Chapter One

Introduction

1.1 Background to the Study

Traffic surveillance is a critical component of the network of intelligent transportation systems (ITS)[1]. Providing reliable, real-time traffic monitoring has a significant impact on highway and roadway efficiency and safety. Increased demand for road transportation as a result of population growth and its impact on traffic safety have been a major concern for transportation agencies. By 2032, the federal highway administration (FHWA) in the United States projects a 23 percent increase in vehicle miles travelled (i.e.1.04 percent annual growth)[2]. According to World Health Organization (WHO) statistics, 1.25 million people die and up to 50 million are injured on the world's roads each year[3].

In 2017, 5,049 people were killed in road traffic accidents in Nigeria. It increased to 5,181 the following year. By 2019, the total number of road traffic fatalities had risen to 5,483, and was expected to reach 5,574 by the end of 2020[4]. To avoid the physical and emotional stress associated with traffic jams, impatient commuters frequently look for ways to circumvent the congestion to the point of violating established traffic laws[5]. Violations of this law have resulted in numerous unprecedented accidents, as well as the loss of valuable property and lives. The majority of road accidents occur as a result of rule violations such as speeding, driving on the wrong side, and driving on the wrong side of the road. To prevent such traffic violations, traffic police must be present on the road and constantly check to see if any vehicle is violating the rule[5]. In Nigeria, the most common violations of road safety laws are the failure to wear seat belts, exceeding expected speed limits, reckless driving, driving with an unauthorized plate number, abandoning the use of a fire extinguisher, overtaking at dangerous points, overloading, texting while driving, and failing to obey traffic lights and signs[5]. Excessive speeding, frequent and unsafe lane changes, failure to signal, lane blocking, tailgating, disregard for traffic control, driving against traffic, aggressive use of horns, provocative gestures, failure to wear a seat belt, and verbal abuse, creation of multiple lanes that narrow into a junction, and on off-road space are all examples of traffic violations[5]. Motorists frequently commit a combination of these offences, putting other people's lives and property at risk, because the majority of drivers have no regard for other road users and use the vehicle to vent their frustrations and anger[5].

In 2013, the United States' National Highway Traffic Safety Administration (NHTSA) reported 32,719 fatalities and 2.313 million injuries, 28 percent of which were caused by speed[6]. Truck-related traffic fatalities are estimated to result in 4,000 deaths and 100,000 injuries each year. With rapid population growth and more than 60% of the world's population projected to live in cities by 2050, cities face significant urban planning challenges[7]. They are not only confronted with a rapidly growing population, but also with social and environmental challenges. To better adapt to change, cities require long-term strategies that lead to sustainability[7]. While there is no universally accepted definition of a smart city, it is frequently defined as the use of information and communication technologies (ICT) to create tools that respond to people's needs through sustainable solutions to social and economic challenges[7].

Road traffic is a complex phenomenon that involves the interaction of numerous entities (pedestrians, cars, trucks, buses, tramps, and bicycles, for example) when they share common infrastructure. Due to infrastructure constraints and an increasing vehicle population, traffic management and control is a complex task that requires the application of dedicated algorithms in conjunction with precise traffic data (both historical and current)[8]. The information on the number and type of vehicles is beneficial in reducing travel times and emissions[8].

Precise traffic data enables us to optimise traffic control effectiveness while also adapting management policies to changing conditions and forecasting infrastructure bottlenecks[9]. For instance, the city of Shenzhen, China, reports a vehicle population of over 2 million[10]. Over 600 traffic surveillance cameras have been installed on roadways to capture images of passing vehicles 24 hours a day[10]. On average, over 1,200 vehicles are captured and transmitted to the central server via the network per second[10]. Additional cameras will be installed on roadways in the near future. When metropolitan areas such as Beijing, Shanghai, New York City, and Tokyo are included, the situation becomes even worse.

Automatic licence plate recognition (AVLPR) has grown in popularity in a variety of applications, including road traffic monitoring, vehicle tracking, parking, and intelligent transportation systems (ITS)[11]. A licence plate (LP) is a metal plate with characters and words that is attached to the exterior of a vehicle and used to identify it[11]. Due to the fact that (LP) is discriminated against in different countries based on its shape, size, language, signs, and colours. Numerous methods have been proposed for (AVLPR), depending on the characteristics and regulations of the (LP) country. Locating a licence plate against a complex background is a difficult task. Thus, several critical factors should be considered in order to obtain a successful (LP) extraction, including plate size, image quality, plate styling, illumination condition, plate location, and background specifics[10].

A typical LPR system consists of four components: acquiring an image of the vehicle, localization and segmentation of the licence plate, character segmentation and standardisation, and character recognition[10]. The locating operation's performance is critical for the entire system, as it has a direct impact on the accuracy and efficiency of subsequent steps. It is, however, a difficult obstacle to overcome due to the variety of illumination conditions and complex backgrounds. Numerous methods for locating licence plates have been proposed, including the edge detection method, line sensitive filters to extract plate areas, the window method, and mathematics morphology method[12,13]. While these algorithms are capable of processing the licence plate's location, they have significant disadvantages such as sensitivity to brightness, a longer processing time, and a lack of adaptability to changing environmental conditions.

(LPR) plays an important role in numerous applications such as unattended parking lots, security control of restricted areas, and traffic safety enforcement[14]. This task is quite challenging due to the diversity of plate formats and the non-uniform outdoor illumination conditions during image acquisition, such as backgrounds, illumination, vehicle speeds, and distance ranges between the camera and the vehicle. Therefore, most approaches work only under restricted conditions such as fixed illumination, limited vehicle speed, designated routes, and stationary backgrounds. A typical system for LPR consists of four parts, i.e., obtaining an image of the vehicle, license plate localization and segmentation, character segmentation and standardization, and character recognition[14]. The performance of the locating operation is crucial for the entire system, because it directly influences the accuracy and efficiency of the subsequent steps. However, it is also a difficult obstacle to overcome because of different illumination conditions and various complex backgrounds. Researchers have proposed many methods of locating the license plates, such as the edge detection method, line sensitive filters to extract the plate areas, the window method, and the mathematics morphology method. Although these algorithms can process the license plate’s location, they possess formidable disadvantages such as sensitivity to brightness, longer processing time, and lack of versatility in adapting to the varying environment. Character segmentation has, in the past, been accomplished by such techniques as projection, morphology, relaxation labelling, and connected components. There have been a large number of character recognition techniques reported such as model match, Bayes' classifier, artificial neural networks, fuzzy c-means, support vector machine, Markov processes, and K-nearest neighbour classification[14]. Although these algorithms can process the license plate segmentation and recognition, most of them only process a single-line character segmentation.

1.2 Statement of Problem

As cities grow in sizes, the numbers of roadway traffic increases dramatically. Speeding, licence plate cloning, vehicular robbery, and driving under the influence of alcohol are becoming increasingly serious violations and offences that can result in severe accidents, significant property loss, and even jeopardize personal safety. Thus, in order to maintain a safe driving environment, it is necessary to detect various traffic violations, put an end to them, and apprehend the violators as soon as possible. As a result, a real-time monitoring system capable of detecting licence plates and reporting traffic violations is in high demand.

In recent years, research on vehicular traffic flow analysis has been conducted, utilizing precise traffic data provided by traffic monitoring systems that are typically integrated with road infrastructure. These systems enable the detection and classification of vehicles in specific areas by utilizing data from sensors (inductive loops, video detectors, magnetometers, and others), IP cameras capable of providing real-time traffic monitoring reports, and a variety of others that are costly and difficult to maintain. Additionally, numerous research studies have been conducted over the last decade to improve the process of licence plate recognition (LPR) by using some traditional image preprocessing algorithms such as plate localization, character extraction, and pattern recognition. A significant disadvantage of these infrastructure-integrated solutions is their lack of flexibility and high installation and maintenance costs. To address these shortcomings, this work considers the application of new technologies in traffic monitoring through the use of a web-based license plate number identification system with low-cost inputs and maintenance. Hence, this research.

1.3 Research Aim and Objectives

The aim of this research is to develop a web based prototype for traffic monitoring and license plate detection system using low-cost inputs. The specific objectives are to:

i. develop a license plate detection system prototype device using camera (phone) and wireless technology can capture and detect plate numbers.

ii. perform extraction and recognising of image of vehicles number plate using machine learning algorithms such as ML5.js and OpenCV.js for plate identification and detection.

iii. wrap up web app in mobile app using the tunnel server

iv. evaluate the performance of the system

1.4 Research Question

i. How can we make traffic monitoring and license plate detection system less expensive?

ii. How can we develop a prototype device using camera (phone) and wireless technology?

iii. In what way would a plate number be recognized and detected

iv. Does the designed system work as expected**?**

1.5 Significance of the Study

The findings of this study will serve as a reference point for the government and all other stakeholders when it comes to highway security surveillance monitoring. Since recently, the nation's security has deteriorated due to various forms of kidnapping, robbery, and terrorism. While installing surveillance and security is not cheap in general, this study presents a less expensive, faster, easier, and more durable model that can be used to secure homes, offices, schools, private, and government facilities, among others. Additionally, the proposed design will be able to provide information about vehicle movement and flow within the road network. These details can help identify problem areas and aid in incident management decision-making. The photograph of the vehicle, plate numbers, and driver can be stored, retrieved, and sent to the appropriate surveillance mail in the event that evidence is required in disputes or criminal situations. This will result in an overall increase in the Nation's security and a reduction in insecurity.

Academically, the study will contribute to the body of knowledge regarding surveillance systems. The findings of this study will also serve as a reference for computer science students, lecturers, and researchers. It can also propel further research on the topic. Additionally, findings may result in the development of new theories regarding surveillance systems used for traffic monitoring and plate number recognition.

1.6 Scope of the Study

The project will cover the development of a real time prototype device for vehicle surveillance and monitoring using camera, wireless network and web based technology for License plate recognition and algorithm.

1.7 Operational Definition of Terms

Algorithms: is a set of well-defined instructions to solve a particular problem

Intelligent Transportation Systems: state-of-the-art wireless, electronic, and automated technologies to provide innovative services relating to different modes of transport and traffic management and enable users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks.

Licence Plate (LP): metal plate with characters and words that is attached to the exterior of a vehicle and used to identify it

Licence Plate Recognition: uses optical character recognition on images to read vehicle registration plates to create vehicle location data

Surveillance: monitoring of behaviour, many activities, or information for the purpose of information gathering, influencing, managing or directing.

Endnotes

[1] Z Lv, S Zhang, W Xiu. *Solving the security problem of intelligent transportation system with deep learning*. IEEE Transactions on Intelligent Transportation Systems. 2020 Mar 20;22(7):4281-90.

[2] W Balid, HH Refai. *Real-time magnetic length-based vehicle classification: Case study for inductive loops and wireless magnetometer sensors in Oklahoma state*. Transportation Research Record. 2018 Dec;2672(19):102-11.

[3] A Ditcharoen, B Chhour, T Traikunwaranon, N Aphivongpanya, K Maneerat, V Ammarapala. *Road traffic accidents severity factors: A review paper.* In 2018 5th International Conference on Business and Industrial Research (ICBIR) 2018 May 17 (pp. 339-343). IEEE.

[4] https://www.dataphyte.com/latest-reports/security/road-traffic-crashes-in-nigeria-claims-41709-lives-in-8 years

[5] Aghiomesi, Irunokhai & O., J. & S., O. & O., B. *Analysis of Traffic Light Violation on Nigerian Roads (A CASE STUDY of Sango T Junction, IBADAN, OYO STATE).* **International Journal of Computer Applications.** 176. 8-13. 10.5120/ijca2020920299. (2020)

[6] H Xu, Z Cai, R Li, W Li. *Efficient City Cam-to-Edge Cooperative Learning for Vehicle Counting in ITS.* IEEE Transactions on Intelligent Transportation Systems. 2022 Feb 14.

[7] J Barthélemy, N Verstaevel, H Forehead, P Perez. *Edge-computing video analytics for real-time traffic monitoring in a smart city.* Sensors. 2019 Jan;19(9):2048.

[8] M Lewandowski, B Płaczek, M Bernas, P Szymała. *Road traffic monitoring system based on mobile devices and bluetooth low energy beacons.* Wireless communications and mobile computing. 2018 Jul 17;2018.

[9] Y Wang, X Yang, H Liang, Y Liu. *A review of the self-adaptive traffic signal control system based on future traffic environment.* **Journal of Advanced Transportation.** 2018 Jun 27;2018.

[10] W Liu, J Chen, Y Luo, Z Shi, X Ji, H Zhu. *Study on the Annual Reduction Rate of Vehicle Emission Factors for Carbon Monoxide: A Case Study of Urban Road Tunnels in Shenzhen, China.* Advances in Civil Engineering. 2020 Sep 3;2020.

[11] BB Yousif, MM Ata, N Fawzy, M Obaya. *Toward an optimized neutrosophic K-means with genetic algorithm for automatic vehicle license plate recognition (ONKM-AVLPR).* IEEE Access. 2020 Mar 9;8:49285-312.

[12] P Dhar, S Guha, T Biswas, MZ Abedin. *A system design for license plate recognition by using edge detection and convolution neural network.* In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) 2018 Feb 8 (pp. 1-4). IEEE.

13. Z Selmi, MB Halima, U Pal, MA Alimi. *DELP-DAR system for license plate detection and recognition.* Pattern Recognition Letters. 2020 Jan 1;129:213-23.

14. MA Bennet, B Thamilvalluvan, PP Alphonse, DR Thendralarasi, KJ Sujithra. *Performance and Analysis of Automatic License Plate Localization and Recognition from Video Sequences*. **International Journal on Smart Sensing and Intelligent Systems. 2017 Dec 1;10(5):330.**

Chapter Two

Literature Review

2.1 Conceptual Framework

2.1.1 Intelligent Transportation System

Intelligent transportation system" (ITS) refers to a collection of technologies that have been integrated into the transportation system. These technologies include processing, control, communication, and electronic components. In addition to that, it incorporates cutting-edge methods of managing traffic[1]. The incorporation of the technologies that have been mentioned into ITS is part of an effort to save lives, as well as money and time.

In addition to that, it encompasses a wide range of fields, including transportation engineering, telecommunications engineering, computer science, finance, electronics, and commerce, amongst others. It is possible to say that the goal of ITS is to reap the benefits of the appropriate technologies in order to create roads, vehicles, and users that are more intelligent. Because of the capacity of a computer to eradicate the possibility of error caused by human intervention, ITS will soon be dependent on the technology of computers. In today's world, there is technology that can lead people to their destinations while avoiding traffic jams, and it is expected that this type of technology will continue to advance in the years to come.

In spite of the fact that it may appear to be something from the distant future, intelligent transportation systems (ITS) are currently on track to become a practical reality in the not too distant future[2]. It is essential for the transportation sector to make use of the latest technology in order to reap the benefits of these improvements in human life, which are made possible by the applications of advanced technologies. The Intelligent Transportation System (ITS) has the potential to evolve into something more than a transportation system that relies on the four-way traffic signal as its primary technology[3].

According to the history, the first large-scale application of a computerised signal control system in the world was implemented in Metropolitan Toronto during the early 1960s; however, this ITS field did not start to mature until the early 1990s[4]. The ITS industry has been driven by a number of different forces. Researchers in the field of transportation came to the conclusion that the construction of new roads will never be able to keep up with the rising demand for travel. Some nations, having spent billions of dollars on the construction of road networks and infrastructure, are currently faced with the challenge of refreshing or renewing this enormous network and making the best use of the existing networks and infrastructure before expanding the network and infrastructure. This is a challenge that must be overcome before any additional investments can be made in either the network or the infrastructure. In addition to that, the surrounding environment has emerged as a significant contributor to the ITS field.

2.1.1 ITS Subsystems

Improved utilisation of the available roadway capacity is one of the goals of the Intelligent Transportation System (ITS). It is accomplished by enhancing the distribution of traffic and dynamically rerouting traffic away from the areas of the network that are experiencing congestion in order to cut down on the segments' utilisation of the network. In addition to that, one of the goals of ITS is to increase the capacity of the roads already in existence[1]. This goal can be accomplished through the automation of driving and the elimination of the human behaviour component in general.

2.1.2 License Plate Recognition System

There are currently over a half a billion vehicles operating on the world's roadways. The licence plates, which are present on every vehicle, serve to distinguish one vehicle from another and to identify vehicles. It is obvious that due to the enormous wave of vehicles, the human resources, even on a small scale, would not be sufficient to check all of the vehicles without the use of computers and techniques for signal processing. The use of automation in this sphere is of critical importance and ought to be given some thought. Despite the fact that applications of automatic licence plate detection have only become widespread within the past decade or so, the technology itself has been around for nearly 45 years[5]. Researchers working for the United Kingdom's police scientific development branch created the first working licence plate recognition system in the late 1970s. They started implementing the system at the beginning of the 1980s.

Traffic monitoring, parking management, accident reporting, identifying drivers who cause traffic signal violations or drive faster than the speed limit, for toll collection, or to identify uninsured motorists are some of the application areas for automatic licence plate recognition. In general, the vast majority of licence plate recognition (LPR) systems are expected to conform to the following objectives: A quick processing speed, the ability to recognise licence plate numbers from images that contain noise, the capacity to function with plates that are tilted, and the capacity to work with a variety of font styles and sizes.

Applications that benefit from licence plate recognition (LPR) systems include traffic surveillance, traffic law enforcement, automatic toll collection, vehicle parking identification, and vehicle access control in a restricted area[6]. License plate detection (LPD), licence plate character segmentation (LPS), and licence plate character recognition are the three components that make up an LPR system in most cases (LPR). The reading of a vehicle's licence plate is typically the very first step in the LPR system[7]. Its goal is to locate the licence plate, which will then supply the LPR procedure that comes after it with accurate information about the region. In order to avoid the extremely time-consuming task of processing each individual pixel in the input image of the system, the process of licence plate detection is required prior to the process of licence plate recognition.

2.1.2.1 Licence Plate Extraction Methods

First and foremost, the most important and crucial stage for ANPR systems is the successful extraction of the licence plate number from the image or video[8]. The extraction rate is the ratio of the number of plates that were successfully extracted to the total number of images that were input or vehicles that were detected from the scene[9]. In most cases, there is only one camera installed on each lane; however, some more sophisticated cameras may be able to support more than one lane due to the higher resolutions they offer. Installing multiple lanes requires more readers to identify the licence plates on passing vehicles, as well as additional hardware, which results in increased costs to service and maintain.

It's possible that real-time scenarios will face multiple obstacles. For example, the camera that is installed in a stationary position may acquire images of the vehicle with the number plate characters that are skewed or tilted[8]. It's possible that the licence plate was broken, covered in dirt, or placed in a position where it wasn't visible to the camera (since different types of vehicles have their number plates affixed at different positions of the vehicle body). It can be difficult for the system to effectively extract the licence plate number due to environmental factors such as light, motion blur, reflections, fog, and other conditions of a similar nature. It's possible that algorithms that use geometrical features to extract rectangular-shaped licence plates could run into problems if there are multiple shapes drawn or pasted over the body of the car that are similar to one another[8,10]. In addition to having features with a rectangular shape, additional algorithms are required in order to get rid of the undesirable regions. The algorithms need to be reliable so that they can distinguish between the licence plate and the other things that are visible in the image.

2.1.2.2 LP Extraction Using Color Features

Certain countries and regions have a preference for a particular colour for the licence plates on vehicles. As we investigate the work that has been done up to this point, we are also putting color-based extraction of number plates from images through their paces for ANPR systems[8]. The concept of the colour combination of the number plates is generally involved in the general approach to extracting the number plates. In addition, these characteristics are one of a kind, and such colour contrasts can only be found in a particular region of a plate. In the course of working on a method for the distinct designs of Chinese licence plates this work was done[8,11]. The method used every pixel from the image that was obtained, which were then categorised according to their hue, brightness, and saturation (HLS). Instead of the traditional RGB colour model's six divisions, the HLS colour model divides colours up into 13 distinct categories. The colour model was utilised rather than the grayscale. The colour selection for the division is determined by the formats used for number plates on the China mainland. In the experiment that they conducted, 90 percent of the total images were able to be recognised correctly in a variety of lighting conditions[11].

A similar study found that the recognition of only certain colours, including black, green, and white, are used in number plates. Color edge detectors work only on three types of coloured edges, including combinations of red and white, black and white, and green and white[8]. A total of approximately 1088 images taken in a variety of settings were analysed while conducting tests under a variety of conditions. There is a localization rate of 97.9% with the number colour plate[8]. Following the process of converting the image's true colours into the hue, saturation, and lightness (HLS) colour model, a neural network is used to classify the colours of each individual pixel. White, green, and red are the colours that are output by the neural network. These are the specific colours that are used in Korean number plates. In order to determine which part of the number colour plate contains the highest colour density, the same colour combination is projected in both the horizontal and the vertical directions[8].

In addition, the combination of character colour and number plate colour is used in a study that is very similar to this one in order to produce an edge image. After that, a horizontal scan is performed on the image that was generated. In the event that there is any pixel with a value that falls within the range of the number colour plate, the colour range of the pixels that are immediately to its left and right is investigated. In order for a pixel to be considered an edge pixel, there must be a minimum of two and a maximum of two horizontal neighbours that belong to the same range. In the end, each of the edges in the newly created edge image is scrutinised for analysis. This is done so that the colour plate regions can be determined[8].

When segmenting the colour images into candidate regions for a study, the mean shift algorithm is used as the primary tool[8]. In a later stage, each of these candidate regions will either be designated as a number plate area, or it will be eliminated. A detection rate with an accuracy of 97.6% was measured and recorded. Another algorithm, specifically a fast-mean-shift method, was suggested to be used in the newly developed licence plate extraction framework that is based on fast mean shift[12]. The Mean Shift method sees widespread application in feature analysis, including the segmentation of images and videos. The authors made use of the function that could segment the complicated background into possible segments, and as a result, they were able to isolate the candidate number plate region. The proposed method was applied to a database consisting of 400 images, each of which had a size of 640 by 480 pixels and had been taken in different lighting conditions. It was determined that the accuracy of detection was 92.6%[12].

Some researchers came up with an algorithm based on fuzzy logic in order to address the issues that were caused by the different illuminations. During the process, the HSV colour space is used. When dealing with fuzzy sets, each of the aspects of HSV's component parts are at first mapped in terms of the various membership-functions. After that, the fuzzy classified function is illustrated by fusing the weighted membership degrees of all three HSV components[8]. In order to evaluate how effectively the algorithm works, three distinct image datasets were used. Number plates in Shanghai have a recognition rate of 95.05%, Shenzhen's number plates have a recognition rate of 92.17%, and Beijing's number plates have a recognition rate of 93.23%[8]. The statistical threshold is determined by the colour model of HSI that has been adopted. HSI is an acronym that stands for hue, saturation, and intensity, and it is used to determine candidate regions[8,13]. If the colour of the number plates and the bodies of the cars or vehicles being investigated are identical, then this method can be utilised to locate potential candidates. When attempting to identify yellow and green pixels on a number plate, the standard deviation and mean of hue are both utilised as analytical tools. The intensity and saturation components of the HSI are utilised in order to send the pixels of the number plate that are white, green, and yellow from the images of the relevant vehicles.

The use of colour information in conjunction with the number plate extraction method has a number of beneficial effects. That is to say, it is up to you to identify the licence plates that have either been distorted or tilted[8]. Despite this, there are a few drawbacks associated with it. It can be difficult to define the colour of the pixel by using the RGB value, particularly under certain illumination conditions. The HLS colour model, which is used as a stand-in, is extremely sensitive to the presence of noise. The disadvantage of incorrect detection of the image parts that have the same colour on the number plate as the car body is something that must be dealt with by methods that operate based on colour projection[8].

Utilizing Color characteristics, It is possible to make use of the fact that licence plates from some countries have distinctive colour schemes in order to facilitate the extraction of licence plates from those countries. In the context of scenarios involving the extraction of licence plates, it is assumed that the colour combination of a plate and its characters is one of a kind and that this combination can only be found in a particular plate region[8]. According to the specific formats of Chinese licence plates, all of the pixels in the input image can be categorised into 13 different groups using the hue, lightness, and saturation (HLS) colour space. This was done in accordance with the author's findings.

A Neural Network for the Classification of Each Pixel's Color In HLS colour space, a Neural Network is utilised in order to classify the colour of each individual pixel[8]. The classification index produced by the neural network, which is represented by the colours green, red, and white respectively. These are the colours that are used for licence plates in Korea. Searching for License Plate Color Using a Genetic Algorithm Genetic algorithm (GA) is an algorithm used in computational intelligence for fitness function optimization that is used to search for licence plate color8. During training with pictures taken under a variety of lighting conditions, GA is utilised to determine thresholds for the plate colour.

Through the use of a function, the thresholds are connected to the average brightness. Since the average brightness is calculated for each image that is input, it is possible to determine both the upper and lower thresholds by utilising this function. Each pixel will be labeled if the intensity of its value falls between this pair of thresholds, and the region will be considered a candidate for a plate if the connectivity of these thresholds forms a rectangle that has an eligible aspect ratio[8]. Under a variety of lighting scenarios, attempting to define the pixel colour by using the RGB value is difficult. In addition, this alternative colour model, known as HLS, is very sensitive to background noise. When the colour of the licence plate is not unique to the plate regions, but can be found elsewhere, such as on the body of the vehicle or in the environment, problems may arise for methods that use colour projection[8].

2.1.2.3 LP Extraction Using Edge or Boundary Information

The aspect ratio of the number plate is typically known, and its shape is typically rectangular. In a study that was very similar to this one, the images were initially scaled to a predetermined aspect ratio[8]. The authors evaluated and tested a variety of algorithms that had been proposed in previous research, and then they compared the results by putting each algorithm into practise with their own dataset. The vertical edge information was used as the foundation for one of the number plate extraction methods that they investigated. It does this by employing Sobel operator to locate the vertical edges[14]. The licence plate is localised by comparing the predetermined minimum and maximum lengths with those of the extracted edges and then removing the edges that are not desired. The overall extraction rate for 141 images comes to 65.25 percent, which is significantly lower than the 99.99% that was initially reported in a research8. In a different study, the information from the vertical and horizontal edges of the histogram are used for number plate extraction. After testing 50 images with a variety of fonts and lighting conditions, the extraction accuracy was found to be 90%[8,15].

Finding all of the rectangles in the acquired images is one of the most common tasks for edge detection algorithms, which are commonly used to extract number plates. The majority of the time, there is a distinct colour transition between the car body and the area around the licence plate. Using edge detection filters or algorithms, one can determine which of the two is being discussed by locating the edges of the image[8]. According to the findings of a research study, the successful edge extraction was accomplished with the help of a straightforward algorithm called the Sobel filter. Detecting edges can be done by performing vertical edge detection to get the vertical lines, horizontal edge detection to get the horizontal lines, or simultaneous use of both to get a complete rectangular shape[8]. Vertical edge detection is used to get the vertical lines, and horizontal edge detection gets the horizontal lines. Using the geometric properties, specifically the location of the rectangle lines, it is possible to identify the licence plate number.

Gabor filters are widely regarded as the superior option for structure recognition due to the exceptionally good results they display when it comes to excluding clamour and noise while preserving edges[8]. It is possible to extract the indented number plate region by making use of the magnitude of the vertical edges, which is the extraction feature that is regarded as being the most reliable. The vertical edges are compared in order to obtain the intended rectangles, which are then filtered in order to find the one rectangle that corresponds to the area occupied by the number plate using the aspect ratio that is already known. It is stated that the number plate can be successfully extracted from the remaining edges in the image if the background edges are removed and the vertical edges are obtained first. This is because the number plate is located in the centre of the image. The total processing time for an image with dimensions of 384 by 288 milliseconds is 47.9 ms, and the detection rate for 1165 test images was approximately one hundred percent[8].

It has been suggested that the Vertical Edge Detection Algorithm, also known as VEDA, is the most reliable algorithm for edge detection among those that have been proposed for number plate extraction[17]. The percentage of information that can be extracted from 50 images taken under different lighting conditions is 96%. The horizontal edges open the door to the possibility of making mistakes. The car bumper is primarily to blame for these types of mistakes. The areas that could be used for number plates are those on blocks that have a high edge magnitude. This method can be used to determine the licence plate even if the image is blurry and is independent of the edges that define the boundaries of the plate itself. When using a pair of 180 unique images, the authors achieved a recognition accuracy of 92.5% for the characters[8]. In a similar manner, tests were carried out in a study to check the inspection status of motorcycles by recognising the number plate of each vehicle. The success rate for roadside test images was 95.7%, while the success rate for inspection station test images was 93.7%[18].

In a study, the Hough Transform is applied to the boundaries in order to extract the licence plate number[19]. Finding the number plate in the test image requires locating the straight lines in the image. This transformation has the capacity to recognise straight lines that are inclined by up to thirty degrees. However, it is both difficult to compute and requires a significant amount of memory[19]. The generalised symmetry transform is utilised in an investigation that takes place where. Scanning the image in certain directions allows for the detection of the image's corners, which are formed by the image's edges. By employing the generalised symmetry transform and searching for similarities between these corners, the number plate region can be extracted from the image[8]. When using edge-based methods, which are generally regarded as being straightforward and quick, maintaining the continuity of the edges is an essential consideration. By removing the edges that aren't needed through the application of some morphological steps, the extraction rate can be significantly boosted[20].

Using the Sobel filter to detect edges, which is how the boundaries of licence plates are represented because of the colour transition between the licence plate and the car body[21]. When performing horizontal edge detection, two horizontal lines are located, and when performing vertical edge detection, two vertical lines are located. Both sets of lines are located. When two different sets of lines are located at the same time, the rectangle is considered to be fixed. Gabor filters were also used to detect licence plate regions, and they achieved a good performance when images were of a fixed angle[22]. This was achieved by using a combination of the previous two methods. Because they are sensitive to differences in scale and direction, Gabor filters are useful for analysing textures. This makes them an excellent choice. In addition, the intuitive knowledge that the licence plate is of some shape, most likely rectangular, and that the aspect ratio of the shape is known leads to the development of methods that are commonly used to extract plates from all possible rectangles. Methods of Edge Detection are examples of methods that can locate rectangles. Only vertically or horizontally oriented edges can be detected, and statistical analysis can be used to identify potential regions for licence plate placement [21].

In addition, geometric features can include the height, width, area, aspect ratio of a rectangle, as well as the number of rectangles that are included in another rectangle, among other things. Rectangular elements of licence plates can be identified by employing productive geometric attributes to track down the lines that compose the rectangles[8,23].

Methods that are based on blocks consider individual blocks to be potential licence plate regions. A significant increase in edge magnitude is an indicator of such blocks[8]. Block processing is not dependent on the edges of the licence plate boundary, which is one of the advantages of this method. This allows the method to work well with licence plates that do not have clear boundaries. They used 180 image pairs and found that their method was accurate 92.5% of the time. Transforms Methods of image transformation such as the Hough transform (HT), the wavelet transform (WT), and other transforms such as the Generalized symmetry transform (GST) have been utilised in the process of licence plate detection. It has been described how to perform boundary-based extraction using the Hough transform. The Hough transform is useful for locating licence plates because it can identify straight lines in the image, and one of its advantages is that it can detect lines that are tilted at an angle of up to 30 degrees[8].

The Hough Transform, on the other hand, has the drawback of being both time- and memory-intensive[24]. This is the transform's primary disadvantage. The Hough transform and the contour algorithm are both incorporated into the boundary line-based method. The extraction of licence plates was successful to the extent of 98.8 percent. The licence plates can also be extracted using another transform known as GST. After the image has been edged, it is scanned in various directions to find corners. Then, the generalised similarity measure (GST) is applied to determine whether or not two corners are similar and to create licence plate regions8. In conclusion, methods that make use of edge information to detect licence plates have a number of benefits including being straightforward and quick. However, they require the edges to have a continuous appearance. It is possible that licence plates will not be accurately detected if the edge information is not reliable and complete[8].

2.1.2.4 LP Extraction Using Global Image Information

The image is scanned in the process known as binary image processing, and its pixels are then labelled into components using a method known as Connected Component Analysis (CCA)[25]. This method is based on the connectivity of the pixels. Connected Component Analysis (CCA) is an essential one that is utilised in binary image processing[25]. It sees widespread application in Scene Text Recognition challenges, and more specifically, it sees application in licence plate recognition challenges. CCA examines a binary image and assigns a label to every pixel based on the connectivity of those pixels, with each label standing for a different component[26]. A connected component is formed by a group of pixels that have the same label. Measurements pertaining to space, such as area and aspect ratio, are frequently utilised in the process of licence plate detection[26]. Information about the contour can be incorporated to facilitate the connecting of components. On the binary image, a contour detection algorithm is applied in order to locate connected components. Candidates for these connected components are selected based on whether or not they share the same geometrical characteristics as the plate. If the image is of poor quality, which results in distorted contours, then this method may not work properly[8].

In the work that was presented, a two-step process was proposed for the extraction of number plates[27]. In the first step, Otsu's Threshold Method, which is an effective and straightforward method for adaptive thresholding techniques, is utilised in order to deal with the variable light illumination conditions that are present. After that, the CCA technique is used to search the binarized image for any rectangular shapes that might be present there. In the second step, which is applied to the obtained number plate, edge detection is carried out, and then the closed curve method is used, in order to validate that the image that was generated is in fact a number plate. With this method, more than 2500 images in Moroccan format taken from video sequences were examined, and the success rate was found to be 96%[27]. In a study that was similar to this one, the technique of connected component analysis was used, and it recorded a successful extraction rate of 96.6% on a video of poor quality that was over four hours long[8]. On binary images, techniques for detecting contours are utilised in order to localise the connected objects[8]. The geometrical features that have similarities with the licence plate are selected for further processing because of these similarities. However, if the image that is being acquired is of poor quality, this algorithm may produce errors that result in distortion[8].

In addition, in order to extract the number plate region, a pre-stored number plate template is utilised while carrying out the 2D cross correlation. This ensures that the extraction process is unaffected by the location of the number plate in the image[8]. On the other hand, it is known to be a method that takes a lot of time. Detection of licence plates also makes use of two-dimensional cross correlation[28]. This technique makes use of a previously saved licence plate template. Using this template, a 2-D cross correlation is carried out across the entirety of the image in order to pinpoint areas of the picture that are most likely to contain a licence plate. It is possible for it to remove the plate no matter where the plate is located. Nevertheless, the 2-D cross correlation takes a significant amount of time[8].

2.1.2.5 LP Extraction Using Texture Features

These procedures are determined by the characters in the area for the licence plate number. A grey scale depicts a substantial difference as a result of a colour transition by having a high edge density between the colours of the characters on the plate and the background. In order to extract numbers from licence plates, a Local Binary Pattern (LBP) and a Histogram of Oriented Gradients (HOG) are utilized[29]. LBP is the algorithm that is used to classify the texture and calculate a histogram. This is done while taking into consideration the rectangular shape of the number plate along with HOG. A total of 110 images have been located with an accuracy of 89.7%. However, this method is of no use for photographs that are blurry, poorly lit, or oriented at an angle other than 90 degrees[29].

Utilizing the line weight density map is yet another efficient method for plate detection that is used by a lot of different researchers[30]. In order to achieve even better results, this strategy can be combined with others. In the scan line technique, the peaks are created from the colour transition of the grey scale level, which corresponds to the number of characters on the number plate[8]. This technique is used. In a work that was proposed on the horizontal line scanning with multiple thresholds technique for plate detection in real time/complex images, the technique was described. When compared to the traditional model, which was based on Hough transforms and had a detection accuracy of only 69.8% and a longer processing time of 8–10s, the results of the experiments were found to be superior[8]. In contrast, the line scan technique8 was able to achieve an extraction rate of 99.2%. The amount of time spent executing the process of locating the plate was drastically cut down to 0.3–0.5 seconds on average[8].

Texture-based techniques can be computationally challenging when it comes to extracting number plates from these kinds of images. This is especially true if the image contains an excessive number of edges, the background contains multiple elements, or the light illumination is insufficient. The methods that make use of texture features are predicated on the assumption that licence plates contain characters.

These characters cause a significant difference in the grey-level between the characters and the background of the licence plate, or they cause a high edge density area due to colour transition. There is a good reason to suppose that characters are included in licence plates, and it is possible that a variety of techniques were utilised[31].

a. Vector Quantization: The Vector Quantization (VQ) representation provides insight into the components that make up various image regions. These smaller blocks frequently map areas with greater contrast and a greater number of details[32]. VQ is utilised in the process of licence plate detection, and the findings reveal that the detection rate is 98% and that the processing time is approximately 200ms[33].

b. Sliding Concentric Window: A sliding concentric window method is proposed, in which licence plates are regarded as irregularities in the image's texture, and thus abrupt changes in local characteristics indicate the presence of potential licence plates[8].

c. Image Transformations: Image transformations are one of the most popular techniques for extracting licence plates[8]. One of the most important tools for conducting texture analysis is called a Gabor filter. This is due to the fact that it can analyse features on an unlimited scale and in any orientation. It is utilised in a work that achieved a result of 98% when applied to images of a particular fixed angle; however, the drawback is that it requires a significant amount of time to complete[8]. The Discrete Fourier Transform, also known as DFT, is utilised in the process of spectrum analysis to determine the spatial frequency. Row-wise detection is typically used to determine the horizontal position of the plate, while column-wise detection is utilised to determine the vertical position. Another type of transform, known as the wavelet transform (WT), is utilised in the context of a study to identify licence plates. In WT, there are four subbands, and the subimages HL and LH describe the information about the vertical edge information and the horizontal information in two separate ways. A scan of the LH image is used to decide the maximum change in horizontal edges, and this change is indicated by a reference line. Below this line, the vertical edges are projected horizontally to determine the position based on this maximum projection[35].

d. AdaBoost: Adaptive boosting (AdaBoost) is applied in a study to detect license plates. It is incorporated with Haar-like features to obtain cascade classifiers, which are invariant to brightness, color, size, and position of the license plates36. Also, cascade classifiers use gradient density in the first layer and then Haar-like features. The accuracy is 93.5% in license plate detection. AdaBoost is also used but without Haar-like features. They achieved an accuracy of 99% in detecting license plates that are from images with different formats, sizes, various lighting conditions[8].

In conclusion, the detection of licence plates using methods that make use of texture features are advantageous even if the boundary is unclear or has been deformed. This is the case. The fact that they require a lot of computation is, unfortunately, a drawback.

2.1.2.6 LP Extraction Using Character Features

Techniques that are based on the extraction of character features have also been proposed in a variety of research studies that are associated with the process of locating the characters on the plate. These methods look through the image to see if there are any characters on the licence plate. After the relevant characters have been identified, the region of the number plate that contains those characters is then extracted. Instead of directly applying the properties of the number plate, the algorithm used in the study identifies all of the character areas from the image. The character-like region dependent method is utilised in the execution of this process[8,37]. All of the character-based regions that have been identified are discussed and classified with the help of a neural network.

In a separate piece of research, it was found that the aspect ratio of binary image objects was the same as that of the characters, and that approximately 30 image pixels were labelled. In order to determine which lines are straight, the Hough transformation is applied. These binary labelled objects have their upper and lower halves transformed in the same way using the same transformation[8,38]. The area that lies between two lines that are parallel to one another, lie within a given range, and contain the same number of objects as the characters is referred to as the number plate area. This area exists when two lines lie within the same range. In another study, the scale-space analysis was utilised to decipher the licence plate numbers. Using this method, blob-type figures of large size are extracted, along with line-type figures of smaller size, for consideration as character candidates [8,39].

Methods for the detection of licence plates that are based on the localization of characters or the recognition of character features have also been proposed. The goal of these methods is to locate the characters, the regions of which are then extracted to form the licence plate region.

a. Repeating Contrast Changes: The approach used in a study that scanned the input image horizontally, on a scale of 15 or more pixels, in order to look for repeating contrast changes. The assumption here is that there is a good deal of contrast between the characters and the background, and that there are at least three to four characters whose smallest possible vertical size is 15 pixels. When combined with differential edge detection, this method achieves an accuracy of 99%.[8].

b. Same Aspect Ratio As Characters: In a work in which binary objects are labelled based on two characteristics, namely the aspect ratio and the number of pixels, the work in question is a binary one. The components that are connected together are potential candidates for the character. Hough Transform on both the upper side and the lower side, if there are two parallel straight lines detected and the number of connected components is comparable to that of a licence plate, the region will be considered to be a licence plate region[8].

c. SVM-trained SIFT Descriptors: A study proposed a method to classify the regions of the input image using thirty-six AdaBoost classifiers to determine possible character regions, highly-unlike character regions, and regions that are highly unlikely to contain character. In the second stage of the process, a support vector machine (SVM) that has been trained on scale-invariant feature transform (SIFT) descriptors is used to perform a further classification on the possible character regions that were obtained from the first stage. After all of the stages have been completed, the regions that have been positively labelled will be the licence plate region detected[8].

d. MSER: The method known as maximally stable extremal region, or MSER, is applied to a study in order to identify text or character regions on licence plates[40]. An overly simplistic heuristic-based filter will be used to eliminate highly improbable candidate regions from the large amount of regions detected by MSER as potential candidates. The remaining regions will be further filtered by checking the number of positively classified SIFT points; if the number of positively classified SIFT points in those regions is greater than a threshold, then those regions will be considered licence plate regions[8,40].

e. Combing Two or More Features. An additional study combines the characteristics of licence plate detection systems, including the rectangle shape feature, the texture feature, and the colour feature[41]. On a total of 1176 images taken in a variety of settings, the detection rate was found to be 97.3%. Wavelet analysis, an improved version of HLS colour decomposition, and Hough line detection are all components of the method that has been proposed to detect licence plates[8,42].

2.1.2.7 LP Extraction Using Feature Learning

Few extraction methods search for at least two or more characteristics of the number plate, which is necessary for the effective detection of number plates. In this particular scenario, the extraction procedures are categorised as hybrid extraction methods[8].

A work has made use of the object detector known as You Only Look Once (YOLO), which is based on the Convolutional Neural Network (CNN) technology. It is a two-step process that makes use of straightforward data enhancement strategies such as flipped characters and inverted number plates[8,42]. At each stage, the CNNs undergo training and are fine-tuned to improve their performance. The resulting model produces accurate predictions for both of the separate datasets. The first database, which was created by the Smart Surveillance Interest Group (SSIG) and is called the SSIG-SegPlate Database, contains 2000 frames from approximately 101 vehicle videos. The system is able to recognise faces with an accuracy of 93.53% at a frame rate of 47 frames per second, outperforming both the commercial systems OpenALPR and Sighthound, which have recognition rates of 93.03% and 89.80%, respectively, and producing significantly better results than the older techniques, which only managed to achieve an accuracy of 81.80%[8]. The other dataset used has images with varying conditions, which is analogous to how things are in real time. The acronym UFPR-ALPR describes the name of this open database. The dataset known as UFPR-ALPR belongs to the Laboratory of Vision, Robotics, and Imaging (VRI) at the Federal University of Paraná in Brazil. It includes approximately 4500 still images and 150 videos that were captured while both the vehicles and the camera were moving. The vehicle types represented in the dataset range from automobiles to motorcycles to trucks. Also included are buses. This system operates smoothly at 35 frames per second and has an identification rate of 78.33%, in contrast to the test versions of commercial systems, which produced a rate of recognition that was lower than 70%.[8]. In another study, the NP recognition was accurate to a greater degree thanks to the concurrent implementation of character recognition and segmentation using Hidden Markov Models (HMMs). Specifically, the Viterbi algorithm was utilised to arrive at a conclusion regarding the most likely NP[8,44].

For the purpose of a study, colour and texture characteristics are analysed with the assistance of two different neural networks. One of the neural networks is trained to detect plate textures by making use of multiple plate edges during the training process. The other is utilised for the determination of colour. The outputs of these neural networks are utilised in the identification of the candidate regions[45]. A single neural network is used for image scanning in a work with a H x W window, which is the same size as a vehicle plate[8]. This size is comparable to the dimensions of the window. Within this window, this network is used to sense the edges and colour in order to determine whether or not the area in question is a candidate for containing a number plate. In addition, the neural network in another study is used to horizontally scan an HLS image using a window size of 1 x M. Here, M denotes a value that is roughly representative of the width of the licence plate, and the vertical scanning is carried out using a window that is N minus one, where N denotes the height of the plate. The hue value of each pixel in the image is what is used to denote the colour details, and the intensity value of each pixel is what is used to represent the texture details[8,46]. The results of both the horizontal and the vertical scans are combined, and the areas of the plate that are candidates for extraction are found.

In a study, Time-Delay Neural Network (TDNN) processing is utilised for the extraction of number plates. Two of these TDNNs are implemented in the colour and texture analysis of the number plate by checking small windows of the image's horizontal and vertical cross sections[8]. The region with the highest edge density is isolated as the number plate because the pixel values in this region are most similar to those of the number plate. Plate extraction has also been accomplished with the help of a covariance matrix. The spatial information and the statistical data are brought together to form its foundation. Each matrix contains an adequate amount of information, and this information is sufficient to match the area from multiple perspectives. In order to train the neural network effectively and efficiently in order to detect the number plate area, this matrix is used[8].

In another study, the Modified Census Transform (MCT) was used to compute local structure patterns, which was then used to detect licence plate numbers. Following that, two different post-processing steps are utilised in order to lower the number of false-positive results[47]. One of the post-processing steps involves using a position-based technique to differentiate between a vehicle licence plate and a false positive. A false positive has the same general local structure patterns, such as radiators or headlights. The other one is the color-dependent method, which uses the comprehensive colour details of the licence plates. ANPR systems that are based on

Deep Learning (DL) techniques generally address character identification and segmentation as a whole.

A CNN architecture was proposed in a piece of work for the purpose of character segmentation and recognition. The investigation was carried out utilising a dataset that was open to the general public. With their method, 99% of the characters were successfully segmented, but the accuracy of reading the segmented characters was only 93%[48]. Nevertheless, despite the remarkable progress that DL techniques have made in ANPR, there is still a significant demand for ANPR datasets that include annotations of cars/vehicles and NPs. The incremental improvement in the performance of DL methods can be attributed to the training data set. The training of data-hungry deep neural networks will benefit from the availability of a large amount of training data, as will the utilisation of more robust network architectures, in addition to additional layers and parameters[8].

2.1.3 License Plate Segmentation

A pre-processing procedure called licence plate segmentation comes before licence plate recognition (LPR) and is frequently required because the majority of LPR algorithms only accept single-character inputs. On the basis of the information that was used in the segmentation process, various methodologies have been utilised, including the utilisation of pixel connectivity, the utilisation of projection profiles, the utilisation of prior knowledge of characters, and the utilisation of character contours[49].

a. Using Pixel Connectivity. A study that utilised On the binary image that was derived from the input image, segmentation is performed by labelling the connected pixels into connected components. These connected components will then be analysed in aspects such as size and aspect ratio to determine whether or not they belong to licence plate characters[50]. The drawback, on the other hand, is that it is unable to process characters that are either joined or broken.

b. Using Projection Profiles: Because people can tell the difference between characters and plates based on their colours, the values assigned to them in the binary image should also be distinct from one another[51]. Some methods, such as those proposed, are used to project the extracted binary licence plate both vertically and horizontally in order to determine the starting and ending positions of the characters and the position of each character, respectively[52]. The binary information that was previously used in the projection has been replaced with character colour information in the review. The fact that the segmentation of these kinds of methods is not dependent on the positions of the characters is one of their advantages. One of their disadvantages, on the other hand, is that it is dependent on the quality of the image that is being input, as any noise could affect the projection value. In addition, in order to use these kinds of methods, one must be familiar with the total number of characters in advance.

c. Using Prior Knowledge of Characters: The prior knowledge of characters can be useful for plates from some countries because their plates are standardized and therefore not versatile, such as Chinese license plates53. The layout is fixed except for some special plates such as military vehicle plates. An approach is proposed in a study which provides a solution in detecting vehicle license plates that are severely degraded. The plate is first located by the using color collocation, then dimensions of each character are used for segmentation. The layout of the Chinese license plates provides information for the classifier to recognize characters afterwards. The advantage of such methods is its simplicity. Nevertheless, the extracted license plate must not be of any shift of the ground truth license plate location, otherwise extraction results may be in background instead of characters[8].

d. Using Character Contours: Additionally, contour information is utilised in the segmentation of licence plates. In a study that was published in, the researcher developed a shape-driven active contour model that makes use of a variational fast marching algorithm[44]. In the first stage, the approximate location of the licence plates is determined by using a standard fast marching technique in conjunction with a speed function that depends on both the gradient and the curvature of the terrain. In the second stage, a specialised fast marching method is utilised in order to derive the precise boundaries of the licence plates.

e. Using Combined Features: Two or more characteristics of the characters may be utilised in order to segment the licence plate in a manner that is more effective. Dynamic programming was the method that was utilised in an analysis to segment the primary numerical characters that are found on a licence plate (DP). It is very rapid because it uses the bottom-up approach of the DP algorithm, and it is robust because it minimises the use of environmental-dependent features like colour and edges55. Both of these attributes contribute to its speed.

2.1.3.1 Number Plate Segmentation Methods

The success of the number plate extraction stage is absolutely necessary for the character segmentation stage to proceed. The target image or scene must first be completely analysed. There may be problems with the contrast of the number plate that is isolated, as well as conditions of illumination that vary, and the plate may also be oriented at different angles. Before segmenting the characters, it may be necessary to first apply pre-processing techniques, such as de-skewing, de-blurring, or any other methods, depending on the circumstances surrounding the number plate[8]. This step can take place either during the extraction stage or after getting an isolated candidate area; the specifics of how the step is carried out are determined by the approach that is taken. A preprocessing method such as bilinear transformation is utilised in order to deal with images of tilted licence plates. The isolated licence plate number is mapped onto a shape that is straight and rectangular. A least square method is used to correct the angle of tilted number plates. To calculate the angle of vertical tilte, it is suggested to use the K-means cluster based line fitting, the least squares based line fittings, and the K-L transform[56]. Each of these methods has their own advantages and disadvantages. Even though the threshold application seems straightforward while converting to a binary image, this stage of the process is actually one of the most difficult parts. Following is an analysis of the various techniques for segmenting number plates, organised according to the features that are taken into account.

a. **NP Segmentation Using Connected Components**: Segmentation in research can be accomplished through the use of pixel connectivity. The connected component labelling technique is applied to 958 high definition (HD) images of varying conditions, and the accuracy of the segmentation is measured at 99.75%. The character pixels in the binary image are given labels according to their connectivity, and the aspect ratio and size of the characters on the licence plate are compared with those of the character pixels. However, it does not appear that this method will be successful for either joined or broken characters[8]. A small dataset consisting of fifty images was used to test the connected component labelling and morphological method, and the results showed a segmentation rate of 91 percent. By utilising high-definition images captured under a variety of lighting and atmospheric conditions, an accuracy rate of 99.5% was measured using connected components analysis. In a different study, the extracted plate area is binarized and labelled so that the researchers can get the numbers. Several different label layout patterns were utilised in order to successfully identify the segment that was labelled as a number. The system had a total recognition rate of 99% across the board. In order to segment the data, several different approaches, including a combination of connected components and blob colouring, were considered. Their system had an accuracy of 93.7% on average across the board[8,46].

b. **NP Segmentation Using Vertical/Horizontal Projection:** The characters and the background of a number plate each have their own unique colour. A number plate is required to have a background. Because of this, the binary image of the licence plate that was produced contains values that are differentiated for both the character and the background of the plate. Character segmentation was accomplished in a study by employing pixel projection in both the vertical and horizontal planes simultaneously. Projection techniques are used[46]. After applying vertical projection to the binary number plate in order to extract individual characters, the start and end points of each character are analysed next. This step follows the application of vertical projection. The horizontal projection method is then utilised to extract the individual characters. In order to extract characters from a character sequence, the vertical projection method is utilised, in addition to performing an analysis of the character sequence in order to eliminate noise. This technique can achieve an accuracy of up to 99.2% while keeping the processing speed within 10–20 milliseconds even when applied to more than 30,000 images[8,46]. A method of profile projection that was used in another study is evaluated with the help of a database that contains 560 individual photographs. The rate of segmentation that was accomplished was 95.4%, and it was accomplished by successfully recognising multiple number plates that were present in a single image. This allowed for the successful completion of the segmentation. It is possible, given the information that has been provided, to arrive at the conclusion that the method that utilises horizontal and vertical pixel projections is the one that can be put into practise with the least amount of difficulty. The projection techniques appear to produce promising results for the segmentation of characters. This is likely due to the fact that the results are not dependent on the positions of the characters. There is a possibility that the projection values will be affected by the image quality as well as the noise[8].

c. **NP Segmentation Using Characters Features:** The process of segmenting vehicle licence plates is made easier by having primary knowledge regarding characters. In a study, character isolation is achieved through the utilisation of the RGB colour extractor[8]. The rate of segmentation is 98.5% for 255 different colour images that were examined. For the purpose of segmentation, the YOLO models, YOLOv2, Fast-YOLO, and Classification-Regression Network (CR-NET), all of which are based on neural networks, are utilised[8]. According to the findings of a study, the extracted licence plate should be resized into a known template proportion[58]. It is known where all of the characters are supposed to go within this template. After the resizing process is complete, the original locations will continue to function as the characters. It is common knowledge that this method is straightforward. On the other hand, when the extracted number plate is shifted, this process produces background rather than characters[8]. A possible solution has been proposed8 for number plates that have suffered severe damage. The colour combination is used in order to identify the licence plate of the vehicle within the image. When determining how to segment a character, each of the character dimensions is taken into consideration. The layout of Chinese licence plates was used in the construction of recognition classifier[8]. According to the findings of one piece of research, Taiwanese licence plates all share a similar colour distribution, with a white background and black characters[59]. When the licence plate is scanned in a horizontal direction, the number of colour transitions from white to black or vice versa can be as high as 14 and as low as 6 at the absolute minimum. The Hough transform is utilised so that the rotation issue can be resolved. For character segmentation on dirty number plates, a hybrid binarization method is applied. In the final step, the feedback procedure is put into place so that the parameters can be managed. For the purposes of their experiments, approximately 332 distinct images are captured at a range of distances and under a variety of lighting conditions. Both segmentation and localization have a rate of 96.4% and 97.1% respectively across the board[8,59].

d. **NP Segmentation Using Boundary Information**: Modeling contours is another method that can be used to achieve character segmentation. In a study, the method of vertical edge detection along with the removal of the long edge is used[8]. A closed curved technique was used by some researchers, while others used vertical histogram character segmentation. Additionally, a segmentation process based on an adaptive morphology approach was proposed for the purpose of extracting severely degraded number plates[8]. In the work of a different author, the process of morphological thickening was utilised to locate the reference lines that were necessary to separate the overlapping characters from one another[60]. The morphological thinning algorithm can determine the baseline for connected character segmentation. An image set containing 1189 degraded images had a segmentation rate of 84.5%, which resulted in the correct segmentation of approximately 1005 images[60].

2.1.3.2 Type of LPR Systems

LPR system is divided into fixed and mobile types. Here, recognizes differences between the systems and their strengths and weaknesses are described61.

Fixed Systems: LPR systems that are utilised in a fixed location are intended to be immobile and should not be relocated from the location in which they are installed[61]. The speed of this method and the ease with which it can be designed are two of its many advantages; however, one of its disadvantages is the high cost of using sophisticated equipment.

Mobile Systems: Mobile systems are designed in such a way that law enforcement officers can view the results of licence plate readings in real time. Cameras can be attached to this kind of system and then mounted in a variety of different positions[61]. It is abundantly clear that this type is economical, but due to the high volume of procedures and algorithms, it is significantly slower than the approaches that came before it.

2.1.4 LPR Application

Toll gates, parking lots, the entrances to secured buildings, and many other locations are some of the many places where LPR systems have been installed. These systems are helpful due to the fact that they can automate the management of parking lots, increase the safety of parking lot operators, do away with the need for parking tickets, restore the flow of traffic during rush hours, and identify vehicles that are travelling too fast on highways. The following eight scenarios are also controlled by automatic licence plate recognition systems: country borders, traffic monitoring, law enforcement, traffic management, extensive parking, highways tax, identification of stolen vehicles, and vehicle tracking. The following will contain explanations of some of the more applicable practical applications.

Parking LPR System: When vehicles enter the parking lot, the licence plates are read, and the system also computes the amount of time spent parked using the information. This procedure not only reduces costs and saves time, but it also improves the accuracy and efficiency of the acceleration process, which in turn attracts customers as well[61]. A vehicle makes its way into a parking spot in Figure 2.1. The licence plate of the vehicle is read, and if the owner of the vehicle has been charged parking fees, the gate will open on its own shortly after the payment has been processed.

Figure 2.1: LPR in Parking[61].

Access Control: The number plate number is read by the LPR system, and the events that take place are recorded on a database. This allows the entrance gate to be opened for authorised members in the secured area. As shown in Figure 2.2, the entrance gate will automatically raise for authorised vehicles, providing assistance to the security guard.

Figure 2.2: LPR in Access Control [61].

Tolling Control: As shown in Figure 2.3, the LPR system is utilised in the process of calculating the travel fee that must be paid at the toll road gates. When a vehicle enters the toll lane and presents a pass card, the system will read the licence plate of the vehicle. The details of the passes are retrieved from the database, and then those details are compared with the information about the vehicles. In the event that there is fraud, the operators will be informed[61].

Figure 2.3: LPR in Tolling Control[61].

Border Control: The information on people's licence plates will be entered into the central databases of the countries' borders, and those databases, along with the borders themselves, will be monitored. It is able to put restrictions on the movement of people across national borders, and this particular installation will also protect the borders of the entire state. This will be utilised to track almost all crossings of national borders [61].

Stolen Cars: The list of vehicles that have been reported stolen will be used to notify on any passing "black" vehicles. This "black list" is immediately updated, and it provides a rapid alarm system to the police force[61]. This LPR system has been implemented for the roadside, and it is capable of performing a new realtime match between the passing vehicle and the checklist. By using these calculations, burglars will be located, and then police personnel will be notified about detected vehicles in order to put a stop to the stolen vehicle.

Traffic Control: Highways frequently suffer from the issue of backed-up traffic. Traffic management is an essential component of road control, and as a result, various kinds of web digital cameras, timers, and sensors are installed along roadways in order to record information regarding traffic congestion[61].

2.1.4.1 Elements of LPR System

The procedure begins when a sensor recognises the presence of a vehicle and sends a signal to the system cameras instructing them to take a picture of the vehicle as it travels by[62]. The image is then sent to a computer that is connected to a wireless network. The software that is running on the computer will then extract the number plate number from the image. When a licence plate is recognised, it is cross-checked against a record in a database, and the tax on the vehicle is determined. Figure 2.4 illustrates that a licence plate recognition system needs to have four primary components: a light source to illuminate the licence plate, a video camera to capture images of 11 passing vehicles, a computer with image processing software, and a wireless network[61]. These components are required for the system to function properly.

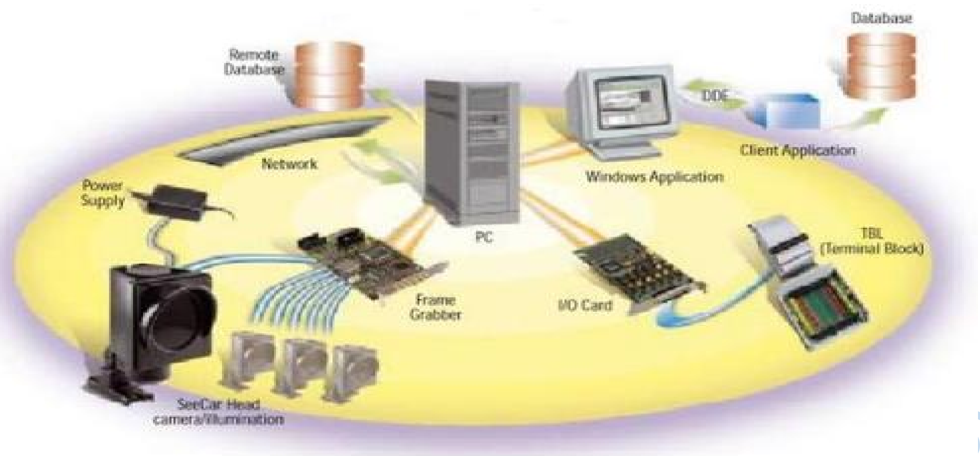


Figure 2.4: LPR System Unit[61].

2.1.4.2 Difficulties and Obstacles of Development

According to this point of view, the rate at which licence plate recognition is performed accurately and quickly is influenced by a large number of interconnected factors, including political and cultural conditions, rules and regulations, climatic and geographical conditions, scientific and technological conditions, and many more. Some of the parameters that were mentioned earlier will be explained here.

The Influence of Rules and Regulations: Because the laws and regulations in different countries can be quite different from one another, the locations of licence plates can vary. Because of this, the system needs to collect knowledge and information about the character distribution, as well as font styles, size, colour, and the amount of space between characters, among other things. Common locations and characters will be able to solve a wide variety of research issues, but researchers in this field are still confronted with numerous types of plates with a variety of designs, colours, and combinations of characters.

The Effect of Climatic and Geography Conditions: Image quality is directly related to the speed and accuracy of the results of recognition, as is obvious. There is no question that atmospheric and environmental factors, such as the angle of sunlight, fog, humidity, rain, and dust, are some of the things that can affect the quality of the recorded images. Ambient light during the day and night is another factor that can have an effect. It is anticipated that cameras with high performance with regard to geographical conditions will be able to solve this issue.

2.1.5 Automatic Number Plate Recognition

The Automatic Number Plate Recognition system is now ingrained in our daily lives and appears to be here to stay, with the potential to be incorporated into future transportation technologies. The advent of the idea of autonomous vehicles has opened the door to a plethora of opportunities for fundamental change in transportation systems. The use of ANPR technology is already making a contribution to the development of intelligent transportation systems and is doing away with the necessity of human intervention. It is not just the camera on the side of the road or the one at the entrance to the parking lot anymore. Over the course of the years, it has evolved into a mobile system, initially being installed in moving vehicles. More recently, with the development of technology for smart phones, many ANPR systems have evolved into hand held devices as well. In the toll and parking lot industries, ANPR is frequently a choice because it has lower costs associated with its provisioning. In contrast to the Ultra High Frequency Radio Frequency Identification (UHF-RFID) systems8, the Automatic Number Plate Recognition (ANPR) system is able to recognise the registered number plate without the need for an additional transponder. This is the primary reason. A significant step forward in the development of our modern world is the accelerated urbanisation of rural areas. The majority of people choose to live in urban areas rather than rural ones as they migrate away from rural areas. As a result of the increase in traffic in these areas, local governments frequently fail to recognise the mobility needs of residents and visitors, both currently and in the future. ANPR is being utilised more and more frequently in order to investigate the unimpeded flow of traffic, which enables intelligent transportation8.

Not only can today's ANPR cameras read licence plates, but they can also provide helpful additional information such as counting, direction, groups of vehicles, and the speed at which they are traveling[8,63]. The ability of automatic number plate recognition (ANPR) technology to detect and read large volumes of rapidly moving vehicles has led to its implementation in a variety of different areas of today's digital landscape. Even though ANPR technology can be packaged in a wide variety of ways, its fundamental purpose remains the same: to provide an automated, highly accurate method of reading a vehicle's licence plate without the need for any assistance from a person. Access control, parking management, tolling, user billing, delivery tracking, traffic management, policing and security services, customer services and directions, the red light and lane enforcement, queue length estimation, and many other services are some of the many applications that make use of it[8,63]. The fundamental system diagram of both a fixed and mobile ANPR technology is depicted here in Figure 2.5.

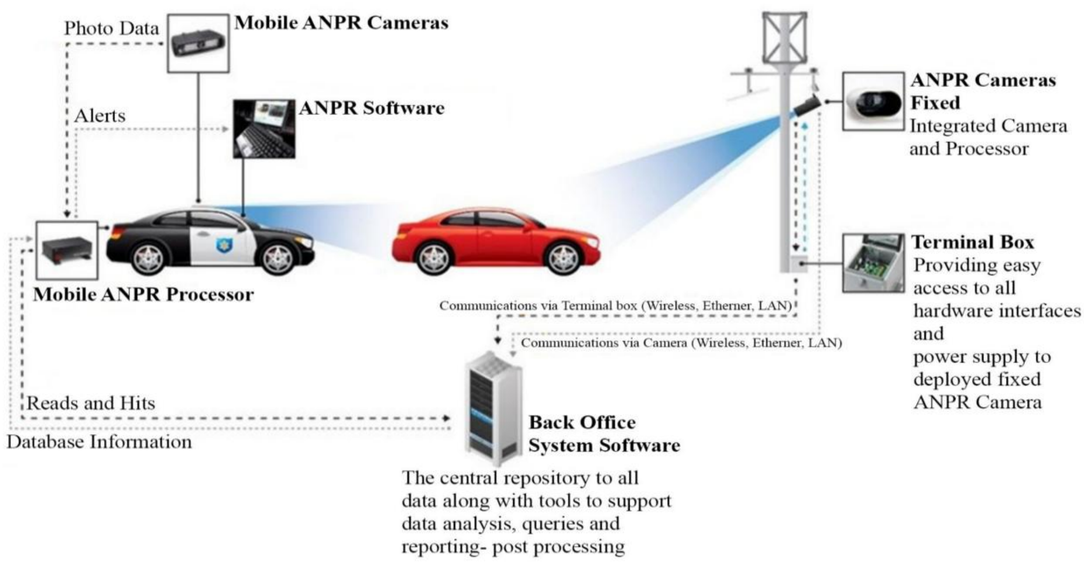


Figure 2.5: System Diagram of a Fixed and Mobile ANPR Technology[8].

The "automatic number plate recognition," or "ANPR," refers to a system that is designed to automatically recognise and store number plate data on vehicles that pass through a particular point. The modern ANPR system makes use of the most recent and cutting-edge technologies, such as the utilisation of high speed cameras that are able to detect and capture the number plate of a vehicle while the vehicle is moving at a high rate of speed and the development of ANPR software that is able to process the number plate that was captured in a very short amount of time. The majority of the time, the number plates of vehicles are used to identify the vehicles.

These licence plates are simple enough for humans to read, but they are unintelligible to machines. Number plate is just an image for the machine, which can be defined as a two-dimensional function called f (x, y). Here, x and y are the spatial coordinates of a picture element, also known as a pixel, and f is the light intensity at that point[8]. Because of this problem, it is essential to develop ANPR software that is able to transform the data between the information system and the real world environment. According to the earlier research, an ANPR system typically has a recognition rate of between 50 and 90 percent of all vehicles at each camera location[65]. Because the likelihood of a successful reading of a number plate is primarily dependent on the characteristics of the vehicle, including the 10 system utilised, the quality of installation, and the weather condition, the vehicle whose number plate was successfully read at an upstream point will most likely be successfully detected at a downstream point. This is because the probability of a successful reading of a number plate is primarily dependent on the vehicle.

However, the recognition rate is likely to be lower in an urban environment due to the fact that the distance between vehicles is closer together and larger vehicles may obscure the number plates of smaller vehicles. To verify the ANPR system's recognition rate in an urban environment, additional research is required. The rate of recognition will be variable depending on a number of factors including the speed of the vehicles that are being recorded, the varying conditions of the ambient lighting, the conditions of the weather, and a number of other factors. The Automatic Number Plate Reader (ANPR) system is made up of several essential parts, the most important of which are the hardware and the software components. Within the scope of this study, discussions of both software and hardware components are included. On the other hand, more attention is being paid to the development of software for ANPR rather than the components of the hardware.

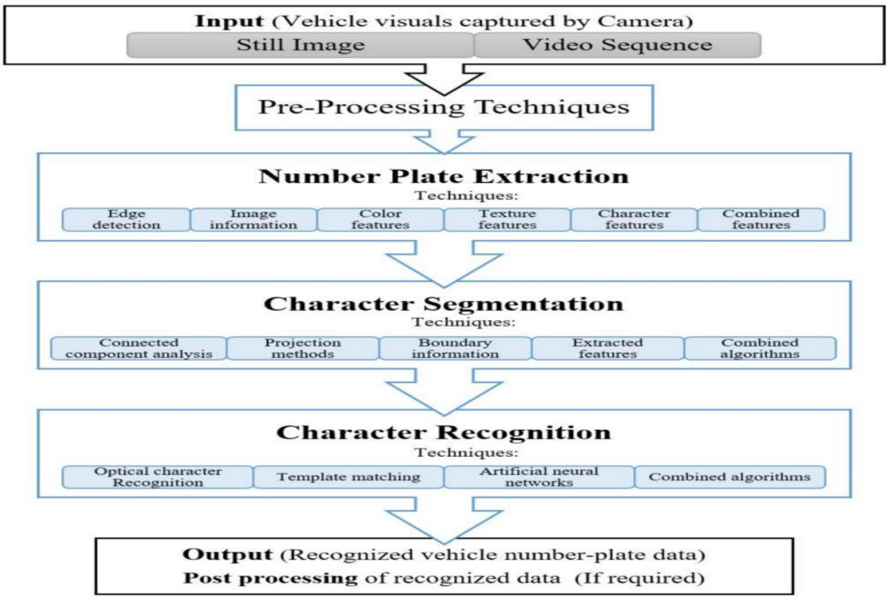


Figure 2.6: A Typical ANPR System[8]

Image acquisition (as input to the system), number plate extraction (as output from the system), character segmentation, and character recognition (as output from the system) are the general processes that are carried out by a typical ANPR system[8]. After the vehicle has been successfully recognised, the data can be accessed and utilised for post-processing operations as necessary. The information collected from the vehicles is transmitted to the connected back office system software. This software serves as the central repository for all data and comes equipped with tools to support data analysis, queries, and reporting as appropriate. Since ANPR systems not only visually capture the vehicle images but also record the metadata in their central repository, the data that is collected can be used for a variety of other intelligent transportation applications[8]. This may include the recognition of vehicles by stamping the current date and time on their data as well as their precise location, all while storing a comprehensive database of the movement of traffic. These data may prove useful in the modelling of various transportation systems and the analysis of those systems.

Depending on the type of camera that was used, its resolution, lightening/illumination aids, the mounting position, area/lanes coverage capability, complex scenes, shutter speed, and other environmental and system constraints, the image that was taken from the scene may experience some complexities[8].

The system uses plate localization functions to extract the number plate from the image of the vehicle when it detects a vehicle in the scene or image. This is a process that is commonly referred to as Number Plate Extraction, and it is performed by the system. After the number plate has been extracted, the characters on the plate will be segmented before the recognition process begins[8].

Character segmentation is a type of algorithm that determines where the alphabetic and numeric characters are located on a licence plate. Afterwards, the optical character recognition (OCR) techniques are utilised in order to convert the segmented characters into an alphabetic and numeric text entry[66]. Character recognition is accomplished with the assistance of algorithms such as template matching and neural network classifiers. 67,68. The efficiency of a given ANPR system's individual stages directly impacts the overall performance of the system. The performance-rate, also known as the success-rate, is a parameter that is used to quantify the entire process.

This rate is the ratio of the number of licence plates that have been successfully recognised to the total number of input images that have been taken. The performance rate takes into account all three stages of the recognition process namely, the extraction of the number plate, the segmentation, and the character recognition..

2.1.6 License Plate Deflection

License plate tilt consists of vertical tilt, horizontal tilt and both. These tilts will undoubtedly result in character distortion and adversely affect character recognition[69]. Therefore, if the pose and part deformation of the object can be disentangled from the texture and shape, it will facilitate the subsequent prediction, for example, local max-pooling layers in CNNs. The process of license plate correction can be regarded as the process of affine transformation, which needs to find out a mapping from tilted image to corrected image[69]. The distorted image is transformed into the corrected image through affine matrix.

Once the input images was obtained, which could learn invariance to translation, scale, rotation and more generic warping. Unlike pooling layers, where the receptive fields are fixed and local, the spatial transformer module is a dynamic mechanism that can actively spatially transform an image (or a feature map) by producing an appropriate transformation for each input sample[69]. The spatial transformer module combines the localization network and sampling mechanism. This kind of spatial transformer can be incorporated into other convolutional neural networks, which effectively improved the representation of deep network and improve the recognition accuracy of convolutional neural network[69]. Therefore this transformer is a method that was trained with tilted images and normal images to automatically find a mapping between two kinds of images, and usually incorporated by many license plate recognition algorithms. Multiple spatial transformers can also be used simultaneously to identify multiple objects in a single image. This space transformers can not only affine the whole license plate, but also affine several characters in the license plate[69].

2.1.6.1 Image with Noise

The introduction of noise into an image can occur as a result of a variety of processes, including image acquisition, transmission, and compression. Additionally, there are a variety of types of noise, which can be categorised as pepper and salt noise, gaussian noise, and so on[70]. In the majority of cases, in order to facilitate licence plate recognition in the real scene, the licence plate.

The image will be distorted as a result of noises such as rain lines, snow lines, and other noises, and some licence plates may be obscured. There is currently no effective method for image denoising. In practise, it is more about finding a balance between the effect and the amount of computational complexity that is required. Some denoising algorithms, such as the bilinear filtering and the median filtering, directly use the values of adjacent pixels to calculate the average value, whereas others regard the noisy image as the superposition of noise and clear image, decompose the image into the detail layer and the base layer, and then further separate the noise streaks from the detail layer by using the network[70].

2.1.6.2 Fuzzy License

Because of noises like rain lines, snow lines, and other noises, the image will be distorted, and it is possible that some licence plates will be obscured. At the moment, there isn't a noise removal technique that works particularly well. In actual application, it comes down to striking a balance between the desired effect and the level of computational complexity that must be met in order to achieve it. Some denoising algorithms, such as the bilinear filtering and the median filtering, directly use the values of adjacent pixels to calculate the average value, whereas others regard the noisy image as the superposition of noise and clear image, decompose the image into the detail layer and the base layer, and then further separate the noise streaks from the detail layer by using the network[70]. For example, the bilinear filtering and the median filtering both use the values of adjacent pixels to calculate the average value.

As a result, it is necessary to find a method to effectively improve the resolution of small licence plate targets. Perceptual GAN was used in a study to internally lift representations of small objects to ones that were "super-resolved," achieving similar characteristics as large objects and making detection more discriminative as a result[69]. In contrast to the GAN in its most basic form, the discriminator network is organised into two distinct branches: the adversarial branch and the perception branch[69]. The adversarial branch is utilised in order to differentiate between the super resolved representation that was generated for the large object and the representation that was used initially. And the perceptual branch is what's used to justify the detection accuracy that comes from the representation that's been generated[69].

The generator network was developed to be a deep residual learning network with the purpose of enhancing the representations of small objects into ones that are super-resolved. The weighted sum of two different parts makes up the loss function of the discriminator network. The log loss function was implemented in the adversarial branch's initial stage of development[69]. The smooth L1 loss function is utilised by the perceptual branch in the second part of the analysis. In order to obtain the parameters of the generator network, optimization of the discriminator network loss function was performed[69]. Researchers looked into the use of a cascade generation model as a possible solution to the issue of model absence in training. This was done due to the challenging nature of training GANs. They intend to accomplish this by making the counter generation network more reliable and simpler to tune[69]. A dual directed capsule network called DirectCapsNet was also proposed in this study to recognise very low resolution images. This network, which was given the name DirectCapsNet, was able to achieve a recognition accuracy of over 95% when 16*16 images were matched with 80*80 images[69].

2.1.8 Nigerian Licence Plate

One report estimates that there are more than 11 million cars and trucks on the roads of Nigeria[71]. As a result of the rise in the number of vehicles on the roads of Nigeria, there is a pressing requirement to create an efficient system for the monitoring of traffic[72]. Utilizing the number plates on vehicles is one method that can be used to keep track of them. In order to aid in the management of traffic and to more easily keep track of vehicles, each vehicle is required to have a licence number that is also required to be assigned to the vehicle's number plate. In order for a vehicle to participate in public traffic, it is required to have a licence number, which is also referred to as a vehicle identification number, or "VIN." This number also acts as the primary identifier for the vehicle once it has been registered[72].

Table 2.1 – Categories of Vehicle in Nigeria with their Attributes[72].

Year	Vehicle Type	Background Colour	Foreground Colour
1976	Private	Black	White
1992	Commercial	Green	White
1992 till Date	Private	White	Blue
	Commercial	White	Red
	Government	White	Green

Nigeria uses the North American standard of 14cm by 31cm for number plate as depicted in Figure 2.8

Figure 2.8: The Nigerian Plate Number[72].

The Nigerian licence plate displays the country's flag in the upper left-hand corner of the plate. The name of the state as well as its slogan are shown in the top centre of the plate, and the Federal Republic of Nigeria are written at the bottom. The outline of the map of Nigeria can be seen in the background. Characters for the plate number are made up of stamped alphabetic and numerical character ridges, with a maximum of nine ridges per character. The first two ridges are reserved for the prefix alphabets, then there will be a maximum of three ridges for a maximum of three numerals, then there will be a hyphen ridge, and finally there will be a three ridge character abbreviation of the Local Government Area (LGA) where the vehicle was registered.

However, according to the findings of researchers, very little research had been done on the development of an ANPR system that will function optimally for vehicles in Nigeria. This was the conclusion reached by the researchers. While some worked on the number plate systems of other nations, others were responsible for the previous Nigerian numbering system. This study has the potential to close this knowledge gap.

2.2 Theoretical Framework

2.2.1 Licence Plate Algorithm

In the licence plate location stage, there's a need to extract from the input car image a portion of the image that contains the potential licence plate. Without a licence plate location, direct recognition would have a difficult time differentiating licence plate numbers from other text blocks such as traffic signs and phone numbers on storefronts. In addition, licence plates might only occupy a small portion of the image, so they might be ignored[8]. The subsequent performance of licence plate recognition will be impacted by the accuracy of the location of the plate. In this stage, the image that contains the car serves as the input, and the stage's output consists of four values that represent the location of the licence plate. It is common practise to represent using the coordinates of the upper left corner as well as the height and width, which are written as (x, y, w, h)[8].

The traditional algorithm is based on the manual extraction of features to achieve the location, such as the colour features of the licence plate, the texture, and other features that can be single or combined. However, there is a lack of background information, which means that the information regarding the bigger picture cannot be utilised to its full potential. Deep learning can extract features from input images based on the pixel information in those images, and deeper networks are able to extract more detailed features than shallower networks. In addition, features that are difficult to extract manually can be obtained through the use of various feature extraction models[8].

Some examples of these types of features include multi-scale information and fine-grained information. The feature extraction layer and the parameter regression layer are the typical components that make up the entirety of the network structure. The regression layer is typically the full connection layer, and the number of neurons in this layer is modified according to the number of prediction parameters, which in most cases is 4[8].

Traditional licence plate location algorithms can be broken down into five categories, based on the intuitive features they use: text-based detection, color-based detection, character-based detection, and connected-component detection8. Each of these categories has its own set of subcategories. These intuitive features are easily influenced by their surroundings, whereas deep learning can extract deeper features based on the information contained within pixel data. It is possible to estimate where the licence plate is located by combining the output values of all of the image's subregions. This is done in such a way that the centre of the licence plate is situated closer to the left or right subregions that have the highest score. Its detection accuracy can reach up to 0.87, and its recall rate is 0.83. The amount of time it takes to process data is 0.23 seconds[8]. In a study that proposed a unified deep neural network for end-to-end training to simultaneously locate and identify licence plates, the researchers came up with an interesting idea[73]. It is made up of a few convolutional layers, a region proposal network for the generation of licence plate proposals, a proposal integrating and pooling layer, multi-layer perceptions for plate detection and bounding box regression, and RNNs with CTC for the recognition of licence plates[8,73].

The feature extraction network is altered so that it more closely resembles the vgg-16 network. The convolutional layer is maintained, but the number of pooling layers and the full connection layer are both removed. A set of potential bounding boxes was output by utilising the modified RPN, and 6 scales with an aspect ratio of 5 were designed to generate 6 anchors at each position of the input feature maps[73]. The inception-RPNzl algorithm served as inspiration for the application of two 256-dimensional convolutional filters simultaneously across each sliding position. The extracted features were concatenated along the channel axis to obtain 512-dimensional feature vectors. These feature vectors were then fed into two separate fully convolutional layers, one for plate/non-plate classification and the other for box regression[8]. Both of these fully convolutional layers were fed the 512-dimensional feature vectors. The detection model only needs 300 milliseconds to process each image, despite being able to achieve an accuracy of 98.15 percent. In a separate piece of research, the authors presented an effective and lightweight full convolutional network for licence plate detection from complex scenes. This network was able to downscale input images to significantly accelerate the process and reduce the amount of computational effort required[74]. In order to achieve an even higher level of accuracy in the predictions, dense connections and dilated convolutions have been implemented for the purpose of combining multi-level and multi-scale vision features, and a fusion loss structure has been included in the training process.

In order to further improve prediction accuracy, an additional fusion loss structure is appended during training. The network is divided into two separate but parallel branches. The image is sampled in the foreground branch down to 1/8 of its original size, and the background branch is built with dense blocks, each of which contains a series of connected convolutional layers[75]. The main part of the background branch is built with dense blocks. A 3x3 convolution with a stride of 2 is used for pooling in each block to subsample the feature maps in order to reduce the amount of time spent on calculation while maintaining the accuracy of the results. On the datasets provided by Caltech, the detection accuracy could reach 93.47 percent, and the processing time required for each image was 28.33 milliseconds and 75 microseconds[75].

Because of the complicated environment, it is not always easy to directly locate the licence plate. This is especially true in situations in which the licence plate target is either too small or partially shielded[8]. Even if it is not immediately apparent where the licence plate is located, a person with sufficient experience will be able to estimate its general location. This is due to the fact that the licence plate is physically attached to the body of the vehicle. As a result, some researchers take advantage of the prior knowledge between the licence plate and the car body, such as the position relationship between the rear lights and the licence plate, to transform the issue into a target that is simpler to detect. For example, the position relationship between the rear lights and the licence plate.

A scholar read the licence plate by employing a cascaded framework, which, in order to do so, first identified the character region and then extracted the licence plate frame. In the first step of the process, a four-layer, 37-class CNN classifier is used in a sliding window fashion across the entire image to detect the presence of the text and generate a text saliency map. Next, the candidate bounding boxes are generated independently at each scale by using the run length smoothing algorithm (RLSA) and connected component analysis (CCA)[8]. After that, the generated boxes are filtered using geometric constraints, and the edge feature of the licence plate is used to further refine them. Last but not least, a second plate/non-plate CNN classifier was applied in order to validate the remaining bounding box.

This detection model had the capability of reaching a precision of approximately 97% and a recall of greater than 95%. An analogous piece of work extracted licence plate number 76 by employing a cascade structure that was made up of a fast region proposal network and an R-CNN network. In the first step, a lightweight RPN network was given the downsampled image to use as input in order to generate potential licence plate candidates. After that, the sampler takes the original high-resolution image and extracts the regions of interest (ROIs) from it[8,76]. Additionally, the patches that were extracted are input into an R-CNN network in order to classify the candidate plate and regress the four corners of the licence plate. This licence plate detector is more accurate than 96% of its faster R-CNN counterpart while also being

1.5 times faster and 57 times more compact. Another study merged the classes of cars and buses on the PASCAL-VOC dataset while ignoring other classes8. This study also used the YOLOv2 network to detect vehicles, but it did not make any changes to the network or attempt to improve it in any way before using it.

WPOD-NET was proposed to detect licence plates in a variety of different distortions. It regresses coefficients of an affine transformation that unwarps the distorted licence plate into a rectangular shape resembling a frontal view[8]. These insights came from YOLO, SSD, and STN. In order to realise multidirectional car licence plate detection, a CNN-based method was proposed and given the name MDYOLO[8,77]. This method was inspired by the YOLO framework. Each input image was divided into regular 7x7 grid cells, much like YOLO does, and the cell in which the centre of the car licence plate is located was used to detect the licence plate, as well as to predict two bounding boxes and a confidence score for each cell. This was accomplished by using the cell in which the licence plate centre was located. The difference between MD-YOLO and YOLO is that MD-YOLO introduces angle information and guides the model to regress in order to determine the angle of rotation of a given car licence plate image. YOLO does not do either of these things[77].

In addition, the angle deviation penalty factor, abbreviated as ADPF, was suggested as a means of approximating the intersection ratio between the predicted value and the tag value8. In addition, rather than using the ReLU function as the activation function, the leaky function and the identity function were selected. This was done so that negative rotation angle values could be found. In light of the fact that the licence plate is typically quite diminutive, a prepositive CNN attention model known as ALMDYOLO was used in the time leading up to the implementation of MDYOLO[8].

On a GPU GTX980, this detection model was able to achieve an accuracy of more than 99% despite having a processing time of only 5 milliseconds. The system that was conceived was implemented by a work that evaluated and optimised different Yolo models with various modifications, with the goal of achieving the best speed/accuracy tradeoff at each stage[78]. Because the licence plate might only occupy a very small portion of the image, and other textual blocks like traffic signs might be confused with licence plates, the vehicles were first detected, and then their respective licence plates were detected in vehicle patches. This was done for the licence plate detection stage. On a total of 8 different datasets, this system was able to achieve an average precision of 98.37% and an average recall of 99.92%[78].

2.3 Review of Related Studies

According to the findings of a study that used region-based convolutional neural networks to detect vehicle licence plates[79]. In this study, a novel strategy for solving this problem is proposed, and it involves treating the LP vehicle as an object. The following activities associated with the difficulty of LP detection are the primary focus of this study, which aims to address them: (1) the detection of LPs in each individual frame of a video sequence, (2) the detection of partial LPs, and (3) the detection of LPs using moving cameras and vehicles. In this work, the most cutting-edge techniques for object detection, such as convolutional neural networks with region proposal (RCNN), its successors (Faster-RCNN and Faster-RCNN), and the exemplar-SVM, are used to provide solutions to the aforementioned problem. These techniques include: The suggested research produces superior outcomes in comprehensive tests and comparisons when compared to other traditional methods.

Utilising the YOLO detector as the foundation for a dependable real-time automatic licence plate recognition system[80]. This article presents a dependable and productive

ALPR system that is based on the most recent iteration of the YOLO object detector. For each stage of the ALPR process, the Convolutional Neural Networks (CNNs) are trained and fine-tuned in order to ensure that they are resilient in a variety of settings (e.g., variations in camera, lighting, and background). The authors design a two-stage approach specifically for character segmentation and recognition, employing simple data augmentation tricks such as flipped characters and inverted License Plates (LPs). This allows us to segment and recognise characters more accurately. The ALPR approach that was developed as a result achieved remarkable success in both datasets. First, in the SSIG dataset, which is made up of 2,000 frames from 101 vehicle videos, our system was able to achieve a recognition rate of 93.53% and 47 Frames Per Second (FPS). This performance was superior to that of the commercial systems Sighthound and OpenALPR (which achieved 89.80% and 93.03%, respectively), and it significantly surpassed the results obtained in the past (81.8%). Second, in order to simulate a situation that is more representative of the real world, we present a larger public dataset 1 dataset that is intended for ALPR. This dataset consists of 150 videos and 4,500 still images taken while both the camera and the vehicles being studied were moving. Additionally, this dataset includes a variety of vehicle types (cars, motorcycles, buses and trucks). The evaluation versions of commercial recognition systems achieved recognition rates lower than 70% in the dataset that we proposed. On the other hand, the performance of our system was superior, with a recognition rate of 78.33% and 35 frames per second (FPS)[80].

In a related study, researchers used a single neural network to detect and recognise different types of licence plates[81]. In order to detect and recognise mixed-style long plays, the author of this article suggests using a singular neural network known as ALPRNet. In ALPRNet, two fully convolutional one-stage object detectors are used to detect and classify LPs and characters simultaneously. These detectors are followed by an assembly module, which is responsible for outputting the LP strings. ALPRNet treats LPs and characters equally, and its object detectors directly output bounding boxes of LPs and characters with corresponding labels. Because of this, ALPRNet is able to avoid the recurrent neural network (RNN) branches of optical character recognition (OCR) that are present in the existing recognition approaches. We evaluate ALPRNet using a dataset with mixed LP style examples as well as two datasets with single LP style examples. The experimental results demonstrate that the proposed network achieves state-of-the-art results with a simple one-stage network.

Increasing CNN performance through the use of GAN-based synthetic data augmentation in Vehicle Number Plate Recognition[82]. The current investigation explores the idea of an automatic parking system that recognises a vehicle's licence plate or number plate and parks the vehicle for the driver automatically. By removing the need for human interaction, it will make the process more efficient while also reducing the amount of hassle involved. It will also result in an improvement in the safety of vehicles, as it will eliminate the need for a slip or a magnetic card to be used when entering and exiting a parking spot, which is currently the standard method for registering vehicles. The researcher employs image processing algorithms in order to automatically enter information about the parking spot into the database. Automatic Vehicle Number Plate Recognition, or AVNPR for short, is the technology that is utilised for the purpose of identifying the plates' respective numbers. Deep learning algorithms such as CNN (convolutional neural networks) and RNN (recurrent neural networks) are unable to correctly recognise the misidentification of the numbers on the vehicle plate as a result of noise issues. The GAN (Generative adversarial networks) algorithm was utilised by the authors in order to solve this issue and bring about the desired results. GAN makes it possible to create images with a high resolution from a single image with a low resolution. Following the implementation of the GAN, the classification of the licence plate will be carried out by the CNN. The proposed method achieves a recognition accuracy of 99.39% for a vehicle number plate when tested in practise. Therefore, the system that has been proposed is suitable for automatically identifying the numbers that are on the vehicle's number plate. In addition, when the proposed system was compared to other models that already existed, it was discovered that it had achieved a higher level of accuracy than the other models.

In addition, applying a novel method for the recognition of characters on licence plates featuring a variety of fonts and designs[83]. An innovative licence plate recognition network was designed by the author of the plan to precisely locate and classify characters and LP regions simultaneously. The plan also includes the addition of an assembly layer, which is responsible for combining the characters into licence plates and producing licence plate strings. According to the findings of the experiments, the proposed method is capable of achieving a recognition rate of 98.57% for multi-style LPs when applied to real-world applications. In addition, in order to test the proposed method for licence plate recognition, the authors select the standard licence plate datasets, which only contain single-style licence plates. The corresponding results show that the proposed method achieves competitive performance[83].

A comprehensive analysis of the applicable algorithms for automatic number plate recognition was conducted[84]. The work provides a comprehensive review of recent developments and techniques used in Automatic Number Plate Recognition (ANPR) systems, as well as a comprehensive performance comparison of numerous real-time tested and simulated algorithms, some of which involve computer vision (CV). The Automatic Number Plate Reader (ANPR) technology has the capability to detect and recognise vehicles based on the number plates that are attached to them. Even with the most advanced algorithms, a successful deployment of an ANPR system may require additional hardware in order to achieve the highest possible level of accuracy. The condition of the number plate, non-standardized formats, complex scenes, camera quality, camera mount position, tolerance to distortion, motion-blur, contrast problems, reflections, processing and memory limitations, environmental conditions, indoor/outdoor or day/night shots, software-tools or other hardware-based constraints may all be factors that hinder its performance. Researchers find ANPR to be an interesting field because of its inconsistency, challenging environments, and other complexities. The Internet of Things is starting to have an impact on the future of a wide variety of industries and is creating new opportunities for ITS. Integrating ANPR with RFID-systems, GPS, Android platforms, and other platforms and technologies that perform a similar function is an effective way to use ANPR. Deep learning strategies are increasingly being used in the computer vision field in an effort to improve detection rates.

To conduct research on the Saudi Arabian licence plates' automatic number plate recognition[85]. In the experiment that was researched for this paper, a total of fifty images were examined to look for Saudi licence plates. After the preprocessing stage, the canny edge method was used to detect the car edges, and various threshold techniques were utilised to cut down on the amount of noise. During the segmentation process, horizontal projection was used to split the plate into its component parts. After that, a masking technique was applied so that the region of interest in the image could be located and isolated from the rest of the picture. The processed images were given an OCR treatment so that the characters and numbers written in English and Arabic could be read independently. After reshaping the Arabic letters using the tool provided, the next step is to combine the English and Arabic text. At long last, a rendering of the results of text superimposed on images was carried out down the plate regions. The clever algorithm, in conjunction with the projection technique, in conjunction with the appropriate preprocessing for images, generates results with an accuracy of 92.4% for the Arabic language and 96% for the English language[85].

In a live-video automatic number plate recognition (ANPR) system on an android smartphone using convolutional neural network (CNN), data labelling was performed[86]. The purpose of this paper is to discuss a live-video ANPR system that was developed on an Android smartphone using CNN. The camera on the smartphone has a limited resolution and limited processing power. The system was developed based on the standards for Malaysian licence plates. In terms of system performance, the recognition works perfectly with a computational time of 0.635 seconds when it is performed in an ideal outdoor setting with adequate lighting and a direct or slightly skewed camera angle. In this scenario, the environment is outside and the lighting conditions are ideal. Nevertheless, this performance is hindered by inadequate lighting, an extremely crooked angle of licence plates, and rapid vehicle movement.

In a system for recognising licence plates, taking large camera shooting angles into consideration[87]. This study demonstrates an automatic licence plate recognition system that improves recognition accuracy even at wide camera angles. The technology behind the system allows for the recognition of images through the application of convolutional neural networks that are exceptionally accurate. Taking into account wide camera angles, the proposed system enhances the stages of normalisation and segmentation of an image of a licence plate. The accuracy of recognition is intended to be increased as a result of these improvements. The affine transformation of the image is carried out during the stage of normalisation, prior to the stage in which the histogram is equalised. Mask R-CNN is utilised both during the segmentation process and the recognition stage. Selective search has been chosen to serve as the primary segment-search algorithm. In order to speed up the process of training and classifying the network, the combined loss function is utilised. In order to solve the problem of interclass segmentation, an additional module has been added to the convolutional neural network. The generated feature tensor87 is what goes into this module as its input.

In a study titled "Improved Automatic License Plate Recognition System in Iraq for Surveillance System Using OCR"[88]. The findings of this study point to a method that can be used in Iraq for the identification and detection of automobile licence plates. Every single facet of the system's logic is predicated on morphological operations and an approach to OCR edge detection, with the end goal of building and developing effective image processing methods and strategies to position a licence plate. This was done with the intention of achieving the aforementioned goal. Detection of highway speeds, automated charging systems, security, manuscript papers, and lighting infractions are some of the possible applications that could profit from this technology. It is possible to read the licence plate of a vehicle and find out who the owner of the vehicle is by using a combination of auto plate recognition hardware and software. Automatic licence plate recognition (ALPR) has a great deal of unfavourable repercussions as a result of its wide range of effects, which include light and speed, amongst others. To put this into perspective, the accuracy of our findings is greater than that of currently available systems that include plate detection and character segmentation as components of their process for character recognition.

In a study on the creation of a virtual vehicle identification for the purpose of tracking vehicles involved in hit-and-run accidents[89]. The development of a virtual vehicle identification tracking system is the focus of this body of work. This system makes use of wireless communication interfaces to facilitate the transfer of data that is helpful for the investigation of road accidents and the monitoring of traffic. For the purpose of assisting the vehicle identity tracking system, this system makes use of vehicle access points and the Vehicular Ad Hoc Network, also known as VANET. Within the beacon frames, the Internet of Things development board performs a scan of all of the vehicle Wi-Fi access points. It is difficult to accurately determine the offender's vehicle identity due to the characteristics of different positions of signal strength and the distance of the station to the access point. As a result, the purpose of this paper is to propose a hybrid tracking method that tracks vehicle identity by combining pre-accident tracking methods with post-accident tracking methods. In addition, the research presented in this paper demonstrates that specific Wi-Fi access point identities, such as Service Set Identifier (SSID) and Media Access Control (MAC) addresses, can be utilised as virtual vehicle identities for the purposes of vehicle tracking and traffic surveillance systems. In general, the result demonstrates that this system is able to positively detect the identity of the suspect vehicle and track it. It is possible for the system to track a vehicle access point signal from a distance of up to 45 metres away and is functional at driving speeds of more than 50 kilometres per hour.

According to the findings of a study conducted in Vietnam on an innovative system for the recognition of licence plates in various scenarios involving traffic violations[90]. A brand new system that is able to identify traffic violations, as well as output the licence plate numbers of the vehicles that broke the law and evidence, was presented in this body of work. Image processing methods and deep neural networks were integrated into the development of this system. A dataset was collected, and within it are thousands of photographs taken by public cameras in the city of Hue, which depict traffic vehicles. On the basis of this dataset, three different deep learning networks were developed in order to perform distinct tasks within the overall pipeline process. A model that can detect four different types of vehicles, including cars, buses, trucks, and motorcycles with riders, was developed with the help of a pretrained model of the YOLOv56 neural network. This model was used to create the model. Vehicle tracking and the identification of traffic violations were two of the applications that made use of the so-called "Deep SORT" model. For the purpose of licence plate recognition, a model based on RetinaFace and using MobileNet as its backbone was trained. In order to rectify the detected licence plates, the appropriate procedures were carried out. In the end, a model with CRNN architecture that had been pre-trained was used to develop an OCR system that could recognise the numbers on licence plates. The results of the experimental evaluations demonstrate that all of the models that were developed are both very lightweight and have a high level of accuracy. The YOLOv56 manages to get a mAP of 87.8%, the RetinaFace manages to get a maximal mean square error of only 1.6, and the OCR model manages to get a very high accuracy of 99.72%. Our system is very computationally efficient, as evidenced by the findings of the implementation of the pipeline on embedded hardware, specifically the Jetson Xavier Development Kit, which can be found in the previous sentence. The entire computation takes only 41 milliseconds, and the amount of RAM that is used is significantly less than 3 gigabytes. The findings have significant implications for a variety of practical settings.

The objective of this project is to investigate the Hardware–software co-simulation of vehicle licence plate detection using the ZedBoard SoC platform[91]. This chapter's objective is to present a demonstration of the system-on-chip (SoC) implementation of VLPDS, which is a technique for recognising licence plates that is based on a histogram technique of edge processing. Following the completion of the design phase on the Xilinx Zynq-7000 ZedBoard System-on-a-Chip (SoC) using MATLAB Simulink and Xilinx System Generator (XSG), the system is put into operation. The most widespread application of XSG is image processing, which not only simplifies the process of structural design but also enables hardware-software co-simulation, which is possibly a feature that is exclusive to this technology. Furthermore, the accuracy of the algorithm is evaluated for various sets of input pictures, and significant performance gains are revealed, culminating in the best SoC-based hardware implementation of VLPDS[91].

Image processing for the purpose of automatic licence plate recognition using combined methods was the subject of a research study[92]. This research paper makes a proposal to automate the process of licence plate recognition by combining four algorithms from the three methods described in the previous sentence. These algorithms are Adaptive Thresholding, Otsu's Thresholding, Canny Edge Detection, and Morphological Gradient applied to Edge Detection. The goal of this research paper is to improve the accuracy of licence plate recognition. The objective was accomplished by obtaining the best possible binary image using those methods, and the statistical method that was utilised in is the median of the intensity of each pixel in all of the output images that were obtained using the four different methods. In addition, this research provides a comparative study on thresholding techniques in order to select the most effective method for binarizing an image, which is the initial and most important step in the Automatic License Plate Recognition Process.

In an analogous study on licence plate recognition algorithms based on deep learning in complex environments, the researchers looked at complex environments[93]. In this paper, the authors discuss how deep learning can be applied to the process of licence plate recognition. The primary contributions of this work are as follows: 1) Introduce the most advanced algorithms from the three main technical difficulties, which are licence plate skew, image noise, and licence plate blur; 2) According to the process, the deep learning algorithms are classified into direct detection algorithms and indirect detection algorithms, and the advantages and disadvantages of the current licence plate detection algorithms and character recognition algorithms are analysed; 3) The differences in data sets, workstations, and accuracy are discussed.

According to the findings of a study on the detection of real-time licence plates based on vehicle regions and text detection[94]. The purpose of this paper is to present a novel method for the detection of real-time licence plates based on vehicle regions and text regions. In the first step of the process, the single shot multibox detector (SSD) framework is used to extract vehicle regions. Second, the multichannel maximally stable extremal regions (MSER) algorithm is applied to the vehicle regions in order to generate character candidates. This paper first eliminates false character candidates by analysing the properties of vehicle regions and then constructs licence plate candidate combinations using the characters that are still available. The subsequent step involves exploiting the correlation between the dimension of the vehicle and the licence plate in order to eliminate any false licence plate candidates. In the end, the licence plate candidates that are still available are put through a word/no-word classifier in order to select the winning licence plate. In this paper, the MobileNets architecture is selected for deep CNN configurations because it can be made to run in real time on embedded systems. The experimental findings on both the publicly available test dataset and the newly collected dataset indicate that the proposed method can be applied to a variety of licence plates with a performance that is superior to that of the state-of-the-art methods currently in use.

In a recent study based on R-CNN that looked at the detection of vehicle licence plates[95]. The purpose of this paper is to provide an overview of a robust licence plate detection method that makes use of smart mixing of faster R-CNN and image processing operations. In the survey that is being proposed, a vehicle will first be identified at the input by employing the coloured RGB images of the Faster R-CNN. After that, the image with the vehicle detected is sent to our already established License Location Plates Module (LPLM) so that a potential licence life Platform can be checked. The results of the experiments show that the proposed survey is very effective in dealing with a wide variety of images under a wide variety of conditions. These conditions include complex scenes, changes in illumination, distances, and various weather conditions, amongst others. Intelligent Transportation System; machine learning; R-CNN; faster R-CNN are some of the keywords that come to mind.

According to the findings of a study that used deep learning and font evaluation to detect licence plates[96]. There are two aspects to this work. On the one hand, we are going to propose using a Deep Learning technique (more specifically, You Only Look Once, or YOLO) in the LPD. On the other hand, the authors propose analysing the characteristics of the font within the context of the LP. This study makes use of two distinct datasets, UFPR-ALPR and the CENPARMI datasets, which were recently developed. The authors proposed an adaptive algorithm that is based on YOLO and has its parameters tuned in order to improve its performance. In addition to reporting the results of the recall ratio, this work will also conduct a comprehensive error analysis in order to provide some insights into the different kinds of false positives. A single YOLO network was all that was required for the proposed model to achieve a competitive recall ratio of 98.38%. While there are fonts that are difficult for humans to read, there are also fonts that are difficult for computers to recognise. In this section, we present two sets of findings regarding font evaluation: the findings regarding font anatomy and the findings regarding the recognition of commercial products. Both Mandatory and Driver Gothic are taken into consideration when presenting anatomy results. In addition, we make use of two commercial products called OpenALPR and Plate Recognizer to analyse how the fonts used in each dataset interact with their respective environments. The findings of the font anatomy comparison revealed several significant confusion cases as well as several qualities shared by both fonts. In contrast to the Mandatory font, the results that were obtained show that the Driver font does not have any severe instances of confusion.

To evaluate the effectiveness of the performance improvement method for multiple licence plate recognition in difficult environments[97]. A two-step strategy for plate localization in difficult conditions is presented as a potential solution in this paper. In the first step of the process, an algorithm called Faster-Region-based Convolutional Neural Network (Faster R-CNN) is used to detect all of the vehicles in an image. This produces scaled information that can then be used to locate licence plates. In the second stage of the process, morphological operations are used to cut down on the amount of non-plate areas. While this is going on, plates are being localised in the HSI colour space based on their geometric properties. This approach not only speeds up the processing time but also improves accuracy. The look-up table (LUT) classifier that uses adaptive boosting and modified census transform (MCT) as a feature extractor is used for character recognition. In terms of precision and recall for multiple plate recognition, the proposed methods for plate detection and character recognition have significantly outperformed conventional approaches.

In addition, a similar study was conducted on the detection and perspective rectification of vehicle licence plates in the year[98]. The authors of this work investigated an efficient framework for the perspective rectification and homography correction of images of vehicles as part of their research for this work. As a result of the many different movements, the images of the vehicle that were captured might be skewed in a vertical, horizontal, or both vertical and horizontal mix direction. An application of a polynomial fitting-based homography correction method is made in order to rectify the tilted VLPs. This helps to achieve reasonable high identification results. An investigation is carried out to discover a method for locating the four corner points of the rotated VPLs. The homography correction algorithm makes use of these four detected corner points at various points. Rotating the VLPs that have been detected in a variety of directions, such as horizontally, vertically, and in a mix of horizontal and vertical orientations, is done for the purpose of conducting an all-encompassing performance evaluation of the proposed framework. For the purposes of the experiments, real images of the vehicles were captured in their natural environments outside, looking in a variety of directions and from a range of distances. With the help of our proposed method, we were able to achieve an accuracy of 97 percent for the simulated images and 95 percent for the real images that were captured.

In the course of research carried out in Oman on the subject of automatic licence plate recognition[99]. The work discusses an idea for an automatic licence plate recognition system that could be implemented in Oman. In order to accomplish this goal, this chapter presents a comparison, both theoretically and analytically, between several earlier works in this field in order to gain an understanding of which algorithms are the most appropriate. Actual licence plates are used in the practical assessment that is carried out. According to the findings of this investigation, the recognition system can be broken down into three primary categories: number plate detection, number plate recognition, and character recognition. There are additional subpreprocessing operations and deep learning algorithms used in each subsequent level of processing. For instance, using morphological operations on number plate detection, using thresholding operations to extract binary images in the level of number plate recognition, and using convolutional neural networks in the character recognition level are some examples. The effectiveness of the recognition operation is judged using a variety of metrics, including classification accuracy, logarithmic loss, F1 score, precision, and recall, among others. Overall, the number plate extraction from vehicle images had an accuracy of 71.5%, while character recognition based on extracted characters had an accuracy of 96–99%. (depending on the type of character).

Using the more efficient R-CNN for automatic licence plate reading in motor vehicles[100]. The purpose of this paper is to propose the use of a Faster Recurrent Neural Network (R-CNN) in order to detect the number plate in the vehicle from the surveillance camera that is placed on the traffic areas etc. The developed system is put to use to record a video of the vehicle, which is followed by the extraction of the licence plate from the recorded video by means of frame segmentation and image interpolation for improved accuracy. For the purpose of number recognition, the resulting image is put through a process known as optical character recognition so that the numbers can be read. These numbers are used as input into the database in order to retrieve information such as the name of the vehicle, the owner's name, the address, and the owner's mobile number, among other things. The effectiveness of this system is evaluated using a graphical modelling tool (graph model). The system that is being proposed is capable of achieving an accuracy of 99.1% when detecting the number plate of the vehicle and displaying the information regarding the vehicle's owner.

During the high-speed tracking-by-detection process that did not make use of image information[101]. This article presents a novel method for high-accuracy real-time car licence plate detection that is based on the use of convolutional neural networks (CNN). There are a lot of modern techniques for detecting licence plates on cars, and many of them are reasonably effective, but only under very specific conditions or very strong assumptions. However, when the assessed images of car licence plates have a degree of rotation, as a result of manual capture by traffic police or deviation of the camera, they demonstrate poor performance. This can happen for a number of reasons. In light of this fact, the authors proposed a CNN-based MD-YOLO framework for the detection of car licence plates in multiple directions. The rotational issues that arise in real-time scenarios can be deftly handled by our proposed method thanks to its accurate rotation angle prediction and its lightning-fast intersection-over-union evaluation strategy. A number of tests have been carried out to demonstrate that the proposed method is superior to other existing methods that are considered to be state-of-the-art in terms of producing more accurate results at a lower cost to the computational system.

2.4 Literature Summary and Gap

This section presented a review or literature related and relevant to the research topic under investigation. The section started with the conceptual review relevant concepts: intelligent transportation system, license plate recognition system, license plate segmentation, license plate application, automatic number plate recognition, license plate deflection and Nigerian licence plate were defined and clarified. This was followed by the theoretical framework which discussed licence plate algorithms. Empirical findings both local and international on license plate recognition algorithm (region-based convolutional neural networks, YOLO detector, single neural network, RFID-systems, OCR, RetinaFace, MobileNet, ZedBoard SoC platform, R-CNN, GAN-based synthetic data, maximally stable extremal regions (MSER) algorithm and others) were discussed and presented.

Various researchers have worked and contributed to the existing literature on License plate recognition and vehicular violation monitoring using various algorithms[82,82,83,84]. Some also worked on recognising licence plates, taking large camera shooting angles, polynomial fitting-based homography correction method[87,98]. However, most of this investigations are done in other countries and there is a dearth in Literature on license plate recognition in Nigeria. Also, most of the studies are not realtime which lacks efficient monitoring and reporting. This study therefore tend to develop a web based prototype high way vehicular monitoring and reporting system having low-cost inputs.

Endnotes

1. AE Gorev, O Gasilova, BA Sidorov. *Surface transportation engineering technology: Prerequisite for accident-free traffic at signal-controlled intersections.* Architecture and Engineering. 2021;6(1):73-80.

2. J Tang, L Wan, J Schooling, P Zhao, J Chen, S Wei. *Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases. Cities.* 2022 Oct 1;129:103833.

3. S Raj, Y Gupta, R Malhotra. *License Plate Recognition System using Yolov5 and CNN.* In2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) 2022 Mar 25 (Vol. 1, pp. 372-377). IEEE.

4. P Dhar, S Guha, T Biswas, MZ Abedin. *A system design for license plate recognition by using edge detection and convolution neural network.* In2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) 2018 Feb 8 (pp. 1-4). IEEE.

5. N Mufti, SA Shah. *Automatic number plate Recognition: A detailed survey of relevant algorithms.* Sensors. 2021 Apr 26;21(9):3028.

6. Y Kessentini, MD Besbes, S Ammar, A Chabbouh. *A two-stage deep neural network for multi-norm license plate detection and recognition.* Expert systems with applications. 2019 Dec 1;136:159-70.

7. B Mindula, M Ranasinghe, R Ahamed, J Tennakoon, C Silva, G Wimalarathne. *Image and Video Processing based Expressway Traffic Rules Violation Detection.* In2021 8th International Conference on ICT & Accessibility (ICTA) 2021 Dec 8 (pp. 1-6). IEEE.

11. J Tang, L Wan, J Schooling, P Zhao, J Chen, S Wei. *Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases.* Cities. 2022 Oct 1;129:103833.

12. HG Daway, EG Daway, HH Kareem. *Colour image enhancement by fuzzy logic based on sigmoid membership function.* **International Journal of Intelligent Engineering and Systems.** 2020;13(5):238-46.

13. X Wu, Z Wei, Y Hu, L Wang. *Traffic Sign Detection Method Using Multi-Color Space Fusion.* In2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) 2020 Jun 27 (pp. 314-319). IEEE.

14. A Menon, B Omman. *Detection and recognition of multiple license plate from still images.* In2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET) 2018 Dec 21 (pp. 1-5). IEEE.

15. P Arora, VM Kapse, S Sinha, S Gera. *Number Plate Recognition System Using Convolutional Neural Network.* In2021 9th International Conference on Reliability,

Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) 2021 Sep 3 (pp. 1-5). IEEE.

16. AS Agbemenu, J Yankey, EO Addo. *An automatic number plate recognition system using opencv and tesseract ocr engine.* **International Journal of Computer Applications.** 2018 May;180(43):1-5.

17. MY Arafat, AS Khairuddin, U Khairuddin, R Paramesran. *Systematic review on vehicular licence plate recognition framework in intelligent transport systems.* IET Intelligent Transport Systems. 2019 May;13(5):745-55.

18. Y Kulkarni, S Bodkhe, AKamthe, A Patil. *Automatic number plate recognition for motorcyclists riding without helmet.* In2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) 2018 Mar 1 (pp. 1-6). IEEE.

19. PS Mellinda, F Sthevanie, KN Ramadhani. *Detection Of Vehicle Number Plate Using Probabilistic Hough Transform.* eProceedings of Engineering. 2020 Aug 1;7(2).

20. M Mafi, H Rajaei, M Cabrerizo, M Adjouadi. *A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering.* IEEE Transactions on Image Processing. 2018 Jul 18;27(11):5475-90.

21. J Shashirangana, H Padmasiri, D Meedeniya, C Perera. *Automated license plate recognition: a survey on methods and techniques.* IEEE Access. 2020 Dec 29;9:11203-25.

22. V Tadic, Z Kiraly, P Odry, Z Trpovski, T Loncar-Turukalo. *Comparison of Gabor filter bank and fuzzified Gabor filter for license plate detection.* Acta Polytechnica Hungarica. 2020 Jan 1;17(1):1-21.

23. MK Hossen, AC Roy, MS Chowdhury, MS Islam, K Deb. *License plate detection and recognition system based on morphological approach and feed-forward neural network.* **IJCSNS International Journal of Computer Science and Network Security**. 2018 May 30;18(5):36-45.

24. M Kročka, P Dakić, V Vranić. *Automatic License Plate Recognition Using Open CV.* In2022 12th International Conference on Advanced Computer Information Technologies (ACIT) 2022 Sep 26 (pp. 530-535). IEEE.

25. F Spagnolo, S Perri, P Corsonello. *An efficient hardware-oriented single-pass approach for connected component analysis.* Sensors. 2019 Jul 11;19(14):3055.

26. IA Znamenskaya, IA Doroshchenko. *Edge detection and machine learning for automatic flow structures detection and tracking on schlieren and shadowgraph images.* **Journal of Flow Visualization and Image Processing.** 2021;28(4).

27. I Slimani, A Zaarane, A Hamdoun, I Atouf. *Vehicle license plate localization and recognition system for intelligent transportation applications.* In2019 6th

International Conference on Control, Decision and Information Technologies (CoDIT) 2019 Apr 23 (pp. 1592-1597). IEEE.

[28.] G Sharma. *Performance analysis of vehicle number plate recognition system using template matching techniques.* **Journal of Information Technology & Software Engineering.** 2018 Apr;8(2):1-9

[29.] WVEl-Tarhouni, A Abdo, A ELmegreisi. *Feature fusion using the Local Binary Pattern Histogram Fourier and the Pyramid Histogram of Feature fusion using the Local Binary Pattern Oriented Gradient in iris recognition.* In2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA 2021 May 25 (pp. 853-857). IEEE.

[30.] Z Selmi, MB Halima, U Pal, MA Alimi. *DELP-DAR system for license plate detection and recognition*. Pattern Recognition Letters. 2020 Jan 1;129:213-23.

[31.] R Laroca, LA Zanlorensi, GR Gonçalves, E Todt, WR Schwartz, D Menotti. *An efficient and layout-independent automatic license plate recognition system based on the YOLO detector.* IET Intelligent Transport Systems. 2021 Apr;15(4):483-503.

[32.] J Pang, X Pu, C Li. *A Hybrid Algorithm Incorporating Vector Quantization and One-Class Support Vector Machine for Industrial Anomaly Detection.* IEEE Transactions on Industrial Informatics. 2022 Jan 25.

[33.] MC Wijaya. *Research of Indonesian license plates recognition on moving vehicles.* EUREKA: Physics and Engineering. 2022 Nov 29(6):185-98

[34.] MS Al-Shemarry. *Developing new techniques to improve licence plate detection systems for complicated and low quality vehicle images* (Doctoral dissertation, University of Southern Queensland).

[35.] L Tarekegn. *Faculity Of Informatics Shape And Texture Based Inter-Specific Hybrid Fish Species Image Recognition* (Doctoral dissertation, University of Gondar).2019

[36.] MR Niluckshini, MF Firdhous. *Automatic Number Plate Detection using Haar-Cascade Algorithm Proposed for Srilankan Context.* In2022 2nd International Conference on Advanced Research in Computing (ICARC) 2022 Feb 23 (pp. 248-253). IEEE.

[37.] AN Titus. *Vehicle License Plate Localization based on Local Binary Pattern Features.* In2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC) 2019 Mar 7 (pp. 1-5). IEEE.

[38.] DM Rhee, FT Lombardo, J Kadowaki. *Semi-automated tree-fall pattern identification using image processing technique: Application to Alonsa, MB tornado.* **Journal of Wind Engineering and Industrial Aerodynamics.** 2021 Jan 1;208:104399.

[39] XA Davix, CS Christopher. *Edge based marker controlled watershed algorithm for automatic car licence plate localization. Journal of Computational and Theoretical Nanoscience.* 2017 Nov 1;14(11):5539-51.

[40] J Joseph, A Prasad, and Jithina, L Mary. *A Study on Localization Techniques for Automatic License Plate Recognition System*. In2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) 2019 Jul 5 (Vol. 1, pp. 1090-1094). IEEE.

[41] MY Arafat, AS Khairuddin, U Khairuddin, R Paramesran. *Systematic review on vehicular licence plate recognition framework in intelligent transport systems*. IET Intelligent Transport Systems. 2019 May;13(5):745-55.

[42] J Shashirangana, H Padmasiri, D Meedeniya, C Perera. *Automated license plate recognition: a survey on methods and techniques.* IEEE Access. 2020 Dec 29;9:11203-25.

[43] TA Pham. *Effective deep neural networks for license plate detection and recognition*. The Visual Computer. 2022 Jan 21:1-5.

[44] BE. *Person detection and tracking using omnidirectional cameras, and rectangle blanket problem.2019.*

[45] N Darapaneni, K Mogeraya, S Mandal, A Narayanan, P Siva, AR Paduri, F Khan, PM Agadi. *Computer vision based license plate detection for automated vehicle parking management system.* In2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) 2020 Oct 28 (pp. 0800-0805). IEEE.

[46] P Rani, S Kotwal, J Manhas, V Sharma, S Sharma. Machine learning and deep learning based computational approaches in *automatic microorganisms image recognition: methodologies, challenges, and developments.* Archives of Computational Methods in Engineering. 2022 May;29(3):1801-37.

[47] J Zhuang, S Hou, Z Wang, ZJ Zha. *Towards human-level license plate recognition.* InProceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 306-321)

[48] J Pareek, D Singhania, RR Kumari, S Purohit. *Gujarati handwritten character recognition from text images.* Procedia Computer Science. 2020 Jan 1;171:514-23..

[49] SB Suthar, AR Thakkar. *Hybrid Deep Resnet With Inception Model For Optical Character Recognition In Gujarati Language.* Reliability: Theory & Applications. 2022;17(1 (67)):194-209.

[50] K Kumar, S Sinha, P Manupriya. D-*PNR: deep license plate number recognition.* InProceedings of 2nd International Conference on Computer Vision & Image Processing 2018 (pp. 37-46). Springer, Singapore.

[51] R Peter, AK Grosselfinger, D Münch, Arens M. *Automated license plate detection for image anonymization.* InCounterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III 2019 Oct 7 (Vol. 11166, pp. 218-231). SPIE.

[52] Y Yang, W Zhang, Z He, D Li. *High-speed rail pole number recognition through deep representation and temporal redundancy.* Neurocomputing. 2020 Nov 20;415:201-14.

[53] G Ning. *Vehicle license plate detection and recognition (Doctoral dissertation,* University of Missouri--Columbia).2013.

[54] FN Khan, Q Fan, C Lu, AP Lau. *Machine learning methods for optical communication systems and networks.* InOptical fiber telecommunications VII 2020 Jan 1 (pp. 921-978). Academic Press.

[55] N Mufti, SA Shah. *Automatic number plate Recognition: A detailed survey of relevant algorithms.* Sensors. 2021 Apr 26;21(9):3028.

[56] T Jain, VK Verma, P Garg, M Jangid. *An Improvised Model for High-Security License Plate Detection and Recognition for Indian Vehicle to Enhance Detection.* Computational Network Application Tools for Performance Management. 2019 Oct 18:109.

[57] CJ Lin, CC Chuang, HY Lin. *Edge-AI-Based Real-Time Automated License Plate Recognition System.* Applied Sciences. 2022 Jan 28;12(3):1445.

[58] E Altinsoy, J Yang, C Yilmaz. *Fully-automatic raw G-band chromosome image segmentation.* IET Image Processing. 2020 Jul;14(9):1920-8.

[59] S Ghasempour. *Automatic License Plate Recognition (ALPR) (Master's thesis, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ).*2015.

[60] K Manaa, M Rabee'a, L Khalaf. *Traffic control by digital imaging cameras. InEmerging Trends in Image Processing*, Computer Vision and Pattern Recognition 2015 Jan 1 (pp. 231-247). Morgan Kaufmann.

[61] J Tang, L Wan, J Schooling, P Zhao, J Chen, S Wei. *Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases.* Cities. 2022 Oct 1;129:103833.

[62] TK HS, JR DK, K Priyadharsini. *An experiment analysis on tracking and detecting the vehicle speed using machine learning and iot.* In2021 Smart Technologies, Communication and Robotics (STCR) 2021 Oct 9 (pp. 1-5). IEEE.

[63] MY Amri. *Development of automatic number plate recognition software and journey time measurement/Amri Mohd Yasin* (Doctoral dissertation, University of Malaya)

[64] MA Awel, AI Abidi. *Review on optical character recognition.* **International Research Journal of Engineering and Technology (IRJET)**. 2019 Jun;6(6):3666-9.

[65] I Shafi, I Hussain, J Ahmad, PW Kim, GS Choi, I Ashraf, S Din. *License plate identification and recognition in a non-standard environment using neural pattern matching.* Complex & Intelligent Systems. 2022 Oct;8(5):3627-39

[66] Y Sun, X Mao, S Hong, W Xu, G Gui. *Template matching-based method for intelligent invoice information identification*. IEEE access. 2019 Feb 27;7:28392-401

[67] W Weihong, T Jiaoyang. *Research on license plate recognition algorithms based on deep learning in complex environment*. IEEE Access. 2020 May 14;8:91661-75.

[68] HS Gharraf, G Cansever, AS Ahmed. *Image Filtering of Impulsive Noise Using Biologically Inspired Algorithms.* In2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2022 Oct 20 (pp. 58-65). IEEE.

[69] C Okafor, C Ajaero, C Madu, K Agomuo, E Abu. *Implementation of circular economy principles in management of end-of-life tyres in a developing country (Nigeria).* AIMS Environ Sci. 2020 Oct 12;7:406-33.

[70] TS Ibiyemi, JS Owotogbe, BA Adu. *A Comparative Study of Vehicle Number Plate Recognition Systems.* **African Journal of Management Information System**. 2020;2(1):10-23.

[71] W Wang, J Yang, M Chen, P Wang. A *light CNN for end-to-end car license plates detection and recognition.* IEEE Access. 2019 Nov 28;7:173875-83.

[72] H Xiang, Y Zhao, Y Yuan, G Zhang, X Hu. *Lightweight fully convolutional network for license plate detection*. Optik. 2019 Feb 1;178:1185-94.

[73] H Xiang, Y Zhao, Y Yuan, G Zhang, X Hu. *Lightweight fully convolutional network for license plate detection.* Optik. 2019 Feb 1;178:1185-94.

[74] M Dong, D He, C Luo, D Liu, W Zeng. *A CNN-Based Approach for Automatic License Plate Recognition in the Wild*. InBMVC 2017 Sep.

[75] J Shashirangana, H Padmasiri, D Meedeniya, C Perera. *Automated license plate recognition: a survey on methods and techniques.* IEEE Access. 2020 Dec 29;9:11203-25.

[76] R Laroca, LA Zanlorensi, GR Gonçalves, E Todt, WR Schwartz, D Menotti. *An efficient and layout-independent automatic license plate recognition system based on the YOLO detector.* IET Intelligent Transport Systems. 2021 Apr;15(4):483-503.

[77] MA Rafique, W Pedrycz, M Jeon. *Vehicle license plate detection using region-based convolutional neural networks.* Soft Computing. 2018 Oct;22(19):6429-40.

[78] R Laroca, E Severo, LA Zanlorensi, LS Oliveira, GR Gonçalves, WR Schwartz, D Menotti. *A robust real-time automatic license plate recognition based on the YOLO*

detector. In2018 **international joint conference on neural networks (IJCNN)** 2018 Jul 8 (pp. 1-10). IEEE.

79. Q Huang, Z Cai, T Lan. *A single neural network for mixed style license plate detection and recognition*. IEEE Access. 2021 Jan 28;9:21777-85.

80. V Kukreja, D Kumar, A Kaur. *GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition.* In2020 4th international conference on electronics, communication and aerospace technology (ICECA) 2020 Nov 5 (pp. 1190-1195). IEEE.

81. Q Huang, Z Cai, T Lan. *A new approach for character recognition of multi-style vehicle license plates*. IEEE Transactions on multimedia. 2020 Oct 14;23:3768-77.

82. N Mufti, SA Shah. *Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors*. 2021 Apr 26;21(9):3028.

83. R Antar, S Alghamdi, J Alotaibi, M Alghamdi. *Automatic Number Plate Recognition of Saudi License Car Plates.* Engineering, Technology & Applied Science Research. 2022 Apr 9;12(2):8266-72.

84. SF Abd Gani, MF Miskon, RA Hamzah, N Mohamood, Z Manap, MF Zulkifli. *A live-video automatic number plate recognition (anpr) system using convolutional neural network (CNN) with data labelling on an android smartphone.*

85. H Kuchuk, A Podorozhniak, N Liubchenko, D Onischenko. *System of license plate recognition considering large camera shooting angles.* Radioelectronic and Computer Systems. 2021 Nov 29(4):82-91.

86. YD Salman, HS Alhadawi, AS Mahdi, AL-Dhief FT. *Improved Automatic License Plate Recognition System in Iraq for Surveillance System Using OCR.* InInternational Conference on Emerging Technologies and Intelligent Systems 2023 (pp. 270-277). Springer, Cham.

87. KB Sheng, AA Saad, MK Ishak. *Development of a Virtual Vehicle Identification for Tracking Hit-and-Run Vehicle.* In2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) 2022 Sep 13 (pp. 1-6). IEEE.

88. H Nguyen-Xuan, D Hoang-Nhu, K Kim-Van, T Dang-Minh. *A New System for License Plate Recognition in Traffic Violation Scenarios in Vietnam.* In Intelligent Systems and Networks 2022 (pp. 287-297). Springer, Singapore.

89. S Chhabra, S Saini, K Lata. *Hardware–software co-simulation of vehicle license plate detection on the ZedBoard SoC platform.* In Advances in Image and Data Processing using VLSI Design, Volume 1: Smart vision systems 2021 Dec 1. IOP Publishing.

[90.] N Hamdoun, D Mentagui. *Image Processing in Automatic License Plate Recognition Using Combined Methods.* Serdica Journal of Computing. 2022 Jul 4;16(1):1-23.

[91.] W Weihong, T Jiaoyang. *Research on license plate recognition algorithms based on deep learning in complex environment.* IEEE Access. 2020 May 14;8:91661-75.

[92.] H Nguyen. *Real-Time License Plate Detection Based On Vehicle Region And Text Detection.* **Journal of Theoretical and Applied Information Technology**. 2020 Feb 15;98(03).

[93.] J.K Denny, J Denny, A Satheesh. and M METS. *Recent Study on Vehicle Licence Plate Detection Based on R-CNN.* **IJIRIS: International Journal of Innovative Research in Information Security,** Volume VII, 53-56, 2020

[94.] R Al-qudah. *License Plate Detection using Deep Learning and Font Evaluation (Doctoral dissertation, Concordia University*

[95.] K Khan, A Imran, HZ Rehman, A Fazil, M Zakwan, Z Mahmood. *Performance enhancement method for multiple license plate recognition in challenging environments.* **EURASIP Journal on Image and Video Processing.** 2021 Dec;2021(1):1-23.

[96.] M Alhussein, K Aurangzeb, SI Haider. *Vehicle license plate detection and perspective rectification.* Elektronika ir Elektrotechnika. 2019 Oct 6;25(5):47-56.

[97.] M Al Awaimri, S Fageeri, A Moyaid, C Thron, A ALhasanat. *Automatic Number Plate Recognition System for Oman.* InArtificial Intelligence for Data Science in Theory and Practice 2022 (pp. 155-178). Springer, Cham.

[98.] NP Ap, T Vigneshwaran, MS Arappradhan, R Madhanraj. *Automatic Number Plate Detection in Vehicles using Faster R-CNN.* In2020 International Conference on System, Computation, Automation and Networking (ICSCAN) 2020 Jul 3 (pp. 1-6). IEEE.

[99.] E Bochinski, V Eiselein, T Sikora. *High-speed tracking-by-detection without using image information.* In2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS) 2017 Aug 29 (pp. 1-6). IEEE.

Chapter Three

Methodology

3.1 Research Approach

The research study will generate a real-time low cost high way vehicular monitoring and reporting system having low-cost inputs.The prototype of the vehicular monitoring system node will be created using a mobile phone (with camera detection capability of 15-20 metres at coverage of 180 degrees and, 1200x900 pixels resolution), Laptop, and licence plate' monitoring model softwares was developed and installed in the mobile phone and Laptop respectively. A mobile app, a tunnelling server, web application and number plate detection unit were used in the process of developing the model software[1]. A functional https connection will be established between the tunnelling server and the mobile application by the server.

3.2 System Design

The prototype system was developed using electronic architecture. This include: Network Unit (Tunnel Server), Mobile Unit (Mobile App), and number plate detection unit.

3.2.1 The Tunnel Server

The tunnel is the software that help us communicate with the local web server through our phones(mobile app). The tunnel we employed for this project is ngrok.io. it's a command line application that help generate active https link between our server and the mobile app. ngrok exposes local networked services behinds NATs and firewalls to the public internet over a secure tunnel. Ngrok is a cross-platform tool that allows developers to rapidly and simply connect a local development server to the Internet. The software allows the locally hosted web server seem to be hosted on the ngrok.com subdomain, removing the requirement for the local computer to have a public IP address or domain name.[3]

To open local networked resources behind NATs and firewalls to the public internet, Ngrok employs a safe tunnel. Build/test webhook consumers, self-host personal service, share local websites. Ngrok launches a small client process that builds a private connection tunnel to the cloud service to get around access limitations.[3] A remote user can access the localhost development server by mapping it to a ngrok.io sub-domain. Ngrok enables a web server running on a local system to be exposed to the internet. Hence, the command to generate the webhook is ngrok http 80.
Moreover, command "ngrok http 8000 was typed on the tunneling to replace the http 80. This will then generate two protocols (http and https). For this research the secured tunnel (protocol) was used

The mobile app for this project was built on flutter. This is used to capture and send images to a backend API url. Flutter is Google's mobile app SDK, which includes a platform, widgets, and tools to help developers build and launch visually attractive, fast Android and iOS applications.[4] Flutter simplifies the method of designing cross-platform smart phone software. It is based on Dart, a swift and easy-to-learn object-oriented programming language.[4] Flutter widgets, which are rendered with a high-performance rendering engine, are also supported. They are simple, appealing, and adaptable.[4]

3.2.2 The Mobile App

The mobile app for this project was built on flutter. This is used to capture and send images to a backend API url. Flutter is Google's mobile app SDK, which includes a platform, widgets, and tools to help developers build and launch visually attractive, fast Android and iOS applications.[4] Flutter simplifies the method of designing cross-platform smart phone software. It is based on Dart, a swift and easy-to-learn object-oriented programming language.[4] Flutter widgets, which are rendered with a high-

performance rendering engine, are also supported. They are simple, appealing, and adaptable4. The snippet codes used in designing the mobile app is shown below. The full programming code is available are in Appendix AI

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_inappwebview/flutter_inappwebview.dart';
```

The mobile app consists of 2 pages as shown in Figure 3.1; the tunnel url input and the web view. The tunnel url input helps get the dynamically generated tunnel url to be viewed through a web view. With the help of tunneling link, the mobile app can access what's on the server as this will entail that both the server and the mobile app must be connected to the internet. This is done by copying the forwarded secured url from the tunnel, paste in the mobile app and clicking the next tab as shown in figure 3.1 (b). Once it runs, it goes to a web browser which is a canvas. The canvas hence opens the link of the detection unit which is where the detection takes place as shown in figure 3.2.

A web viewer synchrinise the canvas (web browser) and mobile app in such a way that the camera of the mobile app can be used to work as the camera for the mobile detection system using the snippet code below

```
class InAppWebViewPage extends StatefulWidget {
 final String url;
 InAppWebViewPage(this.url);
 @override
```

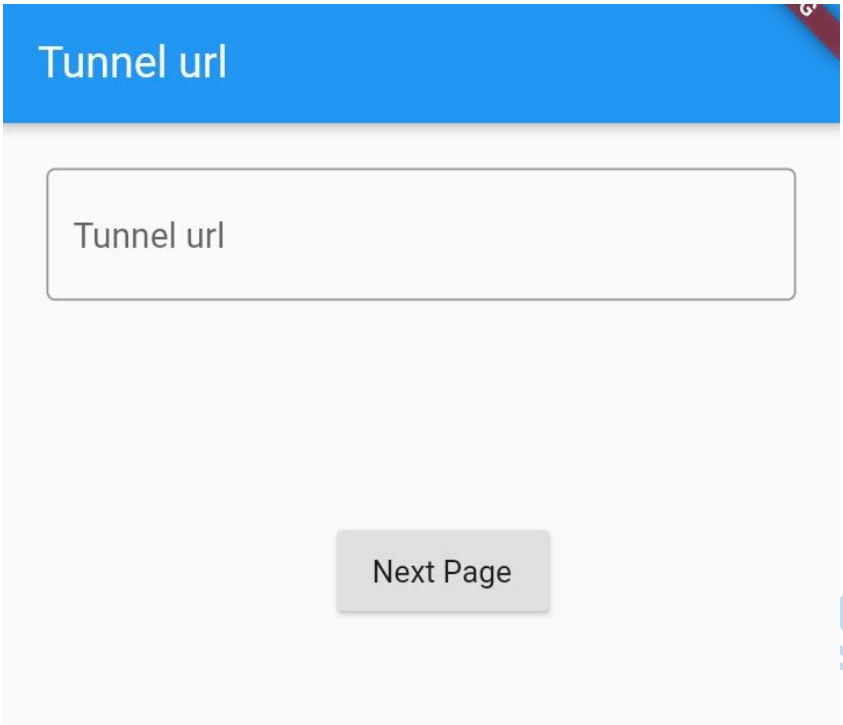


Figure 2.8: The Tunnel URL

3.2.3 Detection Unit

The detection unit is divided into three

1. Video source capture
2. ML5.js
3. OpenCV.js

Video Capture Unit: The video source capture uses navigator.getUserMedia facility in a browser to get access to the camera of the device. Since a browser is used on the app, the app is loading a browser in on its own. But since the browser is expected to use a camera, the phone camera now becomes the access camera of the detection unit. Instead of using the web cam of the system, the mobile app accesses the camera of the mobile phone which serves as the camera for the detection unit. The camera is initiated with the following codes.

```
const videoElement = document.createElement('video');
videoElement.setAttribute("style", "display: none;"); videoElement.width = width;
videoElement.height = height;
```

```
document.body.appendChild(videoElement);
```

ML5.js: A video element was created and get a navigator devices "navigator.mediaDevices.getUserMedia".

```
// Create a webcam capture
const capture = await navigator.mediaDevices.getUserMedia({ video: true })
videoElement.srcObject = capture;
videoElement.play();
return videoElement
```

Everything that is seen by the video element is being checked by the ML5. ML5.js is an open-source JavaScript library that provides a high-level interface for machine learning in the browser. It's built on top of TensorFlow.js[1]. ML5.js simplifies the process of using pre-trained machine learning models for various tasks, such as image classification, object detection, text generation, and more. In this project, ML5 is used such that every frame the camera is recording is passed through the library *"Objectdetector = await ml5. objectdetector ('cocossd', startDetecting).* Where it looks for anything in its library for example, if it's a car, or person, it will identify it.

OpenCV.js: This is used to find the number plate using the find number plate function *"function findPlateNumber (Image)"* which will read the image and convert it to gray scale, edge detection, finding contours to find if there is a number plate there based on the characteristics and others and if there is a number plate, it should give a coordinate of the number plate (x, y position, width and height) and show on the canvas.

OpenCV.js is a JavaScript binding or port of the popular computer vision library OpenCV (Open Source Computer Vision Library)[2]. OpenCV is a library designed for computer vision tasks. It provides a wide range of tools and functions for image and video analysis, object detection, feature extraction, facial recognition, and more. It extends the accessibility of OpenCV to web applications, making it possible to perform computer vision tasks in web browsers[2]. OpenCV.js is built with performance in mind. It leverages hardware acceleration through WebGL when available, which can significantly speed up computer vision tasks when running in a web browser[2].

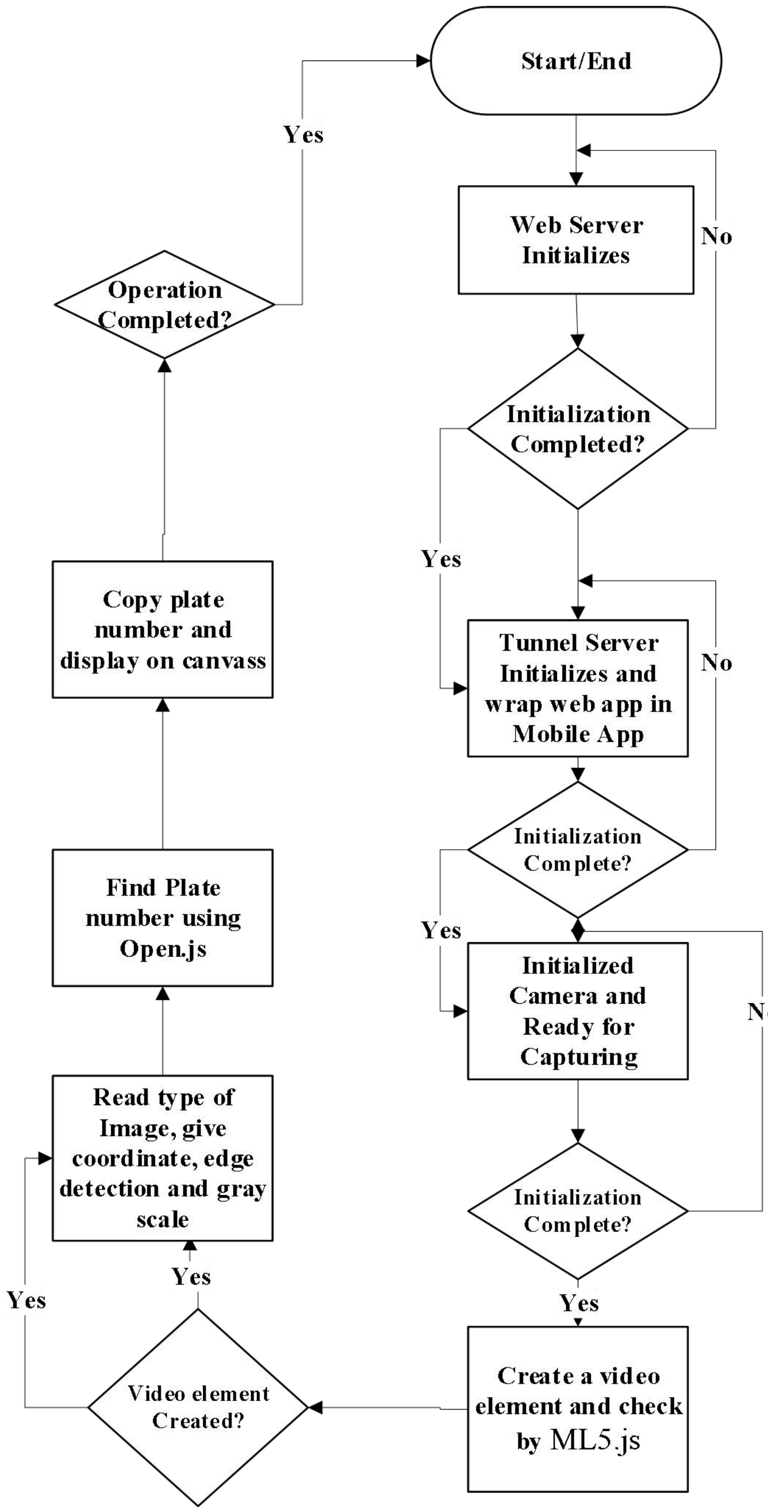


Figure 3.2: Flowchart of Vehicular Monitoring and Reporting System.

3.3 Requirement Specification

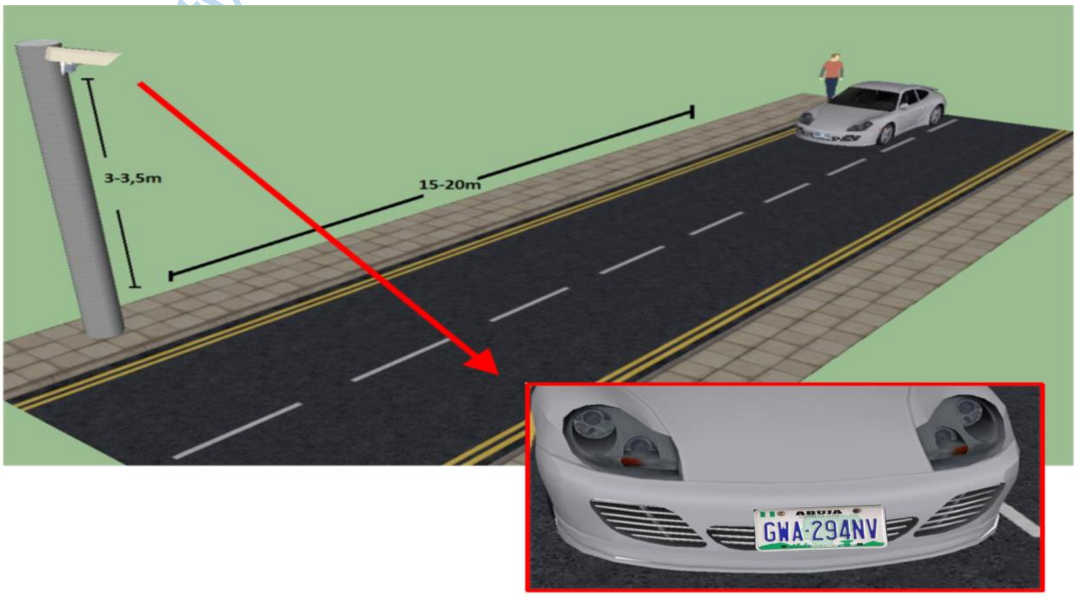
Hardware Minimum Requirements: hardware Specification for the prototype system consist of a processor Intel®Pentium®CPU, Random Access Memory RAM of 4GB or more capacity, Read Only Memory ROM of 250G or more. The android phone used for the mobile app is a Kernel Version 4.4.147+, Android Version 10 with 1 GB, RAM

Software Requirements: The software specification consist of an operating system OS with windows 10 of 64bit. The Web App was a WAMP Server (Windows, Apache, Msql, Php), built on Laravel. Tunnel server is Ngrok.io and the mobile app is developed using Flutter and Matlab R2015a for OCR.

3.4 Research Method

3.4.1 License Plate Capturing

The prototype camera that will be used has a resolution of 1200x900 pixels and 13 megapixels (with an aperture of f/2.2). However, in order to determine how far the camera on the mobile phone will be able to see, a field view formula was developed. This formula was used to determine the distance at which an license plate can be identified.



3.4.2 License Plate Detection

An HTML file is created to set up the structure of of the web application. JavaScript libraries, such as ML5.js and OpenCV.js, were included in your HTML file. Uing HTML5's `getUserMedia` API to access the user's camera and capture video. Display the video stream on an HTML5 `<video>` element.

```
<video id="videoElement" autoplay></video>
```

ML5.js, was used which includes pre-trained machine learning models, for object detection. Alos a pre-trained object detection model (e.g., COCO-SSD) and the video element was specified as the input source. Thus detecting objects, including license plates, in each video frame.

```
// Initialize the video capture
const videoElement = document.getElementById('videoElement');
const detector = ml5.objectDetector('cocossd', {}, () => {
  console.log('Model is ready');
  detector.detect(videoElement, gotResults);
});
  // Process the detected objects, including license plates
  // Implement logic to filter license plates from other objects
}
```

Once the number plate was detected Optical Character Recognition (OCR) can be performed to recognize the characters on the plate.

The image was processed with OpenCV.js for additional image processing tasks such as filtering, or perspective transformations to improve the quality of the license plate.

3.5 Performance Evaluation of the Developed System

In order to accomplish the goals of this study, Black Box Testing method was used, which is devoid of any functional aspects. A form of software testing known as "black box testing" examines a program's functionality without taking into account the program's internal structure or the code it was written in. Testing by hand can be carried out in a number of different ways, including white box testing, black box testing, and grey box testing. The tester first selects a function in order to validate that it is operational, and then examines the output of the function to determine whether or not it generates the expected result. On the developed prototype of the system, performance and scalability testing was conducted, as well as usability testing, compatibility testing, and reliability testing.

Endnotes

1. https://ml5js.org/

2. https://opencv.org/

3. P Venkateswari, EJ Steffy, DN Muthukumaran. *License Plate cognizance by Ocular Character Perception'*. **International Research Journal of Engineering and Technology.** 2018 Feb;5(2):536-42.

Chapter Four

Analysis, Results and Discussion

4.1 Result on Electronic Vehicle Monitoring and Reporting System

4.1.1 Tunnel Server

To generate result for the design, a command "php artisan serve" was written that the ngrok http port 80(secure public URL for port 80 web server) highlighted on the tunneling server as shown in figure 4.1 is forwarded to 8000 as shown in figure 4.2 . Hence starting the laravel development server. The terminal was cleared and show the status with two forwarding http and https addresses.

Moreover, command "ngrok http http://localhost:8000/ was written on the tunneling server as shown in figure 4.3 to replace the http 80. This will then generate a secured tunnel (highlighted) as shown in figure 4.4. When ngrok runs an HTTP tunnel, it opens endpoints for both HTTP and HTTPS traffic. The connection tunnel established by ngrok is secure and can only transmit data to the localhost when port is open. This is where https URL will be copied, so as to access the application from the mobile app.

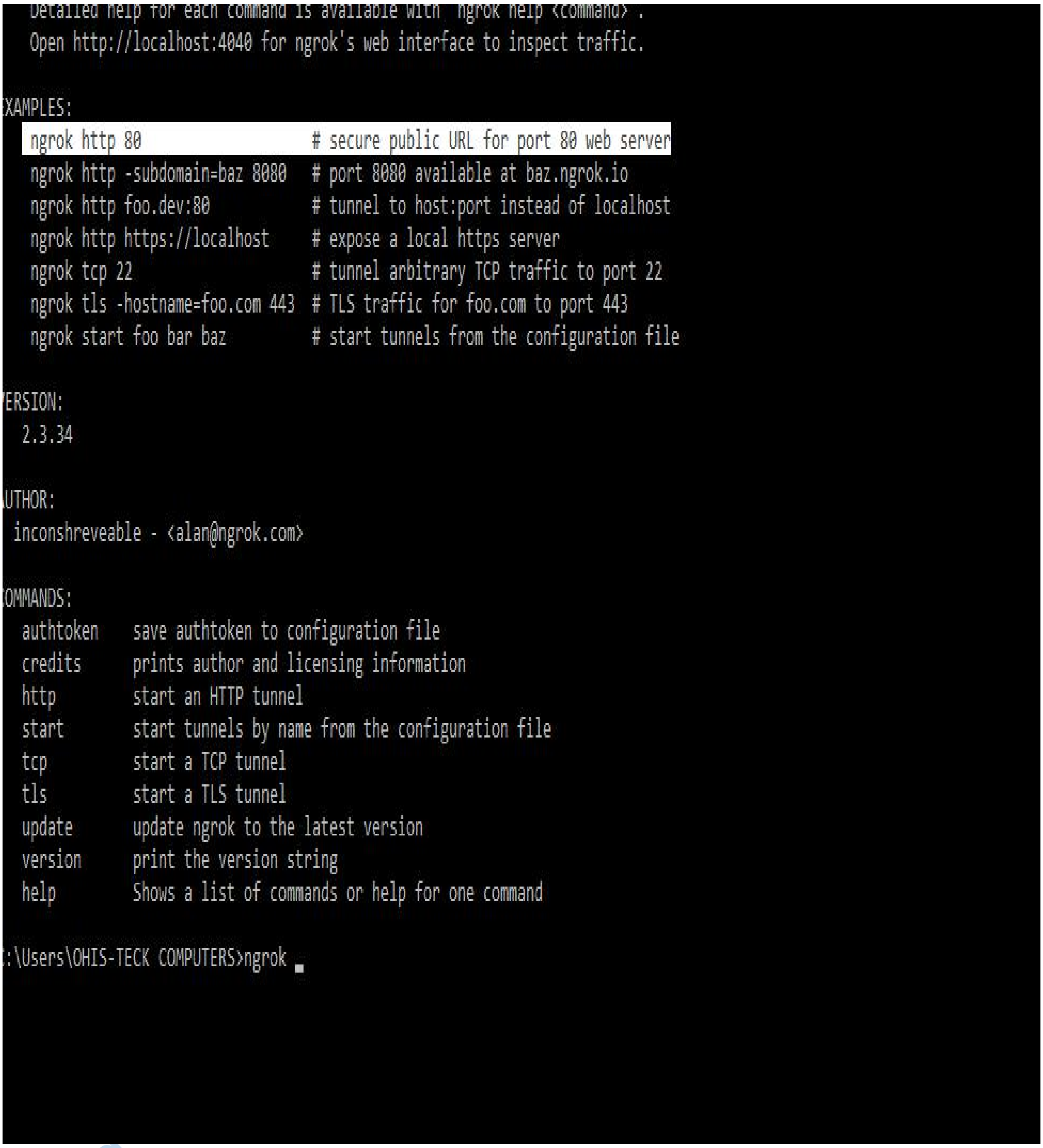


Figure 4.1: Tunneling Server Showing http port 80
Source: Research Work, 2023

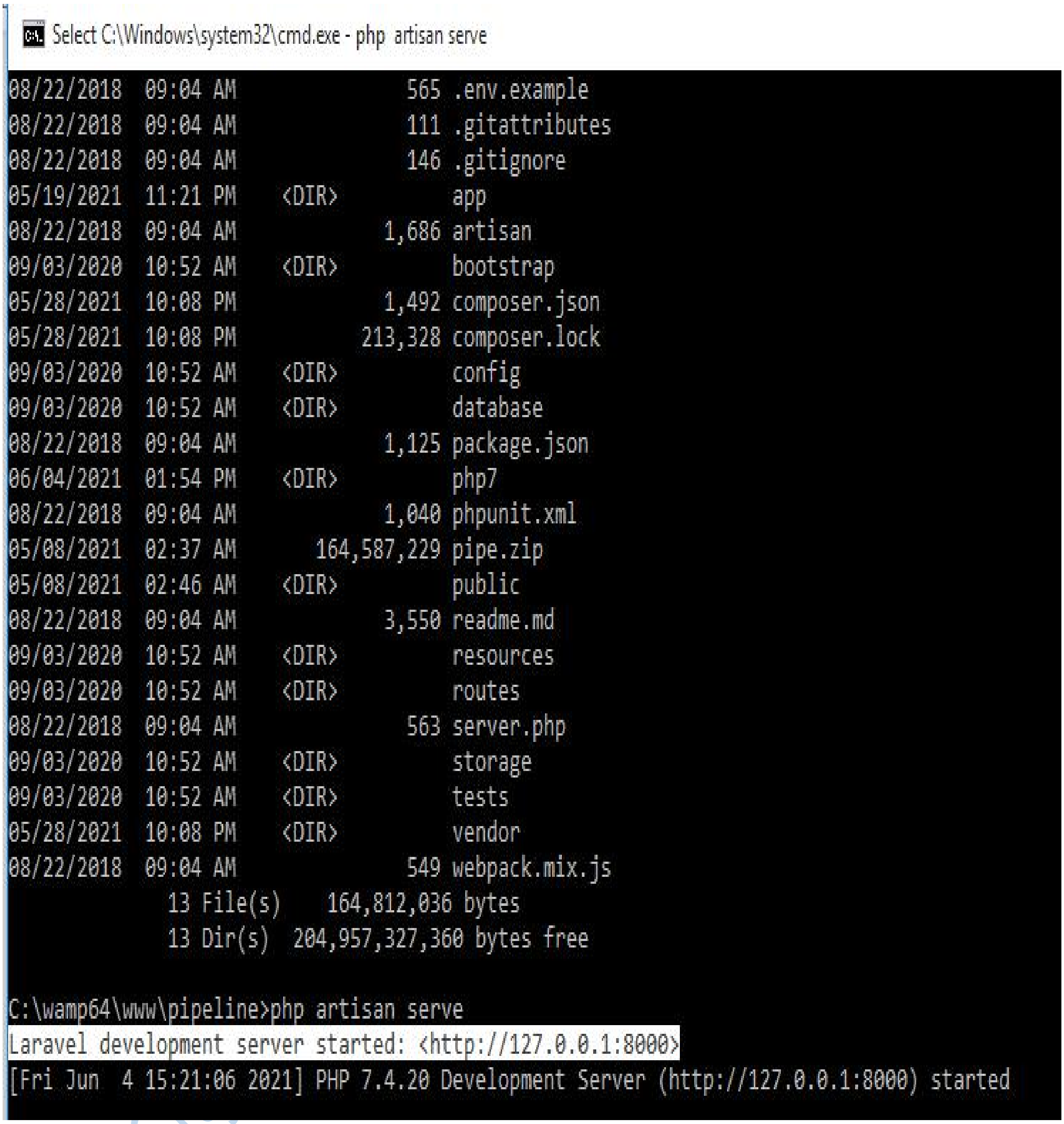


Figure 4.2: Php Artisan Serve Showing the Port 8000 that Ngrok Port 80 Will be Forwarded
Source: Research Work, 2023

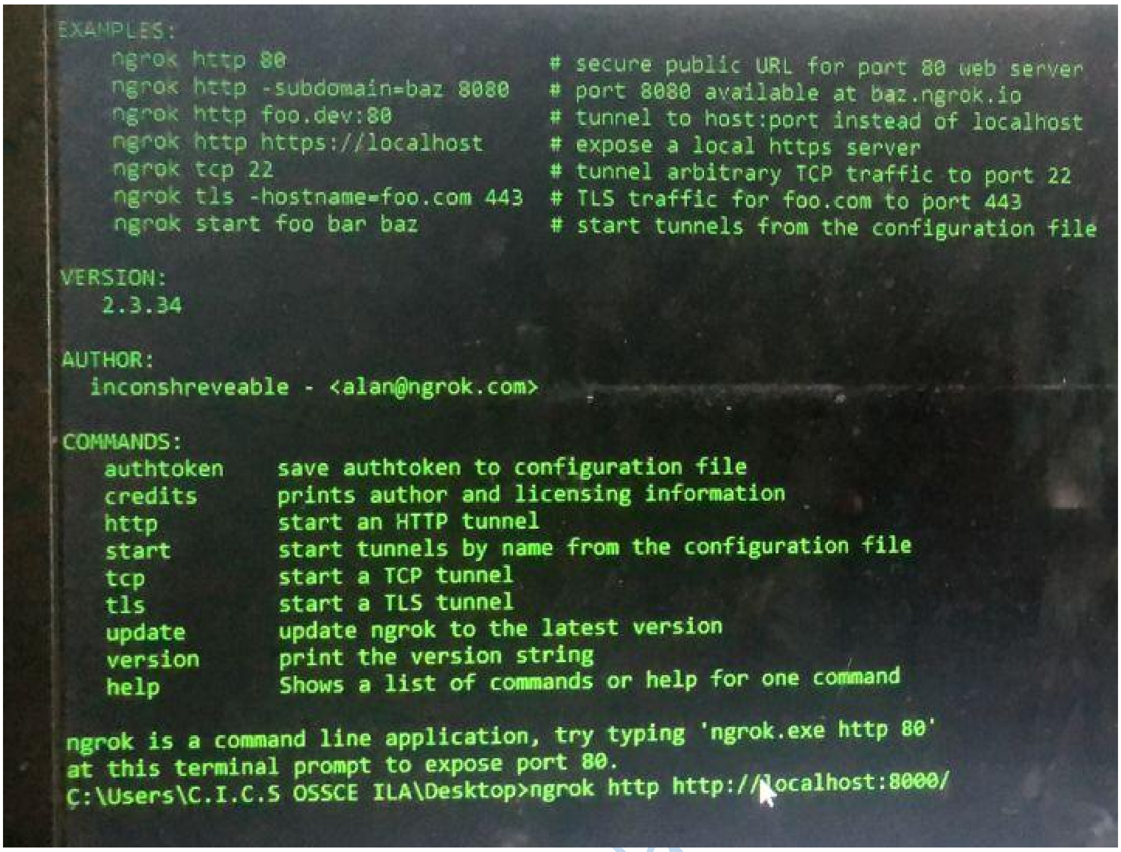


Figure 4.3: Snapshot of a Command Ngrok Http 8000 Written on the Tunnel Server
Source: Research Work, 2023

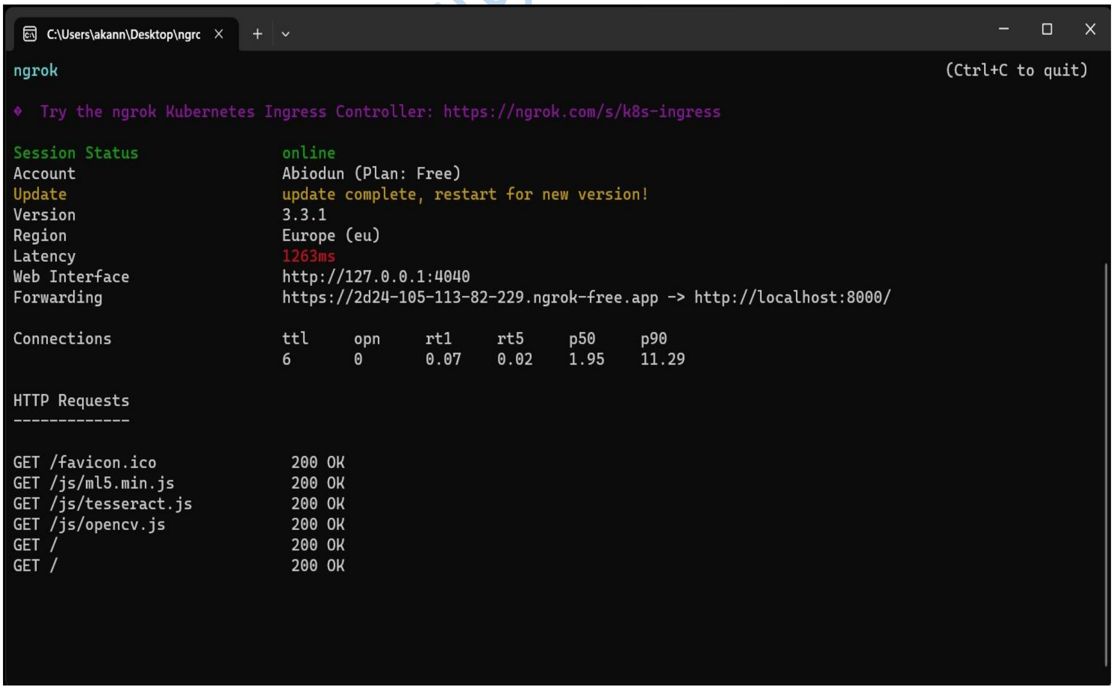


Figure 4.4: Snapshot of The Newly Generated Https That Will Be Copied On The Mobile App
Source: Research Work, 2023

4.1.2 Mobile App

The newly generated secured tunnel i.e http://2d24-185-113-82-229.ngrok-free.app -> http://localhost:8000/ will be copied and carefully typed on the mobile app built as shown in figure 4.5. The HTTPS is HTTP with encryption. The only difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses. As a result, HTTPS is far more secure than HTTP. Also, the prototype design can run only on on a secured protocol.

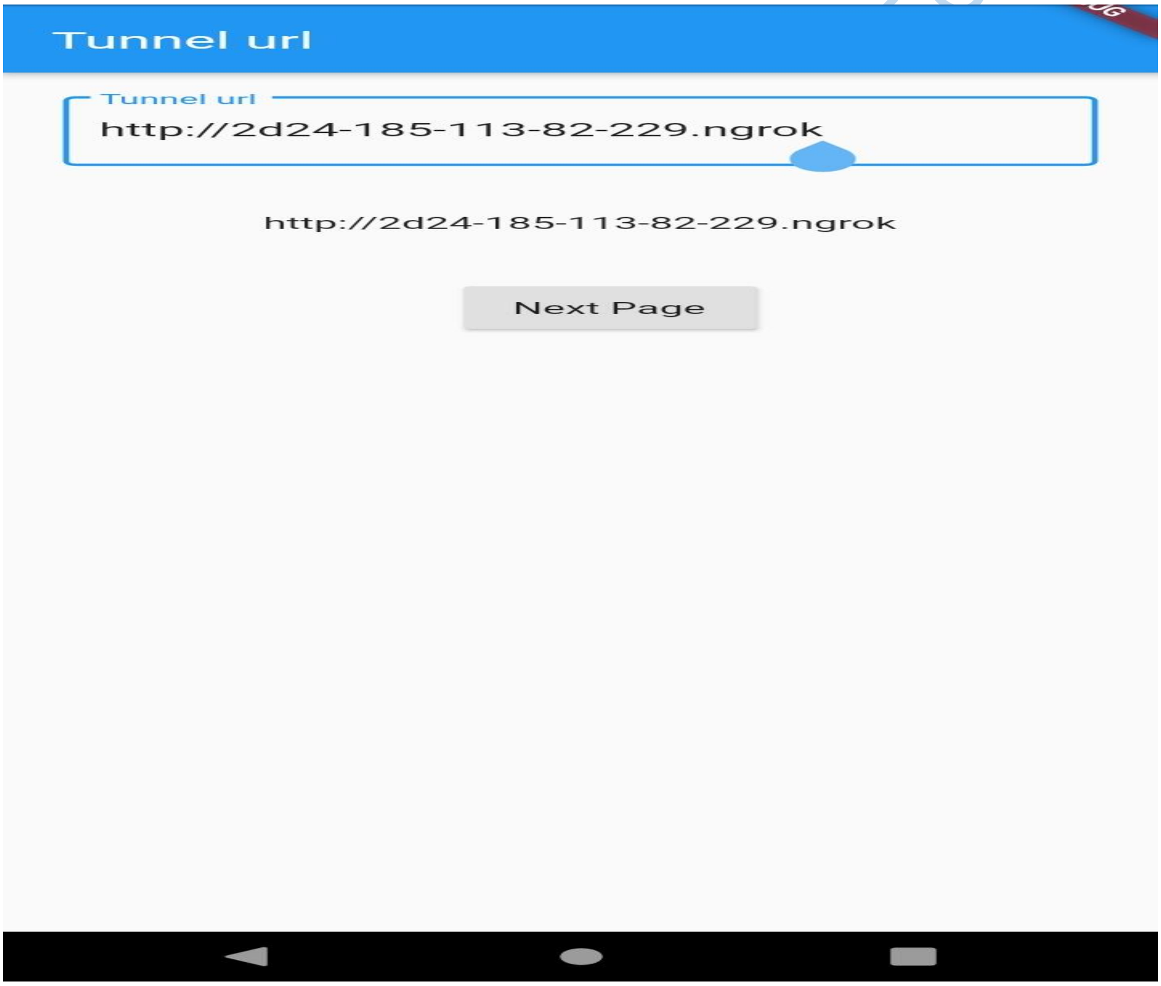


Figure 4.5: Snapshot of Tunnel Url Input Section of the Mobile APP Where the Secured Tunnel URL Will be Copied
Source: Research Work, 2023

4.2 Detection Unit

Immediately the next tab is clicked and processed, it goes to a web browser which is a canvas. The canvas hence opens the link of the detection unit which is where the detection takes place. A web viewer synchronize the canvas (web browser) and mobile app in such a way that the camera of the mobile app can be used to work as the camera for the mobile detection system as shown in figure 4.6.

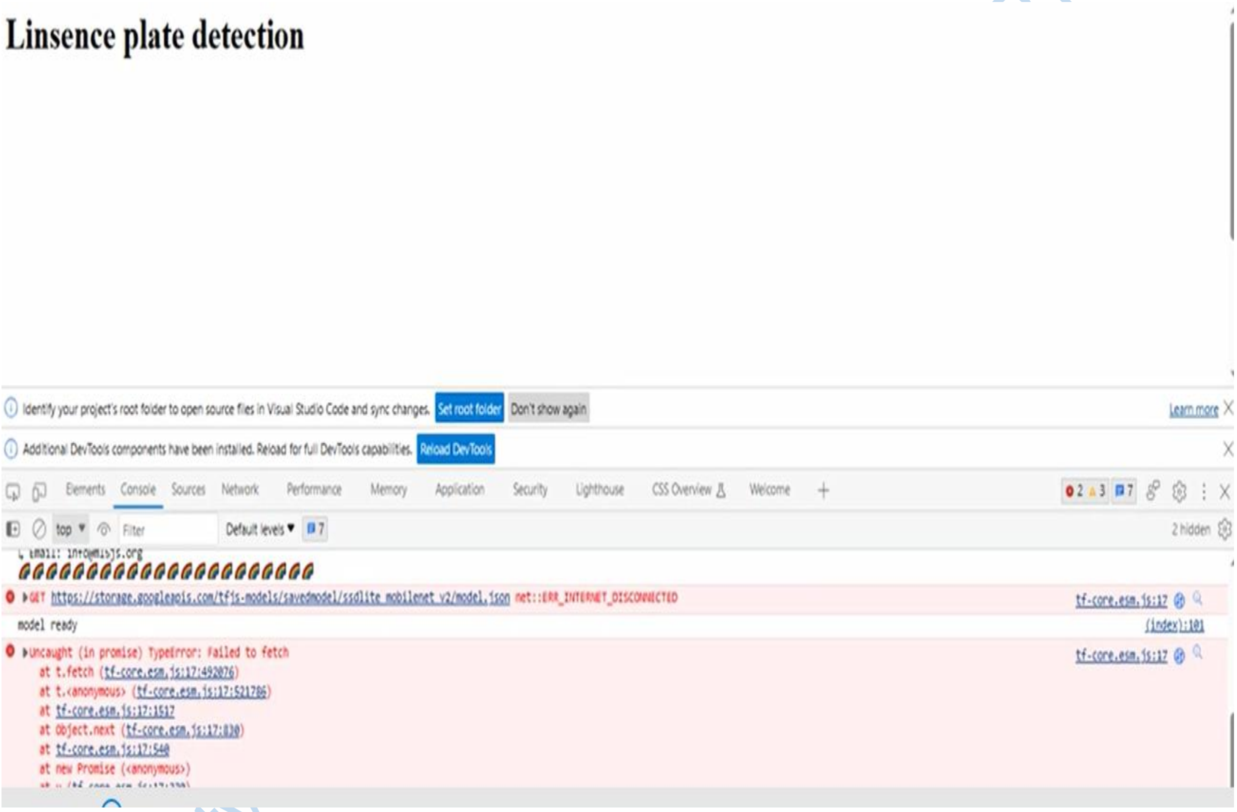


Figure 4.6: License Plate Detection Canvas (Web Browser)
Source: Research Work, 2023

Since we are using a browser on the app, the app is loading a browser in on its own. But since the browser is expected to use a camera, the phone camera now becomes the access camera of the detection unit. Instead of using the web cam of the system, the mobile app accesses the camera of the mobile phone which serves as the camera for the detection unit. The camera is initiated with the following codes. Everything that is seen by the video element is being checked by the ML5.

ML5 was used such that every frame the camera is recording is passed through the library. Where it looks for anything in its library for example, if it's a car, or person, it will identify it. Once the initialization is done, it will be passed to a detect function. In the detect function, the video was passed and draw anytime an object is detected in this case, a car. It crops the car with a green line and identify shown in figure 4.7 and find the number plate, show it on the canvas and use OCR (image to text module). Everything that is seen by the video element is being checked by the ML5.

ML5 is used such that every frame the camera is recording is passed through the library *"Objectdetector = await ml5. objectdetector ('cocossd', startDetecting).* Where it looks for anything in its library for example, if it's a car, or person, it will identify it.

The *'cocossd'* has some predefined object detection module. Once the initialization is done, it will be passed to a detect function. In the detect function, the video was passed and draw anytime an object is detected. A command is written that if a car, bus, truck or any vehicle is passing, it should crop it with a green line and identify if it's a car or truck as shown in figure 4.7 and find the number plate and show it on the canvas and use OCR (image to text module)

Figure 4.7: Object Detection Module Showing Type of Object, Using Detect Function
Source: Research Work, 2023

The number plate was found using the find number plate function (Open.js) which will read the image and convert it to gray scale, edge detection, finding contours to find if there is a number plate there based on the characteristics and others and if there is a number plate, it should give a coordinate of the number plate (x, y position, width and height) and show on the canvas. Here, an image like a car was placed in front of the mobile app, and it starts detecting the type of image, the coordinate of the car. Copied the license plate number and pasted it on the canvas.

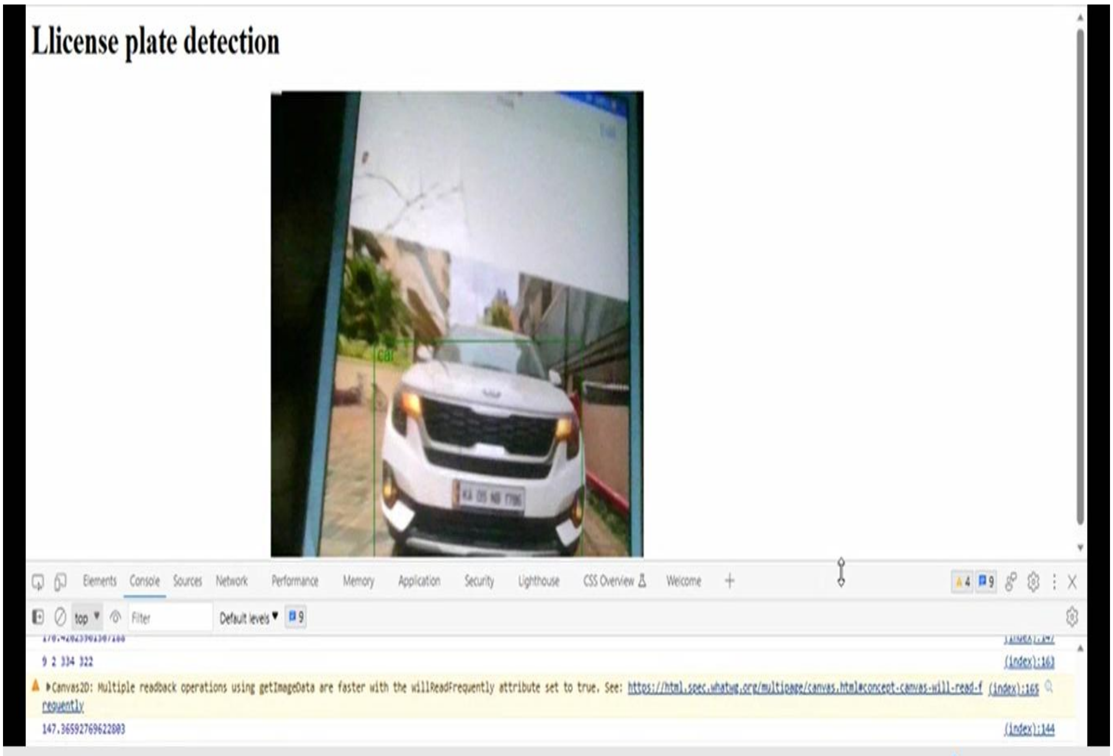


Figure 4.8: Display of Car Plate Number
Source: Research Work, 2023

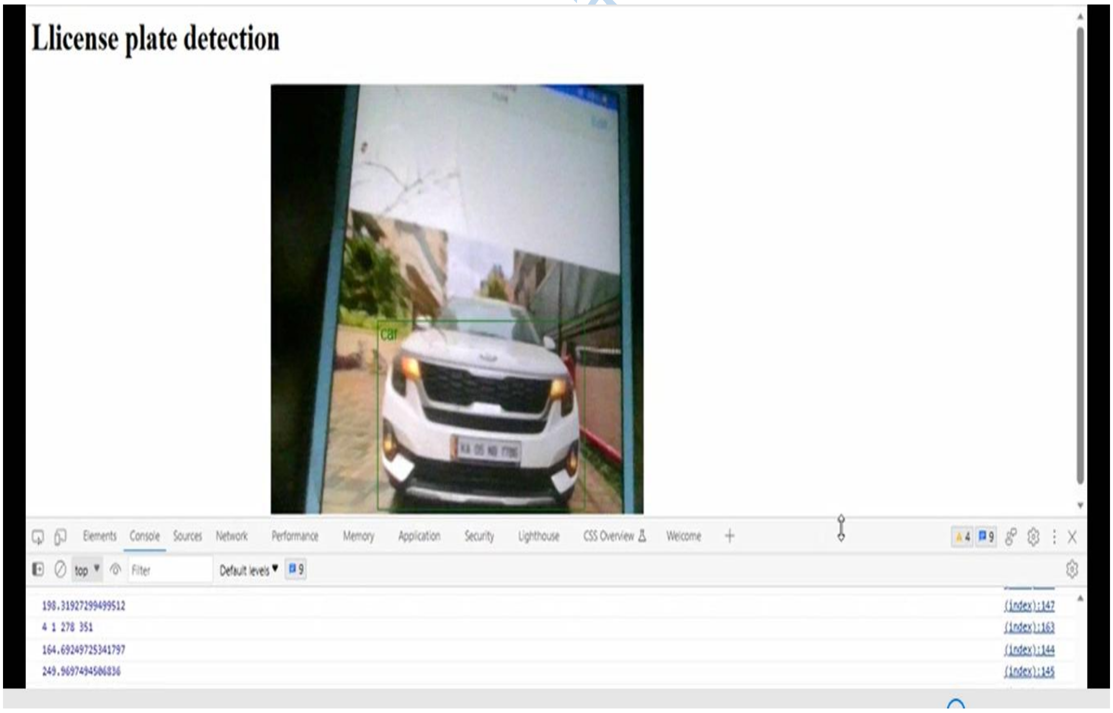


Figure 4.9: Display of Car Plate number Showing Coordinates
Source: Research Work, 2023

Figure 4.9: Display of Car Plate Number Showing x, y Position, Width And Height
Source: Research Work, 2023

Table 4.3: Performance Evaluation Table[2].

S/N	Test Type		Score (%)
1	Performance	v. Response Time	80
		vi. Stability	80
		vii. Load	70
		viii. Reliability	80
2	Usability	i. Easy to Understand	90
		ii. Easy to Access	80
		iii. Faster to Access	80
		iv. Effective Navigation	85
3	Compatibility	i. Software	90
		ii. Hardware	80
		iii. Network	80
		iv. Mobile	80
4	Scalability	i. Throughput	85
		ii. Memory Usage	1.8
		iii. CPU Usage	2
		iv. Network Usage	2

Source: Research Work, 2023

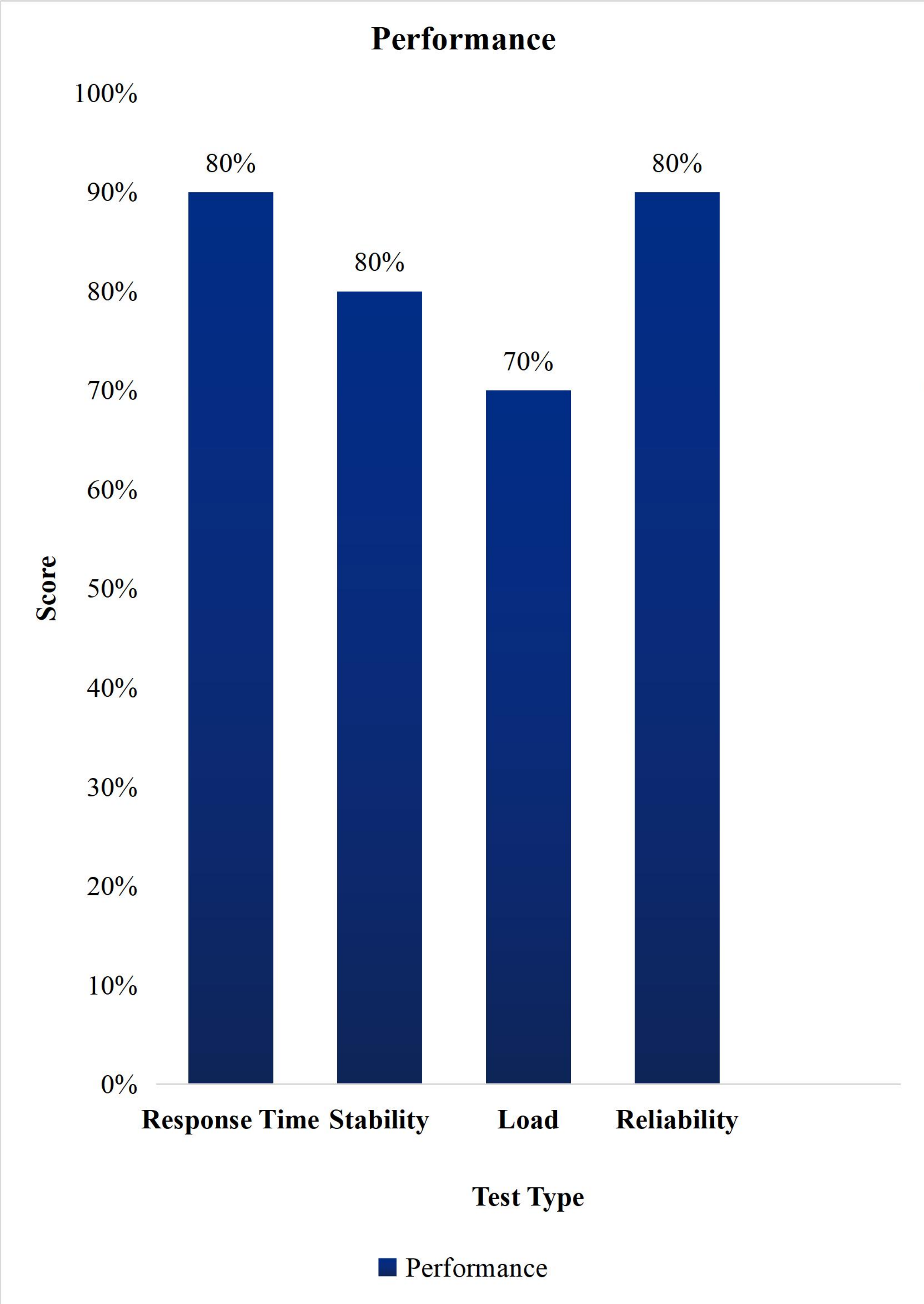


Figure 4.10: Performance Testing Chart of Developed System
Source: Research Work, 2023

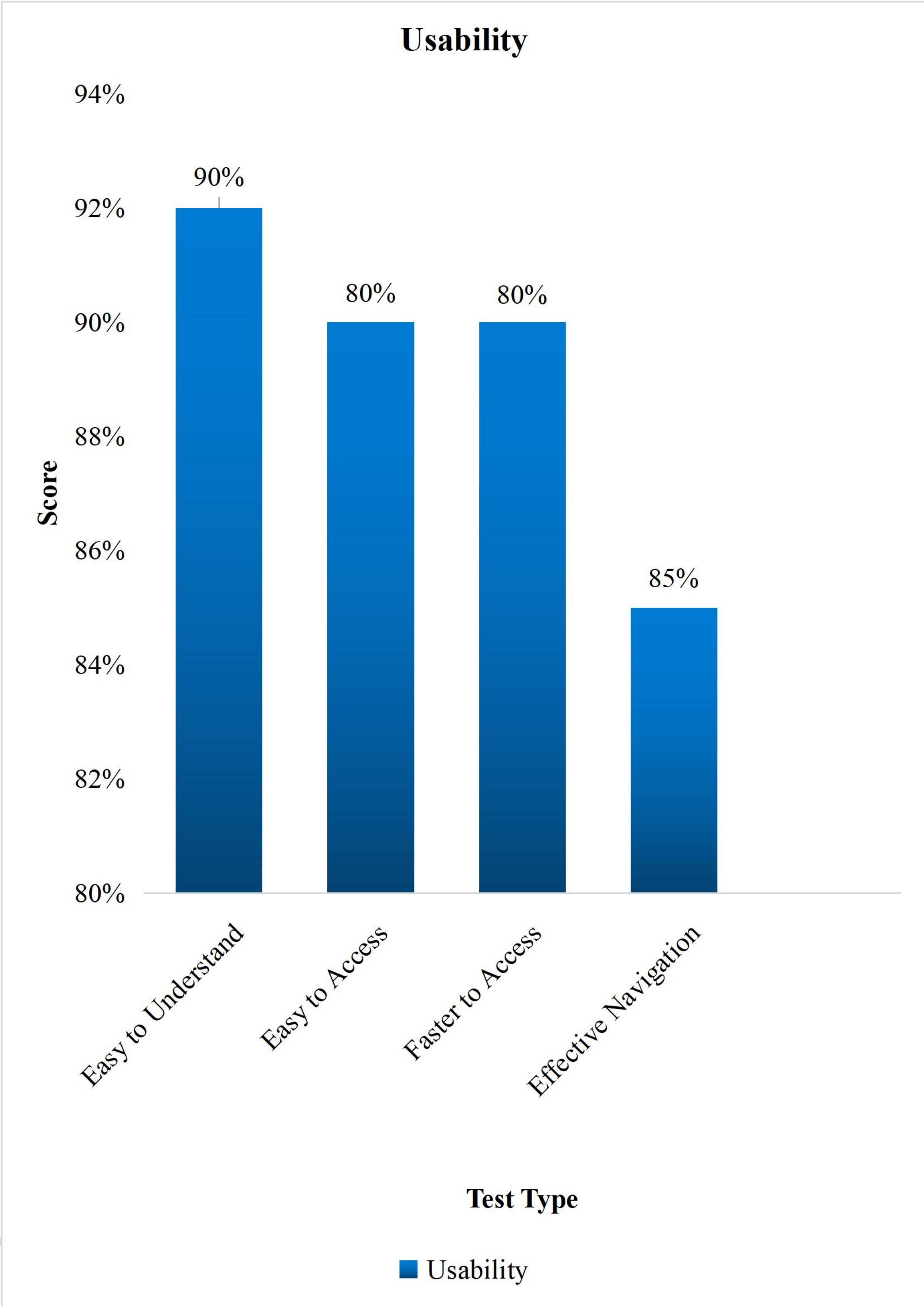


Figure 4.11: Usability Testing Chart of Developed System
Source: Research Work, 2023

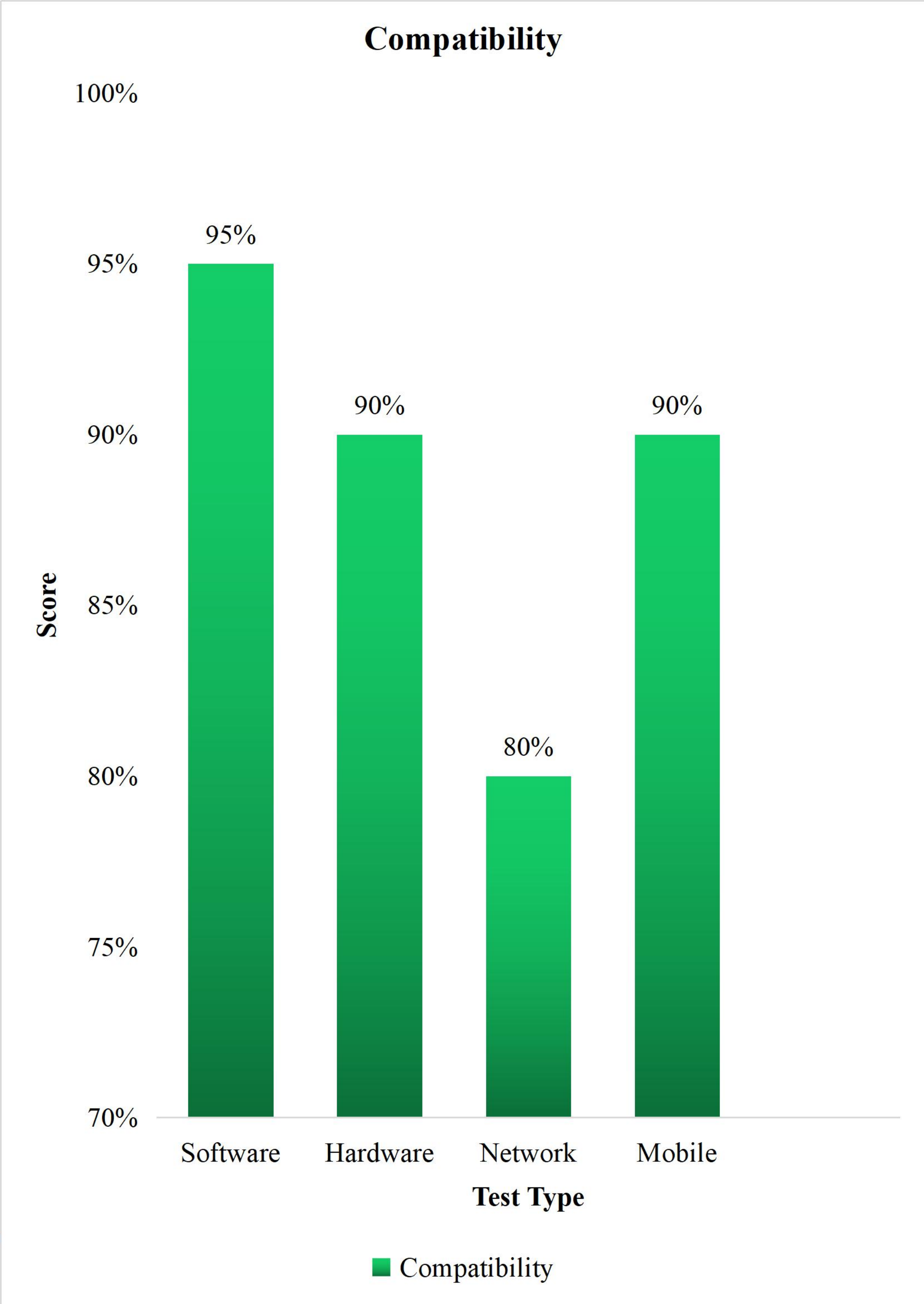


Figure 4.22: Compatibility Test Chart of Developed System
Source: Research Work, 2023

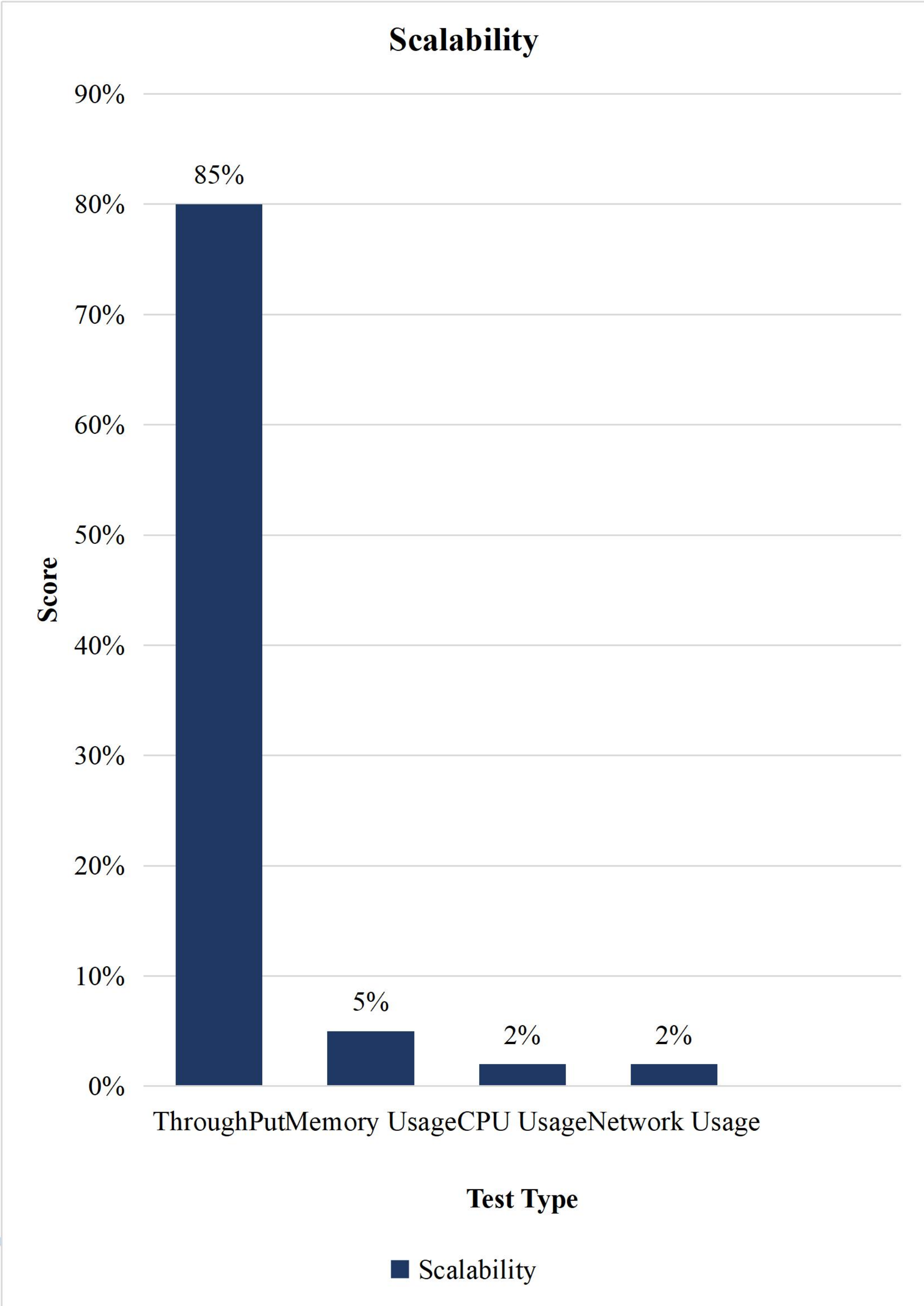


Figure 4.23: Scalability Test Chart of Developed System

4.4 Discussions of Results

The developed system is made consists of a web application enclosed within a mobile application using a tunnelling server. The system that has been developed possesses the capability to capture licence plate numbers through the utilisation of plate number detection on the mobile application, employing the camera module. According to the information provided in Figure 4.2, the tunnelling server was configured to use ngrok with the default settings, specifically for the purpose of exposing a local HTTP server. This was done in order to package the web application into a mobile application. To achieve this, the command "php artisan serve" was executed, resulting in the forwarding of port 80 to port 8000, as depicted in Figure 4.2. Consequently, the server for the developed application was successfully initiated.

Furthermore, as shown in Figure 4.3, a command "ngrok http 8000" was executed to establish an HTTP tunnel, thereby enabling access to the application from the mobile app. This command facilitated the opening of endpoints for both HTTP and HTTPS traffic, as illustrated in Figure 4.4. The URL that was securely obtained was carefully copied within the mobile application for the purpose of collecting, detecting, and analysing using the ML5.js and OpenCV.js frameworks. The ML5 library was utilised to process each frame captured by the camera. The system conducts a search inside its library, enabling it to identify various objects such as cars or individuals. After the completion of the initialization process, the resulting output was afterwards transferred to a detect function. The number plate was identified through the utilisation of the find number plate function (Open.js). This function operates by analysing the image, converting it to grayscale, performing edge detection, and subsequently identifying contours to determine the presence of a number plate based on its distinctive features. If a number plate is detected, the function provides the coordinates (x, y position), as well as the width and height of the number plate. Additionally, the function visually displays the identified number plate on the canvas. In this scenario, a visual representation resembling a car was positioned in the foreground of the mobile application. Subsequently, the application commences the process of identifying the nature of the visual representation, as well as determining the precise location of the automobile inside the image. The number plate number was replicated and afterwards pasted onto the canvas.

Figures 4.10 to 4.12 depict the test chart utilized for the evaluation of the developed system. The chart displays a graphical representation of the relationship between test scores and the corresponding test types. The evaluation was conducted by a professional test engineer in order to assess the performance of the design.

Performance Testing

Response Time: This metric measures the time it takes for a system or application to respond to a specific user action or request. It's an essential metric for user experience, as users generally expect quick responses. A lower response time is usually better. The detection process was about 3-5 seconds which showed a good response time of 80%. However the response time is dependent of the network used for internet.

Stability: Stability assesses the reliability of a system or application over time and under varying conditions. A stable system is one that doesn't crash or produce unexpected errors frequently. High stability is desirable. The design was stable for the load at the response time of 3-5 second. However, this depends on the RAM of the system used. For this study the stability was 80%.

Load: Load refers to the system's ability to handle a specific level of concurrent users or a workload without degrading performance. Systems should be designed to handle the expected load efficiently. A higher load capacity is generally better. The prototype design was tested with 10 mobile apps running at the same time. The goal is to respond immediately with the numbers of users. However, it showed the capacity to be stable with a score of 70%.

Reliability: Reliability measures the system's ability to consistently perform its functions accurately without failures. A reliable system is one that can be trusted to work correctly. The design was reliable at a good network speed with attenuation .Hence a score of 80%

Usability Testing

This is the checking the user-friendliness, efficiency, and accuracy of the design. The design is easy to understand in terms of user interaction. It requires little or no basic training and can be used even if one is not a professional

Easy to Understand: This metric evaluates how easily users can comprehend and make sense of the system or application. A higher score, such as 90, indicates that the system is very easy for users to understand, and its user interface is likely clear and intuitive.

Easy to Access: This metric measures how readily users can access the system or application. A score of 80 suggests that accessing the system is relatively easy, but there might be some room for improvement to make it even more accessible.

Faster to Access: This metric pertains to the speed at which users can access the system or application. A score of 80 indicates that access is reasonably fast, but there might be some room for optimization to make it even quicker.

Effective Navigation: Effective navigation assesses how well users can move through the system or application to find what they need. An 85 suggests that navigation is generally effective, but there could be some enhancements to improve the user's journey.

In this context, higher scores represent better usability. For instance, a score of 90 for "Easy to Understand" implies that users find the system very intuitive and user-friendly in terms of comprehensibility.

These usability metrics and scores are valuable for evaluating the user experience and making improvements to enhance usability further. Usability testing, user feedback, and iterative design processes can help refine the system to ensure it meets or exceeds usability expectations.

Compatibility Testing

Software: The design is compatible with different operating systems, both for forward compatibility and backward compatibility having a score of 95%. Also with different browsers like Google Chrome, Firefox, and Internet Explorer[1].

Win 7 → Win 8 → Win 8.1 → Win 10 (Forward Compatibility)

Window XP → Vista → Win 7 → Win 8 → Win 8.1 (Backward Compatibility).

Hardware: The design is compatible with hardware of different sizes such as RAM, hard disk, processor, and the graphic card with a score of 80%

Mobile: The design is compatible with mobile platforms such as iOS and Android. The mobile app for the design does not run on all Android phones less than Android 9, runs efficiently on iOS and all global phones. Thus having a score of 80%

Network: Compatibility with different network parameters such as operating speed, bandwidth, and capacity with a score of 80%.

Scalability

Scalability metrics assess a system's ability to handle increased load and demand. Throughput: Throughput measures the rate at which a system can process a certain number of tasks or transactions per unit of time. An 85 suggests that the system has good throughput and can handle a significant volume of tasks efficiently.

Memory Usage: Memory usage assesses how much system memory (RAM) is consumed during operation. A score of 1.6 indicates that memory usage is relatively low, which is generally a positive characteristic as it allows the system to operate without consuming excessive memory resources.

CPU Usage: CPU usage measures the amount of central processing unit (CPU) resources used during system operation. A score of 2 suggests that the system's CPU usage is moderate, indicating that it doesn't overly tax the CPU.

Network Usage: Network usage assesses the amount of network resources consumed during operation, including data transfer and communication. A score of 2 implies that network usage is also moderate, indicating efficient use of network resources. "Throughput" represent better performance in terms of processing tasks efficiently. Lower scores for "Memory Usage," "CPU Usage," and "Network Usage" are generally positive, as they indicate efficient resource utilization without overloading memory, CPU, or network capacities. These scalability metrics and scores are valuable for understanding how well the system can handle increased loads and demands. Maintaining good throughput while efficiently managing memory, CPU, and network resources is essential for a scalable and high-performance system.

ThroughPut: The design has a Throughput of 80%, because it can send several images in a second effectively and efficiently.

Memory Usage: The application used for the design (Ngrok, windows command processor and Php CLI used a memory of about 0.9%, 0.7% 0.2% as shown in figure 4.23. However, CPU and Network Usage of the application for the design is at 0%

out of the total 8% of the entire CPU usage and as shown in figure 4.3. This shows that the CPU and network utilization of the design is very low.

Endnotes

1. https://www.javatpoint.com

2. R A Badru, A A Waheed, O A Akinmoluwa, O R Obayemi. *Generation of Surveillance Networked Nodes for Oil Pipelines' Theft* **International Journal of Recent Engineering Science**, 8(5), 21-26.

Chapter Five

Conclusion

5.1 Summary of Results

The developed system was designed to capture license plate numbers using plate number detection on a mobile application with a camera module. A tunnelling server, configured with ngrok, was used to expose a local HTTP server to package the web application into the mobile application. The server for the application was successfully initiated, enabling communication between the web and mobile components.ML5.js and OpenCV.js frameworks were used to process images captured by the camera. ML5 library was used to process frames and identify objects such as cars or individuals. A "find number plate" function in OpenCV.js analyzed the image, converted it to grayscale, performed edge detection, and identified contours to detect number plates. If a number plate was detected, its coordinates and visual representation were displayed on the canvas.

Response time was measured and found to be about 3-5 seconds, indicating a good response time of 80%. However, it was noted that response time depends on the network used. Stability was tested and found to be stable with a score of 80%, although it depends on the system's RAM. Load testing with 10 concurrent mobile apps showed that the prototype design was stable with a score of 70%. Reliability was rated at 80% when tested under good network conditions. Also, the system was found to be easy to understand (90%) and easy to access (80%). Faster access received a score of 80%, and navigation was rated at 85% effectiveness.

Further, the system was compatible with various operating systems and browsers, achieving a score of 95% for software compatibility. It was also compatible with different hardware configurations (80%) Mobile compatibility was established for iOS and Android platforms, with some restrictions on Android versions (80%). Network compatibility included parameters like operating speed, bandwidth, and capacity, with a score of 80%. The Throughput received a score of 85%, indicating good performance in processing tasks. Memory usage was relatively low, with a score of 1.6. CPU and network usage were moderate, with scores of 2 each. The developed system effectively captures license plate numbers and performs well in terms of response time, stability, and usability. It exhibits good compatibility and scalability. However, it's important to note that some performance aspects may depend on network conditions and system resources.

5.2 Recommendations

The developed license plate detection system represents a significant technological achievement in the intersection of web and mobile applications, computer vision, and artificial intelligence. This system successfully captures license plate numbers using a mobile application integrated with web technologies and sophisticated image processing frameworks like ML5.js and OpenCV.js. The system demonstrated commendable performance. Response times, though dependent on network conditions, averaged 3-5 seconds, achieving a response time score of 80%. Stability and reliability were also strong points, with scores of 80%. However, it's worth noting that stability could be influenced by system RAM, and response time might vary based on internet speed. The system received high usability ratings. It was considered easy to understand (90%), easy to access (80%), and had efficient navigation (85%). This indicates that users found the system intuitive and user-friendly. The system exhibited excellent compatibility, supporting various operating systems, browsers, hardware configurations, and mobile platforms. The compatibility score for software compatibility was an impressive 95%. Scalability metrics showed promise.

Throughput was rated at 85%, indicating efficient task processing. Memory usage was low (1.6), and CPU and network usage were moderate (2), reflecting efficient resource utilization. The developed license plate detection system represents a significant technological achievement with promising performance and usability. With further refinement and expansion of capabilities, it holds the potential to contribute significantly to various applications, including security, transportation, and automation.

Based on the findings and the conclusion drawn from the evaluation of the developed license plate detection system, the following recommendations are proposed:

1. Network Optimization: Given the system's sensitivity to network conditions, it's advisable to explore strategies for network optimization. This includes implementing mechanisms to handle low bandwidth and high-latency situations more gracefully. Additionally, consider providing offline functionality for scenarios where network connectivity is unreliable or unavailable.
2. Resource Efficiency: While the system's memory and CPU usage are relatively low, ongoing efforts to optimize resource utilization should continue. This can involve refining algorithms, compressing data, or implementing more efficient coding practices. Resource-efficient design not only enhances performance but also extends the system's compatibility to devices with varying resource constraints.
3. Compatibility Testing: Continue rigorous compatibility testing to ensure that the system remains compatible with a wide range of operating systems, browsers, hardware configurations, and mobile platforms. Regular updates and testing can help address compatibility issues as new technologies and devices emerge.

4. Object Recognition Expansion: Expanding the system's capabilities beyond license plate detection. Explore opportunities to incorporate object recognition for various applications, such as identifying vehicles, traffic signs, or security threats. Expanding the system's object recognition capabilities can increase its versatility and value.
5. Testing and Quality Assurance: Maintain a robust testing and quality assurance process to identify and rectify any issues promptly. Regularly test the system under various conditions to ensure its stability, reliability, and performance.
6. Documentation: Continuously update and improve system documentation. Clear and comprehensive documentation aids users and developers in understanding, configuring, and troubleshooting the system effectively.

These recommendations aim to further enhance the functionality, usability, and reliability of the license plate detection system, ensuring its continued effectiveness in real-world scenarios and its ability to meet evolving user needs.

5.3 Contribution to Knowledge

The developed license plate detection system contributes to academic knowledge in several key areas: The system leverages computer vision techniques and image processing algorithms to detect and analyze license plates in real-time. This contribution enhances the field's understanding of practical applications for computer vision, particularly in the domain of vehicle recognition and surveillance. The system also integrates machine learning models, specifically ML5.js and OpenCV.js, to improve the accuracy of license plate detection. This demonstrates the synergy between machine learning and computer vision, showcasing their combined potential in real-world scenarios.

The fusion of a web application within a mobile application, as described in the system, provides insights into mobile app development practices. It contributes to the knowledge of mobile app architecture, deployment, and usability. The system's use of a tunnelling server, ngrok, and port forwarding for web application packaging highlights network and server configuration techniques. This knowledge is valuable for those interested in server-side development and networking. The usability testing metrics and scores provide valuable insights into the user-friendliness and efficiency of the system. This contributes to the field of human-computer interaction (HCI) and usability engineering. The system's potential for real-world applications, such as license plate recognition for law enforcement or surveillance, demonstrates the practicality of computer vision and machine learning technologies in enhancing security and efficiency.

The developed license plate detection system contributes to academic knowledge by bridging various domains, including computer vision, machine learning, mobile app development, network configuration, usability testing, and ethical considerations. Its real-world applicability and performance evaluation metrics offer valuable insights and serve as a foundation for further research and advancements in these areas.

To further enhance the system's capabilities and real-world applicability, future research and development efforts could focus on the following areas:

Enhanced Accuracy with Deep Learning: Investigate the use of deep learning techniques, such as Convolutional Neural Networks (CNNs), for license plate detection. These models have shown remarkable accuracy in image recognition tasks and could further improve the precision of plate detection.

Real-time Processing: Explore methods to optimize license plate detection for real-time processing. This includes reducing latency in image capture, analysis, and response, which is crucial for applications like traffic monitoring and toll collection.

Multi-Language Support: Extend the system's capabilities to detect license plates with characters from multiple languages and scripts. This would be particularly valuable for international applications and could involve training models on diverse character sets.

Low-Light and Adverse Conditions: Research how the system can perform under challenging conditions, such as low-light environments, adverse weather conditions, or when license plates are partially obscured. Develop algorithms that can adapt to these scenarios.

Integrate OCR for Text Recognition: Extend the system to include Optical Character Recognition (OCR) to extract and interpret the text on license plates. This could enable applications like vehicle tracking and parking management.

Mobile App Enhancements: Improve the mobile application's user interface and accessibility features. Consider incorporating augmented reality (AR) elements to enhance user experience.

5.4 Limitation of Research

While the developed license plate detection system demonstrates significant potential and functionality, it is essential to acknowledge its limitations. The system's reliance on network connectivity can be a limitation, particularly in areas with poor or unreliable internet connections. Although the system's memory and CPU usage is relatively low, it may still face challenges on devices with limited resources. Also, the accuracy of the license plate detection heavily relies on the underlying machine learning models (ML5.js and OpenCV.js). If these models are not updated or retrained regularly, the system's ability to detect license plates accurately may decrease over time. The system's primary focus is license plate detection. While it can identify other objects, it may not be as accurate or versatile in recognizing a wide range of objects or scenarios. Expanding its object recognition capabilities may require additional development efforts.

Biblography

Conference Proceedings

Ap NP, Vigneshwaran T, Arappradhan MS, Madhanraj R. *Automatic number plate detection in vehicles using faster R-CNN.* In2020 International Conference on System, Computation, Automation and Networking (ICSCAN) 2020 Jul 3 (pp. 1-6). IEEE.

Arora P, Kapse VM, Sinha S, Gera S. Number *plate recognition system Using Convolutional Neural Network.* In2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) 2021 Sep 3 (pp. 1-5). IEEE.

Bochinski E, Eiselein V, Sikora T. *High-speed tracking-by-detection without using image information.* In2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS) 2017 Aug 29 (pp. 1-6). IEEE.

Darapaneni N, Mogeraya K, Mandal S, Narayanan A, Siva P, Paduri AR, Khan F, Agadi PM. *Computer vision based license plate detection for automated vehicle parking management system.* In2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) 2020 Oct 28 (pp. 0800-0805). IEEE.

Dhar P, Guha S, Biswas T, Abedin MZ. *A system design for license plate recognition by using edge detection and convolution neural network.* In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) 2018 Feb 8 (pp. 1-4). IEEE.

Ditcharoen, Chhour B, Traikunwaranon T, Aphivongpanya N, Maneerat K, Ammarapala V. *Road traffic accidents severity factors: A review paper.* In 2018 5th International Conference on Business and Industrial Research (ICBIR) 2018 May 17 (pp. 339-343). IEEE.

Gharraf HS, Cansever G, Ahmed AS. *Image filtering of impulsive noise using biologically inspired algorithms.* In2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2022 Oct 20 (pp. 58-65). IEEE.

Joseph J, Prasad A & Jithina LM. *A study on localization techniques for automatic license plate recognition system*. In2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) 2019 Jul 5 (Vol. 1, pp. 1090-1094). IEEE.

Kukreja V, Kumar D, Kaur A. *GAN-based synthetic data augmentation for increased CNN performance in vehicle number plate recognition.* In2020 4th international conference on electronics, communication and aerospace technology (ICECA) 2020 Nov 5 (pp. 1190-1195). IEE

Kulkarni, Bodkhe S, AKamthe, Patil A. *Automatic number plate recognition for motorcyclists riding without helmet.* In2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) 2018 Mar 1 (pp. 1-6). IEEE.

Kumar K, Sinha S, Manupriya P. D-*PNR: deep license plate number recognition.* InProceedings of 2nd International Conference on Computer Vision & Image Processing 2018 (pp. 37-46). Springer, Singapore.

Laroca R, Severo E, Zanlorensi LA, Oliveira LS, Gonçalves GR, Schwartz WR, Menotti D. *A robust real-time automatic license plate recognition based on the YOLO detector.* In2018 international joint conference on neural networks (IJCNN) 2018 Jul 8 (pp. 1-10). IEEE.

M Kročka, P Dakić, V Vranić. *Automatic License Plate Recognition Using Open CV.* In2022 12th International Conference on Advanced Computer Information Technologies (ACIT) 2022 Sep 26 (pp. 530-535). IEEE.

Menon, B Omman. *Detection and recognition of multiple license plate from still images.* In2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET) 2018 Dec 21 (pp. 1-5). IEEE.

Mindula B, Ranasinghe M, Ahamed R, Tennakoon J, Silva C, Wimalarathne G. *Image and video processing based expressway traffic rules violation detection.* In2021 8th International Conference on ICT & Accessibility (ICTA) 2021 Dec 8 (pp. 1-6). IEEE.

Niluckshini NR, Firdhous MF. *Automatic number plate detection using Haar-Cascade Algorithm proposed for Srilankan context.* In2022 2nd International Conference on Advanced Research in Computing (ICARC) 2022 Feb 23 (pp. 248-253). IEEE.

Raj S, Gupta Y, Malhotra R. *License plate recognition system using Yolov5 and CNN.* In2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) 2022 Mar 25 (Vol. 1, pp. 372-377). IEEE.

Salman YD, Alhadawi HS, Mahdi AS, AL-Dhief FT. *Improved automatic license plate recognition system in Iraq for surveillance system using OCR.* InInternational Conference on Emerging Technologies and Intelligent Systems 2023 (pp. 270-277). Springer, Cham.

Sheng KB, Saad AA, MK Ishak. *Development of a virtual vehicle identification for tracking hit-and-run vehicle.* In2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) 2022 Sep 13 (pp. 1-6). IEEE.

Slimani I, Zaarane A, Hamdoun A, Atouf I. *Vehicle license plate localization and recognition system for intelligent transportation applications.* In2019 6th International Conference on Control, Decision and Information Technologies (CoDIT) 2019 Apr 23 (pp. 1592-1597). IEEE.

Tarhouni W, Abdo A, ELmegreisi A. *Feature fusion using the local binary pattern histogram fourier and the pyramid histogram of feature fusion using the local binary pattern oriented gradient in iris recognition.* In2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA 2021 May 25 (pp. 853-857). IEEE.

Titus AN. *Vehicle license plate localization based on local binary pattern features.* In2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC) 2019 Mar 7 (pp. 1-5). IEEE.

Wu X, Wei Z, Hu Y, Wang K. *Traffic sign detection method using multi-color space fusion.* In2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) 2020 Jun 27 (pp. 314-319). IEEE.

Zhuang J, Hou S, Wang Z, Zha ZJ. *Towards human-level license plate recognition.* InProceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 306-321)

Journals

Abd Gani SF, Miskon MF, Hamzah RA, N Mohamood, Z Manap, MF Zulkifli. *A live-video automatic number plate recognition (anpr) system using convolutional neural network (CNN) with data labelling on an android smartphone.*

Agbemenu AS, Yankey J, Addo EO. *An automatic number plate recognition system using opencv and tesseract ocr engine.* International Journal of Computer Applications. 2018 May;180(43):1-5.

Aghiomesi, Irunokhai & O., J. & S., O. & O., B. *Analysis of traffic light violation on Nigerian roads (a case study of Sango T junction, Ibadan, Oyo State).* International Journal of Computer Applications. 176. 8-13. 10.5120/ijca2020920299. (2020)

Al Awaimri M, Fageeri, Moyaid MA, Thron C, ALhasanat A. *Automatic number plate recognition system for Oman.* InArtificial Intelligence for Data Science in Theory and Practice 2022 (pp. 155-178). Springer, Cham.

Alhussein M, Aurangzeb K, Haider SI. *Vehicle license plate detection and perspective rectification.* Elektronika ir Elektrotechnika. 2019 Oct 6;25(5):47-56.

Altinsoy E, Yang J, Yilmaz C. *Fully-automatic raw G-band chromosome image segmentation.* IET Image Processing. 2020 Jul;14(9):1920-8.

Antar R, Alghamdi S, AlotaibiJ, M Alghamdi. *Automatic Number Plate Recognition of Saudi License Car Plates.* Engineering, Technology & Applied Science Research. 2022 Apr 9;12(2):8266-72.

Arafat MY, Khairuddin AS, Khairuddin U, Paramesran R. *Systematic review on vehicular licence plate recognition framework in intelligent transport systems.* IET Intelligent Transport Systems. 2019 May;13(5):745-55.

Awel MA, Abidi. *Review on optical character recognition.* International Research Journal of Engineering and Technology (IRJET). 2019 Jun;6(6):3666-9.

Badru RA, Waheed AA, Akinmoluwa OA, Obayemi OR. *Generation of surveillance networked nodes for oil pipelines' theft International Journal of Recent Engineering Science*, 8(5), 21-26.

Balid W, Refai H.H. *Real-time magnetic length-based vehicle classification: Case study for inductive loops and wireless magnetometer sensors in Oklahoma state*. Transportation Research Record. 2018 Dec;2672(19):102-11.

Barthélemy J, Verstaevel N, Forehead H, Perez P. *Edge-computing video analytics for real-time traffic monitoring in a smart city.* Sensors. 2019 Jan;19(9):2048

Bennet MA, Thamilvalluvan B, Alphonse PP, Thendralarasi DR, Sujithra KJ. *Performance and analysis of automatic license plate localization and recognition from video sequences*. International Journal on Smart Sensing and Intelligent Systems. 2017 Dec 1;10(5):330.

Chhabra S, Saini S, Lata K. *Hardware–software co-simulation of vehicle license plate detection on the ZedBoard SoC platform.* In Advances in Image and Data Processing using VLSI Design, Volume 1: Smart vision systems 2021 Dec 1. IOP Publishing.

Davix XA, Christopher CS. *Edge based marker controlled watershed algorithm for automatic car licence plate localization.* Journal of Computational and Theoretical Nanoscience. 2017 Nov 1;14(11):5539-51.

Daway HG, Daway EG, Kareem HH. *Colour image enhancement by fuzzy logic based on sigmoid membership function.* International Journal of Intelligent Engineering and Systems. 2020;13(5):238-46.

Denny K, Denny J, Satheesh A & Mets M. *Recent Study on Vehicle Licence Plate Detection Based on R-CNN.* IJIRIS: International Journal of Innovative Research in Information Security, Volume VII, 53-56, 2020

Dong M, He H, Luo C, Liu D, Zeng W. *A CNN-Based approach for automatic license plate recognition in the wild*. InBMVC 2017 Sep.

Gorev AE, Gasilova O, Sidorov BA. *Surface transportation engineering technology: Prerequisite for accident-free traffic at signal-controlled intersections.* Architecture and Engineering. 2021;6(1):73-80.

Hamdoun N, Mentagui D. *Image processing in automatic license plate recognition using combined methods.* Serdica Journal of Computing. 2022 Jul 4;16(1):1-23.

Hossen MK, Roy AC, Chowdhury MS, Islam MS, Deb K. *License plate detection and recognition system based on morphological approach and feed-forward neural network.* IJCSNS International Journal of Computer Science and Network Security. 2018 May 30;18(5):36-45.

Huang Q, Cai Z, Lan T. *A new approach for character recognition of multi-style vehicle license plates*. IEEE Transactions on multimedia. 2020 Oct 14;23:3768-77.

Huang Q, Cai Z, Lan T. *A single neural network for mixed style license plate detection and recognition*. IEEE Access. 2021 Jan 28;9:21777-85.

Ibiyemi TS, Owotogbe JS, Adu. BA *a comparative study of vehicle number plate recognition systems.* African Journal of Management Information System. 2020;2(1):10-23.

Jain T, Verma VK, Garg P, Jangid M. *An improvised model for high-security license plate detection and recognition for indian vehicle to enhance detection*. Computational Network Application Tools for Performance Management. 2019 Oct 18:109.

Kessentini Y, Besbes MD, Ammar S, Chabbouh A. *A two-stage deep neural network for multi-norm license plate detection and recognition.* Expert systems with applications. 2019 Dec 1;136:159-70.

Khan FN, Fan Q, Lu C, Lau AP. *Machine learning methods for optical communication systems and networks.* InOptical fiber telecommunications VII 2020 Jan 1 (pp. 921-978). Academic Press.

Khan M, Imran A Rehman,GX A Fazil, M Zakwan, Z Mahmood. *Performance enhancement method for multiple license plate recognition in challenging environments.* EURASIP Journal on Image and Video Processing. 2021 Dec;2021(1):1-23.

Kuchuk H, Podorozhniak A, N, Onischenko D. *System of license plate recognition considering large camera shooting angles.* Radioelectronic and Computer Systems. 2021 Nov 29(4):82-91.

Laroca R, Zanlorensi LA, Gonçalves GR, Todt E, Schwartz WR, D Menotti. *An efficient and layout-independent automatic license plate recognition system based on the YOLO detector.* IET Intelligent Transport Systems. 2021 Apr;15(4):483-503.

Laroca R, Zanlorensi LA, Gonçalves GR, Todt E, Schwartz WR, Menotti D. *An efficient and layout-independent automatic license plate recognition system based on the YOLO detector.* IET Intelligent Transport Systems. 2021 Apr;15(4):483-503.

Lewandowski M, Płaczek B, Bernas M, Szymała P. *Road traffic monitoring system based on mobile devices and bluetooth low energy beacons.* Wireless communications and mobile computing. 2018 Jul 17;2018.

Lin CJ, Chuang CC, Lin HY. *Edge-AI--based real-time automated license plate recognition system.* Applied Sciences. 2022 Jan 28;12(3):1445.

Liu W, Chen J, Luo Y, Shi Z, Ji X, Zhu H. *Study on the annual reduction rate of vehicle emission factors for carbon monoxide: A case study of urban road tunnels in Shenzhen, China.* Advances in Civil Engineering. 2020 Sep 3;2020.

Mafi M, Rajaei H, Cabrerizo M, Adjouadi M. *A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering.* IEEE Transactions on Image Processing. 2018 Jul 18;27(11):5475-90.

Manaa K, Rabee'a M, Khalaf. *Traffic control by digital imaging cameras. InEmerging Trends in Image Processing*, Computer Vision and Pattern Recognition 2015 Jan 1 (pp. 231-247). Morgan Kaufmann.

Mellinda PS, Sthevanie F, Ramadhani KN. *Detection of vehicle number plate using probabilistic hough transform.* eProceedings of Engineering. 2020 Aug 1;7(2).

Mufti N, Shah SA. *Automatic number plate recognition: A detailed survey of relevant algorithms.* Sensors. 2021 Apr 26;21(9):3028.

Nguyen H. *Real-time license plate detection based on vehicle region and text detection.* Journal of Theoretical and Applied Information Technology. 2020 Feb 15;98(03).

Nguyen-Xuan H, Hoang-Nhu D, Kim-Van, Dang-Minh K. *A new system for license plate recognition in traffic violation scenarios in Vietnam.* In Intelligent Systems and Networks 2022 (pp. 287-297). Springer, Singapore.

Okafor, C Ajaero, Madu C Agomuo, E Abu. *Implementation of circular economy principles in management of end-of-life tyres in a developing country (Nigeria).* AIMS Environ Sci. 2020 Oct 12;7:406-33

Padmasiri BE. Person detection and tracking using omnidirectional cameras, and rectangle blanket problem.*2019.*

Pang J, Pu X, Li C. *A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection.* IEEE Transactions on Industrial Informatics. 2022 Jan 25.

Pareek J, Singhania D, Kumari RR, Purohit S. *Gujarati handwritten character recognition from text images.* Procedia Computer Science. 2020 Jan 1;171:514-23.

Peter R, Grosselfinger AK, Münch D, Arens M. *Automated license plate detection for image anonymization.* InCounterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III 2019 Oct 7 (Vol. 11166, pp. 218-231). SPIE.

Pham TA. *Effective deep neural networks for license plate detection and recognition*. The Visual Computer. 2022 Jan 21:1-5.

Priyadharsini J. *An experiment analysis on tracking and detecting the vehicle speed using machine learning and iot.* In2021 Smart Technologies, Communication and Robotics (STCR) 2021 Oct 9 (pp. 1-5). IEEE

Rafique MA, Pedrycz W, Jeon M. *Vehicle license plate detection using region-based convolutional neural networks.* Soft Computing. 2018 Oct;22(19):6429-40.

Rani P, Kotwal S, Manhas J, Sharma V, Sharma S. *Machine learning* and deep learning based computational approaches in *automatic microorganisms image recognition: methodologies, challenges, and developments.* Archives of Computational Methods in Engineering. 2022 May;29(3):1801-37.

Rhee DM, Lombardo FT, Kadowaki J. *Semi-automated tree-fall pattern identification using image processing technique: Application to Alonsa, MB tornado.* Journal of Wind Engineering and Industrial Aerodynamics. 2021 Jan 1;208:104399.

Selmi Z, Halima MB, Pal U, Alimi MA. *DELP-DAR system for license plate detection and recognition.* Pattern Recognition Letters. 2020 Jan 1;129:213-23.

Shafi, Hussain I, Ahmad J, Kim PW, Choi G, Ashraf I, Din S. *License plate identification and recognition in a non-standard environment using neural pattern matching*. Complex & Intelligent Systems. 2022 Oct;8(5):3627-39

Sharma G. *Performance analysis of vehicle number plate recognition system using template matching techniques.* Journal of Information Technology & Software Engineering. 2018 Apr;8(2):1-9

Shashirangana J, Padmasiri H, Meedeniya D, Perera C. *Automated license plate recognition: a survey on methods and techniques.* IEEE Access. 2020 Dec 29;9:11203-25.

Spagnolo F, Perri S, Corsonello P. *An efficient hardware-oriented single-pass approach for connected component analysis.* Sensors. 2019 Jul 11;19(14):3055.

Sun Y, Mao X, Hong S, Xu, G Gui. *Template matching-based method for intelligent invoice information identification*. IEEE access. 2019 Feb 27;7:28392-401

Suthar SB, Thakkar AR. *Hybrid deep resnet with inception model for optical character recognition in Gujarati language.* Reliability: Theory & Applications. 2022;17(1 (67)):194-209

Tadic V, Kiraly Z, Odry P, Trpovski Z, Loncar-Turukalo T. *Comparison of Gabor filter bank and fuzzified Gabor filter for license plate detection.* Acta Polytechnica Hungarica. 2020 Jan 1;17(1):1-21.

Tang J, L Wan, J Schooling, P Zhao, J Chen, S Wei. *Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases.* Cities. 2022 Oct 1;129:103833.

Venkateswari P, SteffyE J, Muthukumaran DN. *License plate cognizance by ocular character perception'.* International Research Journal of Engineering and Technology. 2018 Feb;5(2):536-42

Wang Y, Yang X, Liang H, Liu Y. *A review of the self-adaptive traffic signal control system based on future traffic environment.* Journal of Advanced Transportation. 2018 Jun 27;2018.

Wang,W Yang J, Chen M, P Wang. A *light CNN for end-to-end car license plates detection and recognition.* IEEE Access. 2019 Nov 28;7:173875-83.

Weihong W, Jiaoyang T. *Research on license plate recognition algorithms based on deep learning in complex environment*. IEEE Access. 2020 May 14;8:91661-75.

Wijaya MC. *Research of Indonesian license plates recognition on moving vehicles.* EUREKA: Physics and Engineering. 2022 Nov 29(6):185-98

Xiang H, Zhao Y, Yuan Y, Zhang G, Hu X. *Lightweight fully convolutional network for license plate detection*. Optik. 2019 Feb 1;178:1185-94

Xu H, Cai Z, Li R, Li W. *Efficient city cam-to-edge cooperative learning for vehicle counting in ITS.* IEEE Transactions on Intelligent Transportation Systems. 2022 Feb 14.

Yang Y, Zhang W, He Z, Li D. *High-speed rail pole number recognition through deep representation and temporal redundancy.* Neurocomputing. 2020 Nov 20;415:201-14.

Yousif BB, Ata MM, Fawzy N, Obaya M. *Toward an optimized neutrosophic K-means with genetic algorithm for automatic vehicle license plate recognition (ONKM-AVLPR).* IEEE Access. 2020 Mar 9;8:49285-312.

Zhu L, Yu FR, Wang Y, Ning B, Tang T. *Big data analytics in intelligent transportation systems: A survey.* IEEE Transactions on Intelligent Transportation Systems. 2018 Apr 23;20(1):383-98.

Znamenskaya IA, Doroshchenko IA. *Edge detection and machine learning for automatic flow structures detection and tracking on schlieren and shadowgraph images.* **Journal of Flow Visualization and Image Processing.** 2021;28(4)

Thesis

Al-qudah R. *License plate detection using deep learning and font evaluation* (Doctoral dissertation, Concordia University). 2019

Al-Shemarry MS. *Developing new techniques to improve licence plate detection systems for complicated and low quality vehicle images* (Doctoral dissertation, University of Southern Queensland).

Amri MY. *Development of automatic number plate recognition software and journey time measurement/Amri Mohd Yasin* (Doctoral dissertation, University of Malaya)

Ghasempour S. *Automatic license plate recognition (ALPR) (Master's thesis, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ).*2015.

Ning G. *Vehicle license plate detection and recognition (Doctoral dissertation,* University of Missouri--Columbia).2013.

Tarekegn L. *Faculty of informatics shape and texture based inter-specific hybrid fish species image recognition* (Doctoral dissertation, University of Gondar).2019

Website

https://ml5js.org/

https://opencv.org/

https://www.javatpoint.com

Appendix

Appendix I: Source Code

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_inappwebview/flutter_inappwebview.dart';
import 'package:permission_handler/permission_handler.dart';

Future main() async {
```

```
  WidgetsFlutterBinding.ensureInitialized();
  await Permission.camera.request();

  runApp(MyApp());
}


class MyAppp extends StatefulWidget {

  @override
  _MyApppState createState() => _MyApppState();
}

class _MyApppState extends State<MyAppp> {

  TextEditingController nameController = TextEditingController();
  String UserName = '';


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
          appBar: AppBar(
            title: Text('Tunnel url'),
          ),

          body: Center(child: Column(children: <Widget>[
            Container(
                margin: EdgeInsets.all(20),
                child: TextField(
                  controller: nameController,
                  decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: 'Tunnel url',
                  ),
                  onChanged: (text) {
                    setState(() {
                      UserName = text;
                      //you can access nameController in its scope to get
                      // the value of text entered as shown below
                      //UserName = nameController.text;
                    });
                  },
                )),

            Container(
              margin: EdgeInsets.all(20),
              child: Text(UserName),
            ),
```

```
          new Container(
            padding: const EdgeInsets.only(left: 20.0, top: 20.0),
            child: new RaisedButton(
              child: const Text('Next Page'),
              onPressed: () {
                Navigator.of(context).push(new MaterialPageRoute(
                    builder: (BuildContext context) => new
InAppWebViewPage(this.UserName)));
              },
            )),
        ])),
    );
  }

}




class MyApp extends StatefulWidget {

  @override
  _MyAppState createState() => new _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        debugShowCheckedModeBanner: false,
        title: ' License project',
        home: MyAppp()
    );
  }
}

class InAppWebViewPage extends StatefulWidget {
  final String url;
  InAppWebViewPage(this.url);
  @override
  _InAppWebViewPageState createState() => new
_InAppWebViewPageState(this.url);
}

class _InAppWebViewPageState extends State<InAppWebViewPage> {
  final String url;
  _InAppWebViewPageState(this.url);
  InAppWebViewController _webViewController;
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text("License plate detector")
        ),
        body: Container(
            child: Column(children: <Widget>[
              Expanded(
                child: Container(
                  child: InAppWebView(
                      initialUrl:url ,
                      initialOptions: InAppWebViewGroupOptions(
                        crossPlatform: InAppWebViewOptions(
                          mediaPlaybackRequiresUserGesture: false,
                          debuggingEnabled: true,
                        ),
                      ),
                      onWebViewCreated: (InAppWebViewController controller) {
                        _webViewController = controller;
                      },
                      androidOnPermissionRequest: (InAppWebViewController controller,
String origin, List<String> resources) async {
                        return PermissionRequestResponse(resources: resources, action:
PermissionRequestResponseAction.GRANT);
                      }
                  ),
                ),
              ),
            ]))
    );
  }
}
```

```
const videoElement = document.createElement('video');
 videoElement.setAttribute("style", "display: none;");
 videoElement.width = width;
 videoElement.height = height;
 document.body.appendChild(videoElement);

 // Create a webcam capture
 const capture = await navigator.mediaDevices.getUserMedia({ video: true })
 videoElement.srcObject = capture;
 videoElement.play();

 return videoElement
```

```
import 'dart:async';
import 'package:flutter/material.dart';
```

```
import 'package:flutter_inappwebview/flutter_inappwebview.dart';
import 'package:permission_handler/permission_handler.dart';
Future main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Permission.camera.request();
  runApp(MyApp());
}
class MyAppp extends StatefulWidget {
  @override
  _MyApppState createState() => _MyApppState();
}
class _MyApppState extends State<MyAppp> {
  TextEditingController nameController = TextEditingController();
  String UserName = '';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
          appBar: AppBar(
            title: Text('Tunnel url'),
          ),
          body: Center(child: Column(children: <Widget>[
            Container(
                margin: EdgeInsets.all(20),
                child: TextField(
                  controller: nameController,
                  decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: 'Tunnel url',
                  ),
                  onChanged: (text) {
                    setState(() {
                      UserName = text;
                      //you can access nameController in its scope to get
                      // the value of text entered as shown below
                      //UserName = nameController.text;
                    });
                  },
                )),
            Container(
              margin: EdgeInsets.all(20),
              child: Text(UserName),
            ),
            new Container(
                padding: const EdgeInsets.only(left: 20.0, top: 20.0),
                child: new RaisedButton(
                  child: const Text('Next Page'),
                  onPressed: () {
                    Navigator.of(context).push(new MaterialPageRoute(
```

```
                    builder: (BuildContext context) => new
InAppWebViewPage(this.UserName)));
                },
              )),
        ]))),
    );
  }
}
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => new _MyAppState();
}
class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'Pipeline project',
        home: MyAppp()
    );
  }
}
class InAppWebViewPage extends StatefulWidget {
  final String url;
  InAppWebViewPage(this.url);
  @override
  _InAppWebViewPageState createState() => new
_InAppWebViewPageState(this.url);
}

class _InAppWebViewPageState extends State<InAppWebViewPage> {
  final String url;
  _InAppWebViewPageState(this.url);
  InAppWebViewController _webViewController;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Pipeline face detector")
        ),
        body: Container(
            child: Column(children: <Widget>[
              Expanded(
                child: Container(
                  child: InAppWebView(
                      initialUrl:url ,
                      initialOptions: InAppWebViewGroupOptions(
                        crossPlatform: InAppWebViewOptions(
                          mediaPlaybackRequiresUserGesture: false,
                          debuggingEnabled: true,
```

```
            ),
          ),
          onWebViewCreated: (InAppWebViewController controller) {
            _webViewController = controller;
          },
          androidOnPermissionRequest: (InAppWebViewController controller,
String origin, List<String> resources) async {
            return PermissionRequestResponse(resources: resources, action:
PermissionRequestResponseAction.GRANT);
          }
        ),
      ),
    ),
  ]))
 );
 }
}


        var ctrack = new clm.tracker();
        ctrack.init();
        var trackingStarted = false;

        function startVideo() {
          // start video
          vid.play();
          // start tracking
          ctrack.start(vid);
          trackingStarted = true;
          // start loop to draw face
          drawLoop();
        }

        function drawLoop() {
          requestAnimFrame(drawLoop);
          overlayCC.clearRect(0, 0, vid_width, vid_height);
          //psrElement.innerHTML = "score :" + ctrack.getScore().toFixed(4);
          if (ctrack.getCurrentPosition()) {
      capture();
            ctrack.draw(overlay);
      console.log('found face');


          }
        }

// update stats on every iteration
        document.addEventListener('clmtrackrIteration', function(event) {
          stats.update();
        }, false);
```

```
    // start video
        vid.play();
        // start tracking
        ctrack.start(vid);
        trackingStarted = true;
        // start loop to draw face
        drawLoop();

   function capture() {
     // var canvas = document.getElementById('overlay');
     // var video = document.getElementById('videoel');
     // canvas.width = video.videoWidth;
     // canvas.height = video.videoHeight;
     overlay.getContext('2d').drawImage(vid, 0, 0, vid.width, vid.height);
     // canvas.toBlob() = (blob) => {
     //   const img = new Image();
     //   img.src = window.URL.createObjectUrl(blob);
     //   console.log(img.src);
     // };

     // canvas.toBlob(blob => {
     //   const img = new Image();
     //    img.src = window.URL.HTMLMediaElement.src(blob);
     // });
     img = new Image();
     img = overlay.toDataURL('image/png');
    let url = img.replace(/^data:image\/png/,'data:application/octet-stream');
    var name= getFileName();
    //var file = dataURLtoFile(img,name);
    // console.log(file);
    // Split the base64 string in data and contentType
    var block = img.split(";");
    // Get the content type of the image
    var contentType = block[0].split(":")[1];// In this case "image/gif"
    // get the real base64 content of the file
    var realData = block[1].split(",")[1];// In this case "R0lGODlhPQBEAPeoAJo
sM...."

    // Convert it to a blob to upload
    var blob = b64toBlob(realData, contentType);


    // saveAs(img,'testing.png');

    // Create a FormData and append the file with "image" as parameter name
    var formDataToUpload = new FormData();
    formDataToUpload.append("image", blob);
    formDataToUpload.append("name", getFileName());
    formDataToUpload.append("_token", $('meta[name="csrf-
token"]').attr('content'));
```

```
            /**
 * The following code should send 2 post parameters:
 * filename: providen by the text input
 * image: a file, dinamically added from a base64 string using javascript
 *
 * Is up to you how to receive the file in the Server side.
 */
$.ajax({
   url:"/detectface",
   data: formDataToUpload,// Add as Data the Previously create formData
   type:"POST",
   contentType:false,
   processData:false,
   cache:false,
   dataType:"json", // Change this according to your response from the server.
   error:function(err){
      console.error(err);
   },
   success:function(data){
      console.log(data);
   },
   complete:function(){
      console.log("Request finished.");
   }
});
```

Appendix A3: Code For Recapturing

```
<!doctype html>
<html lang="en">
   <head>
      <title>Face Tracker</title>
      <meta charset="utf-8">
      <style>
         @import url(https://fonts.googleapis.com/css?family=Lato:300italic,700italic,
300,700);

         body {
            font-family: 'Lato';
            background-color: #f0f0f0;
            margin: 0px auto;
            max-width: 1150px;
         }

         #overlay {
            position: absolute;
            top: 0px;
            left: 0px;
         }
```

```
        #container {
            position : relative;
            width : 700px;
            height : 500px;
            /*margin : 0px auto;*/
        }

        #content {
            margin-top : 70px;
            margin-left : 100px;
            margin-right : 100px;
            max-width: 950px;
        }

        #convergence {
            display : inline;
        }

        h2 {
            font-weight : 400;
        }

        .btn {
            font-family: 'Lato';
            font-size: 16px;
        }

        .hide {
            display : none;
        }
    </style>

</head>
<body>
    <script src="../pipe/js/libs/utils.js"></script>
    <script src="../pipe/js/libs/dat.gui.min.js"></script>
    <script src="../pipe/build/clmtrackr.js"></script>
    <script src="../pipe/js/libs/Stats.js"></script>

    <link rel="stylesheet" type="text/css" href="./styles/imgareaselect-default.css" />
    <script src="../pipe/js/libs/jquery.min.js"></script>
    <script src="../pipe/js/libs/jquery.imgareaselect.pack.js"></script>

    <div id="content">
        <h2>Face tracking on a pipeline </h2>
        <div id="container">
            <canvas id="image" width="625" height="500"></canvas>
            <canvas id="overlay" width="625" height="500"></canvas>
        </div>
```

```
        <br/>
        <!--
 <input type="button" class="btn" value="start" onclick="animateClean()"></input> -
->
        <input type="button" class="btn" value="manually select face" onclick="selec
tBox()"></input>
        <input type="file" class="btn" id="files" name="files[]" />
        <p id="convergence"></p>

        <script>

          var i =1;
          //alert(getLatestFile('pipeline'));

          var cc = document.getElementById('image').getContext('2d');
          var overlay = document.getElementById('overlay');
          var overlayCC = overlay.getContext('2d');

          var img = new Image();
          img.onload = function() {
            cc.drawImage(img,0,0,625, 500);
            //cc.drawImage(img,312.5,225,312.5, 225);
          };
          img.src = '../pipeline/'+i+'.jpg';
          cc.drawImage(img,0,0,625, 500);
           $.ajax({
          type: 'POST',
           url: 'api/counterpipe ',
           success: function (data) {
            i= data.count;
              //alert(i);
              img.src = '../pipeline/'+i+'.jpg';
          }

        });
          var ctrack = new clm.tracker({stopOnConvergence : true});
          ctrack.init();

          stats = new Stats();
          stats.domElement.style.position = 'absolute';
          stats.domElement.style.top = '0px';
          document.getElementById('container').appendChild( stats.domElement );

          var drawRequest;

          function animateClean() {
            ctrack.start(document.getElementById('image'));
            drawLoop();
          }
```

```
function animate(box) {
   ctrack.start(document.getElementById('image'), box);
   drawLoop();
}

function drawLoop() {
   drawRequest = requestAnimFrame(drawLoop);
   overlayCC.clearRect(0, 0, 720, 576);
   if (ctrack.getCurrentPosition()) {
      ctrack.draw(overlay);
   }
}

// detect if tracker fails to find a face
document.addEventListener("clmtrackrNotFound", function(event) {
   ctrack.stop();
   //alert("The tracking had problems with finding a face in this image. Try s
electing the face in the image manually.")
}, false);

// detect if tracker loses tracking of face
document.addEventListener("clmtrackrLost", function(event) {
   ctrack.stop();
   //alert("The tracking had problems converging on a face in this image. Tr
y selecting the face in the image manually.")
}, false);

// detect if tracker has converged
document.addEventListener("clmtrackrConverged", function(event) {
   document.getElementById('convergence').innerHTML = "Face found acr
oss the pipeline";
   document.getElementById('convergence').style.backgroundColor = "#00
FF00";
   // stop drawloop
   cancelRequestAnimFrame(drawRequest);
}, false);

// update stats on iteration
document.addEventListener("clmtrackrIteration", function(event) {
   stats.update();
}, false);

// manual selection of faces (with jquery imgareaselect plugin)
function selectBox() {
   overlayCC.clearRect(0, 0, 720, 576);
   document.getElementById('convergence').innerHTML = "";
   ctrack.reset();
   $('#overlay').addClass('hide');
   $('#image').imgAreaSelect({
      handles : true,
```

```
        onSelectEnd : function(img, selection) {
          // create box
          var box = [selection.x1, selection.y1, selection.width, selection.heig
ht];
          // do fitting
          animate(box);
          $('#overlay').removeClass('hide');
        },
        autoHide : true
      });
    }
    // function to start showing images
    function loadImage() {
      if (fileList.indexOf(fileIndex) < 0) {
        var reader = new FileReader();
        reader.onload = (function(theFile) {
          return function(e) {
            // check if positions already exist in storage
            // Render thumbnail.
            var canvas = document.getElementById('image')
            var cc = canvas.getContext('2d');
            var img = new Image();
            img.onload = function() {
              if (img.height > 500 || img.width > 700) {
                var rel = img.height/img.width;
                var neww = 700;
                var newh = neww*rel;
                if (newh > 500) {
                  newh = 500;
                  neww = newh/rel;
                }
                canvas.setAttribute('width', neww);
                canvas.setAttribute('height', newh);
                cc.drawImage(img,0,0,neww, newh);
              } else {
                canvas.setAttribute('width', img.width);
                canvas.setAttribute('height', img.height);
                cc.drawImage(img,0,0,img.width, img.height);
              }
            }
            img.src = e.target.result;
          };
        })(fileList[fileIndex]);
        reader.readAsDataURL(fileList[fileIndex]);
        overlayCC.clearRect(0, 0, 720, 576);
        document.getElementById('convergence').innerHTML = "";
        ctrack.reset();
      }
    }
    // set up file selector and variables to hold selections
```

```
            var fileList, fileIndex;
            if (window.File && window.FileReader && window.FileList) {
                function handleFileSelect(evt) {
                    var files = evt.target.files;
                    fileList = [];
                    for (var i = 0;i < files.length;i++) {
                        if (!files[i].type.match('image.*')) {
                            continue;
                        }
                        fileList.push(files[i]);
                    }
                    if (files.length > 0) {
                        fileIndex = 0;
                    }
                    loadImage();
                }
                document.getElementById('files').addEventListener('change', handleFileS
elect, false);
            } else {
                $('#files').addClass("hide");
                $('#loadimagetext').addClass("hide");
            }
            function autoRefresh_div() {


            }
            //setInterval(autoRefresh_div, 5000); // every 5 seconds
            //autoRefresh_div(); // on load */
            animateClean();
            var time = new Date().getTime();
            $(document.body).bind("mousemove keypress", function(e) {
                time = new Date().getTime();
            });

            function refresh() {
                if(new Date().getTime() - time >= 60000)
                    window.location.reload(true);
                else
                    setTimeout(refresh, 10000);
            }
            setTimeout(refresh, 10000);

        </script>
    </div>
  </body>
</html>
```

Bio-data

A. Personal Data

1. **Full Name:** AKANNI ABIODUN AYINDE
2. **Date and Place of Birth:** 14^{th} February 1996.

3. **Nationality:** Nigerian

4. **Marital Status:** Single

5. **No. of Children & their ages:** Nil

6. **Name and Address of Spouse:** Nil

7. **Name and Phone Number:** Mrs Akanni (08131002484)

8. **Faculty:** Natural and Applied Sciences

9. **Department:** Computer Science

B. Educational Background

11/2017 – 07/2019
HIGHER NATIONAL DIPLOMA (H.N.D), COMPUTER SCIENCE, GATEWAY (I.C.T) POLYTECHNIC, SAAPADE. OGUN STATE.

10/2014 – 08/2016
ORDINARY NATIONAL DIPLOMA (O.N.D), COMPUTER SCIENCE, GATEWAY (I.C.T) POLYTECHNIC, SAAPADE. OGUN STATE.

09/2007 – 06/2012
WEST AFRICAN EXAMINATION COUNCIL (WAEC), **BADE UNIK SECONDARY SCHOOL, SAGAMU, OGUN STATE**

09/2000 – 08/2006
FIRST SCHOOL LEAVING CERTIFICATE, **DEEN MASTER FOUNDATION SCHOOL, LAGOS ISLAND, LAGOS STATE.**

C. Work Experience: With Dates

DRAGNET SOLUTIONS LIMITED, LAGOS.	**11/2020 –**
XYZ TECHNOLOGIES LIMITED, ABIA.	12/2019 – 11/2020
ADDOSSER MICROFINANCE BANK, LAGOS.	01/2017 – 11/2017
JONDADE CAFE, LAGOS.	10/2016 – 12/2016

TECHNICAL PROJECTS

- THE DESIGN AND IMPLEMENTATION OF AN **EMPLOYEE LEAVE MANAGEMENT SYSTEM**
- THE DESIGN AND IMPLEMENTATION OF AN **ELECTRONIC HEALTH INFORMATION SYSTEM**

TECHNICAL SEMINARS

- SEMINAR REPORT ON ANDROID

- SEMINAR REPORT ON WORLD WIDE WEB

PROFESSIONAL SKILLS

- Graphics: Corel draw, Photoshop
- MS Office: Word, Excel, PowerPoint, Access
- Networking
- Programming Skills: HTML5, CSS3, PHP, SQL

WORK RELATED SKILLS

- Excellent Interpersonal Skills
- Excellent Verbal and Written Skills
- Critical thinking skills
- Analytical Skills
- Team & Project Management

PROFESSIONAL CREDITS

- Addosser Micro Finance Bank Staff of the month award winner for the month of July, 2017.

LEADERSHIP AND VOLUNTEER ROLES

- Project/Program Director for the Sustainable Development Group during service year

REFEREES

AVAILABLE ON REQUEST

The University Compliance Certification

This is to certify that this thesis by Abiodun AKANNI with Matriculation Number LCU/PG/002226 in the Department of Computer Science, Faculty of Natural and Applied

Sciences, Lead City University, Ibadan is in full compliance with the approval of the University's format and style.

………………………………… ……………………………..

Signature Date