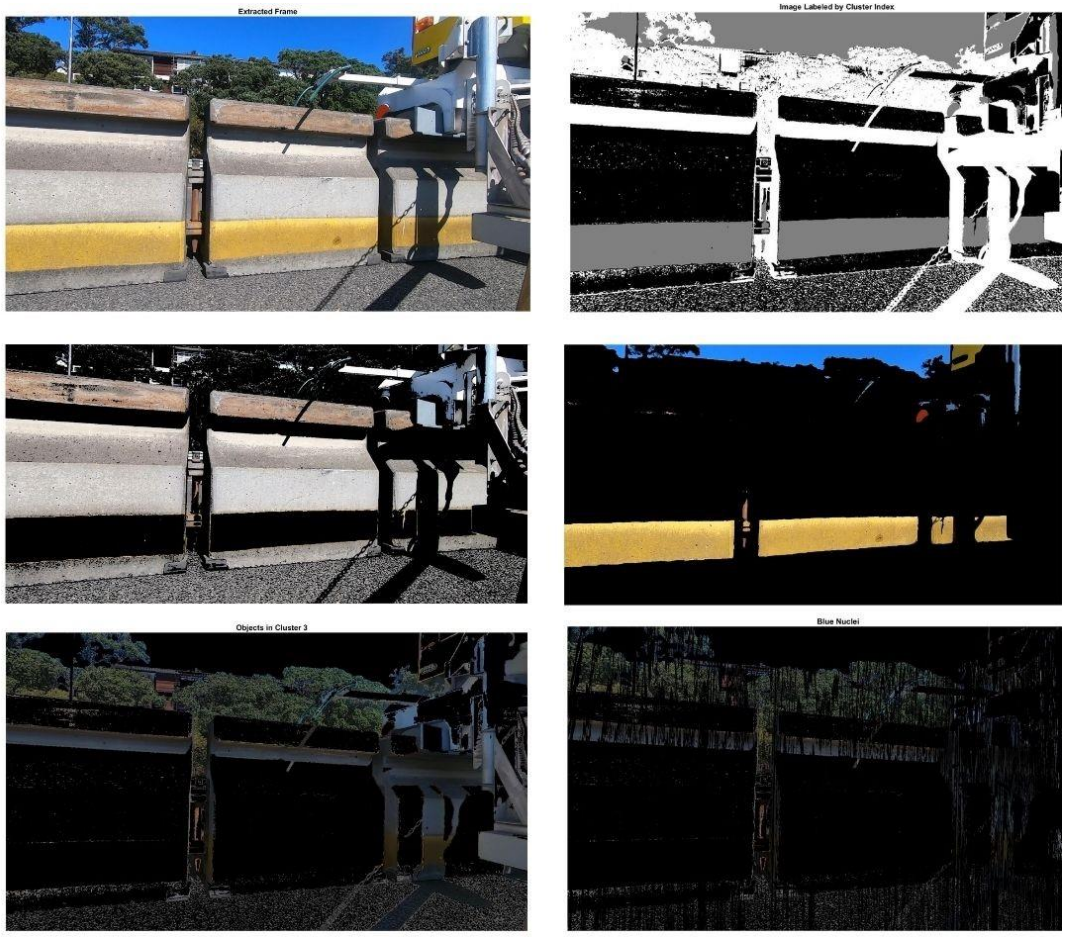

Fig. 24: Example of colour-based segmentation using K-Means clustering.

The video frame is a collection of RGB images. If the RGB colour space is utilised to detect colour, all three channels need to be utilised. Moreover, the colour can also be affected by the lighting condition of the video. During the experiments, we tried LAB and HSV colour spaces with K-Means clustering. However, when transformed to the HSV, the calculations become simpler. The Hue and Saturation are not affected by the brightness because they are calculated in proportion to the chroma.

3.5 Data Labelling

All video frames are labelled using MATLAB ground truth labeller, image labeller apps and other functions. Image and ground truth labelling apps in MATLAB comes with the image processing and computer vision toolbox. These apps are widely used in computer vision models like classification, segmentation, detection, location, and motion recognition, especially the R-CNN series of object detection models.

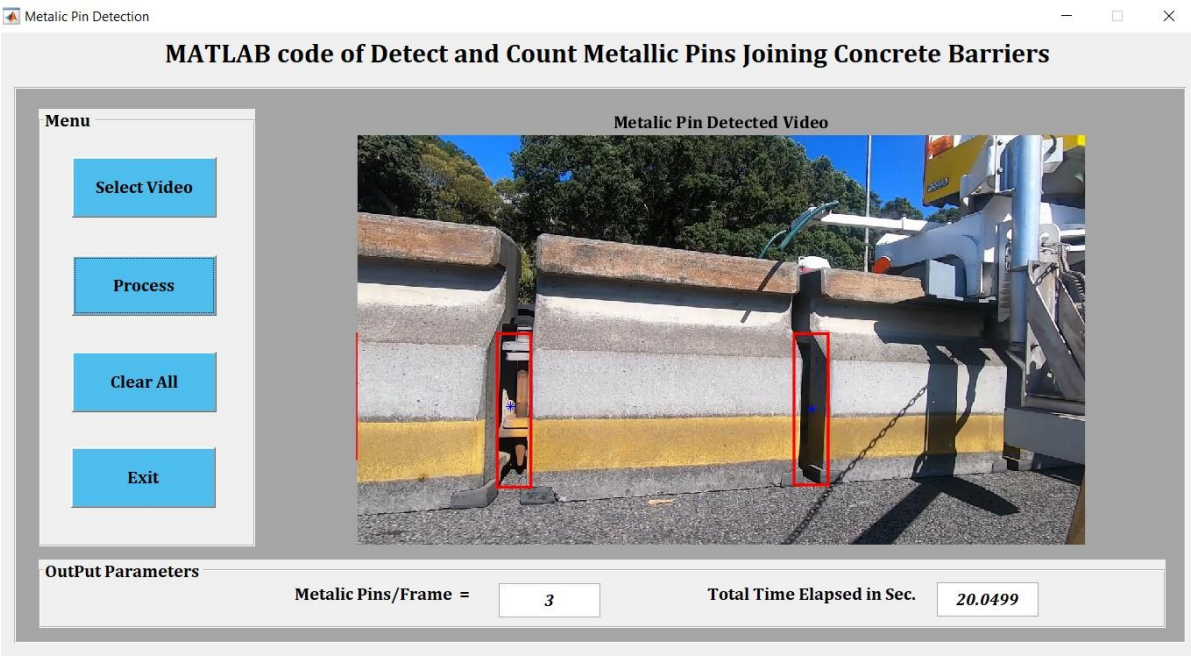

Fig. 25: Automated ROI detection and labelling using *regionprops*. The MATLAB app shown also lets the user export labelling in CSV format.

Since the data is in hundreds of Gigabytes, the author created an automated labelling method using a graphic interface for easy ROI labelling (Fig. 25). The custom automated labelling method used the *regionprops* MATLAB function to measure the properties of image regions. The main properties utilised are " '*Centroid*', '*BoundingBox*', '*MajorAxisLength*', '*MinorAxisLength*'". After detecting the ROI, all data is exported to a CSV file for training purpose.

MATLAB's Ground truth labeller and its automated labelling algorithm with image labelling apps are also used to create labelled information to detect minority class. Two sub-labels (Pin_Ok and Pin_Out) are created under the Pin ROI label.

3.6 Data Augmentation

The data augmentation process helps prevent the networks from overfitting and memorizes the aspects of training images. Moreover, the data augmentation process generates more data from our existing data to help improve the accuracy of the network (Lemley, Bazrafkan, & Corcoran, 2017). According to Perez et al., There are two types of image data enhancement techniques in deep learning, data augmentation based on artificial experience and data enhancement based on machine learning (J. Wang & Perez, 2017).

The experience-based data augmentation includes geometric transformation, Affine transformation, noise injection and random erasing, etc. Geometric transformation carries out transformation based on the original image data, changes the image pixel position, and ensures that the features remain unchanged (Wu, Wu, Cox, & Lotter, 2018).

The *randomAffine2d* function used to create a randomized 2-D affine transformation. The affine transformation is a linear transformation from two-dimensional coordinates. The author used Matlab functions to augment images with rotation, resizing, reflection, translation and shear transformations. To be specific, the author used functions from the Image Processing Toolbox to apply common styles of image augmentation.

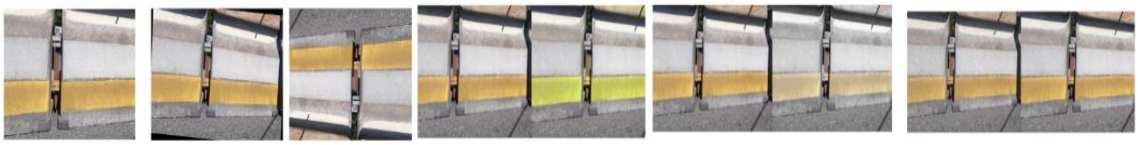

Fig. 26: Affine transformations for image augmentation.

This transformation is a two-dimensional linear transformation. The transformations using the 2X3 matrix, a linear transformation of two-dimensional coordinates using a matrix is

$$\begin{matrix} u \\ v \end{matrix} = \begin{bmatrix} a1 & b1 \\ a2 & b2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c1 \\ c2 \end{bmatrix} \quad 3\text{-}5$$

If the linear transformation matrix R, it is defined as:

$$R = \begin{bmatrix} a1 & b1 \\ a2 & b2 \end{bmatrix}, t = \begin{bmatrix} c1 \\ c2 \end{bmatrix}, T = [R \quad t] \quad 3\text{-}6$$

Hence, the transformation is linear with translation. This image transformation contains several processing methods, including translation, rotation, scaling, and flipping. They correspond to different transformation matrices. In this thesis, we randomly rotate the image by a given angle and change the orientation of the image, as shown in Fig. 26.

The author used two other methods of adding blur and noise. These are two typical operations of image processing used with deep learning applications in MATLAB. The augmentation process randomly distorts the RGB of every pixel. Gaussian and Salt and pepper noises are some of the popular noise modes. MATLAB enables one to apply the synthetic noises to an image using the *imnoise* function. We can also tune the strength of the noises. Randomized Gaussian blur can also be applied to an image; we used the *imgaussfilt* function.

In this research, the author studied and followed Box-Muller's algorithm to generate Gaussian noises using MATLAB. This method is based on two uniformly distributed (0,1) sets of independent and random numbers A and B. These numbers are applied to generate two sets of independent and standard normal distribution random variables X and Y:

$$X = \sqrt{-2\ln A}\cos 2\pi B$$
$$Y = \sqrt{-2\ln A}\sin 2\pi B \qquad 3\text{-}7$$

An exponential random variable can generate the Chi-square distribution of two degrees of freedom. Therefore, the random variable B is employed to select an angle that surrounds the circle uniformly; the exponential distribution is applied to select the radius and then transformed into x and y coordinates.

Fig. 27: Colour transformation, Synthetic noises and blurs.

The author studied and took inspiration from a novel colour transformation technique that combines a table lookup method and a 3D colour space interpolation method using four neighbourhood points (Kanamori, Kawakami, & Kotera, 1990). This algorithm allows realizing a simple structured real-time colour processor applied to various colour transformations with high quality. Similarly, the MATLAB function *jitterColorHSV* can randomly adjust the brightness, contrast, hue and saturation of a colour image. Moreover, various Colour transformations are included, and the user can specify the range of transformation parameters. In addition, users can use basic math functions to adjust the contrast and brightness of the grayscale images randomly.

3.7 Training and Validation

Transfer learning is adopted as the starting point to classify safe and unsafe classes for pin detection. The Transfer learning approach is commonly utilised in deep learning applications and can take a pre-trained neural network and retrain pin detection using previously extracted features.

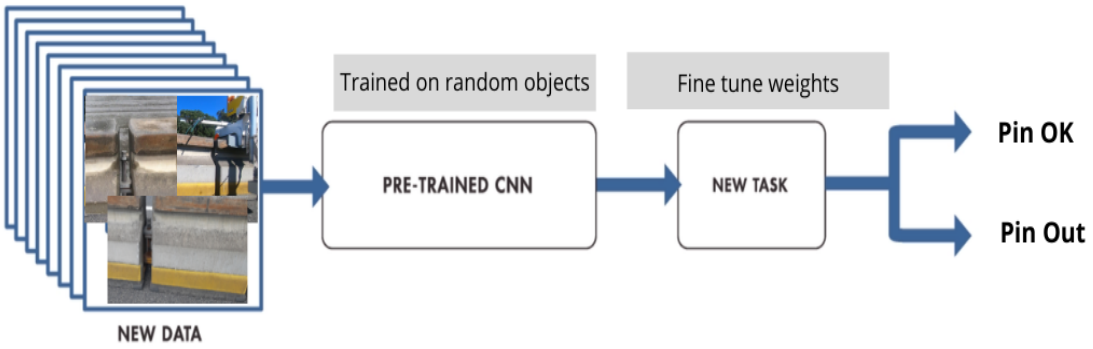

Fig. 28: Transfer learning process for pin status classification.

Fine-tuning the network using transfer learning is faster and easier than training a network from scratch using randomly initialised weights (The MathWorks, 2020). Furthermore, transfer learned features could easily be assigned to a new task using fewer training images. To reduce dependence on domain expertise in feature-engineering and finally produce a robust solution in non-ideal lighting settings (such as cloudy overcast weather, daytime, reflected or low-in-the-horizon sun glare and rain) or colour-based identification, the author used deep learning relying on pre-trained neural networks.

Various deep neural networks can be utilised for object detection on a single frame, including all in Table 9: A comparison of various deep neural networks. Adopted from (MathWorks) However, as a design decision within the scope of the research, the author intended to use less-computationally intensive pretrained CNN, which still would provide the solution to detect video frames in Pin_OK or Pin_Out categories.

Table 9: A comparison of various deep neural networks. Adopted from (MathWorks).

Pretrained Deep Neural Networks				
Model	**Input Image Resolution**	**Parameters (1,000,000)**	**Depth**	**Size**
AlexNet	227x227	61	8	227
SqueezeNet	227x227	1.24	18	5.2
GoogleNet	224x224	7	22	27
Inception v3	299x299	23.9	23.9	48
MobileNet v2	224x224	3.5	. 3.6	53
Resnet 50	224x224	25.6	50	96

The software used is MATLAB with a neural network and computer vision toolbox, using MATLAB version 2020a. In addition, MATLAB includes the function of "*trainFasterRCNNObjectDetector*()" to train a faster RCNN network. A development system that has become available for final demo software development, training, testing and other modelling tasks is shown in Table 10.

Table 10: A development system using NVIDIA GPU parallel processing architecture.

System configuration	
Processor	Intel Core i7 Processor
Memory	32GB RAM
Hard Drive	512GB Solid State Drive
Graphics	NVIDIA GeForce RTX2070 Super 8GB GFX,
Operating System	Windows 10

Resnet 50 with Imagenet is utilised as the core network. As the name suggests, this network includes 50 convolutional layers. The initial experiment established that the GPU RAM is sufficient enough for training purposes. Therefore, we

selected Resnet 50 out of a selection of networks experimented with, such as Squeezenet (18 layers), GoogLeNet (22 layers), InceptionV3 (23.9 layers) and Mobilenetv2 (3.6 layers).

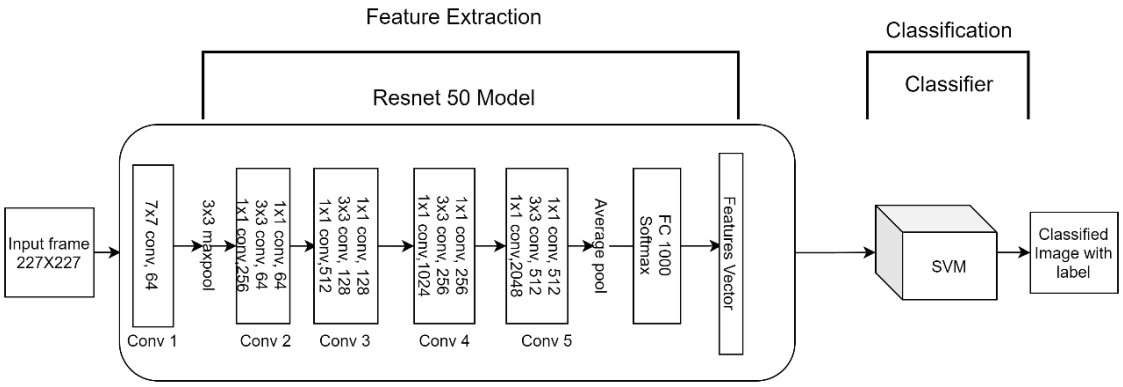

Fig. 29: Classification process using Resnet 50.

Different parts of the six videos with visible metal pin are used for data collection. All of these videos are recorded under different weather conditions, as shown in Table 4. This variation of weather and lighting conditions makes the network model perform more robustly under different light conditions. Initially, around 2300 metal pin block (MCB) images with safe pin positions are collected from around 30,000 frames. All these images include visible pin positions with sharp frames. These 2300 images also include 210 synthetic pin block images with unsafe pin positions. These 210 synthetic images with unsafe pin positions were used for validation mixed with another 200 safe pin position images.

Table 11: Details of used data

Training video session no. (refer to Table 4)	Training Data	Validation data	
	Pin_OK images	Pin_OK Images	Pin_Out Images
1	194	40	40
2	906	100	100
3	1000	40	50
4	500	20	20

These 2300 MCB frames are labelled as Pin_OK and Pin_Out using the MATLAB "Image Labeller" APP. Fig 29 shows an example of Pin_OK labelling using the image labeller app. In addition, the region-of-interest (RoI) is labelled on MCB images with a rectangle label. This establishes the non-RoI part of the images as the potential background.

Fig. 30: An example of Pin_OK labelling.

Once the manual labelling of approx 2300 training images, the labels and images are further augmented using techniques described in 3.5. The augmentation is done because the camera positions varied depending on camera angle and distance. Therefore, a lot more video data is available that can also be used. However, more data processing require a substantial time to train the network because of the hardware limitation. Therefore in this research, there are 2300 MCB images with labels are used for training.

3.7.1 Pin Detection, Pin_OK and Pin_Out Identification

The pretrained Resnet 50 network is utilised as the core network for network training. The training used the 2300 labelled MCB images with 50 epochs. A MATLAB function is used to shuffle the images randomly. The detection overlap with ground truth > 0.5. Each network takes around 2 hours to get trained. Learning rate is a vital training operation parameter as different learning rates give different results.

The training process produced a pin status detection network on MCB images. The trained pin detection network inputted 410 validation images (200 Pin_OK and 210 Pin_Out images) for pin detection testing. The network successfully detected both safe and unsafe pin positions on the validation images, as shown in Fig. 31.

Fig. 31: Pin detection examples.

The network model produces bboxes showing detection of Pin status on images. The bounding box labels the MCB pin images with “Pin_OK, Pin_Out” and a confidence score. After this pin status detection, the image can be classified as to whether the pin is unsafe or not. Depending on camera distance, a video frame can show up to two metal pins. Therefore, if an MCB image is detected with at least one pin out of position, this image shows a positive, identified frame with an alert warning.

In Fig. 31, (2) and (3) are positive or Pin_Out results. Image (2) is a true-positive result; however, image (3) is a false positive result because the network model falsely detected BTM’s part as Pin_Out of position. Images (1) and (4) are negative results because the network does not detect Pin_Out in either of them, despite the image (4) has a Pin_Out of position. Therefore, image (4) is a false-negative result.

Table 12: Yolo v2 detector training process

Algorithm 3: Train Yolov2 detector using Resnet50
Input: labelled video frames
Parameters:
Batch Size = 64
Epoch = 100
Learning Rate = 0.0001
Optimizer = SGDM, ADAM
Output: trained pin status detector
1: Load labelled dataset
2: Set image size and classes
3: Load residual network
4: add last identity connection and plot relu31 layer
5: graph
6: Activate relu_40 for feature extraction
7: Train yolov2 network
8: Test trained model with live feed and video

The thresholding of the bbox confidence scores relates to the Pin_Out and Pin_OK identifications. A low threshold produces more positives and increases the true-positives in the Pin_Out images. However, this also increases the number of false-positives among the Pin_OK images. On the other hand, a high threshold causes more negative results and increases true-negatives among Pin_OK images but increase false-negative results among the Pin_Out images. Thus, an ideal confidence threshold should produce a satisfactory balance of specificity and sensitivity to identify Pin_Out and Pin_OK images.

The author applied the ROC curve to see the effect of thresholding. Then, the network with different thresholds is used on the validation images, producing different results for specificity and sensitivity. The ROC curve details and the learning rates are discussed in the following section.

3.7.2 Learning Rate Selection

The network training process is repeated with the different learning rates. The ROC results can help choose the ideal learning rate. The learning rate tried were 0.01, 0.001, 0.0001 and 0.00001. A ROC curve can be produced with different score thresholds from 0 to 1 to show the network's performance. Observe the learning rate and network model choice flowchart in Fig. 32. The initial training operations are decided, and the training images are input into the YOLO v2 + Resnet 50 network (Fig. 34). The training produces a network model for pin detection.

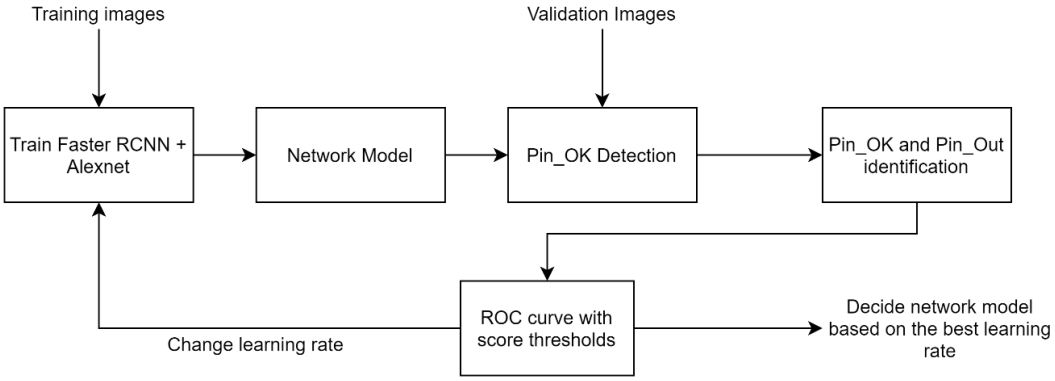

Fig. 32: Training process to find best network model based on learning rate.

This model detects pin status on validation images. First, the results utilised to identify Pin_Out and Pin_OK images. Afterwards, the ROC curves created with different confidence score thresholds using different learning rates. Finally, the ROC curves compared with the good/bad ROC curve model (Fig. 33). The better curve equals the better learning rate, so the better learning rate network is chosen as the pin status detection model.

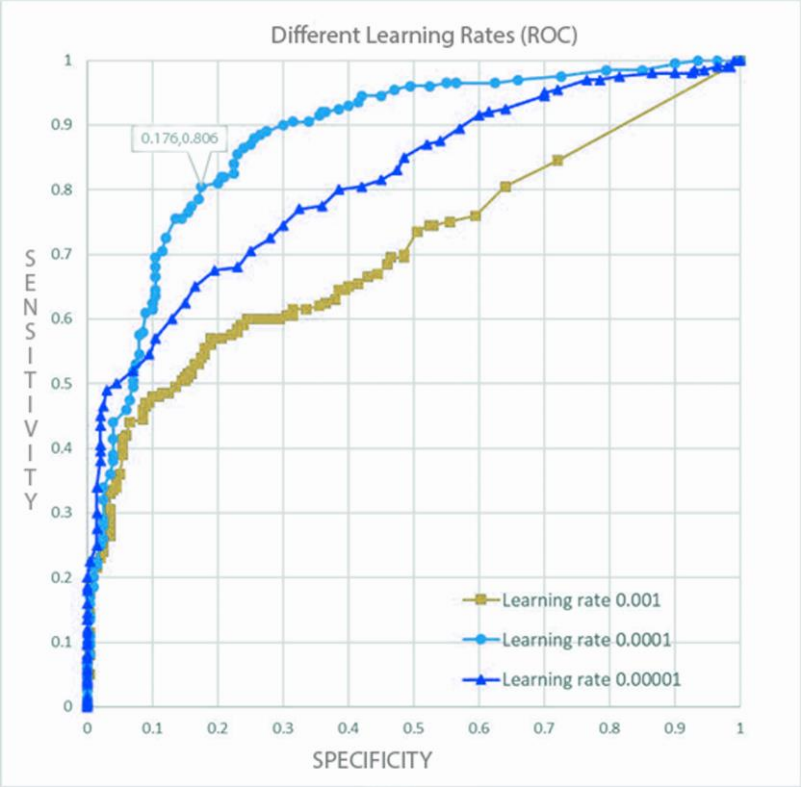

Fig. 33: Learning rate selection based on ROC curve.

The best ROC curve appears closest to the best classification (0,1) point, as shown in Fig. 33, establishing that the 10^{-4} learning rate is optimal. On top of this, the ROC curve with the learning rate of 10^{-4} is smoother than the other two. Therefore, the curve closest to the perfect classification point is (0.176, 0.806). Therefore, the score of 0.37 is chosen as the threshold for detection.

Table 13: Learning rate confidence threshold.

Learning Rate	Threshold	TP	FN	FP	TN	Sens	SPC	Precision	Accuracy	Shrt Distance	Youden Idx
10^{-4}	0.37	162	38	34	166	0.806	0.824	0.820	0.815	0.262	0.63

3.7.3 Pin Status Detection on Videos

After the training completes, the pin status detection model is used with video frames to identify Pin_Out and Pin_OK images. This model works, as shown in Fig. 34.

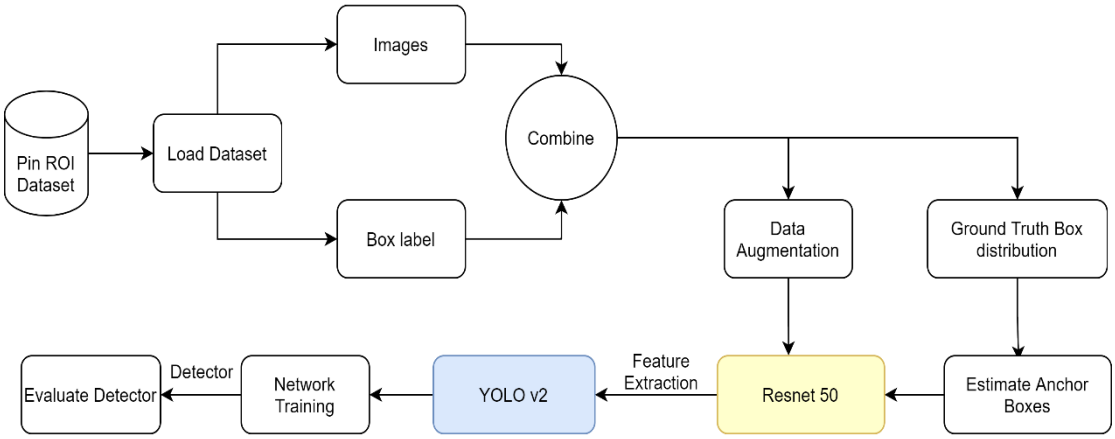

Fig. 34: The pin status detection model.

As shown in Fig. 34, an MCB image goes into the network for pin status detection. First, the network model produces a pin status detection result. Then the result identifies whether the pin staus is Pin_Out or Pin_OK. The model's output of the Pin_Out count combined with the pin tracking model's single-tracking count calculates the pin number with Pin_Out status on the video. The pin status detection model from Fig. 34 combined with the pin ROI tracking model from Fig. 34 produces a pin count based alert system.

Once a pin ROI is detected, the MCB image is sent to the deep learning network model to detect if the MCB image has Pin_Out or Pin_OK. If the MCB image is detected as Pin_Out, the alert is raised with the pin number from the total count. Otherwise, the pin count keeps increasing from the prior video frame. At the end, the total Pin_Out numbers are shown with the corresponding pin number.

4 RESULTS AND DISCUSSION

The pin detection and alert system is discussed in the last chapter. This research includes pin ROI detection and tracking and pin status detection and alert. This chapter begins with the pin status detection and tracking results using the various methods outlined in previous chapters.

4.1 Preparations

Pin ROI tracking is evaluated by choosing and viewing the recorded video frame by frame. However, Some conditions may affect the result.

- If the conditions are overcast, the system does not have enough light to detect the pins between concrete blocks. Therefore, the detection and counting process becomes unpredictable under overcast condition.
- The moving vehicles in the background can create a problem. Although the greyish colour of road and concrete blocks somewhat simplifies the background. Sometimes, the big passing vehicles can make the background noisy. In addition, the broken concrete barrier parts and metal barrier blocks can be merged with pin ROIs, which may reduce the tracking accuracy.
- MCB shadows falling on the pin ROIs, which may be detected as moving objects. Darker shadows under the bright sun might affect the detection, and tracking may not work well.
- BTM speed can distort the image. If the BTM is moving over 6km/h, too much vibration is generated. The vibration can create background noise and distort the frames. The speed can blur the frames to create additional issues.

Two videos taken in the sunny environment were chosen to demonstrate the results and negate the weather conditions. The first video is taken from the front

arm of the BTM and has a larger field of view. There were about two ROIs in a framed view. This video was taken in the bright clear afternoon. The second video was taken from the back arm of the BTM and has a shorter field of view.
Moreover, only one pin ROI is in the frame and that too partially in some frame. Both videos have a different background because these videos were shot with cameras facing different directions.

4.1.1 Creating Synthetic Frames

The author floated the idea of creating synthetic frames during the first semester of masters studies. The frame cloning technique adopted by the group was not robust enough and resulted in an unusbale minority dataset (Fig. 17), which forced the research group to abandon using classification techniques and go for the traditional mathematical approaches.

Before starting PGR 1 research, the author consulted some graphic experts and learned the new way of creating better synthetic frames. The initial frames were created using photoshop. Later on, the author used both Photoshop and Gimp. During the initial phase of producing a proof of concept, 40 synthetic frames were created using frames extracted from a video recorded by the supervisor from his mobile phone. During the second phase of research, the author created around 200 synthetic frames using sharper video frames. The quality difference can be observed between old frames (A) and new Frames (B) in Fig. 35 below.

The process is time-consuming, and a new method is needed to automate the cloning process. The author tried using a photoshop action panel, but the background proved too tricky to automate the process. MATLAB provides some pixel cloning techniques, but more research is needed to find a robust way of automating the synthetic frame cloning process.

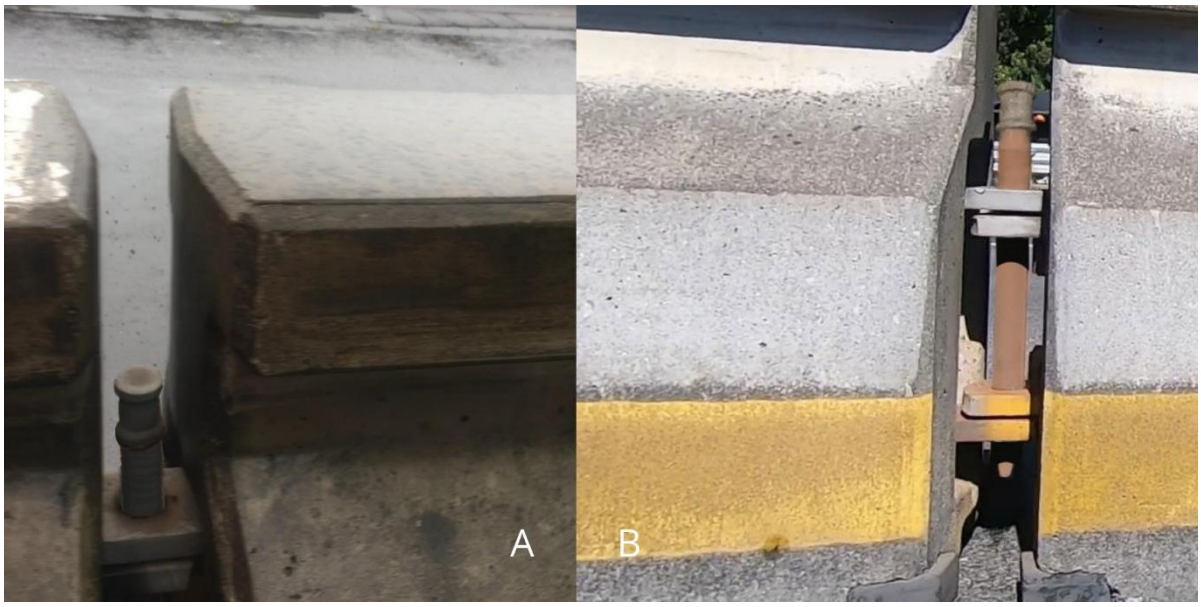

Fig. 35: Quality difference (a) Old synthetic frames vs (b) New synthetic frames

4.2 System Prototype for Pin Tracking, Counting and Alert System

The research assumed that the end-user of the metal pin detection and the alert system would be non-technical, and they need a simplified interface to use the system efficiently. Fig. 36 shows the system protoype with designing pin status detection and alert system app layout and programming its behaviour. The intended prototype system design and codebase integration, was completed in MATLAB App Designer (Mathworks, 2021). As graphical user interaction (GUI) tool and interactive development environment with integrated version of the MATLAB Editor and interactive UI components provided a grid layout manager to efficiently organize user interface and automatic reflow options to alert system app detect and respond. It lets easy distribution of alert system app by packaging components into installer files directly from the App Designer or creating a standalone desktop or web app.

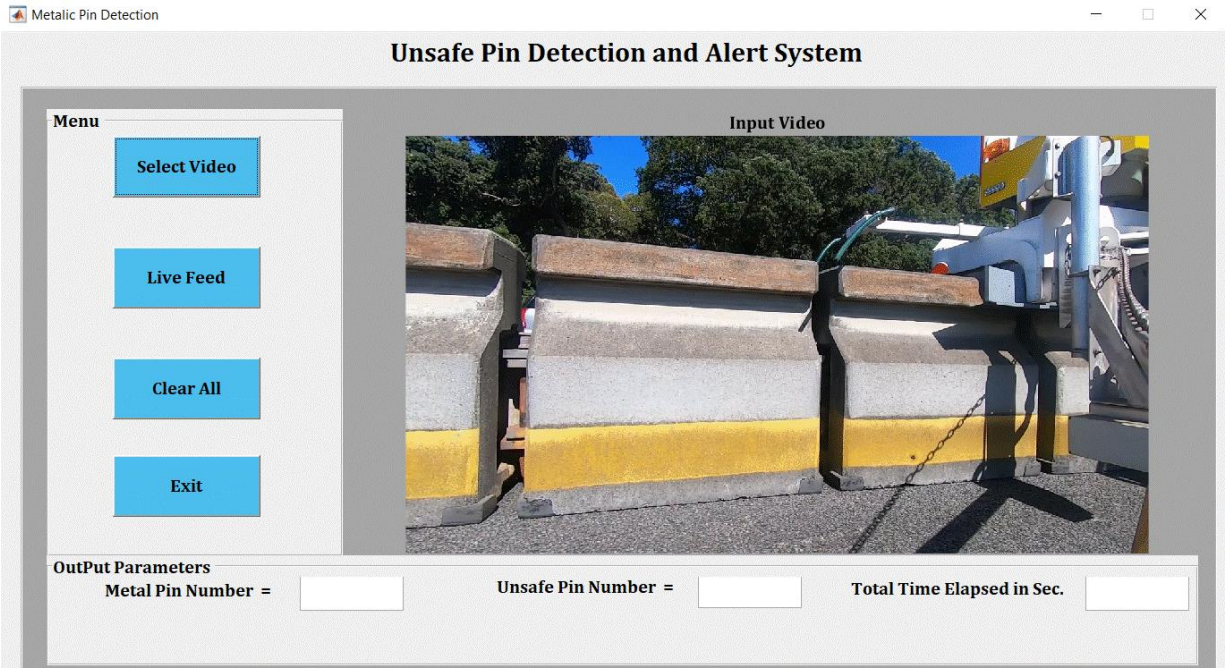

Fig. 36: Interface of metal pin detection and alert app.

The alert system app combines all components of metal pin detection, tracking and alert system. The app interface provides two options:

1. The end-user can use a video or frame sequence to analyse by simply uploading the video to the alert system app. The app loads the deep learning based pin status detection network and starts analysing the video feed. It keeps a single track count of the metal pins. If the alert system detects a metal pin in unsafe position, it raises the alert and flashes the pin number on the screen.

2. The second option is to analyse the live feed. The alert system app provides an option to connect with the live camera feed. Once the live feed starts coming in, the alert system app starts analysing it by initiating the pin status detection network. It keeps track of the actual count of the metal pins. If the alert system detects a metal pin in an unsafe position, it raises the alert and flashes the unsafe pin number on the screen.

The unsafe pin detection and alert system app is easily customisable and can be updated as per NZTA needs. By the time of finishing this thesis, the app is partially working. If NZTA shows further interest in the research, the author will develop it into a viable product and explores the methods to install it onsite properly. However, this must be taken as future work because of limited time and other resource-related limitations.

4.3 Results

The performance of YOLOv2 and ResNet-50 evaluated to find and localize the Pin_Out status. The network model is employed using the workstation provided by AUT with specifications shown previously in Table 10. The configuration is presented inTable 14.

Table 14: Configuration of the detector model

Model	**Batch Size**	**Epoch**	**Learning Rate**	**Optimizer**
Detector	64	100	0.0001	Sgdm
	64	100	0.0001	Adam

The Pin region of interest frame-by-frame tracking is evaluated. Each pin ROI assigned an index and real-time count on the video. If the pin ROI is assigned the same index as in the last frame, then the tracking is correct. However, if the index of pin ROI differs from the last frame, the tracking is not working correctly.

Pin ROI dataset split up into 70 percent training frames, 10 percent validation, and 20 percent testing frames. The configuration of YOLOv2 and ResNet-50 with initial learning rate ($\sigma = 10^{-4}$) and the total number of epochs at 100 as shown in Table 11. The mini-batch size is set to 64. For the optimisation techniques, SGDM

or Stochastic Gradient Descent with momentum (Sutskever, Martens, Dahl, & Hinton, 2013) and Adam (Kingma & Ba, 2014) are chosen optimisation techniques to improve the performance of the detector.

Table 12 show the stats of the training process using SGDM. The SGDM technique achieves a quicker training time than ADAM. Table 12 concludes that the SGDM technique is better than Adam on time and validation Root Mean Square Error and validation Losses.

Table 15: The SGDM training and validation process

Epoch	Iteration	Time Elapsed	Mini-batch (RMSE)	Validation (RMSE)	Minibatch Loss	Validation
10	150	00.15.05	0.91	0.87	0.8285	0.7910
20	300	00.30.20	0.74	0.83	0.5384	0.6761
30	450	00.45.12	0.66	0.78	0.4141	0.6329
40	600	01.00.15	0.62	0.76	0.3968	0.5809
50	750	01.14.50	0.56	0.77	0.3182	0.6150
60	900	01.28.58	0.53	0.76	0.2708	0.6060
70	1050	01.44.55	0.52	0.74	0.2408	0.6116
80	1200	01.58.21	0.50	0.76	0.2018	0.5908
90	1350	02.13.43	0.48	0.75	0.1808	0.5910
100	1500	02.27.23	0.47	0.74	0.1280	0.5819

The results of pin ROI classification and the bounding boxes detected by using Resnet 50 are shown in Fig. 37: The selected pin ROI video frames with bounding boxes. Almost all Pin_OK and Pin_Out have been identified. Also, the pin ROIs that are not in complete view of the camera, the pin ROIs occluded by shadows on top, were identified. There were some instances where pins were too close,

and only half of the pin ROI was shown in the field of view. These frames proved tricky, and the accuracy of the detector dropped in these frames. All in all, the system works robustly in an ideal field of view and good lighting condition. But it needs more labelled data for training on different field of view, lighting conditions and backgrounds. The minority class proved a big challenge, and the author has to spend long hours creating synthetic frames. The automation of synthetic frame cloning needs more work, and after few failed attempts, the author decided to leave it for the future.

Fig. 37: The selected pin ROI video frames with bounding boxes

Table 16 illustrates the classification accuracy, precision, and recall of the two classes after the trained Resnet 50 + YOLOv2 network. The learning rate was set

to 0.0001 and trained the model based on the image resolution 720x1280. It can be observed that the limited minority class created a difference.

Table 16: The performance of Resnet 50 network as a classifier

Classes	Accuracy		Precision		Recall	
	Training	Validation	Training	Validation	Train	Validation
Pin_Ok	0.846	0.814	0.831	0.826	0.802	0.797
Pin_Out	0.623	0.614	0.621	0.612	0.611	0.568

The accuracy of Pin_OK detection achieves the higher accuracy of the two classes (0.846). Meanwhile, Pin_Out detection has a lower accuracy (0.623).

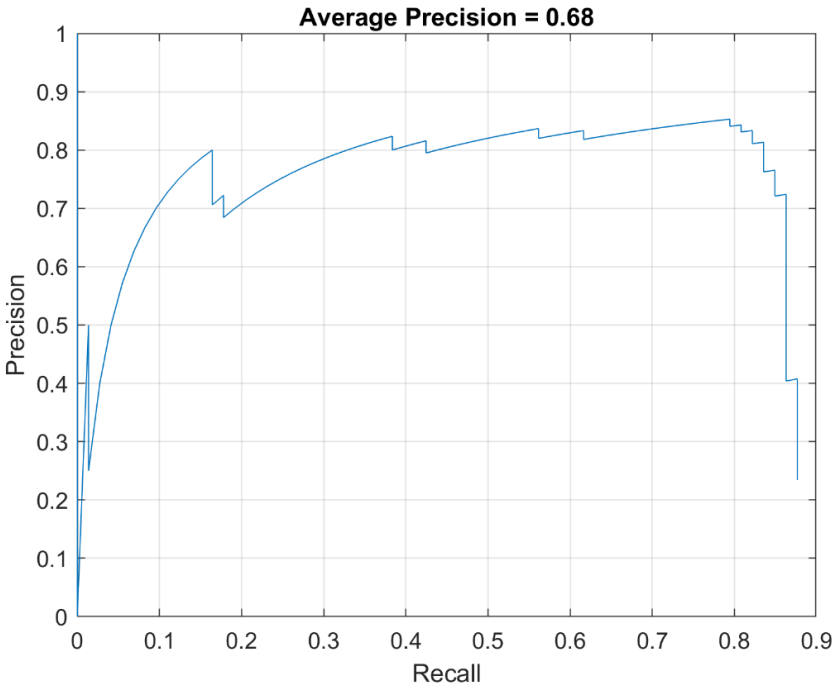

Fig. 38: Overall precision.

Both classes have achieved acceptable results that the average accuracy of the model based on our collected datasets turns out to be 0.826.

4.4 Discussion

The initial recording attempts at high speed by the author from public transportation and personal cars proved challenging. For example, the traffic flow prediction depending on the times of the day may not necessarily come to below the average speed of 20 Km/hr along the bridge, which could allow us to collect frames depicting ideal pin positions at 240 fps. Also, the author tried to rely upon traffic jams for a window of opportunity during rush hours, which also proved unsuccessful due to uneven stoppage time and recording equipment limitations.

Provided that the safety on the Auckland Harbour Bridge (with traffic) is vital, all data collection sessions were conducted under the NZTA supervision. NZTA also provided access to the barrier transfer machine (BTM) and the operation site with safety briefings. For video recordings requiring a clear view of the narrow gap between the movable concrete segments, high-frame-rate cameras(GoPro 5 and GoPro 8) were mounted on the BTM, moving at speeds of around 6 to 9 Km/hr. The videos were recorded under four different weather and lighting condition during different times a day.

However, finding and recording pin out of positions put a hold on the entire research process. After many failed efforts and brainstorming, the synthetic frames were considered as the immediate solution. To demonstrate our progress to NZTA, we delivered a report demonstrating hierarchical clustering and a visual separation of acquired feature vectors related to minority output class by using Pearson and Cosine correlation-based distance measures during the initial stage. Instead of choosing computationally inexpensive distance measures like Euclidean, correlation-based distance measures were chosen because of relatively high feature dimensionality compared to numerous minority class samples, and for computer vision problem areas where extracting relevant information should be independent of light conditions, rain, or a background colour value from a car passing behind a safety pin.

Within this project's scope, we did not try to work with three or more categories of pin positions (including a frame where pin ROI is not visible). However, for the presented system and future specifications (e.g. extending binary classification into the multi-class problem and incremental learning functions), it is possible to include more approaches and inspect the MCB for other forms of damages that might need attention and additional maintenance action. For such situations, we require to collect additional datasets showing additional anomalies in the natural environment and to produce more synthetic data using the previously presented flowchart (Fig. 18: The flowchart illustrating the process for synthetically creating frames with unsafe pin positions.).

The average detection accuracy of 0.826 is good considering the limitations the model training faced. Compared with other region based detectors, YOLO v2 has a higher accuracy comparatively faster speed. The most significant advantage of using YOLO v2 is its speed. It can process 40 to 45 frames per second with accuracy up to 78.6. The YOLOv2 reorganization layer and the depth concatenation layer are used to improve detection by adding low-level image information and improving detection accuracy for smaller objects. The detection subnetwork consists of serially connected convolution, ReLU, and batch normalization layers. A yolov2TransformLayer and a yolov2OutputLayer follow these layers.

The developed MATLAb app presents a platform to expand research work for future applications. It was impossible to utilize all the collected data in the limited time provided by the study period, further cut short by covid lockdowns. The author believes that the collected data can be used further to improve the pin status detection and alert system app.

In the future, the algorithms can be optimized to achieve pin status tracking irrespective of the location of the cameras, for example, the surveillance camera of a floating drone.

5 CONCLUSION AND FUTURE WORK

The Auckland Harbour Bridge is a vital link in Auckland's infrastructure. The traffic flow on the Auckland Harbour Bridge motorway is unevenly balanced but predictable. The uneven morning and evening traffic flow reverse in volume on each workday. The MCBs are considered an adequate solution for managing shorter distance traffic bottlenecks. However, when compared with land motorways, the Auckland Harbour Bridge, with MCBs separating the motorway lanes, is exposed to various types of vibrations due to long bridge movements (e.g. vertical torque), this is most noticeable around its elevated central part and would raise safety concerns for frequent pin inspections.

The preliminary and subsequent experiments confirmed that a device could detect unsafe pin positions connecting concrete barrier segments directly from the live feed and previously recorded video frames. The videos were recorded under ideal and non-ideal lighting conditions (bright sunshine, heavy rain, drizzle and low stratiform clouds). The lack of video frames showing a Pin_Out status was solved by creating synthetic images showing different unsafe pin positions (with different pin heights) needed for modelling purposes. The expected overall system performance (pin region detection, pin frame selection and pin classification into Pin_OK and Pin_Out status) for the prototype was expected to be above 80%, and for the individual models on limited data to be higher. Individual model evaluation results achieved up to 99 percent accuracy on a classifier for safe and unsafe pin position on a limited dataset, as shown in Table 6, which should be further investigated on a larger and more balanced dataset.

The obtained pin status detection and the alert system shows good precision and accuracy (≤ 8.261). The drop in accuracy from initial classification results (approx. 99 percent down to the current state) is expected due to the larger unbalanced database, lighting conditions and data from different camera angles and distances. Moreover, the experimental evidence from a smaller labelled dataset

also suggests that the produced system is a viable product that does not need rigorous manual labelling. Combining the deep learning (Resnet 50 as the pre-trained CNN) to convert video frames into vectors and using SVM for classification facilitates for analysis of intermediate processing and provide flexibility for modelling: (1) on the future data and (2) with less data labelling if the system be enhanced to classify more than two pin position categories. YOLO v2 provided a robust detection network to develop a MATLAB app-based app as a viable solution.

The primary objective of this thesis was to build a computer vision and deep learning based solution to automate the pin detection and alerting process. The pin detection model detects all pin ROIs using *regionprops* function based detection on a sunny day. As a result, the bounding boxes around the pin ROI are mainly detected and labelled correctly. The detection process is also tested on real time videos with partial success. However, conversely, there are also few disadvantages. Firstly, metal pins are not detected visibly, and during the process, some critical parts like the pointy bottom of the pin body may be lost. Furthermore, the pin status detection can be affected by the reflections of vehicles, concrete block shadows falling on the pin block parts, and the pin's angle relative to the camera. Also, the shadows near or connected to pins' bodies are sometimes also detected as part of the pins. These factors change the pin blobs significantly and result in pin status detection as incorrect.

In summary, the main contributions of this thesis are:

- A universal privacy-preserving monitoring system, globally applicable to similar traffic flow regulation and safety contexts with minimal modifications.
- A novel technique to obtain synthetic frames to produce different degrees of unsafe pin positions derived from the original frames for incremental model development and performance tuning.

Practical aspects related to following up work are:

- The presented technological solution provides a low-cost, efficient and near real time automated solution to moveable lane safety and general traffic safety.
- The presented demo system is not meant to replace human inspection but to add another layer of safety inspection. While increasing the frequency of inspections, preserving commuters' privacy and ability to create digital data records providing evidence and potential data analytical insights to answer questions such as: are there regions where unsafe pin positions are more likely to occur?
- Scientific work supporting the transition from MVP to future production systems will involve incremental and adaptive model development and its performance improvements over a more extensive dataset available in the future. Other considerations include additional data visualisation insights such as visual clustering of generated feature space (Fig. 21), other hybrid approaches (including further evaluation of methods from reported literature review needed for expert-driven feature engineering, data preprocessing combining traditional machine and deep learning approaches).
- For our industry partner, NZTA, this thesis allows independent software development such as transcoding, e.g. from Matlab into Python environments with additional literature background and information intended for future project advancements. Furthermore, regarding the production system and target platform, unlike typical IoT infrastructures with resources favouring expert-driven feature engineering with traditional machine learning methods, the author's point of view is that the target platform will have sufficient resources to combine traditional and deep learning approaches included in this thesis.
- Knowing that this is one of the NZTA projects intended to modernise our traffic infrastructures and smart cities in the near future, it is expected that our industry partner will look for code and platform integration on a larger scale.

5.1 Future Work

We are implementing a better video data collection system (with additional photos taken by the NZTA and AHB maintenance teams) than used in initial data collection. We anticipate that the presented system is, as a foundation, universal and applicable globally in similar traffic safety contexts with little modifications. The pin status detection and pin classification results in this experiment are good, but it is expected that the integrative solution combining the pin ROI tracking and alert system performance could be further improved. In the future, better technologies (like Lidar GPS in recent mobile phones) for tracking could be employed to detect the location of pins and their Pin_Out status. The types of Pin classes monitored by the presented PoC are limited into two. In the future, more data can be trained to implement the detection of more categories to aid in AHB traffic management.

REFERENCES

Akbulut, Y., Guo, Y., Şengür, A., & Aslan, M. (2018). An effective color texture image segmentation algorithm based on hermite transform. *Applied Soft Computing, 67*, 494-504.

Ananth, C., Senthilkani, A., Gomathy, S. K., Renilda, J. A., Jebitha, G. B., & Saranya, S. S. (2014). Color image segmentation using IMOWT with 2D histogram grouping. *International Journal of Computer Science and Mobile Computing (IJCSMC), 3*(5).

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence, 33*(5), 898-916.

Bačić, B., Rathee, M., & Pears, R. (2020). Automating inspection of moveable lane barrier for Auckland harbour bridge traffic safety *Springer.* Symposium conducted at the meeting of the International Conference on Neural Information Processing

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition, 13*(2), 111-122.

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International journal of computer vision, 12*(1), 43-77.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem*Springer.* Symposium conducted at the meeting of the Pacific-Asia conference on knowledge discovery and data mining

Busin, L., Vandenbroucke, N., & Macaire, L. (2008). Color spaces and image segmentation. *Advances in imaging and electron physics, 151*(1), 65-168.

Carlson, M. P., & Bloom, I. (2005). The cyclic nature of problem solving: An emergent multidimensional problem-solving framework. *Educational studies in Mathematics, 58*(1), 45-75.

Chen, T.-H., Lin, Y.-F., & Chen, T.-Y. (2007). Intelligent vehicle counting method based on blob analysis in traffic surveillance*IEEE.* Symposium conducted at the meeting of the Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)

Chen, T.-W., Chen, Y.-L., & Chien, S.-Y. (2008). Fast image segmentation based on K-Means clustering with histograms in HSV color space*IEEE.* Symposium conducted at the meeting of the 2008 IEEE 10th Workshop on Multimedia Signal Processing

Chiu, S.-H., & Liaw, J.-J. (2005). An effective voting method for circle detection. *Pattern recognition letters, 26*(2), 121-133.

Cottrell, B. H. (1994). Evaluation of a movable concrete barrier system.

Elgammal, A., Harwood, D., & Davis, L. (2000). Non-parametric model for background subtraction*Springer.* Symposium conducted at the meeting of the European conference on computer vision

Fleyeh, H. (2006). Shadow and highlight invariant colour segmentation algorithm for traffic signs*IEEE.* Symposium conducted at the meeting of the 2006 IEEE Conference on Cybernetics and Intelligent Systems

Friedman, N., & Russell, S. (2013). Image segmentation in video sequences: A probabilistic approach. *arXiv preprint arXiv:1302.1539*.

Ganesan, P., & Rajini, V. (2014). YIQ color space based satellite image segmentation using modified FCM clustering and histogram equalization*IEEE.* Symposium conducted at the meeting of the 2014 International conference on advances in electrical engineering (ICAEE)

Gao, M., Chen, H., Zheng, S., & Fang, B. (2016). A factorization based active contour model for texture segmentation*IEEE.* Symposium conducted at the meeting of the 2016 IEEE International Conference on Image Processing (ICIP)

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence, 38*(1), 142-158.

Ha, J.-E., & Lee, W. (2010). Foreground objects detection using multiple difference images. *Optical Engineering, 49*(4), 047201.

Hassanat, A. B., Alkasassbeh, M., Al-awadi, M., & Esra'a, A. (2016). Color-based object segmentation method using artificial neural network. *Simulation Modelling Practice and Theory, 64*, 3-17.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition Symposium conducted at the meeting of the Proceedings of the IEEE conference on computer vision and pattern recognition

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming & Evolvable Machines, 19*(1/2), 305-307.

heritage, N. Z. m. f. c. a. (1959). *Auckland harbour bridge opens 30 May 1959*. Retrieved from https://nzhistory.govt.nz/the-auckland-harbour-bridge-is-officially-opened

Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., & Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects*IEEE.* Symposium conducted at the meeting of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition

Huang, Z., Wang, X., Wang, J., Liu, W., & Wang, J. (2018). Weakly-supervised semantic segmentation network with deep seeded region growing Symposium conducted at the meeting of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Husein, A., Halim, D., & Leo, R. (2019). Motion detect application with frame difference method on a surveillance Camera*IOP Publishing*. Symposium conducted at the meeting of the Journal of Physics: Conference Series

Ikoma, N., Haraguchi, Y., & Hasegawa, H. (2014). On an evaluation of tracking performance improvement by smc-phd filter with intensity image of pedestrians detection over on-board camera using neural network*IEEE.*

Symposium conducted at the meeting of the 2014 World Automation Congress (WAC)

Iqbal, A., Shah, S. W., & Khan, S. (2014). Non-linear moving target tracking: A particle filter approach. *International Journal of Computer and Communication System Engineering (IJCCSE), 1*(01).

Jain, A., Reddy, H., & Dubey, S. (2014). Automated Driving Vehicle Using Image Processing. *International Journal of Computer Science and Engineering, 2*(4).

Jurio, A., Pagola, M., Galar, M., Lopez-Molina, C., & Paternain, D. (2010). A comparison study of different color spaces in clustering based image segmentation*Springer.* Symposium conducted at the meeting of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems

Kanamori, K., Kawakami, H., & Kotera, H. (1990). Novel color transformation algorithm and its applications*International Society for Optics and Photonics.* Symposium conducted at the meeting of the Image Processing Algorithms and Techniques

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*, 1097-1105.

Kumar, A. (2014). An empirical study of selection of the appropriate color space for skin detection: A case of face detection in color images*IEEE.* Symposium conducted at the meeting of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, H. J., Ullah, I., Wan, W., Gao, Y., & Fang, Z. (2019). Real-time vehicle make and model recognition with the residual SqueezeNet architecture. *Sensors, 19*(5), 982.

Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *Ieee Access, 5*, 5858-5869.

Littmann, E., & Ritter, H. (1997). Adaptive color segmentation-a comparison of neural and statistical methods. *IEEE Transactions on neural networks, 8*(1), 175-185.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector*Springer.* Symposium conducted at the meeting of the European conference on computer vision

Ma, Y., Li, Y., & Zhang, Z. (2015). Moving vehicles detection based on improved gaussian mixture model*Atlantis Press.* Symposium conducted at the meeting of the 2015 International Conference on Electrical, Automation and Mechanical Engineering

MathWorks. *Pretrained deep neural networks(2019)*. Retrieved from https://au.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

Mathworks. (2021). Develop apps using app designer.

Mohapatra, S., Kar, A., Dash, S., Mohanty, S., & Swain, P. (2014). A Novel Approach to Face Detection Using Advanced Support Vector Machine. In *Intelligent Computing, Networking, and Informatics* (pp. 573-578): Springer.

NZTA. *Auckland harbour bridge factsheet*. Retrieved from https://www.nzta.govt.nz/assets/site-resources/content/about/docs/auckland-harbour-bridge-factsheet.pdf

NZTA. (2014). *How to move a concrete motorway barrier*. Retrieved from https://www.nzta.govt.nz/media-releases/how-to-move-a-concrete-motorway-barrier/

Patel, H. A., & Thakore, D. G. (2013). Moving object tracking using kalman filter. *International Journal of Computer Science and Mobile Computing, 2*(4), 326-332.

Peng, X. (2015). Combine color and shape in real-time detection of texture-less objects. *Computer Vision and Image Understanding, 135*, 31-48.

Poe, C. M. (1991). Movable concrete barrier approach to the design and operation of a contraflow HOV lane. *Transportation Research Record*(1299).

Raju, P. D. R., & Neelima, G. (2012). Image segmentation by using histogram thresholding. *International Journal of Computer Science Engineering and Technology, 2*(1), 776-779.

Razavi, N., Gall, J., Kohli, P., & Van Gool, L. (2012). Latent hough transform for object detection*Springer.* Symposium conducted at the meeting of the European Conference on Computer Vision

Rebouças Filho, P. P., da Silva Barros, A. C., Almeida, J. S., Rodrigues, J., & de Albuquerque, V. H. C. (2019). A new effective and powerful medical image segmentation algorithm based on optimum path snakes. *Applied Soft Computing, 76*, 649-670.

Rege, S., Memane, R., Phatak, M., & Agarwal, P. (2013). 2D geometric shape and color recognition using digital image processing. *International journal of advanced research in electrical, electronics and instrumentation engineering, 2*(6), 2479-2487.

Sakthivel, K., Nallusamy, R., & Kavitha, C. (2015). Color image segmentation using SVM pixel classification image. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8*(10), 1919-1925.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85-117.

Schröder, G., Senst, T., Bochinski, E., & Sikora, T. (2018). Optical flow dataset and benchmark for visual crowd analysis*IEEE.* Symposium conducted at the meeting of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)

Shafie, A. A., Hafiz, F., & Ali, M. (2009). Motion detection techniques using optical flow. *World Academy of Science, Engineering and Technology, 56*, 559-561.

Sharma, N., Mishra, M., & Shrivastava, M. (2012). Colour image segmentation techniques and issues: an approach. *International Journal of Scientific & Technology Research, 1*(4), 9-12.

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking*IEEE.* Symposium conducted at the meeting of the Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning*PMLR*. Symposium conducted at the meeting of the International conference on machine learning

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions symposium conducted at the meeting of the Proceedings of the IEEE conference on computer vision and pattern recognition

Tang, J. (2010). A color image segmentation algorithm based on region growing*IEEE.* Symposium conducted at the meeting of the 2010 2nd International Conference on Computer Engineering and Technology

TheGimpTeam. (2020). *GNU Image Manipulation Program*. Retrieved from https://www.gimp.org/

Vallenga, D., Grypdonck, M. H. F., Hoogwerf, L. J. R., & Tan, F. I. Y. (2009). Action research: What, why and how? *Acta Neurologica Belgica, 109*, 81-90.

Varghese, A., & Sreelekha, G. (2017). Sample-based integrated background subtraction and shadow detection. *IPSJ Transactions on Computer Vision and Applications, 9*(1), 1-12.

Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit, 11*.

Wang, K., Liang, Y., Xing, X., & Zhang, R. (2015). Target detection algorithm based on gaussian mixture background subtraction model*Springer.* Symposium conducted at the meeting of the Proceedings of the 2015 Chinese intelligent automation conference

Wang, Z., Jensen, J. R., & Im, J. (2010). An automatic region-based image segmentation algorithm for remote sensing applications. *Environmental Modelling & Software, 25*(10), 1149-1165.

Welch, G., & Bishop, G. (2006). An introduction to the kalman filter. 2006. *University of North Carolina: Chapel Hill, North Carolina, US, 378*.

Wen, Z.-Q., & Cai, Z.-X. (2006). Mean shift algorithm and its application in tracking of objects*IEEE.* Symposium conducted at the meeting of the 2006 International Conference on Machine Learning and Cybernetics

Wikipedia. (2020). *Auckland Harbour Bridge from Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Auckland_Harbour_Bridge

Wilson, S. (2019). *The next harbour crossing: road and rail, or just rail?* Retrieved from https://www.nzherald.co.nz/nz/news/article.cfm?c_id=1&objectid=12210993

Wu, E., Wu, K., Cox, D., & Lotter, W. (2018). Conditional infilling GANs for data augmentation in mammogram classification. In *Image analysis for moving organ, breast, and thoracic images* (pp. 98-106): Springer.

Ye, X., Yang, J., Sun, X., Li, K., Hou, C., & Wang, Y. (2015). Foreground–background separation from video clips via motion-assisted matrix restoration. *IEEE Transactions on Circuits and Systems for Video Technology, 25*(11), 1721-1734.

Yuan, Z.-W., & Zhang, J. (2016). Feature extraction and image retrieval based on AlexNet*International Society for Optics and Photonics.* Symposium conducted at the meeting of the Eighth International Conference on Digital Image Processing (ICDIP 2016)

Yusuf, M. D., Kusumanto, R., Oktarina, Y., Dewi, T., & Risma, P. (2018). Blob analysis for fruit recognition and detection. *Computer Engineering and Applications Journal, 7*(1), 23-32.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks*Springer.* Symposium conducted at the meeting of the European conference on computer vision

Zhang, B., Li, Z., Perina, A., Del Bue, A., & Murino, V. (2015). Adaptive local movement modelling for object tracking*IEEE.* Symposium conducted at the meeting of the 2015 IEEE winter conference on applications of computer vision

Zhang, T., Xu, C., & Yang, M.-H. (2017). Multi-task correlation particle filter for robust object tracking Symposium conducted at the meeting of the Proceedings of the IEEE conference on computer vision and pattern recognition

Zuehlke1a, D. A., Henderson2a, T. A., & McMullenb, S. A. (2019). Machine learning using template matching applied to object tracking in video data.