EXPRESSWAY VEHICLE TRACKING SYSTEM IMAGE PROCESSING

Project Id: 2021-238

Project Final Report

Bojitha Mindula A.M.K

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

October 2021

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

……………………………… ………13/10/2021………

Signature of the student Date

……………………………… ………………

Signature of the supervisor Date

ABSTRACT

Southern expressway system was launched in 2011 as the first expressway system in Sri Lanka and after that, another three expressway projects introduced. Most people tend to get the service of expressways rather than normal ways due to time consumption. However, the number of accidents that occur in expressways are increasing drastically. In the context of Sri Lanka, it has been observed that motor traffic rule violation has become one of the most significant denominators of accidents. Rules have been imposed to control motor traffic rule violations and a separate department called "traffic police have implemented to act upon this matter. Although the existing manual process which conducts only by human observation is not sufficient to detect all the vehicles which violate traffic rules.

Especially in the expressway system in Sri Lanka, most of the issues related to motor traffic rule violations remain unsolved on the grounds because the human eye does not have ability to detect the exact characters of a license plate from vehicles that move fast.

Thus, a surveillance system is required to monitor the vehicles to avoid motor traffic violations and accidents. The objective of the study was to develop a surveillance system to detect motor traffic rule violations in expressway with automatic bill calculating including the penalty fees using image processing and machine learning techniques. The specialty is that it can identify new vehicle models that have arrived in Sri Lanka.
Convolutional Neural Networks will be used for Vehicle identification, classification, and number plate detection to achieve greater accuracy as well as the deep learning technologies will be used to identify new vehicles.

Key Words: Surveillance System, Object Detection, Image Processing, Traffic Rule Violation

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Ms. Chamari Silva and Ms. Geetanjali Wimalarathna, Supervisors, Sri Lanka Institute of Information Technology (SLIIT), for guiding throughout the academic year.

I would like to express sincere gratitude to everyone who helped me to conduct this research project.

Table of Contents

List of Figures

List of Table

TABLES OF EQUATIONS

1.0 INRODUCTION

1.1 Chapter Overview

This chapter acts as the preface to the thesis document of "EXPRESSWAY VEHICLE TRACKING SYSTEM IMAGE PROCESSING". The researcher discusses the problems background of the research project and giving an IT based solution by using latest technologies. The researcher establishes the problem statement, and following to that, proposes the research solution for the problem statement and illustrates rich picture diagrams of the solution. Research aims and objectives are later stated in this chapter.

1.2 Introduction of research project

Video surveillance for monitoring and traffic management in expressway systems is a demanding issue in intelligent transportation. Motor traffic rule violation identification using surveillance video is a helpful computer vision and image processing analysis technique that includes detection of the kind of vehicle, speed of the vehicle, motor traffic rule infractions, and the vehicle's license plate. This project's specialty is auto training for new cars arriving in Sri Lanka. Furthermore, video monitoring is now needed to monitor activities linked to motor traffic and transit, which helps to reduce accidents.

Two types of surveillance systems can be identified.

- Semi-autonomous surveillance systems
- Fully automated surveillance systems

In semi-autonomous surveillance systems, video is captured and transmitted to human specialists for analysis, while in fully automated systems, actions are detected in real time. Non-intelligent video surveillance requires constant human monitoring, which is very difficult.

In Sri Lanka, three expressway systems are presently operational, while another two are in the planned or development stages. The Southern Expressway, Sri Lanka's first expressway, was built from Kottawa to Hambantota, while the Outer Circular Highway and Colombo-Katunayaka Expressway were also created. From Kadawatha to Dambulla, the Ruwanpura Expressway and Central Expressway are still in the planning and building stages. Especially On the expressway, rigorous adherence to traffic regulations is required, since even a little contact between cars or any other item may result in a collision. Not only that, but the degree of the damages also caused by an expressway collision will be greater than in a typical accident. Although non-intelligent video surveillance for traffic rule detection is being used to monitor cars on expressway networks, it is inefficient.

Thus, the authors suggest a new surveillance system for traffic rule breaches that includes the possibility of automatically calculating total service costs for cars that utilize expressway networks, as well as penalty charges for vehicles that break any traffic rule. The authors wanted to simplify the detection of the following situations in a video frame taken from security cameras installed on expressway systems as part of the overall project scope.

- Vehicle identification and classification
- License plate identification
- Traffic Rules Violation Detection
- Speed violation Detection
- Generate the bill at the exit point along with the penalty fees.
- New vehicle types of auto training

Within this broad research area author mainly focused on two aspects as mentioned below.

- Vehicle detection and classification
- License plate detection

1.2.1 Vehicle Detection and classification

The identification and categorization of vehicle types using vision is a critical job for any intelligent surveillance system. As a result, the car accessing the expressway facility will be immediately detected. Additionally, a duplicate instance will be generated for each car detected by the vehicle detection module. In this case, security cameras will be utilized to monitor the vehicle's activities, including its route to the destination. [1]

1.2.2 License Plate Detection

The real-time recognition of license plates is critical in the proposed system. The primary purpose of video processing technology is to identify cars based on their license plates. When a car joins the highway, the vehicle's number plate is recognized, and a real-time vehicle instance is generated. This module will use real-time CCTV camera video and then inspect each frame for the presence of plates. This module's output will be the plate number, which will be used as an input to the vehicle tracking module and the billing module for traffic rule infraction penalties.

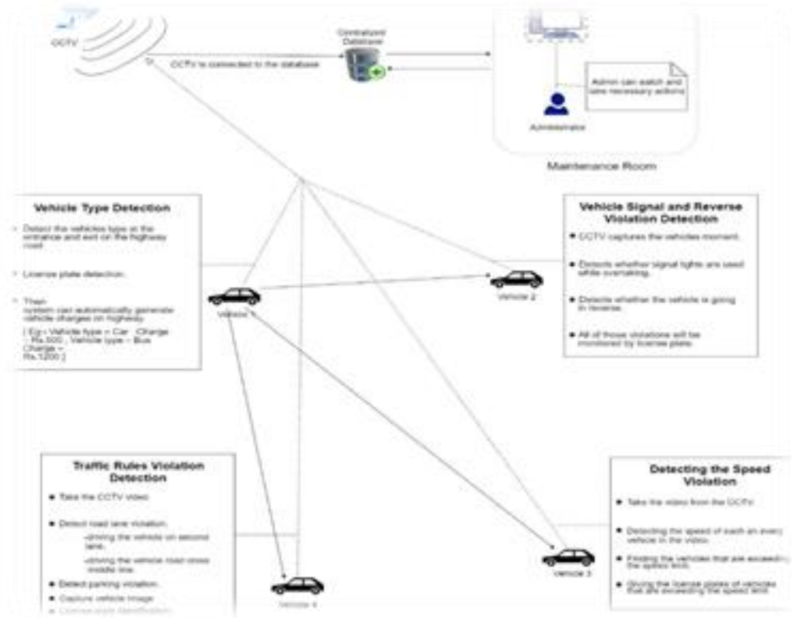


Figure 1 Rich picture diagram

1.3 Background and Literature

With the development of computer vision, machine learning methods, camera hardware, and image processing, concerns about automated video surveillance systems have become more pressing. Manual monitoring has become laborious and time-consuming. As a result, real-time video monitoring for motor vehicle regulation violations has become a hot topic. In a data-driven world, video feeds from CCTV cameras are just as essential as data produced by social media, sensors, and medical systems. Expressway facilities should have the most modern and comprehensive security and monitoring systems available. In Sri Lanka, the surveillance systems for traffic rule breaches are non-autonomous CCTV systems that need constant monitoring by humans. This solution will include an automated intelligent surveillance system that will identify traffic rule breaches and will have the option of automatically generating a bill with penalty costs at the departure point.

Vehicle detection is critical since it serves as the first stage in real-time surveillance systems for motor vehicle offences. Vehicle detection accuracy influences all subsequent stages in the process. Numerous methods have been discovered, along with their associated benefits and drawbacks. Although many studies and initiatives have been performed in this field of intelligent transportation, relatively few studies have been conducted in the Sri Lankan context regarding the intelligent transportation and surveillance sectors.

1.3.1 Image processing

Picture processing is a technique that is extensively used to identify any kind of item or pattern, as well as to conduct operations on the recorded image and recover useful information. Digital image processing is classified into four categories: image enhancement, picture restoration, image analysis, as well as image compression. A high-resolution camera is the primary component utilized in image processing. In addition, an image processing unit or computer having on-board image processing algorithms. A gigabyte of storage to hold the pixels of the pictures while they are being processed.

- **Image enhancement -** A modified picture (ex: median filtering, linear contrast modification) may assist a human viewer in extracting important and usable information from it.
- **Image restoration -** A method of restoring a picture from a degraded form that is often blurry or noisy. A picture may be deteriorated for a variety of causes, such as blur caused by camera shaking. Image noise is often produced by environmental factors like rain, snow, and even thermal signals.
- **Image analysis -** Images with relevant information would be used to outline and characterize things. Picture analysis techniques include edge extraction, image segmentation, texture analysis, and motion analysis.

- **Image compression -** Compressing a picture reduces the number of bytes while maintaining the image's quality. Compression has the benefit of allowing for the storage of more pictures per unit of disk or memory.

1.3.2 Object detection

An image or video may be searched for specific items using the computer vision method known as object detection. In comparison to pictures, the segmented box may assist a person or driver quickly recognize and find things. Object detection's purpose is to transfer this knowledge to a machine.

Object detection may be accomplished using a variety of methods. The popular deep learning methods based on CNNs, like YOLO and SSD, use counterrevolutionary neural networks to learn to identify objects automatically within frames. More information on machine learning is provided below to help explain object detection in more detail.

1.3.3 Machine Learning

Decision-making and prediction are the emphasis of machine learning, a subfield of Artificial Intelligence (AI). Primarily, the goal is to enable the computer to continue developing on its own, without the need for any human input. There are two types of machine learning algorithms: supervised and unsupervised. Supervised algorithms use labeled examples to apply what they've learned in the past to fresh data.

1.3.4 Supervised learning

When using supervised learning, some data is labeled before being fed into the algorithm, which uses the input to categorize the label based on what it has learned. To approximate a mapping function, supervised learning attempts to take fresh input datasets and anticipate the correct output. Until it meets an acceptable standard of performance, the algorithm learning process continues repeatedly until it is complete. Classification and regression issues are subsets of the overall issue.

- **Regression -** an illustration of this issue is the inability to accurately anticipate a continuous amount output.
- **Classification -** is an example of a prediction issue for a discrete class label output.

1.3.5 Unsupervised learning

Data is not labeled to categorize or predict in unsupervised learning, which is a machine learning method. Instead, the algorithm looks for patterns in the processed data that aren't immediately apparent, or it gets insight into the data itself. As a result, unsupervised learning may be very helpful when working with large amounts of data, but it is less precise and reliable than supervised learning. Clustering and association issues are subcategories of the problem that arises.

1.3.6 Machine learning evaluation

Machine learning algorithms may be improved by evaluating several elements of the method, such as its speed, accuracy, and precision. mAP, or Mean Average Precision, is used to measure how well an object identification model performs when trained on a given set of input data. It is necessary to utilize Intersection of Union (IoU) to evaluate if a predicted bounds is true positive, false positive, or false negative while doing AP calculations in object detection Some concepts like accuracy and recall need to be defined to provide a more thorough explanation of these measures. Precision evaluates the accuracy of the predictions made ahead of time, whereas recall reflects how well discover all the positives once that start looking for them.

$$Precision = \frac{TP}{TP + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FP} \quad (2)$$

Equation 1 Precision and recall

True Positive (TP) - The prediction was correct, and the actual result is positive.

False Positive (FP) - The prediction which was made is incorrect, as well as the actual result positive.

False Negatives (FN) - The prediction was incorrect, and the real result is negative.

1.3.7 Deep learning

Deep learning is a machine learning method that relies on learning a suitable feature representation from the input data on its own, rather than having to be explicitly programmed. Deep learning designs include deep belief networks (DBNs), stacking auto encoders, and convolution neural networks (CNNs), which are the most successful picture categorization techniques. As a subset of AI Technology, machine Learning refers to methods that allow a machine to behave like a person. It is inspired by the human brain and develops the pattern by imitating the human brain's own layout, which is composed of neurons.

1.3.8 Machine Learning process

When it comes to artificial intelligence, it's critical to choose the right machine learning algorithm for the job at hand. However, when utilizing machine learning to address issues, there are many additional factors to consider. The answer is based on the already collected and labeled data. When a class is overrepresented, a model may have trouble identifying data instances. Especially when specific items are often mistaken, negative data should be provided. Inconsistencies and outliers should be extracted from the data by trimming it. As a result of this, the algorithm may overfit its test data, resulting in subpar results on real-world data. As a result, instead of learning to generalize, it has just remembered the information. Increasing the amount of data available helps in the resolution of this issue.

1.3.9 Convolutional Neural Network (CNN)

Classification, picture identification, segmentation, and object detection have all been made easier using Convolutional Neural Networks, a kind of Neural Network. Many convolutional as well as subsampling layers make up a CNN. Each layer in the CNN's design includes input and output ports. There are many convolutional layers that have been convolved with a multiplication in the secret layers.

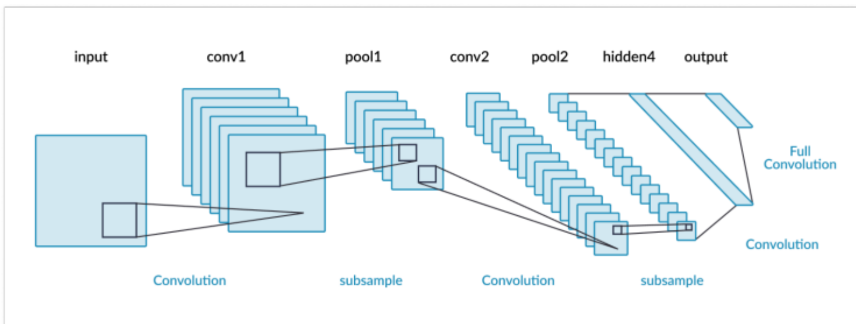


Figure 2 CNN Architecture

In computer vision applications such as image classifications well as detection, CNNs have been used to generate reliable results [5]. CNNs are a type of deep, feed-forward artificial neural networks. Traditional neural networks are like CNNs, however CNNs have more layers. It uses a nonlinear activation to assign weights, biases, and outputs. Neurons in the CNN are organized in a three-dimensional (3D) volumetric) manner. [9]

Neural networks have become a significant area in computer vision-related tasks. Through neural networks, spatial hierarchies of features can be learned automatically and adaptively using backpropagation (building blocks, pooling layers, convolution layers). Also, Neural networks have been identified to be a dominant framework for background subtraction in a video.

Regions with Convolutional Neural Network Features

The use of region-based convolution neural networks (R-CNNs) in object identification is a pioneering technique. The following are two fundamental ideas that have been identified:

1. R-CNN combined with the CNN method.
2. Regional proposals

In the region proposal concept, it is made to locate objects which are dismember using and approach called the bottom to top approach. When the training data is inadequate, supervise trainings will be used for a field of specific fine-tuning process.

Faster Regions with Convolutional Neural Network Features

The deep convolutional neural network is a popular approach. Layers of convolution filter can learn the hierarchical presentation of that input data of images. Low level convolutional layers can learn to capture simple features like textures and lines.

1.3.10 YOLOv3

Real-time object identification algorithm YOLOv3 detects items in video streams, live feeds, or pictures. [6] [7] YOLO detects objects by using deep convolutional neural network characteristics acquired over time. Joseph Redmon and Ali Farhadi are credited with creating YOLO versions 1-3. Hence the authors decided to use the YOLOv3 as most appropriate technology to detect the vehicles.

Version 1 of YOLO was released in 2016; version 3 of YOLO, the one under discussion here, was released in 2018. In comparison to YOLO and

YOLOv2, YOLOv3 is an enhanced version. To put YOLO into action, deep learning libraries Keras and OpenCV are used. Artificial Intelligence (AI) algorithms utilize object categorization methods to identify items within a class as interesting. [8] In pictures, the systems group related things together, but others are ignored until specifically instructed to do so. YOLO is a real-time object detection Convolutional Neural Network (CNN). Using classifiers, CNNs analyze pictures as structured data arrays and look for patterns in the data. When compared to other networks, YOLO is lightning-fast while yet maintaining precision. When the model is tested, it has access to the whole picture and may use that information to make predictions about what will happen next in the image. [9] Convolutional neural network algorithms such as YOLO and others provide a "score" to areas depending on how closely they resemble preset categories.

A high score indicates a positive detection of whichever class it most closely resembles high-scoring areas. According on the areas of the video that score well against preset classes of cars, YOLO may be utilized, for example, in a live traffic stream to identify various types of automobiles.

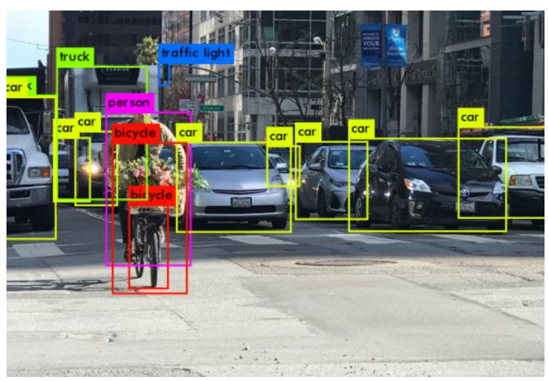


Figure 3 YOLOv3 computer vision example

1.3.11 Speed / Accuracy trade offs

It's difficult to keep them both at a high level, which complicates their relationship even more. These systems are the focus in real-time environments, decisions must be made quickly, which necessitates quick

detection of pedestrians and vehicles. It may choose between making choices quickly and thus having a greater mistake rate or making decisions slowly and therefore having a better accuracy.
The trade-off between detection speed and detection accuracy must be balanced; otherwise, detection speed will be high while detection accuracy will be low. Speed and precision must be matched to avoid sacrificing one for the other.

1.3.12 Transfer learning

Transfer learning, a machine learning technique, is a way for transferring knowledge across domains. Humans' natural ability to transfer information is the inspiration for transfer learning, which utilizes knowledge from such a source domain (a related domain) to enhance learning outcomes. Even with sophisticated computer hardware, training a model from scratch takes time when using transfer learning from those other existing models. Transfer learning makes it possible for the user to make use of previously learned information.

1.3.13 Vehicle type detection

In the areas of highway management, intelligent vehicle identification and has become more essential. However, vehicle detection remains a problem owing to varying vehicle sizes, which has a direct impact on the accuracy of vehicle counts. [1]

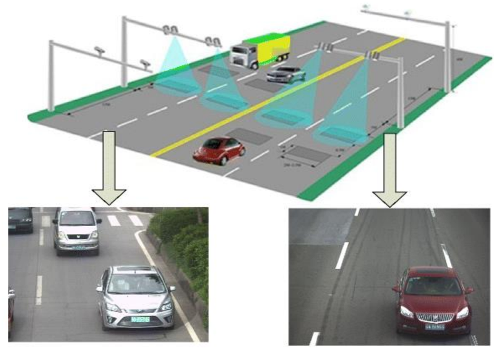


Figure 4 Vehicle type detection rich picture

Mainly, two basic approaches can be identified as in-vehicle detection such as Motivation based approach and appearances-based approach

- <u>Motion-based approach features</u>
  An image's static background will be clearly distinguished from its moving foreground subjects. The pixel difference between two successive frames will be computed using this method. [2] [3]
- <u>Appearance Based approach features</u>
  This method considers an object's colour, shape, texture, and size. Appearance-based methods are capable of identifying and distinguishing fixed items. This kind of approach often makes use of previous data for modelling. The appearance-based feature approach may be used to discuss models such as the part-based model and the feature-based technique.

1.3.14 Classification of vehicles

The Vehicle Classification's goal is to divide recognized vehicles into subclasses like vans, bicycles, and cars. When it comes to categorizing cars, there are many methods to choose from.

- **Geometry-Based Approaches**

  Vehicle detection and classification are performed on highway pictures using fixed cameras. In terms of size, vehicle blobs are employed. Vehicles may be divided into two groups depending on their height and length: buses and non-busses. The Region of Interest characteristics make it difficult to classify cars precisely.

- **Appearance-Based Approaches**

  The appearance-based methodology makes use of characteristics that are dependent on gradient corners and edges. [4] Appearance-based characteristics may be used to identify car models that are based on three-dimensional models, and classification can be accomplished utilizing models.

- **Approaches Based on Texture**

  The texture is another basic parameter that can be used for the classifications. Multi-Block Local Binary Patterns can be used to classify vehicles under this approach. In Local Binary Patterns, it creates a binary string for each pixel.

1.3.15 License plate detection

Each vehicle has a unique identification number that may be used to differentiate it from other cars. It is essential for traffic management and law enforcement to properly recognize the license plate. The same number that appears on the license plate that may be used to get further information about the vehicle's owner.[9]

However, owing to the speed of a moving vehicle, it is impossible to recognize the license plate only by human observation. Identifying a license plate number from an input picture taken from CCTV video is one of the primary goals of this suggested research project. Numerous methods and algorithms for number plate identification and recognition have been discovered, each with its own set of benefits and drawbacks.

Since 2000, Sri Lanka's license plate character format has been standardized, consisting of two English letters followed by four numbers separated by a dash on a white or yellow background, with an imprinted government logo. Additionally, two tiny English letters were added to the license plate to indicate the issuing province, such as SP. (Example: southern province) To get a better perspective, all photos of cars were taken about 3-5 meters distant from the back of the vehicle. Following that, yellow areas were extracted to manage the RGB color space. The authors noted that the RGB color space changes according to ambient illumination conditions and inclement weather. Thus, recorded pictures have been transformed to binary images, with yellow images receiving a value of 1 and other images receiving a value of 0.

Following that, an edge detection method was used to identify discontinuities in intensity data with the assistance of a dexterous edge spotting operator. Following that, the gradient is computed using the Gaussian filter's derivative. Utilizing to determine the edges of recorded images. To reduce noise, pictures having a certain standard deviation are processed using a Gaussian filter. Following that, each point's edge detection and local gradient are calculated. The edge point is described as a spot having a localized high strength. Rise to ridges is indicated by the magnitude image's edge points.

Figure 5 Binarized extracted yellow regions of the image after processing through the edge detection operator.

After the edges have been successfully extracted, a morphological process dilates the image to indicate where the plate is located. To dilate the binary picture, a diamond-shaped structural feature was used. And used a flood-filling algorithm to dilate the photos. It was critical to keep in mind that precise bounding boxes used together with the given regions while creating the model. As a result, bounding boxes were generated for each area. Selecting the highest regions most impacted by bounding boxes resulted in the effective extraction of license plate positions.

1.4 Research gap

Studies on new trends in intelligent transportation have been conducted, and observations and interviews have been conducted on Sri Lanka's current method of graphic rule violation detection and expressway operations to determine where the study needs to be conducted next. Due to the increased difficulty in monitoring fraudulent traffic, it was decided to create a new department called the traffic police. Now every police station has an enforcement section to deal with issues like speeding and other infractions of traffic rules. Even in Sri Lanka's expressway system, authorities utilized the same method and traffic police personnel from the closest branch to watch for traffic rule breaches. According to the current method, police officers are trained and placed in various locations where they observe rule violations and issue penalty papers when they find them. However, this is an ineffective process since human observation is prone to mistake. it is more difficult to get a ticket on the expressway and only when police see a violation will a penalty sheet with the charge and vehicle identification number be issued to each exit point. To top it all off, drivers who want to pay for their tolls on the Expressway may obtain a ticket at the entrance point to do so.

Agents will examine the receipt to determine the particulars, and then manually display the bill in accordance with the receipt information they find. The total cost is calculated based on the distance traveled and the kind of vehicle used. Despite this, each departure point from the incident area must be notified of Chargers-related traffic rule violations. The fee must be manually tacked on by agents at the point of departure.

Even though the Expressway's CCTV cameras are on duty 24 hours a day, seven days a week, human operators are on duty to monitor them as necessary. Because of the existing method' inefficiency and complexity, the authors chose to build a machine learning and image processing-based intelligent automated surveillance system.

Real-time CCTV video of the highway would be used for all detection in the planned system. Detection of the vehicle object and license plate at the entry is possible with the new planned traffic infraction surveillance system. Aside from that, it categorizes the car item to calculate the service fee When a car breaks one of the expressway's regulations, the proposed system would automatically identify it and issue a citation.

Table 1 Existing systems vs developed systems

Existing Systems	Developed System
The Expressway, monitoring is done by human operators when there is an inquiry.	Via an input image taken from CCTV images, recognize a license plate.
Vehicle detection and classification via video sequences.	Vehicle detection and classification via image processing.

Allows to identify with a specific vehicle property.	Detect new vehicle types automatically via front view of the vehicle.
Detection existing vehicle types.	Detection **new** vehicle types.
Identify the traffic patterns via deep learning concepts.	Detect new vehicle types vi deep learning concepts.

1.5 Research problem

How to automate the process of detecting traffic rule violations and calculating service costs, as well as penalty charges, in relation to Sri Lanka's expressway system?

1.6 Research Objectives

1.6.1 The Main Objective

The main objective of this study is to develop an automated surveillance system for the expressway system in Sri Lanka to automatically detect traffic rule violations.

1.6.2 Specific Objectives of the Research Project

Table 2 Specific objectives

ID	Specific Objectives
OB 1	Critically study the current practices and issues in vehicle detection, new vehicle types of detection and number plate detection.
OB 2	Design and develop the functionality of vehicle detection and classification.
OB3	Design and develop the functionality of number plate detection.

OB4	Tests and evaluate the system and compare the functionality against the software requirement specification

2.0 METHODOLOGY

2.0 Chapter Overview

This chapter 'Methodology' is reserved to discuss about the approach to the research and the methodology of conducting research. This chapter put forward, the project approach to alternative solutions. For this purpose, the researcher presents the hypothesis, then going to the discussion of inputs, outputs, process, user, and the features.

2.1 Methodology

2.1.1 Hypothesis

If a proper application format minimizes problems in the business process, it should be easier for them to carry out their work. This is because there are problems with using most of the existing systems. Therefore, a well-analyzed system has been introduced so that all the tasks can be done effectively and efficiently with maximum benefits

The developed traffic rule violation surveillance system has the capability to facilitate for following functionalities.

- Detect the vehicle object.
- Detect the number plate when a vehicle violates a traffic rule. / Detect new vehicle types.

2.1.2 Gathering of Requirements

The software requirements refer to the intended software application's capabilities and characteristics. Requirements Describe the software product's users' expectations. From the user's perspective, the specified needs may be known or unknown, obvious, or hidden, anticipated or unexpected.

Requirement Engineering

Requirement engineering is the method of assembling, analyzing, and documenting software requirements from the client. There are four steps in requirement engineering such as, feasibility study, requirement gathering, software requirement specification and software requirement validation. Requirement engineering is an important part of the project development phase as **"Expressway vehicle tracking system image processing system"** is a real project scenario.

1. **Feasibility Study**

   When the customer comes to develop the desired product, it gives an overview of all the functions that need to be performed and all the features that are expected from any of software. Referring to this information, analysts conduct a detailed study of the expected system and whether it is viable to develop its functionality. The researcher therefore conducted a discussion with the southern expressway officers and conducted a high-feasibility study to identify all the requirements.

2. **Requirement gathering**

   If the feasibility study is positive for accepting the project, the next stage begins with collecting requirements from the system users. So, the researcher communicated systems users to be and some of people who use the Southern expressway as always to find out what the software must provide and what features that software should be included.

Several methods were used to elicit the needs to determine the precise requirements, define the scope, and get acquainted with the current process. The techniques utilized to elicit the requirements were as follows**:**

- Observation
- Interview
- Literature review

✓ **Observation -** Researchers observed and documented the present process that takes place on the highway regarding the system, as well as the gaps and remedies that were discovered.

✓ **Interview -** Interviews with responsible agents at the expressway entrance as well as police officers assigned to the motor traffic unit, who provide important service, were conducted to get a comprehensive knowledge of the existing procedure and its challenges.

✓ **Literature Review -** To obtain a good understanding and collect needs, authors did a domain and existing work literature study.

3. Software Requirement specification

Software Requirement Specification defines how software interacts with hardware, external interfaces, system response time, maintenance capability, security, quality, limitations, and more. In southern express way they primarily demand a user-friendly interface because they are not highly technological people. Maintenance capability, quality, limitations, and much more were considered by the researcher.

4. Software requirement validation.

After developing the requirements specifications, the requirements set out in this document are valid.

2.1.3 Functional Requirements

The functional requirements describe the service that the program should deliver. They provide information on the software system or its components. A function is just the inputs, behavior, and outputs of a software system. It may be a computational, user interaction, data manipulation, business process, or any other function that specifies what a system can do.

The following are the main functional requirements identified by the researcher about the system.

1. Vehicle type detection
2. License plate detection

Those are major roles of the developed system is to identify and classify the vehicle type. It was the key requirements of the research project and the researchers recognized many of its benefits.

2.1.4 Nonfunctional requirements

- User friendly
- Easy to access
- Efficient
- Serviceability
- Reliability

2.1.5 Data Analysis

Data analysis is the process of collecting, translating, refining, and modelling data with the aim of finding the information that authors need.

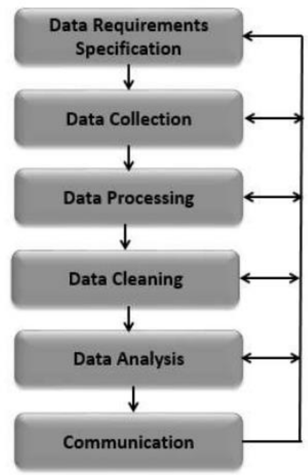


Figure 6 Data analysis process chart

2.1.6 Method

The functional tree diagram is designed to show all the functions required to meet the thesis's objectives.

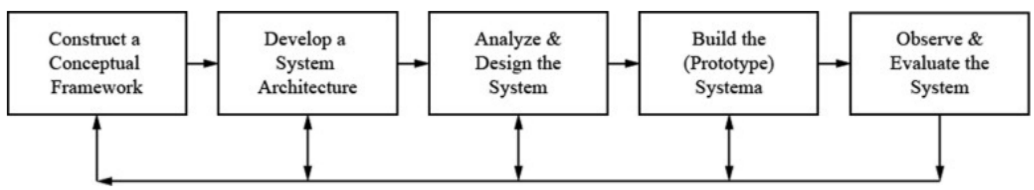


Figure 7 System development research process

- **Construct a conceptual framework**

  Establishing the issue domain and research questions is an essential first step. It will turn up relevant material that will be beneficial to this project. It's also a fantastic chance to learn more about machine learning as well as object recognition, which can subsequently be used to the system's development.

- **Develop a system architecture**

Laptops are used to analyze and identify threats. Real-time video will be captured by the Axis camera mounted on the vehicle's dashboard as it travels along the road. The YOLOv3 algorithm detects objects and divides them into segments using a segmentation technique called bounding boxes that was created. Depending on whatever section when looking at, there are either vehicles or people on the road ahead. The algorithm and identified items may also be monitored by the driver or person utilizing the system and its performance. General Data Protection Regulation requires that no data be kept on the laptop. This should be noted. In addition, the essential stages in the machine learning training process have been discovered.

- **Data collection**

A sufficient data set which includes the video frames captured by CCTV cameras located in the southern expressway will be acquired from the Expressway Operation Maintenance & Management Division. Next, the data were divided into two sections in 70% and 30% ratio respectively for the training and testing purpose.

The data was gathered and determined the model's accuracy depending on quantity and quality. The initial step was to search for annotated datasets for items, such as automobiles, in object identification algorithms. The data were analyzed to understand if there were any red flags, such as missing or incorrectly labeled pictures.

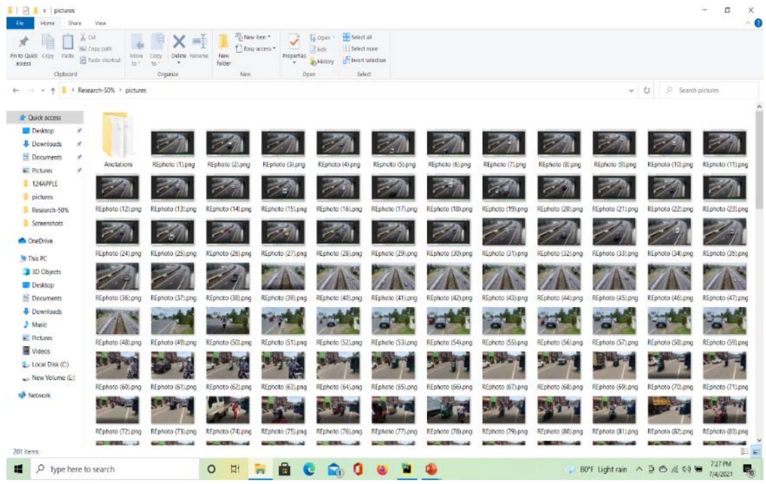


Figure 8 Dataset

Figure 9 Sample labeling

- **Data preparation**

Data preparation will follow once training data has been collected, including putting it into the proper place and getting it ready for machine learning training. Because the sequence of the data should have no effect on what the model learns, the data will be assembled first and then the order will be randomized. The dataset will be deleted if it contains negative data since it has the potential to degrade performance. To better identify objects, the pictures will be re-labeled with new labels based on the new bounding boxes. Most of the dataset will be used to train the model, with the remainder being used to assess the model's performance.

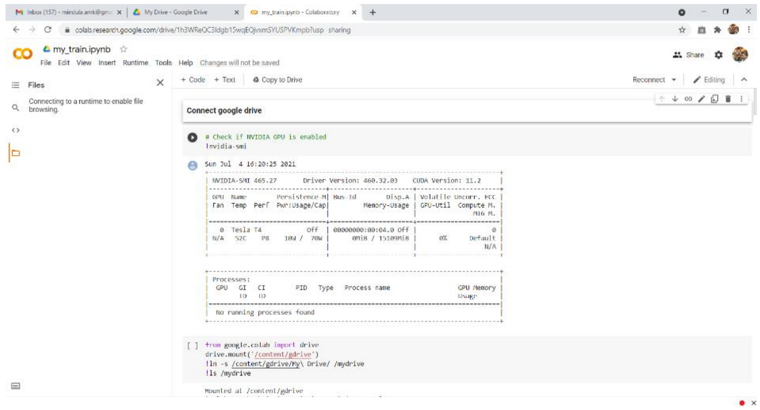


Figure 10 Train the model

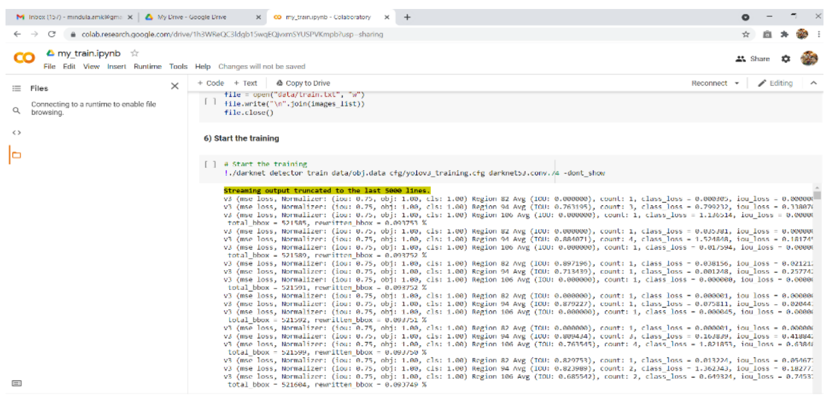


Figure 11 Train the model

- **Choosing a model**

An appropriate algorithm will be selected after examining relevant literature.

- **Training**

Before beginning a machine learning process, familiarize yourself with the preceding stages. A good place to start is by identifying a problem and then developing an algorithm to address it. Data used to train the model must be labeled properly, since objects are often misidentified throughout the training process. Using a model which trains the training data too well may result in overfitting. This may have detrimental effects on the model's performance over time. Adding additional data may help counteract over fitting. For example, if a model is trained to identify trucks, it might be helpful in training another model to detect automobiles. When training a model using fresh data, the goal of training is to improve performance. While working on this project, we enhanced the dataset by include pictures taken from various angles as well as bounding boxes that failed to capture the item.

- **Evaluation**

To see whether the model is accurate enough, it is put through its paces and assessed.

- **Parameter tuning**

The model's input layer was changed. This is the number of steps required to process the whole dataset depending on the batch size. For better accuracy and quicker training, this model is tuned.

- **Predictions**

In this phase, the model will be put to the test in a real-world setting where factors like light as well as weather are unpredictable, in order to see whether it is accurate in its detection and if it has improved over time.

- **Initialize the package/library**

It was chosen to utilize OpenCV-Python since it is a library of Python bindings intended to address computer vision issues because the goal is to identify objects in real-time.

- **Initialize the parameters**

Due to its simpler network and faster detection performance, the YOLOv3 was selected for this application. Because the system must execute object detection in a second or milliseconds, the model's performance must be quick to keep up.

- **Confidence score**

According to the confidence score, it may trigger a response with a value as low as 0. If the matching score falls short of the cut-off point, the system will advise the user to try again later or to try a different query. If the item's confidence score is lower than the threshold, our system will not detect it. The cut-off threshold in our system is configured to activate detection per object.

2.1.7 Application

There are significant initiatives and sophisticated research in vehicle detection, made feasible using license plate detection. As a result, systems like these may be considered, which mainly operate by taking pictures of vehicles and then automatically interpreting the license plate's registration number. The vehicle type detection and recognition system remain an effective tool for automating the laborious, tedious, and physically demanding tasks that workers must deal with daily while also helping to identify vehicles quickly from among the hundreds of thousands of vehicles that are seen during routine patrols. A system that is more vulnerable to false incursions is one that uses License plate detection in collaboration with vehicle shape detection.

2.1.7.1 Vehicle Type Detection and Classification

The vehicle detection will be carried out Using RCNN (Regions with Convolutional Neural Network). To get accurate results this phase will be carried out according to the following steps.

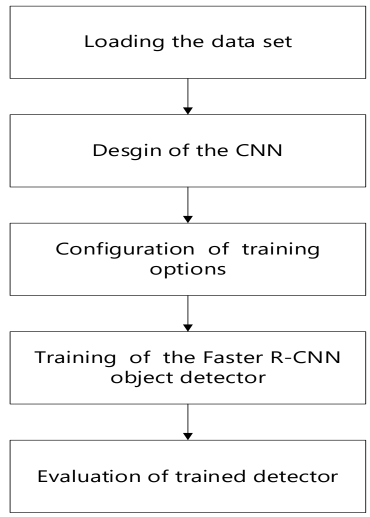


Figure 12 Vehicle type detection

The vehicle object instance created at the entrance will be used as the input to other functionalities. This system's goal is to identify the vehicle's type. A car's form may be divided into a series of different categories.

2.1.7.2 Vehicle type detection data set

The data was collected in the early morning hours of 7:00 a.m. to 9:00 a.m. on several days. This hour was selected due to the large number of cars that enter the parking lot. It was ideal timing to obtain the greatest number of cars in the shortest period. Cameras were stationed at the entrance. The video was then analyzed frame by frame, and the specific frames containing various cars were retrieved.

After creating the database, the images get processed before applying the algorithm. This pre-processing is required to remove the noise that surrounds the vehicle. So, this helps in keeping the image's backdrop constant across the data.

1. Preprocessing

When the picture has a single-color backdrop, the system works. The outlines of the car are clearly apparent on a one-color backdrop. All the extracted pictures were then manually processed such that they all had the same white backdrop. That need two things to obtain the form of a vehicle: foreground and background. The following are the different processes involved in determining the class of any vehicle:

2. Normalizing the image

The photos were shot using a camera that has native resolution pixels. The program's calculation time may be extremely long at this resolution. In addition, the amount of space required to store these pictures will be substantial. As a result, pictures were normalized to specified resolutions to make the system faster as well as fewer resources.

3. Background subtraction

This technique will remove the backdrop from the foreground to create templates for various vehicles. Figure 8 shows the outcome after background removal. After removing the backdrop, the resulting picture is utilized to generate a binary mask. These masks are used to make templates for various forms. These templates are subsequently utilized to determine the form of the vehicle and to obtain the closest match of the vehicle.

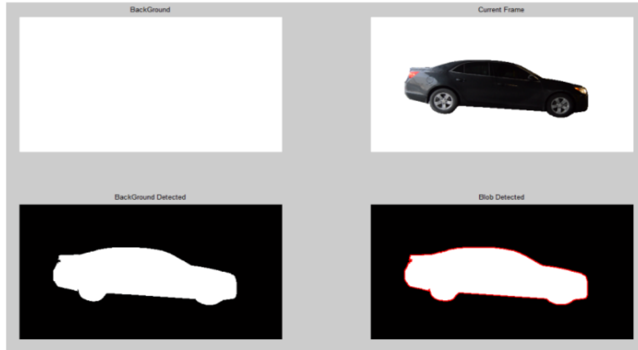


Figure 13 Background subtraction and Blob Detection

4. Template creation

After removing the background these templates are saved in memory so that they may be matched to the fresh batch of pictures to recognize the vehicle's form.

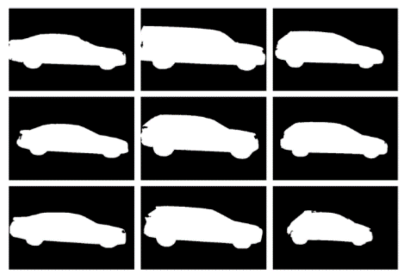


Figure 14 Templates0

5. Template matching

After removing the background these templates are saved in memory so that they may be matched to the fresh batch of pictures to recognize the vehicle's form. The program compares the present vehicle to all of the loaded templates of various forms.

2.1.7.3 License Plate Detection

The methodology for the number plate identification of a vehicle is illustrated in this phase. After detection, the vehicle object captured vehicle image will be the input to this module. The flowchart of the proposed method is illustrated in the following figure.

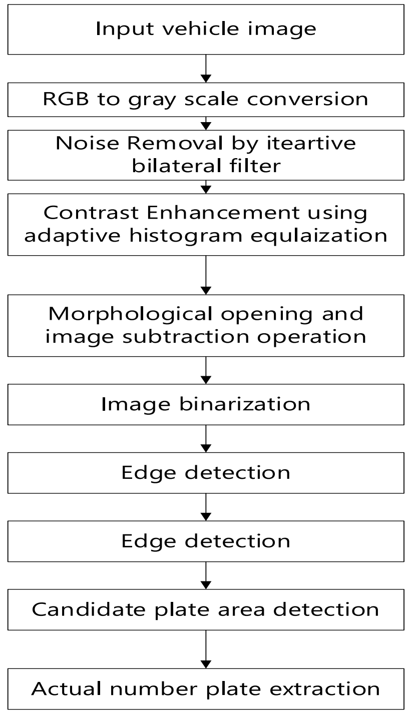


Figure 15 Number plate detection

2.1.7.4 License plate detection Dataset

The datasets for this system were collected in the premises of the southern expressway. Because there is no universal sample on which to test this algorithm's accuracy. The authors added 1000 of datasets to the system training used by Google Collab.

The methodology used in the developed project was divided into four distinct stages, which are detailed below.

- Pre-processing Stage
- Licence Plate or Number Plate Localization
- Character Segmentation
- Optical Character Recognition

1. **Pre-processing stage**

- ✓ Wavelet-based De-noising
- ✓ Conversion image to gray scale
- ✓ Applying the Weiner filter
- ✓ Enhancement of the image contrast

2. <u>Licence plate localization</u>

The License Plate Recognition function's main step is to locate the license plate. [10] [11] The picture recorded for the complex backdrop is filtered in this phase, and the resulting image contains areas of high contrast for the number plate image. Due to the atmosphere included in the pictures, the scene becomes more complicated. There is a need to examine a particular frame to concentrate on specific pictures; this may be accomplished by excluding the backdrop from the image. As a result, a suitable and window frame size should be considered. The window size is determined by the number plate's likely measurements.

The proposed method consists of two modules for number plate localization: rough detection of number plate areas and precise localization of the Region of Interest. The present approach employs the Wavelet Transform (DWT), which emphasizes the vertical edges of the number plates while suppressing background noise. Following this stage, the license plates are extracted and localized using orthogonal projection histogram analysis via a morphological operator. The suggested system enables the extraction of the license plates of a range of automobiles. The findings of the experimental and analytical study indicate that the localization results are of excellent quality, with a short runtime and a high detection rate accuracy.

3. <u>Character Segmentation</u>

Character segmentation serves as a bridge or connection between the number plate localization procedure. This stage necessitates a thorough examination of the vehicle's license plates for any programmed characters. This step's primary purpose is to separate the characters chosen in the candidate area or even from the extracted license plate, so that every character may be sent separately to the Optical Character Recognition module for recognition.

Since histogram computing and thresholding are fundamental to this process, the installation of artificial neural networks has become a more recent progress protocol. Because individual plates use various font styles and alphabets, and number formats are fancy, normalized, or standardized Number plates are essential for an effective segmentation procedure. In nations without standardized license plates, the License Plate Recognition system does not even operate effectively. As a solution to this problem, governments in nations like these could take an effort with specified rules and make high security license plates obligatory for all drivers. Once the localization process is complete, you'll be able to move on to character identification. Since a vehicle's license plate includes a written area with high intensity fluctuation this serves as a foundation for character segmentation. However, because a license plate may include the name of the state as well as other information, it is necessary to remove this information from the plate. The use of morphological operators eliminates the white areas that are left behind after the range rectification procedure, since the shadows and text may occasionally seem like number characters. After that, the OCR module receives the extracted characters and performs optical character recognition on them.

For License Plate Detection and Recognition, character segmentation is essential since it determines how all other procedures work. It's also possible that if the segmentation process fails, the character will not be properly split into two parts or that two characters will be improperly combined, which will lead to the recognition process failing as well. For License Plate Detection and Recognition systems, a single row number plate may be considered. In this case, the segmentation procedure identifies horizontal boundaries between the characters located. The augmentation of segmented characteristics is the second step of segmentation. It's necessary to eliminate everything except the individual characters from the number plate since the plate contains the characters and certain undesirable rudiments, such as noise caused by shadows or flaws in the cameras or other equipment and unwanted space on the character sides.

Using the number plate area as an input, the segmentation procedure uses that region as an output. As an input, the authors use a colored JPEG picture that has been binarized before being used in the segmentation procedure. Unwanted areas in the resulting binary picture may impede the identification process, therefore they should be ignored. To free up more number plate space, the linked components idea is being explored. Subpixel values below this threshold are turned back to background, and therefore not used in this calculation. As a result, what's left is a binarized number plate that's rather quiet, free of unwanted areas, and suitable for segmentation.

4. <u>Character Recognition</u>

The primary goal of this phase is to give classification and full identification to binarized pictures composed of characters acquired from number plate localized areas.

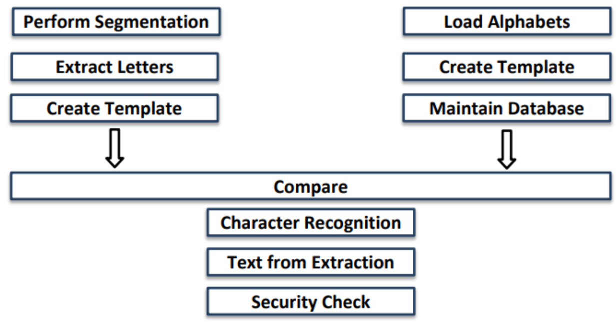


Figure 16 Flowchart representing – Optical Character Recognition

2.1.8 System Architecture

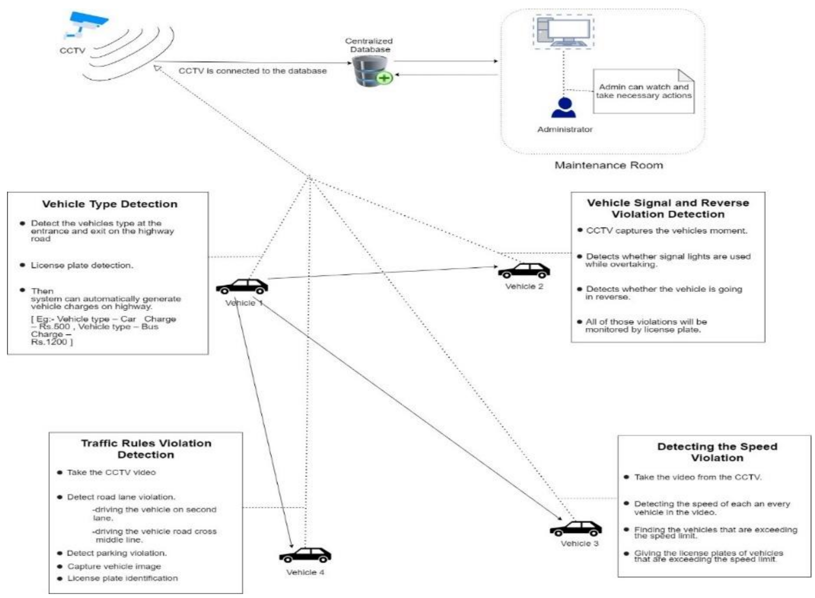


Figure 17 System Architecture – System Diagram

2.1.9 The Architecture of the Functions Assigned to the User.

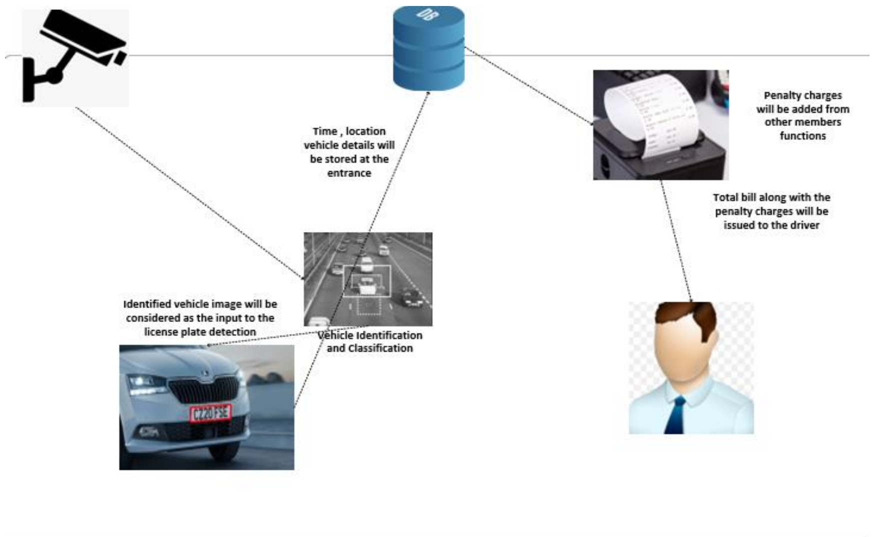


Figure 18 The Architecture of the Functions Assigned to the User

2.1.10 Software development framework

Several project management frameworks have been defined for the IT industry. Authors selected the Agile framework; hence, Agile methodology is designed exclusively for projects where team members have no prior experience in the development of that domain. Thus, the Agile framework provides the learning curve in the development phase. As the author had no prior knowledge and experience of intelligent transportation and surveillance system theories, the Agile framework was selected.

The agile framework consists of Sprints. A sprint is a short life spanned period where developers work a certain amount in developing the total project. The most important point is Agile framework support for unfixed requirements thus, requirement changes can be addressed in the sprints. As the total outcome is an effort of four members each outcome can be tested and view in those sprints.

2.1.11 Programming languages and platforms

1. **Python Language**

   Python is a dynamically semantic, object-oriented, high-level programming language that is translated into machine code. Python's support for modules and packages encourages the modularization of programs and the reusing of code. Free to use it and distribute in source in binary form on all major platforms, the Python interpreter and huge standard library are available. Because of the increased productivity it provides, Python is often adopted by programmers. Since there is no compilation step, the edit test debug cycle is very fast. A defect or erroneous input in a Python program will never cause a segmentation fault, making debugging Python programs a breeze. An exception is thrown when the interpreter encounters a problem. If the program fails to catch the exception, the interpreter shows a stack trace. Using a source-level debugger gives you access to inspect both local and global variables, evaluate arbitrary expressions, set breakpoints, and go line by line through the code. Python was used to create the debugger, showcasing the language's introspective features. A different approach to debugging is by adding a few printing statements to source code since this is the quickest way to test and fix a program in a short time.

2. **PyCharm**

   Many programmers now use Python to build software with clear, concise, and easily understood code bases. Utilizing several Python integrated development environments, they may be able to accelerate the creation of custom applications (IDEs). PyCharm is a well-known Python Integrated Development Environment (IDE). PyCharm was developed by JetBrains only to serve as a Python IDE that runs on a variety of different operating systems. PyCharm works with Python versions 2.x and 3.x and is available for Linux, Windows, and macOS. At the same time, PyCharm's capabilities and features let programmers quickly and efficiently build a broad variety of Python software applications. Developers may even alter the PyCharm UI to suit their preferences and needs. More complex projects may benefit from the IDE's 50+ plug-ins, which let them tailor it to their specific requirements.

The intelligent code editor in PyCharm helps Python programmers writes better code. The editor uses color schemes, automatically inserts indents on new lines, selects the appropriate coding style, and provides context-aware code completion suggestions to help programmers better comprehend their code. As well as extending a code block to include expressions and logical blocks, programmers may use the editor's code-formatting features to look for spelling errors and other inconsistencies, as well as create code on the fly. Aside from that, using an editor streamlines the process for developers of evaluating and detecting errors in their code as they create it.

2.2 Commercialization aspects of the product

To put it another way, commercialization is the process through which all the companies as well able to identify and address issues with new goods before releasing them to consumers. There are many variables that affect or delay a product's release date, and it aids businesses in making an informed decision. Unfavorable market circumstances or customer requests for modifications may both influence the sales of a product.

Critical choices as well as strategic decisions are made by different companies. Choosing where to launch is a critical step in the process. Firms may decide whether or not to sell a product on the local, national, or global market level. Availability of resources, such as operational capabilities and money, is also taken into consideration by them. Market research is also carried out by business executives in order to identify their most important customer segment.

In order to become a successful system product, an idea must be tied to the requirements of consumers. Customers, on the other hand, may not always be aware of their needs or wants. According to the researcher's view, market monitoring, basic research, and consumer behavior are the keys to spotting promising business possibilities. In order to do this, several types of information must be tracked and linked appropriately.

It's critical to hear a variety of viewpoints while working on a development project. When the entire value chain is engaged in the project, it's easiest to bring together the various viewpoints and abilities. In a technological project, the network should encompass, for example, applied research, developers, raw material chain specialists, clients, investors, suppliers, start-up businesses and end users.

In considering the newly implemented vehicle identification and license plate detection system, the following conditions are to be listed by the author,

- The system was developed can be used in express ways.
- The completed system will be given for the users to use it for a travel period.
- After the trail period users can purchase the system.
- If there are any additional customizations required after purchasing the software, then user must pay for those customizations or updates.

The project has been deemed successful in delivering the desired objectives, including the completion of the highway and the creation of new job possibilities for the people of the southern area. All three Southern Province districts have shown an improvement in socioeconomic circumstances. Ultimately, the initiatives had a major effect on the southern region's economy, increasing economic possibilities and improving social circumstances. The whole framework's design and monitoring goals have been met. The project was executed successfully. At the time of assessment and after completion, the project was still relevant to the development objectives and circumstances.

2.3 Testing and Implementation

This chapter contain the details about the testing and the implementation process of the research. The researcher clearly discusses about planes used to conduct the testing process and the results. In the implementation, the researcher discusses about the implementation process of the solution including the purpose, system overview, main steps of implementing the solution with technical requirements, and the changeovers.

After the development of the model, the next stage is the testing phase. This stage helps to identify the various problems that are found after development. This allows you to get the right output without any problems. When the test phase is complete, it is a sign that the user can get a proper output from the developed model.

2.3.1 Testing

This is defined for testing completed software. During the life cycle of software development, it is important to test the system before activating it. Therefore, it helps to identify problems properly and take solutions as the next step. The main purpose of the test is to identify deficiencies in the developed system. Even if a system is fully developed, it may still have various problems. They must be clear and resolved before they can be implemented. Considering the research project, two main modules are integrated into one system such as license plates detection and vehicle type recognition. When all two modules were combined, the researcher examined the entire system and it appeared that there were no major problems.

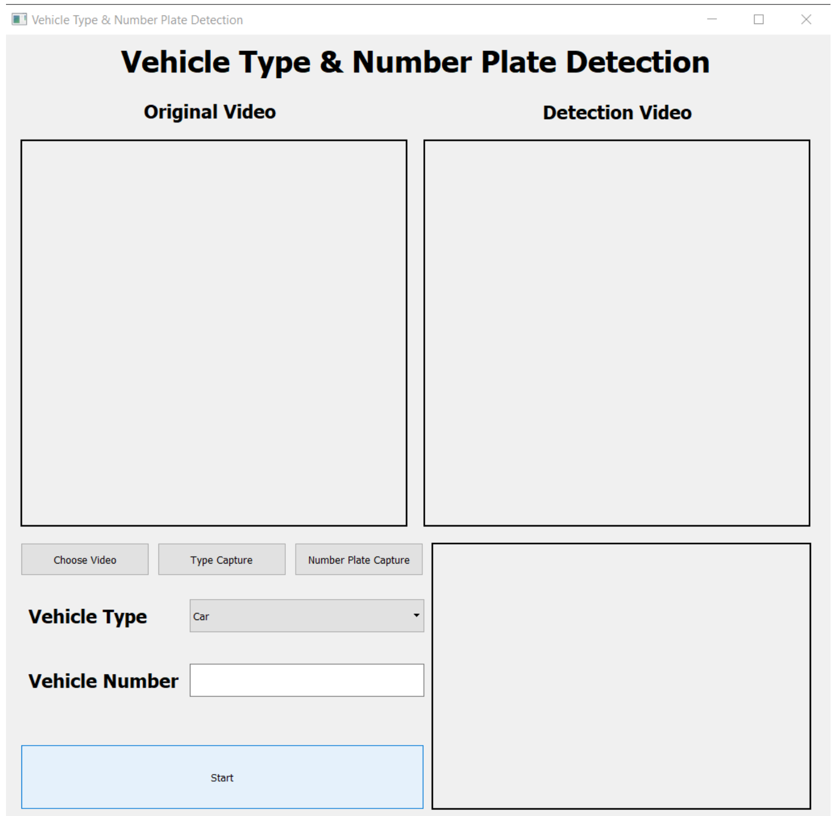


Figure 19 Main dashboard

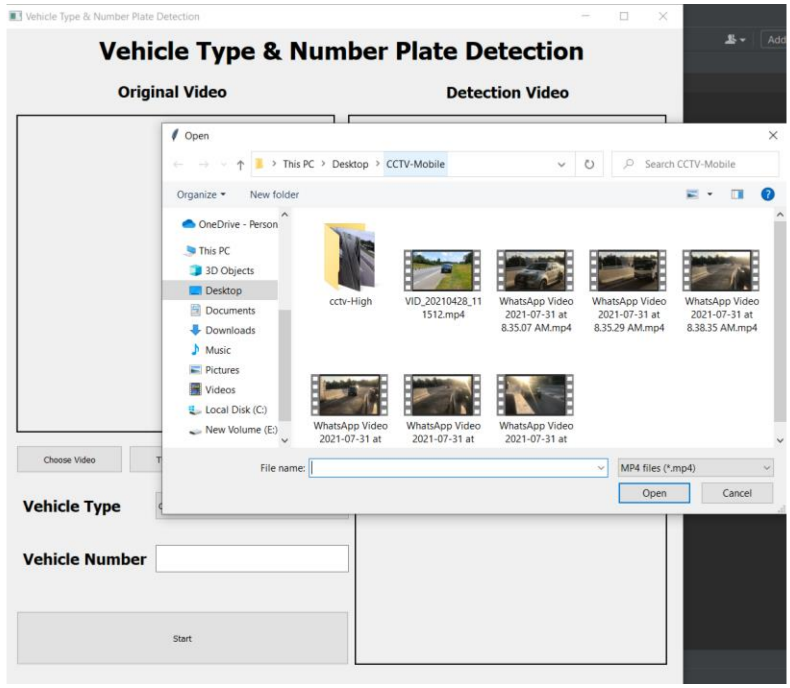


Figure 20 Video Uploading - I

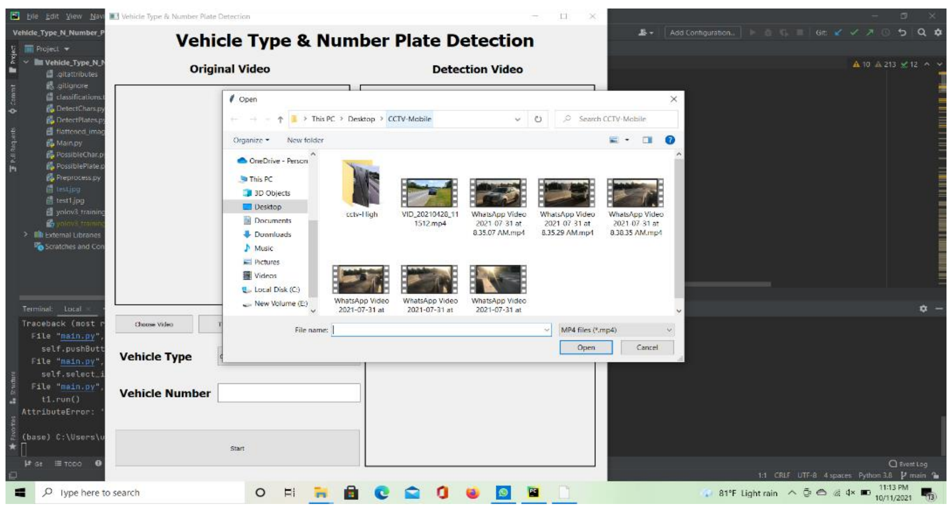


Figure 21 Video uploading II

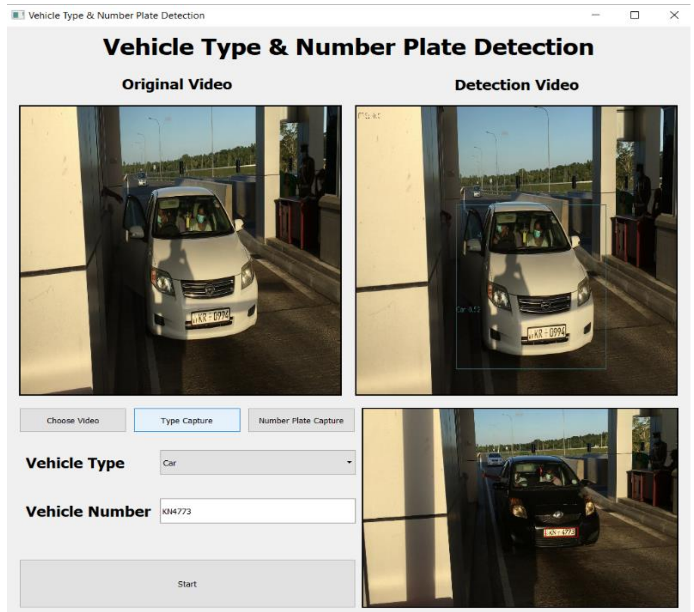


Figure 22 Vehicle type detection

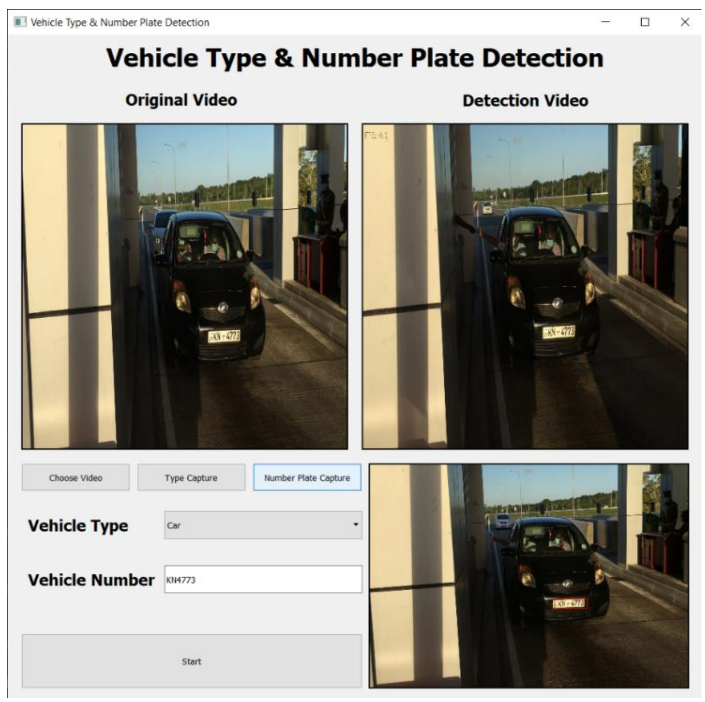


Figure 23 Number plate detection

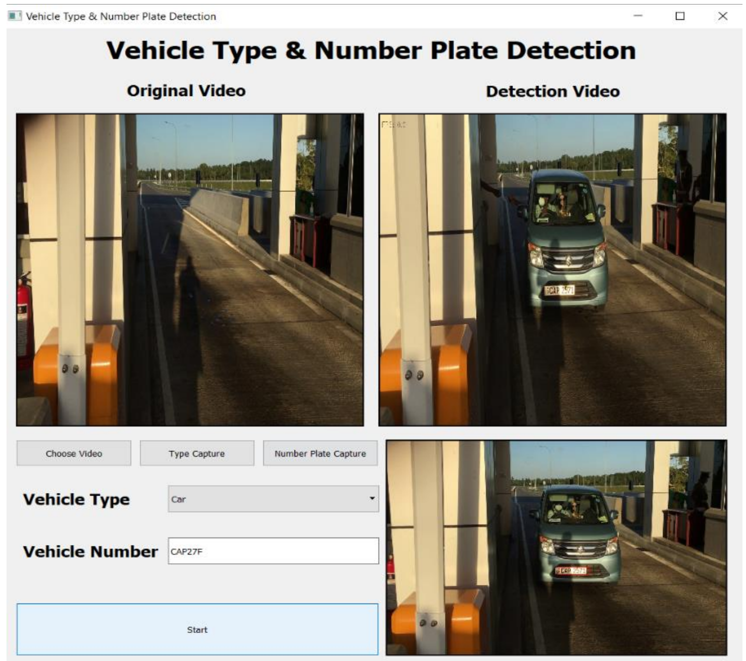


Figure 24 Original video vs detected video

2.3.2 Implementation

Implementation is the most critical stage of a project's life cycle since it is where the created model interacts with the actual world. This occurs after the development testing phase. This level is primarily on quality and performance. During the installation process, many difficulties may emerge, such as troubleshooting, infrastructure concerns, and so on. As a result, the activation step should be carried out with careful attention for the user environment.

2.3.3 Tasks of the system implementation

There are several key points that need high attention in the implementation process. They are listed as follows,

- Overall activation should be well planned.
- Provide training sessions for users and give them an overall idea of the flow.
- Each process and how it works

- Identify the assets required for implementation.
- Ensure system security.

2.3.4 System changeover

This involves how the system changes from the current model to the newly developed model. There are various modes of operation, such as activation, parallel execution, phase activation, etc. The proper procedure must be followed. Different activation methods will have different effects on how the system works. Direct implementation means that the developed method works directly with the current system, but it is somewhat risky. Parallel implementation is the implementation of the newly developed system in parallel with the old system. Phase implementation is the step-by-step implementation of the developed model into the real environment. Before choosing the implementation method to follow, it is important to identify the nature of the existing models. Otherwise, it will adversely affect the whole process. This is not a big deal if the developed model is a new concept, as users will not be confused by switching system usage to each other. After considering all these factors, the researcher suggests installing the new model without completely replacing their existing manual system.

2.3.5 User training

It is very important to consider this part, because even if the system is running successfully, if the user does not understand how it works, it will be a big failure. Therefore, the expressway officers should be given a user manual or proper training on how to use it. Available activities. Although the model works perfectly and effectively, it is useless if users do not have a proper idea of how to use it. Therefore, relevant training should be provided to avoid problems that may arise when using the system.

3.0 RESULTS AND DISCUSSION

3.1 Results

This method cannot be compared to other techniques since there is no single set of rules or unified datasets on that tests can be performed. The images on that the tests were performed are listed in the table below.

Table 3 Test Results

Input Number	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	No of correct	%
WP CAQ7180	w	p	C	A	Q	7	1	8	0	9	100 %
WP CAK7299	W	P	C	A	K	7	2	9	9	9	100 %
18-9685	1	8	-	9	6	8	5			7	100 %
WP GS4060	W	P	G	S	4	0	6	0		8	100 %
WP PM0967	W	P	P	M	0	9	6	7		8	100 %
WP PH7423	W	P	P	H	7	4	2	3		8	100 %
WP CAX6686	W	P	C	A	X	6	6	8	6	9	100 %
WP KM0456	W	P	K	M	0	4	5	6		8	100 %
WP KS6146	W	P	K	S	6	1	4	6		8	100 %

WP CAR5462	W	P	C	A	R	5	4	6	2	9	100 %
WP PB9830	W	P	P	B	9	8	3	0		8	100 %
WP ND3791	W	P	N	D	3	7	9	1		8	100 %
WP KX5340	W	P	K	X	5	3	4	0		8	100 %
WP PH0879	W	P	P	H	0	8	7	9		8	100 %
WP CAP5736	W	P	C	A	P	5	7	3	6	9	100 %
SP GR9898	S	P	G	R	9	8	9	8		8	100 %
UP CBB 3204	U	P	C	B	B	3	2	0	4	9	100 %
WP KI 0659	W	P	K	I	0	6	5	9		8	100 %
UP LL 5996	U	P	L	L	5	9	9	6		8	100 %
WP PC 0103	W	P	P	C	0	1	0	3		8	100 %
SP CAH 5554	S	P	C	A	H	5	5	5	4	9	100 %
WP PF 7343	W	P	P	F	7	3	4	3		8	100 %
WP KW 7431	W	P	K	W	7	4	3	1		8	100 %

WP CAZ 3846	W	P	C	A	Z	3	8	4	6	9	100 %
SP LL 6657	S	P	L	L	6	6	5	7		8	100 %
SP CAS 7063	S	P	C	A	S	7	0	6	3	9	100 %
SP LM 0326	S	P	L	M	0	3	2	6		8	100 %
SP PE 5462	S	P	P	E	5	4	6	2		8	100 %
63-1824	6	3	-	1	8	2	4			7	100 %
WP CBD 7863	W	P	C	B	D	7	8	6	3	9	100 %
SG PG8634	S	G	P	G	8	6	3	4		8	100 %
SP KN4773	S	P	K	N	4	7	7	3		8	100 %
WP KR0994	W	P	K	R	0	9	9	4		8	100 %
SP CAP2571	S	P	C	A	P	2	5	7	1	9	100 %
WP CAZ3574	W	P	C	A	Z	3	5	7	4	9	100 %
SP KW0144	S	P	K	W	0	1	4	4		8	100 %
WP KG3634	W	P	K	G	3	6	3	4		8	100 %

WP CAZ8270	W	P	C	A	Z	8	2	7	0	8	100 %
SP KT8903	S	P	K	T	8	9	0	3		7	87.5 %
SP CBI2022	S	P	C	B	I	2	0	2	2	8	88.8 %
WP CBG7392	W	P	C	B	G	7	3	9	2	8	88.8 %
SP PH6284	S	P	P	H	6	2	8	4		7	87.5 %
WP CBI1998	W	P	C	B	I	1	9	9	8	8	88.8 %
WP DAH2216	W	P	D	A	H	2	2	1	6	8	88.8 %
WP CBJ8763	W	P	C	B	J	7	8	6	3	8	88.8 %
SP KT4902	S	P	K	T	4	9	0	2		7	87.5 %
WP KE1909	W	P	K	E	1	9	0	9		7	87.5 %
SP PH3982	S	P	P	H	3	9	8	2		7	87.5 %
WP PC7421	W	P	P	C	7	4	2	1		7	87.5 %
WP CAO3734	W	P	C	A	O	3	7	3	4	8	88.8 %
WP CBF7785	W	P	C	B	F	7	7	8	5	8	88.8 %

WP CAR6869	W	P	C	A	R	6	8	6	9	7	77.7 %
WP KH1676	W	P	K	H	1	6	7	6		6	75%
WP CAZ2049	W	P	C	A	Z	2	0	4	9	7	77.7 %
301-5367	3	0	1	-	5	3	6	7		6	75%

Accuracy percentage of the images

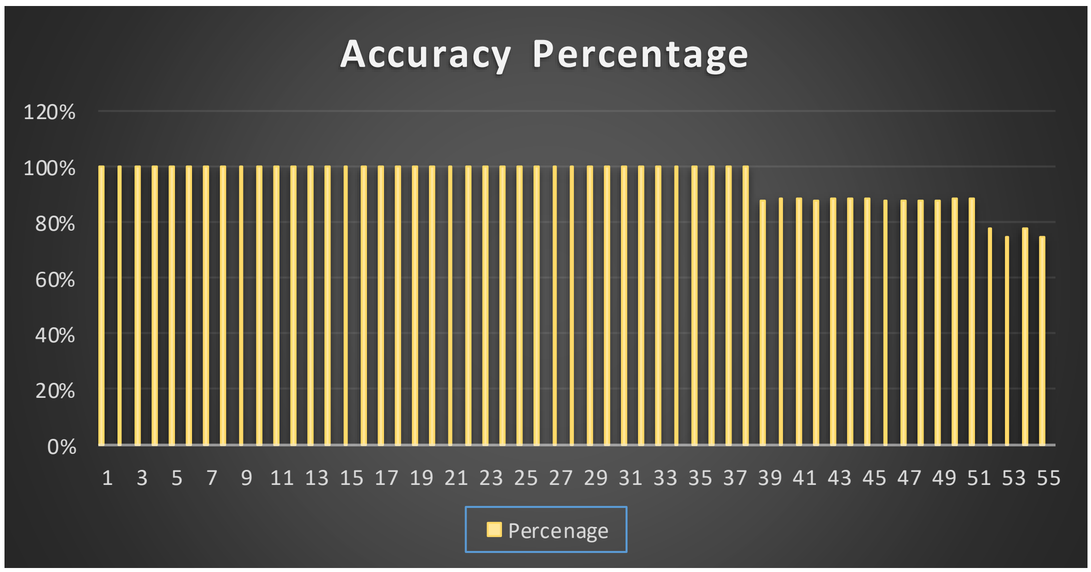


Figure 25 Accuracy Percentage

The following data is derived from the testing:

1. The tests are carried out on 55 different pictures.
2. 7 Images have only 1 out of 9 character wrong.
3. 6 Images have only 1 out of 8 character wrong.
4. 2 Images have only 2 out of 9 character wrong.
5. 2 Images have only 2 out of 8 character wrong.

Total 55 images were used in experiments:

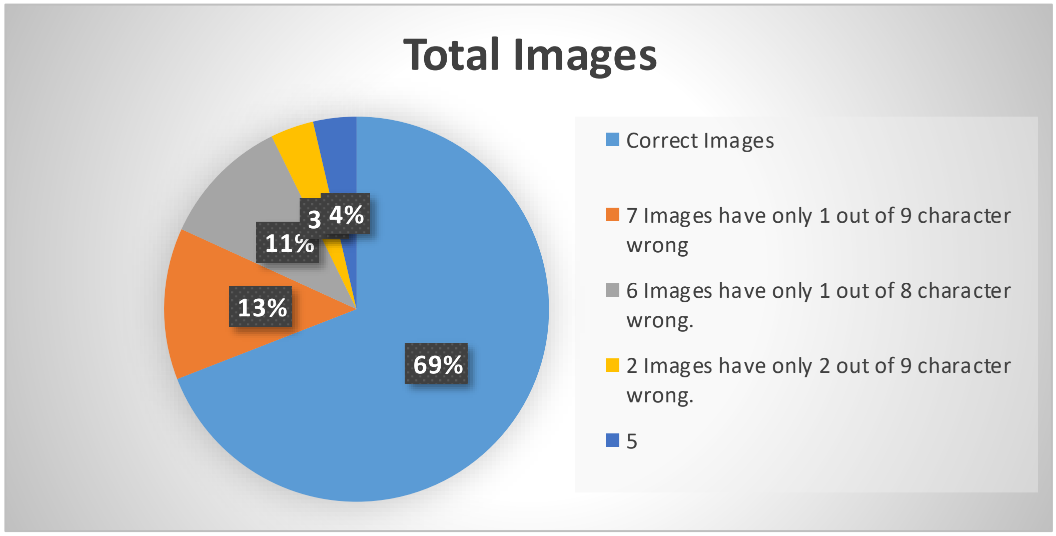


Figure 26 Total Images

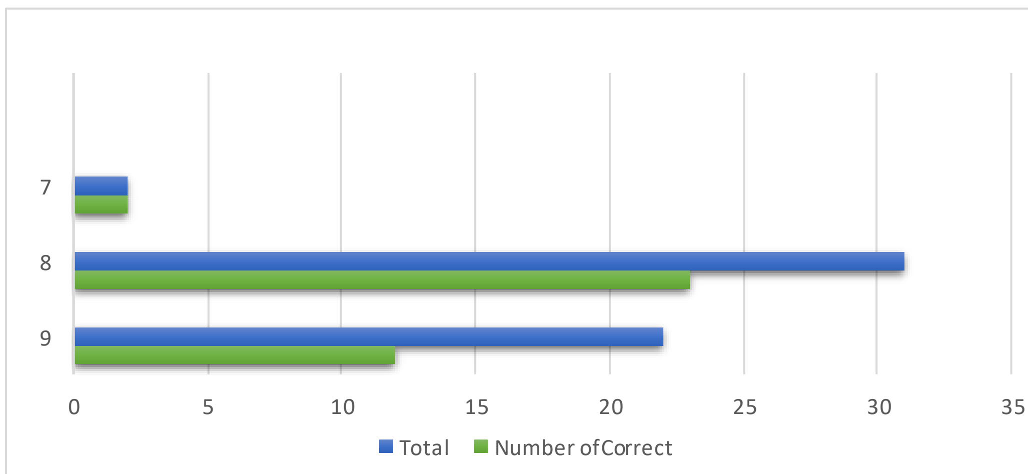


Figure 27 Number of correct images recognized

3.2 Evaluation

3.2.1 What was evaluated?

- Accuracy of the detection of the vehicles in the video.
- Recognizing the license plate numbers of vehicles.

3.2.2 Purpose of the evaluation

To check the accuracy of detect of the vehicles. And to check whether the system is correctly detecting the vehicles that are new model of the express way and to check whether the system is correctly identifying the license plate numbers of those vehicles.

3.2.3 How the evaluation did?

The authors compared the estimated outputs with the actual data. Therefore, researchers got the actual vehicle model and license plate data with the help of the Police Department. And these actual data were collected near the locations of the surveillance cameras which researchers were obtaining data from.

By using above collected actual data, researchers compared the system outputs with the actual new model of the vehicles. And compared the recognized license plate numbers of vehicles with the actual license plate numbers of those vehicles.

3.3 Discussion

In Sri Lanka, the characteristics of the License Plates are strictly maintained. Included in this list are factors such as license plate size and color, as well as the number of characters on the plate and their sizes, colors, and fonts used on each one. While in Sri Lanka, a standard procedure for the License plate is still not used, making the process of localization and identification of the License plate very difficult and time consuming. Other than that, the scripts utilized to write the License plate are not all the same throughout the country.

The various difficulties faced by the system

1. Poor Images: One reason for this is that the Licence plate would be too far away while taking the picture; this results in poor image quality. Another explanation is that the camera used to capture the pictures was of poor quality.

2. Blurred Images: The pictures get smeared as though they were captured in slow motion. A motion blurred picture is one in which the movement of the objects in an immobile image or in a series of images, such as a movie, may be seen as streaking. During the covering or recording of an image, this problem is mainly caused by fast movement, which disturbs the picture.
3. Poor lightning and law contrast images: Overexposure in the surroundings, reflections from different things in the area, or shadows projected onto the license plate are the most common causes of this problem.
4. Object Obscure: Objects or substances that adhere to the plate or a portion of the plate, such as dirt or tow bar wire, make it difficult to take a good picture of the Licence Plate.
5. Problems of having different license plates: Many provinces have distinct readings for the license plates on the front and rear. Some simply ask for the installation of back plates, leaving the owner free to customize the front with their own characters. License plates for the same vehicle will now have unique designs because of this.
6. Trouble caused on account of different fonts used: The Licence Plate has a different font style, which exacerbates the issue.

Some of these flaws can be fixed using software, while the other ones must be addressed using the system's hardware components. Objects covering the License Plate may cause difficulties, which can be avoided by raising the surveillance camera's height. However, it introduces other challenges, such as correcting the License Plate's amplified and expanded skew. The License Plates' faults are due to a variety of small-scale processes. There is a chance that a single character will be incorrect when certain vehicles are allowed entry into prohibited regions. This is since an unauthorized car with a comparable License Plate is not seen as anything special and is seen as quite straightforward. However, this level of error and opacity would be considered unacceptable in a wide range of new System applications. The author suggests the following improvements to the research provided in this thesis to make it even better.

The emphasis of our future research will be night monitoring and the improvement of previously published algorithms. All our proposed system's other parts need to be improved, with a particular emphasis on occlusion management, vehicle matching, and increasing character recognition accuracy by utilizing neural networks and back propagation algorithms. The initial step in this process is to train the network, and to do so, input and goal must be provided. The segmented character on the license plate may now input the neural network to simulate after the network had been trained and effectively used in the real world. In an ideal world, input characters would be compared to neural network training data and the ASCII code for the matching input character would be produced.

4.0 CONCLUSION

This chapter will discuss the overall conclusion of the project. It discusses the overview of the project by analyzing how it meets the desired requirements through the various stages. At the end of the project, there is a lot to learn and experience during this time. Future enhancements will also be discussed to help increase project productivity and efficiency.

Automatic vehicle license plate recognition as well as vehicle type detection play an important role in intelligent transportation systems and have a wide range of practical applications including automatic toll collection, parking fee payment, detection of vehicles exceeding speed limits and thus reducing road accidents, and so on.

Number plate recognition has been successfully accomplished by the researcher. With the help of the algorithm, the number plate area may be effectively detected from the picture. The method was tested on many pictures, and it was shown to be effective at recognizing objects. To improve security, the project was developed with the goal of automating the existing manual number plate detection system. Even though it has limitations in image processing as well as other hardware requirements, this project was completed in recording a vehicle's license plate number.
The methodology demonstrates to be a successful instrument used by legislative as well as consumer safety forums. This data substantially enhances the covert competence of law enforcement organizations, being useful in intelligence and analytical functions. In the present method, a wavelet de-noising procedure with soft thresholding in the preprocessing phase produces a system for successful license plate identification, and binary masking of form templates may successfully identify and categorize a vehicle.

The researcher draws the following conclusions based on the findings:

1. The primary requirement for carrying out the number plate localization procedure effectively is the existence of the appropriate edge among the vehicle License

plates border and the backdrop. This aids the Weiner filter in its detection of the edge.

2. The camera must be placed at a particular distance from the license plate in order to give a sufficient range to the count of pixels inside the Licence plate area and make them consistent in nature.
3. The main disadvantage faced during the character segmentation process is really the existence of fancy scripts and formats, which causes blockage throughout the operation.
4. The backdrop must be constant, and the borders must be clearly seen in order to recognize the form of a vehicle. The camera position should be constant throughout all pictures. Because the distance among the camera and the vehicle may vary, incorrect identification can occur.

The YOLO method was shown to be the most effective for addressing actual object detection in a traffic scenario after extensive testing and analysis. The YOLOv3 version created and reported in this thesis has better mean average accuracy on real-time detection of cars and people. Distance measurements to identified objects are used to train the newly created YOLOv3 object detection model for improved road safety. The range is restricted so that only cars directly in front of it may be detected, thus reducing the number of warning notifications. The model is capable of real-time detection, and all the findings come from tests conducted in a real-world setting where variables like light and weather cannot be controlled. A CCTV camera was used in the real world to test the suggested method. While preserving detecting speed, the project's goal was to improve accuracy.

REFERENCES

[1] S. Kamkar and R. Safabakhsh, ‘Vehicle detection, counting and classification in various conditions’, *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 406–413, Aug. 2016, doi: 10.1049/iet-its.2015.0157.

[2] W. Fang, L. Wang, and P. Ren, ‘Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments’, *IEEE Access*, vol. 8, pp. 1935–1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[3] J. Pedoeem and R. Huang, ‘YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers’, *arXiv:1811.05588 [cs]*, Nov. 2018, Accessed: Oct. 01, 2021. [Online]. Available: http://arxiv.org/abs/1811.05588

[4] V. V. Molchanov, B. V. Vishnyakov, Y. V. Vizilter, O. V. Vishnyakova, and V. A. Knyaz, ‘Pedestrian detection in video surveillance using fully convolutional YOLO neural network’, Munich, Germany, Jun. 2017, p. 103340Q. doi: 10.1117/12.2270326.

[5] R. Laroca *et al.*, ‘A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector’, *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, Jul. 2018, doi: 10.1109/IJCNN.2018.8489629.

[6] N. Wojke and M. Haselich, ‘Moving vehicle detection and tracking in unstructured environments’, in *2012 IEEE International Conference on Robotics and Automation*, St Paul, MN, USA, May 2012, pp. 3082–3087. doi: 10.1109/ICRA.2012.6224636.

[7] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, ‘Vehicle Detection and Tracking in Car Video Based on Motion Model’, *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 2, pp. 583–595, Jun. 2011, doi: 10.1109/TITS.2011.2113340.

[8] S. Sivaraman and M. M. Trivedi, ‘A review of recent developments in vision-based vehicle detection’, in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast City, Australia, Jun. 2013, pp. 310–315. doi: 10.1109/IVS.2013.6629487.

[9] M. Y. Arafat, A. S. M. Khairuddin, U. Khairuddin, and R. Paramesran, ‘Systematic review on vehicular licence plate recognition framework in intelligent transport

systems’, *IET Intelligent Transport Systems*, vol. 13, no. 5, pp. 745–755, May 2019, doi: 10.1049/iet-its.2018.5151.

[10] G. N. Balaji and D. Rajesh, ‘Smart Vehicle Number Plate Detection System for Different Countries Using an Improved Segmentation Method’, vol. 3, no. 6, p. 7, 2017.

[11] M. F. Zakaria and S. A. Suandi, ‘Malaysian Car Number Plate Detection System Based on Template Matching and Colour Information’, vol. 02, no. 04, p. 6, 2010.

[12] C. Gerber and M. Chung, ‘Number Plate Detection with a Multi-Convolutional Neural Network Approach with Optical Character Recognition for Mobile Devices’, *Journal of Information Processing Systems*, vol. 12, no. 1, pp. 100–108, Mar. 2016, doi: 10.3745/JIPS.04.0022.

APENDICES

Charts

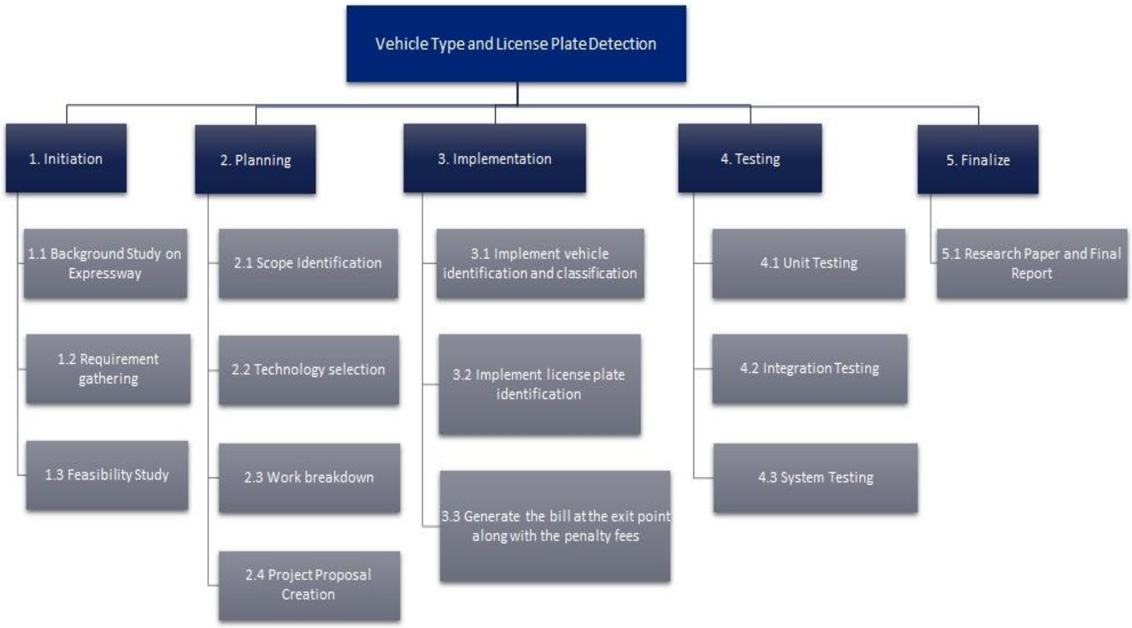


Figure 28 Work Break Down Structure

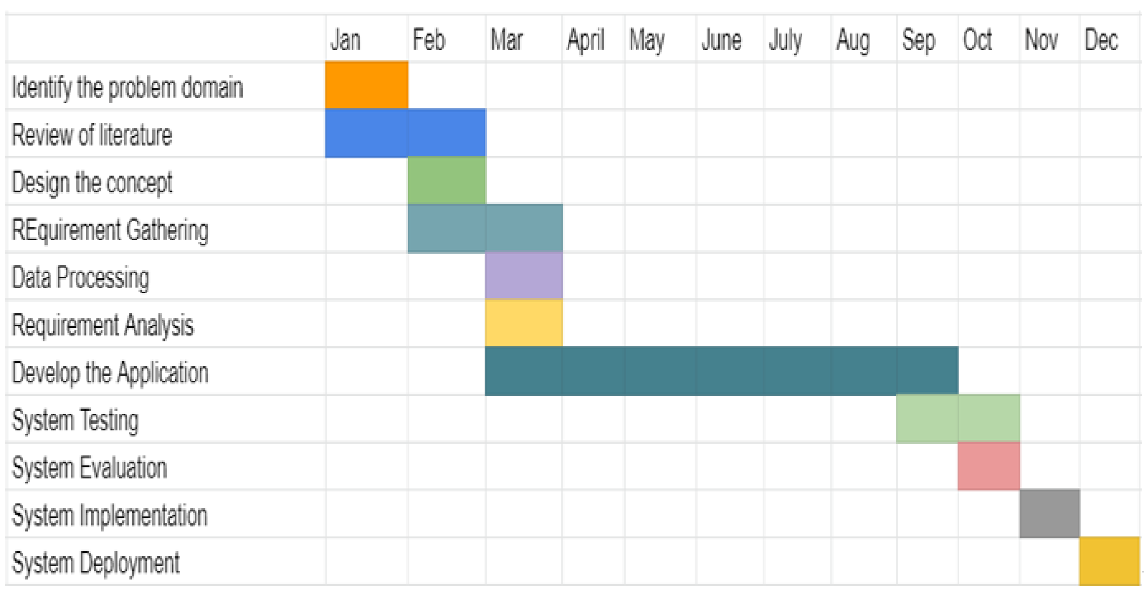


Figure 29 Gannt Char

Images

Figure 30 CCTV Capturing I

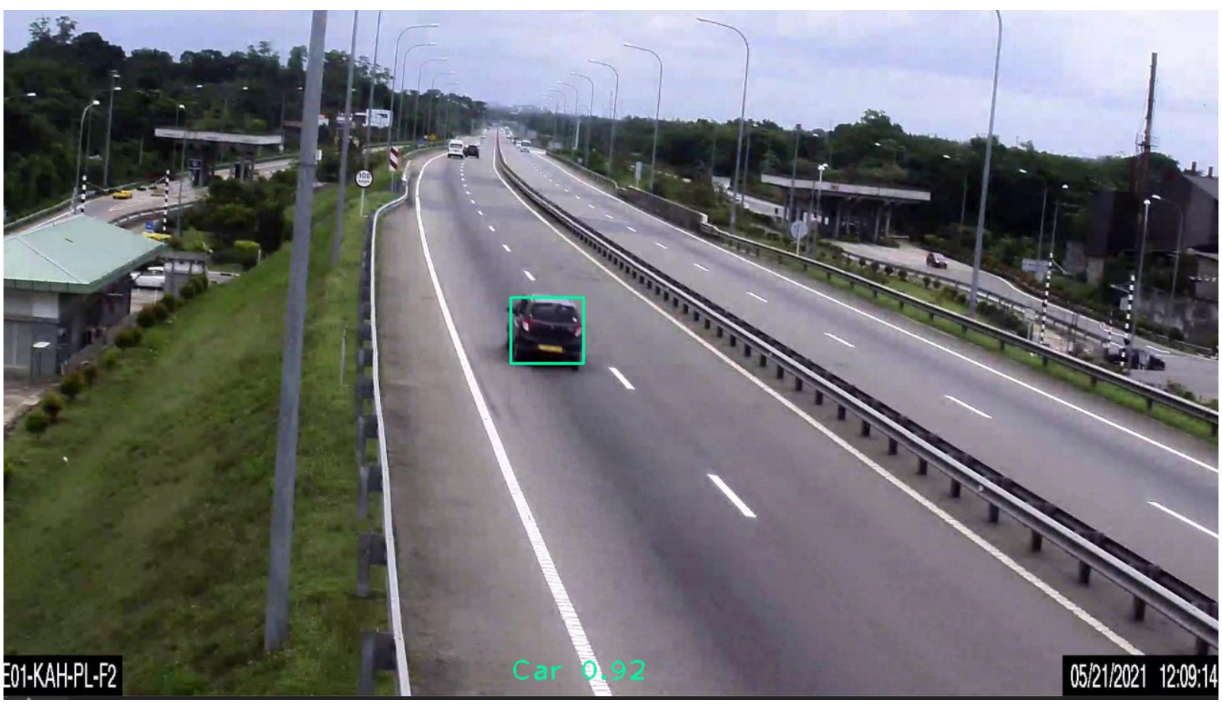


Figure 31 CCTV capturing II

Algorithms

```
Number Plate detection

# DetectPlates.py

import cv2
import numpy as np
import math
import Main
import random

import Preprocess
import DetectChars
import PossiblePlate
import PossibleChar

PLATE_WIDTH_PADDING_FACTOR = 1.3
PLATE_HEIGHT_PADDING_FACTOR = 1.5

def detectPlatesInScene(imgOriginalScene):
    listOfPossiblePlates = []

    height, width, numChannels = imgOriginalScene.shape

    imgGrayscaleScene = np.zeros((height, width, 1), np.uint8)
    imgThreshScene = np.zeros((height, width, 1), np.uint8)
    imgContours = np.zeros((height, width, 3), np.uint8)

    cv2.destroyAllWindows()

    if Main.showSteps == True:
        cv2.imshow("0", imgOriginalScene)

    imgGrayscaleScene, imgThreshScene =
Preprocess.preprocess(imgOriginalScene)

    if Main.showSteps == True:
        cv2.imshow("1a", imgGrayscaleScene)
        cv2.imshow("1b", imgThreshScene)

    listOfPossibleCharsInScene =
findPossibleCharsInScene(imgThreshScene)

    if Main.showSteps == True:
        print("step 2 - len(listOfPossibleCharsInScene) = " + str(
            len(listOfPossibleCharsInScene)))

        imgContours = np.zeros((height, width, 3), np.uint8)

        contours = []

        for possibleChar in listOfPossibleCharsInScene:
            contours.append(possibleChar.contour)

        cv2.drawContours(imgContours, contours, -1,
Main.SCALAR_WHITE)
```

```
        cv2.imshow("2b", imgContours)

    listOfListsOfMatchingCharsInScene =
DetectChars.findListOfListsOfMatchingChars(listOfPossibleCharsInScen
e)

    if Main.showSteps == True:
        print("step 3 - listOfListsOfMatchingCharsInScene.Count = "
+ str(
            len(listOfListsOfMatchingCharsInScene)))

        imgContours = np.zeros((height, width, 3), np.uint8)

        for listOfMatchingChars in
listOfListsOfMatchingCharsInScene:
            intRandomBlue = random.randint(0, 255)
            intRandomGreen = random.randint(0, 255)
            intRandomRed = random.randint(0, 255)

            contours = []

            for matchingChar in listOfMatchingChars:
                contours.append(matchingChar.contour)

            cv2.drawContours(imgContours, contours, -1,
(intRandomBlue, intRandomGreen, intRandomRed))

        cv2.imshow("3", imgContours)

    for listOfMatchingChars in listOfListsOfMatchingCharsInScene:
        possiblePlate = extractPlate(imgOriginalScene,
listOfMatchingChars)

        if possiblePlate.imgPlate is not None:
            listOfPossiblePlates.append(possiblePlate)


    print("\n" + str(len(listOfPossiblePlates)) + " possible plates
found")

    if Main.showSteps == True:
        print("\n")
        cv2.imshow("4a", imgContours)

        for i in range(0, len(listOfPossiblePlates)):
            p2fRectPoints =
cv2.boxPoints(listOfPossiblePlates[i].rrLocationOfPlateInScene)

            cv2.line(imgContours, tuple(p2fRectPoints[0]),
tuple(p2fRectPoints[1]), Main.SCALAR_RED, 2)
            cv2.line(imgContours, tuple(p2fRectPoints[1]),
tuple(p2fRectPoints[2]), Main.SCALAR_RED, 2)
            cv2.line(imgContours, tuple(p2fRectPoints[2]),
tuple(p2fRectPoints[3]), Main.SCALAR_RED, 2)
            cv2.line(imgContours, tuple(p2fRectPoints[3]),
tuple(p2fRectPoints[0]), Main.SCALAR_RED, 2)

            cv2.imshow("4a", imgContours)
```

```
            print("possible plate " + str(i) + ", click on any image
and press a key to continue . . .")

            cv2.imshow("4b", listOfPossiblePlates[i].imgPlate)
            cv2.waitKey(0)

        print("\nplate detection complete, click on any image and
press a key to begin char recognition . . .\n")
        cv2.waitKey(0)


    return listOfPossiblePlates

def findPossibleCharsInScene(imgThresh):
    listOfPossibleChars = []

    intCountOfPossibleChars = 0

    imgThreshCopy = imgThresh.copy()

    contours, npaHierarchy = cv2.findContours(imgThreshCopy,
cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

    height, width = imgThresh.shape
    imgContours = np.zeros((height, width, 3), np.uint8)

    for i in range(0, len(contours)):

        if Main.showSteps == True:
            cv2.drawContours(imgContours, contours, i,
Main.SCALAR_WHITE)

        possibleChar = PossibleChar.PossibleChar(contours[i])

        if DetectChars.checkIfPossibleChar(possibleChar):
            intCountOfPossibleChars = intCountOfPossibleChars + 1
            listOfPossibleChars.append(possibleChar)

    if Main.showSteps == True:
        print("\nstep 2 - len(contours) = " + str(len(contours)))
        print("step 2 - intCountOfPossibleChars = " +
str(intCountOfPossibleChars))
        cv2.imshow("2a", imgContours)

    return listOfPossibleChars

def extractPlate(imgOriginal, listOfMatchingChars):
    possiblePlate = PossiblePlate.PossiblePlate()

    listOfMatchingChars.sort(key = lambda matchingChar:
matchingChar.intCenterX)

    fltPlateCenterX = (listOfMatchingChars[0].intCenterX +
listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterX) / 2.0
    fltPlateCenterY = (listOfMatchingChars[0].intCenterY +
listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterY) / 2.0
```

```
    ptPlateCenter = fltPlateCenterX, fltPlateCenterY


    intPlateWidth =
int((listOfMatchingChars[len(listOfMatchingChars) -
1].intBoundingRectX + listOfMatchingChars[len(listOfMatchingChars) -
1].intBoundingRectWidth - listOfMatchingChars[0].intBoundingRectX) *
PLATE_WIDTH_PADDING_FACTOR)

    intTotalOfCharHeights = 0

    for matchingChar in listOfMatchingChars:
        intTotalOfCharHeights = intTotalOfCharHeights +
matchingChar.intBoundingRectHeight


    fltAverageCharHeight = intTotalOfCharHeights /
len(listOfMatchingChars)

    intPlateHeight = int(fltAverageCharHeight *
PLATE_HEIGHT_PADDING_FACTOR)

    fltOpposite = listOfMatchingChars[len(listOfMatchingChars) -
1].intCenterY - listOfMatchingChars[0].intCenterY
    fltHypotenuse =
DetectChars.distanceBetweenChars(listOfMatchingChars[0],
listOfMatchingChars[len(listOfMatchingChars) - 1])
    fltCorrectionAngleInRad = math.asin(fltOpposite / fltHypotenuse)
    fltCorrectionAngleInDeg = fltCorrectionAngleInRad * (180.0 /
math.pi)

    possiblePlate.rrLocationOfPlateInScene = ( tuple(ptPlateCenter),
(intPlateWidth, intPlateHeight), fltCorrectionAngleInDeg )

    rotationMatrix = cv2.getRotationMatrix2D(tuple(ptPlateCenter),
fltCorrectionAngleInDeg, 1.0)

    height, width, numChannels = imgOriginal.shape

    imgRotated = cv2.warpAffine(imgOriginal, rotationMatrix, (width,
height))

    imgCropped = cv2.getRectSubPix(imgRotated, (intPlateWidth,
intPlateHeight), tuple(ptPlateCenter))

    possiblePlate.imgPlate = imgCropped

    return possiblePlate
```

Number Plate Chars detection

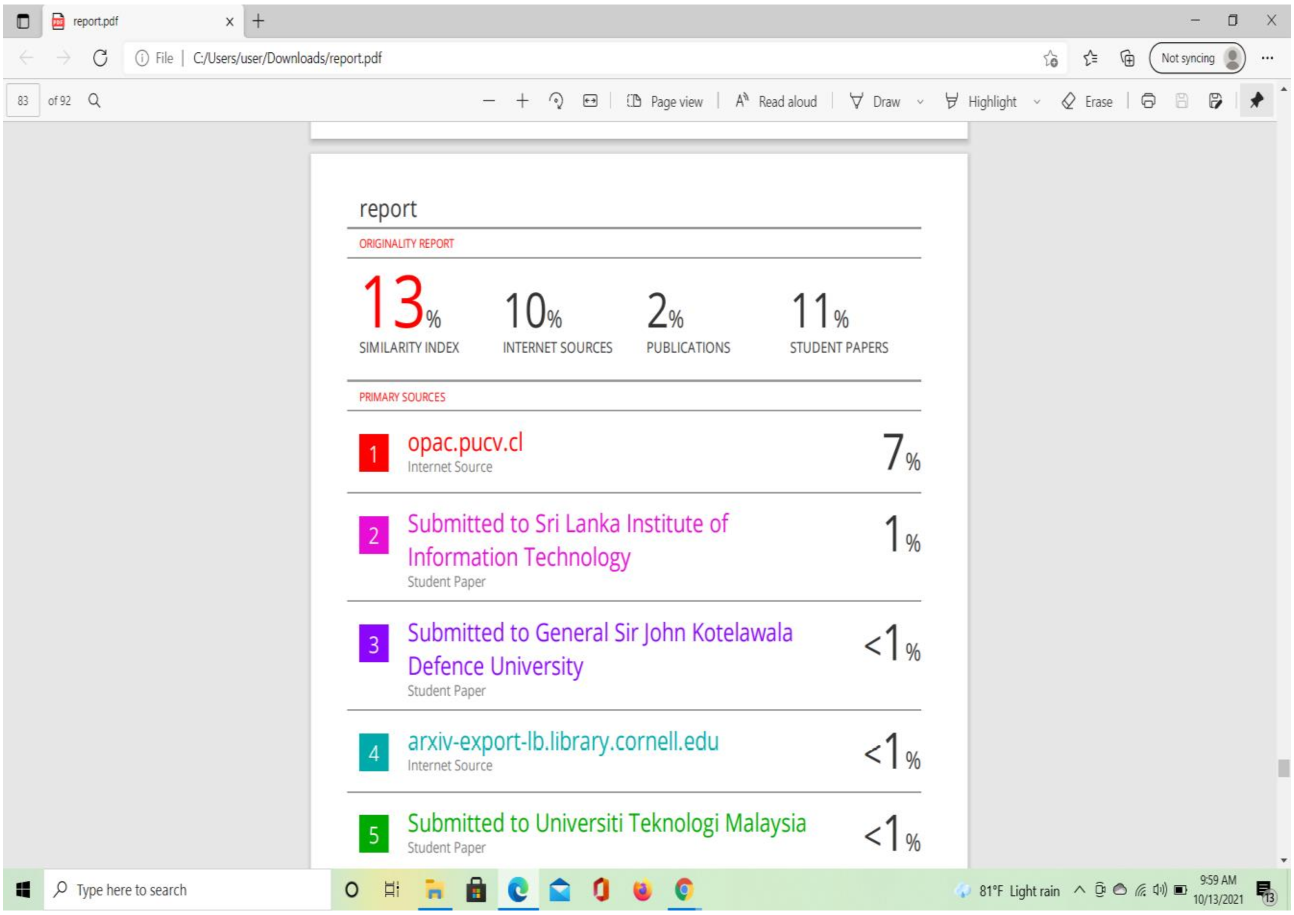
```
import os

import cv2
import numpy as np
import math
import random

import Main
import Preprocess
import PossibleChar

kNearest = cv2.ml.KNearest_create()

MIN_PIXEL_WIDTH = 2
MIN_PIXEL_HEIGHT = 8

MIN_ASPECT_RATIO = 0.25
MAX_ASPECT_RATIO = 1.0

MIN_PIXEL_AREA = 80

MIN_DIAG_SIZE_MULTIPLE_AWAY = 0.3
MAX_DIAG_SIZE_MULTIPLE_AWAY = 5.0

MAX_CHANGE_IN_AREA = 0.5

MAX_CHANGE_IN_WIDTH = 0.8
MAX_CHANGE_IN_HEIGHT = 0.2

MAX_ANGLE_BETWEEN_CHARS = 12.0

MIN_NUMBER_OF_MATCHING_CHARS = 3

RESIZED_CHAR_IMAGE_WIDTH = 20
RESIZED_CHAR_IMAGE_HEIGHT = 30

MIN_CONTOUR_AREA = 100

def loadKNNDataAndTrainKNN():

    try:
        npaClassifications = np.loadtxt("classifications.txt",
np.float32)
    except:
        print("error, unable to open classifications.txt, exiting
program\n")
        os.system("pause")
        return False
    # end try

    try:
        npaFlattenedImages = np.loadtxt("flattened_images.txt",
```

```
np.float32)
    except:
        print("error, unable to open flattened_images.txt, exiting
program\n")
        os.system("pause")
        return False
    # end try

    npaClassifications =
npaClassifications.reshape((npaClassifications.size, 1))

    kNearest.setDefaultK(1)

    kNearest.train(npaFlattenedImages, cv2.ml.ROW_SAMPLE,
npaClassifications)

    return True

def detectCharsInPlates(listOfPossiblePlates):
    intPlateCounter = 0
    imgContours = None
    contours = []

    if len(listOfPossiblePlates) == 0:
        return listOfPossiblePlates

    for possiblePlate in listOfPossiblePlates:

        possiblePlate.imgGrayscale, possiblePlate.imgThresh =
Preprocess.preprocess(possiblePlate.imgPlate)

        if Main.showSteps == True:
            cv2.imshow("5a", possiblePlate.imgPlate)
            cv2.imshow("5b", possiblePlate.imgGrayscale)
            cv2.imshow("5c", possiblePlate.imgThresh)

        possiblePlate.imgThresh =
cv2.resize(possiblePlate.imgThresh, (0, 0), fx = 1.6, fy = 1.6)

        thresholdValue, possiblePlate.imgThresh =
cv2.threshold(possiblePlate.imgThresh, 0.0, 255.0, cv2.THRESH_BINARY
| cv2.THRESH_OTSU)

        if Main.showSteps == True:
            cv2.imshow("5d", possiblePlate.imgThresh)

        listOfPossibleCharsInPlate =
findPossibleCharsInPlate(possiblePlate.imgGrayscale,
possiblePlate.imgThresh)

        if Main.showSteps == True:
            height, width, numChannels =
possiblePlate.imgPlate.shape
            imgContours = np.zeros((height, width, 3), np.uint8)
            del contours[:]

            for possibleChar in listOfPossibleCharsInPlate:
                contours.append(possibleChar.contour)
```

```
            # end for

            cv2.drawContours(imgContours, contours, -1,
Main.SCALAR_WHITE)

            cv2.imshow("6", imgContours)

        listOfListsOfMatchingCharsInPlate =
findListOfListsOfMatchingChars(listOfPossibleCharsInPlate)

        if Main.showSteps == True:
            imgContours = np.zeros((height, width, 3), np.uint8)
            del contours[:]

            for listOfMatchingChars in
listOfListsOfMatchingCharsInPlate:
                intRandomBlue = random.randint(0, 255)
                intRandomGreen = random.randint(0, 255)
                intRandomRed = random.randint(0, 255)

                for matchingChar in listOfMatchingChars:
                    contours.append(matchingChar.contour)

                cv2.drawContours(imgContours, contours, -1,
(intRandomBlue, intRandomGreen, intRandomRed))

            cv2.imshow("7", imgContours)

        if (len(listOfListsOfMatchingCharsInPlate) == 0):

            if Main.showSteps == True:
                print("chars found in plate number " + str(
                    intPlateCounter) + " = (none), click on any
image and press a key to continue . . .")
                intPlateCounter = intPlateCounter + 1
                cv2.destroyWindow("8")
                cv2.destroyWindow("9")
                cv2.destroyWindow("10")
                cv2.waitKey(0)

            possiblePlate.strChars = ""
            continue

        for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
            listOfListsOfMatchingCharsInPlate[i].sort(key = lambda
matchingChar: matchingChar.intCenterX)
            listOfListsOfMatchingCharsInPlate[i] =
removeInnerOverlappingChars(listOfListsOfMatchingCharsInPlate[i])


        if Main.showSteps == True:
            imgContours = np.zeros((height, width, 3), np.uint8)

            for listOfMatchingChars in
listOfListsOfMatchingCharsInPlate:
                intRandomBlue = random.randint(0, 255)
                intRandomGreen = random.randint(0, 255)
                intRandomRed = random.randint(0, 255)
```

```
                del contours[:]

                for matchingChar in listOfMatchingChars:
                    contours.append(matchingChar.contour)


                cv2.drawContours(imgContours, contours, -1,
(intRandomBlue, intRandomGreen, intRandomRed))

            cv2.imshow("8", imgContours)

        intLenOfLongestListOfChars = 0
        intIndexOfLongestListOfChars = 0

        for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
            if len(listOfListsOfMatchingCharsInPlate[i]) >
intLenOfLongestListOfChars:
                intLenOfLongestListOfChars =
len(listOfListsOfMatchingCharsInPlate[i])
                intIndexOfLongestListOfChars = i


        longestListOfMatchingCharsInPlate =
listOfListsOfMatchingCharsInPlate[intIndexOfLongestListOfChars]

        if Main.showSteps == True:
            imgContours = np.zeros((height, width, 3), np.uint8)
            del contours[:]

            for matchingChar in longestListOfMatchingCharsInPlate:
                contours.append(matchingChar.contour)


            cv2.drawContours(imgContours, contours, -1,
Main.SCALAR_WHITE)

            cv2.imshow("9", imgContours)


        possiblePlate.strChars =
recognizeCharsInPlate(possiblePlate.imgThresh,
longestListOfMatchingCharsInPlate)

        if Main.showSteps == True:
            print("chars found in plate number " + str(
                intPlateCounter) + " = " + possiblePlate.strChars +
", click on any image and press a key to continue . . .")
            intPlateCounter = intPlateCounter + 1
            cv2.waitKey(0)


    if Main.showSteps == True:
        print("\nchar detection complete, click on any image and
press a key to continue . . .\n")
        cv2.waitKey(0)
```

Plagiarism Report



report

ORIGINALITY REPORT

13%	10%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	opac.pucv.cl Internet Source	7%
2	Submitted to Sri Lanka Institute of Information Technology Student Paper	1%
3	Submitted to General Sir John Kotelawala Defence University Student Paper	<1%
4	arxiv-export-lb.library.cornell.edu Internet Source	<1%
5	Submitted to Universiti Teknologi Malaysia Student Paper	<1%