Dissertation on

Automatic Vehicle License Plate Detection and Recognition

Thesis submitted towards partial fulfilment of the requirements for the degree of

Master in Multimedia Development

Submitted by

Soumen Hazra

EXAMINATION ROLL NO.: M4MMD23004B

UNIVERSITY REGISTRATION NO.: 160393 of 2021-2022

Under the guidance of

Prof. (Dr.) Matangini Chattopadhyay

School of Education Technology
Jadavpur University

Course affiliated to

Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032
India

2023

Master in Multimedia Development
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **"Automatic Vehicle License Plate Detection and Recognition"** is a bonafide work carried out by **SOUMEN HAZRA** under our supervision and guidance for partial fulfillment of the requirements for the degree of Master in Multimedia Development in the School of Education Technology, during the academic session 2022-2023.

SUPERVISOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DIRECTOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DEAN - FISLM
Jadavpur University,
Kolkata-700 032

Master in Multimedia Development
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

<u>CERTIFICATE OF APPROVAL</u> **

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

Committee of final examination for evaluation of Thesis --

** Only in case the thesis is approved.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of his **Master in Multimedia Development** studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

NAME : **SOUMEN HAZRA**

EXAMINATION ROLL NUMBER: **M4MMD23004B**

THESIS TITLE : **AUTOMATIC VEHICLE LICENSE PLATE DETECTION AND RECOGNITION**

SIGNATURE: DATE:

ACKNOWLEDGEMENT

I feel extremely glad in presenting this thesis at School of Education Technology, Jadavpur University, Kolkata, in the partial fulfilment of the requirements for the degree of Master in Multimedia Development.

I would like to express my sincere gratitude to my guide, Prof. (Dr.) Matangini Chattopadhyay, for her vigilant supervision and constant encouragement. She spent her precious time in reviewing the research work and provided many insightful comments and constructive criticism. Without her enthusiasm, advice and unique support this research work would never have become a reality.

I am really grateful to Dr. Saswati Mukherjee and Mr. Joydeep Mukherjee for their continuous support during the entire course of the research. Their advice and support were highly inspirational and motivating.

I would also like to take this opportunity to thank all my classmates of MMD programme and MTech. IT (Courseware Engineering) programme who motivated me to complete my research work successfully. I do wish to thank all of our departmental support staffs and all of those who were associated with this research contributed in some form or the others.

Lastly but not least, I would like to acknowledge and thank my parents for the strong, unconditional support, inspiration and encouragement that they have provided me, without which I would be unable to complete this research.

Regards,

____

Soumen Hazra
Master in Multimedia Development
School of Education Technology
Jadavpur University, Kolkata

Date:

Dedicated To,

My Parents

CONTENTS

List of Abbreviation

ANPR	Automatic Number Plate Recognition
LP	License Plate
ALPR	Automatic License Plate Recognition
OCR	Optical Character Recognition
OpenCV	Open-Source Computer Vision Library
EasyOCR	Easy Optical Character Recognition
CCA	Connected component analysis
VLP	Vehicle License Plate
CCTV	Closed-Circuit Television

List of Table

LIST OF FIGURES

EXECUTIVE SUMMARY

Automatic Number Plate Recognition (ANPR) is an important application of computer vision and deep learning that has numerous practical applications in areas such as traffic management, law enforcement, and parking management. This ANPR system is a technology that uses optical image recognition to read vehicle license plates. It uses the Open-Source Computer Vision (OpenCV) and Easy Optical Character Recognition (EasyOCR) libraries for this purpose. It can be used in existing closed-circuit television systems, traffic surveillance cameras, or specially designed cameras. After capturing the images from the cameras, the data undergoes preprocessing as the first step. The edges are then found using the Canny edge detection technique. The license plate is detected by analyzing the contours of specific areas. The license plate is recognized using Optical Character Recognition (OCR) technology. We have achieved high accuracy in both the detection and recognition models.

Chapter 1

1.0 INTRODUCTION

Automatic number plate recognition, or ANPR, is a technology that uses pattern recognition to "read" vehicle license plates. ANPR cameras, simply put, capture images of license plates of vehicles as they pass by. The information about the vehicle is obtained by inputting this "photo" into a computer system. ANPR consists of cameras connected to a computer. When a vehicle is passing, ANPR reads the vehicle's number plates - more commonly known as license plates - from digital images captured by mobile device cameras, onboard traffic vehicles, or closed-circuit television (CCTV). The digital image is transformed into data, which is then processed by the ANPR system. We propose a technique that heavily relies on edge detection, optical character recognition (OCR), and the identification of rectangles within the vehicle image. Computer vision in the Intel’s exploration has been producing a fruit called Open-Source Computer Vision (Open CV), which can support the development of computer vision [1].

There are a large number of vehicles in today's generation around the world. So, it's vitally important to keep track of vehicles [2]. In today's environment, we can use computers to track vehicles instead of searching for them manually, which leads to improved accuracy. The technology used to identify the number plate from video footage recorded by the camera is called a vehicle license plate recognition system. It employs techniques such as character recognition, segmentation, and license plate recognition. This system combines hardware and software to capture the license plate and transmit the converted image of the plate. Any gate entrance can benefit from this technology. Therefore, if the image generated from the number plate captured by the camera is clear and visible, this system can be accurate. The image used must have a high resolution.

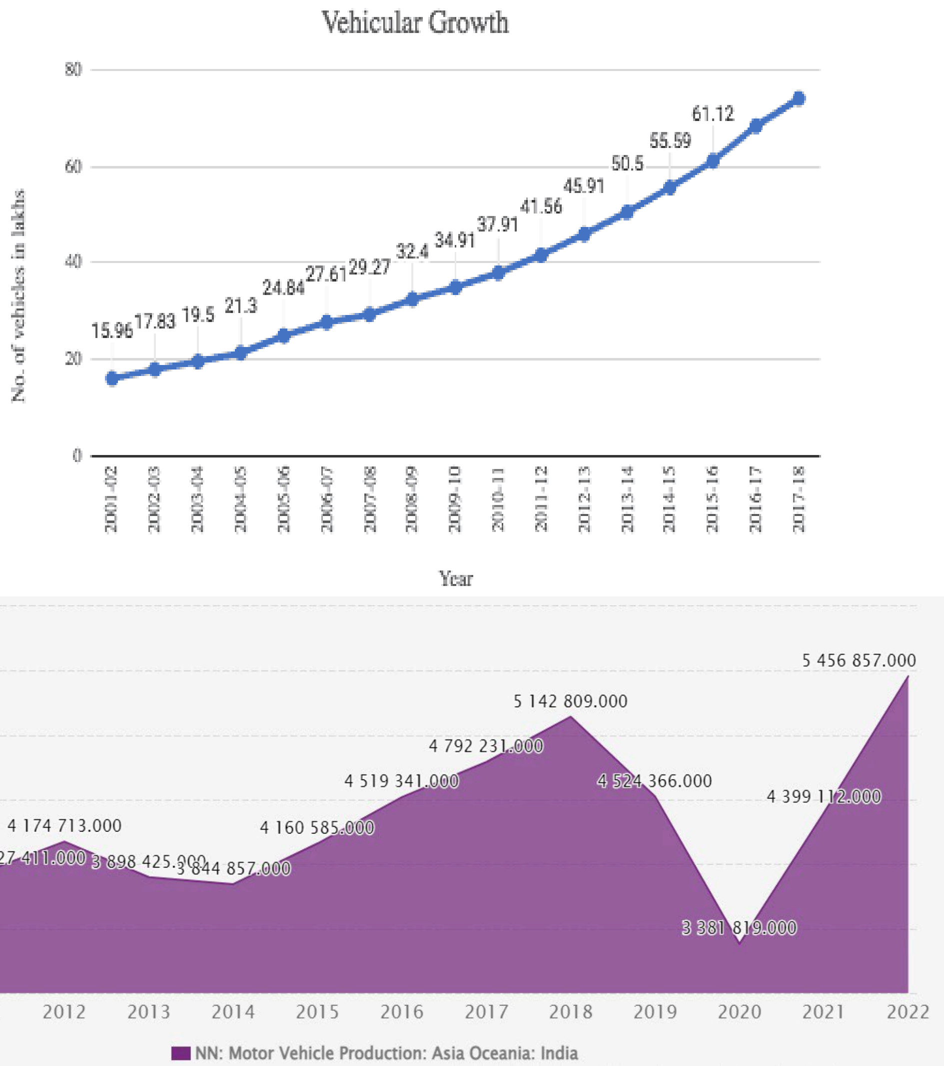

Figure 1: Vehicle growth in India [3]

Currently, the use of vehicles is increasing nationwide. These cars are primarily identified by a unique vehicle identification number. The ID is actually in the license number, which refers to a legal permit to participate in public transportation. Every vehicle in the world needs a unique license plate that must be mounted on its body, at the very least on the rear. They must determine which vehicles are increasing in relation to the total number of vehicles. This identification system helps with safety, automatic switching systems, trace speed discovery, light detection, stolen vehicle recovery, and collection systems for both human and non-human losses. The license plate identification approach comprises three primary challenges in order to achieve accurate identification of individuals. They are: (a) locating the panel of digital photos, (b) segmenting the characters from the panel's photographs, and (c) performing visual character recognition [4]. The first and most important step is to pinpoint the exact location of the license plate in the captured image. The localization of a license plate has been achieved through both structural analysis and colour analysis systems. ANPR is a comprehensive surveillance system that captures vehicle images and recognizes license plate numbers. Some ANPR systems can be configured to store a photograph of the driver, along with the license plate data and the images captured by the cameras. Infrared lighting is commonly used in systems to enable the camera to capture photos at any time of day. The acquisition of digital images generally suffers from undesirable camera shakes and unstable arbitrary camera movements. An important feature of the crossroad monitoring cameras is the inclusion of a flash. This flash serves two purposes: to illuminate the picture and to make the lawbreaker apprehensive of their mistake. Due to variations in license plate formats across different regions, Automatic Number Plate Recognition (ANPR) technology is often designed to be specific to a particular area.

1.1 Problem Statement

To design and develop automatic vehicle license plate detection and recognition system.

1.2 Objectives

This research work aims to develop smart and innovative infrastructures for the automated platform of Automatic Number Plate Recognition (ANPR) retrieval, utilizing high-definition (HD) and real-time technology. The main objectives of this research work are as follows:

- To detect the license plate and accurately isolate the region of interest from the recorded image.
- To accurately determine the coordinates (bounding box) of the license plate.
- To recognize and extract the alphanumeric characters from the license plate.
- To optimize the processing time of the system.
- To learn Python libraries such as OpenCV, EasyOCR, and Imutlis.
- To develop a system that supports various license plate formats, as different nations and regions have unique license plate designs.
- To develop a system that can perform well with a wider range of image qualities.

1.3 Assumptions and Scope

Assumptions

- The system assumes that the input license plate images or video frames are of sufficient quality for detection and recognition. Proper lighting, minimal motion blur, and a clear view of the license plate are all important factors to consider.
- The system assumes that the license plates are standard and not heavily customized or obscured. Non-standard or heavily modified license plates may be difficult to identify.
- Depending on the application, the system requires access to an adequate number of computational resources (CPU/GPU) for real-time or near-real-time processing.

Scopes

- The main purpose of the system is to identify and locate license plates in the input images or video frames. This involves the use of computer vision techniques, such as object detection algorithms (e.g., YOLO, SSD) and image segmentation.
- Once a license plate is detected, the system aims to recognize the alphanumeric characters on the plate. This is commonly achieved using optical character recognition (OCR) techniques.
- In addition to license plate detection and recognition, some systems may include an optional feature to identify the type of vehicle (car, truck, motorcycle, etc.).

- The system may need to integrate with other systems, such as security databases, access control systems, or law enforcement databases, for further processing or action based on the detected and recognized license plates.
- Environmental factors, such as varying lighting conditions, weather, and road surface conditions, may need to be considered in the scope.

1.4 Concept and Problem Analysis

Automatic Vehicle License Plate Detection and Recognition is a technology-driven solution that automatically identifies and reads license plates on vehicles. This system utilizes a combination of computer vision, machine learning, and image processing techniques to identify license plates within images or video streams, extract alphanumeric characters from the plates, and recognize the text.

- License plates can be located in images with complex backgrounds, and lighting conditions can vary dramatically. This can make it challenging to accurately detect and recognize license plates.
- Images of moving vehicles can suffer from motion blur and distortion, which can impact the accuracy of plate detection and recognition.
- In many applications, these systems must operate in real-time, meaning they need to process images or video frames quickly and efficiently.
- Systems need to efficiently handle a large volume of vehicles, which necessitates scalable architecture and hardware.

1.5 Organization of the thesis

- The first chapter of this thesis contains the introduction, which includes an overview, problem statement, research objectives, assumptions, scope, concept and problem analysis.
- The second chapter covers the literature surveys conducted for the research work.
- The third chapter contains a background study that should be understood before conducting this research.
- The fourth chapter contains the proposed approach, dataset, and algorithm of this research work.
- The fifth chapter describes the experimental results and provides a comparative analysis.
- The sixth chapter contains the conclusion and future scope of this research work.
- The references section includes all of the citations.
- All the code snippets are provided in the appendix section.

Chapter 2

2.0 LITERATURE SURVEY

Over the past decade, with the advancement of information technology and the growth of the population, numerous methods for developing license plate detection systems have been developed and researched.

- A. Menon and B. Omman [5] proposed that their system consists of two steps: number plate detection and recognition. In case of detection, there are several steps involved. These include gray scale conversion, filtering the image using a Sobel filter, performing threshold operation, applying morphological operation, detecting the area using the find contour function, and finally pre-processing and segmenting all possible parts of the image by applying classification methods. To predict whether each region is a plate or not, the Support Vector Machine (SVM) algorithm is applied by them. For license plate recognition, an Artificial Neural Network (ANN) is used. First, they apply OCR segmentation by using the OCR function with a histogram equalization. They apply the Multi-Layer Perceptron (MLP), which is the most commonly used artificial neural network (ANN) algorithm.

- Milan Samantaray et al. [6] proposed a system that captures the license plate image using a camera and processes it in the following ways. First, the captured image is uploaded into the PyCharm IDE, which is recommended for image processing tasks. In the alternate phase, the image processing unit reuses its functions by utilizing the "imutils" library tool. In the third phase, the license plate discovery and segmentation are performed using the "findContours()" function from the "imutils" library. This phase has just segmented the number plate area. In the final phase, the characters are extracted from the detected number plate using the "pytesseract" tool, which provides optical character recognition capabilities. So, it will read or recognize the text embedded on the number plate.

- This article [7] provides a detailed overview of the current Automatic Number Plate Recognition (ANPR) systems and algorithms, comparing their performance in real-time testing and simulations. The aim is to promote ANPR technology based on computer vision (CV) algorithms. This involves the use of extraction, segmentation, detection analysis techniques, and recognition techniques. The goal is to provide guidance for future trends in this field. Additional hardware may be required to maximize accuracy, even with the best algorithms.

- The main purpose of this study is to investigate problems related to license plate segmentation and recognition [8], and to develop a framework for recognition along with alternative solutions. It involves three steps: finding the license plate and extracting it, removing the disk area from the larger scene image, and using the alphanumeric characters in the background license plate area to present them for identification in the OCR system. The vehicle license plate has been successfully identified and is visible in the image captured by a recording system, such as a video or image camera.
- Automatic license plate detection and recognition systems have been proposed and implemented over the years. Easy and fast monitoring of traffic rules by vehicles. This paper [9] provides a concise overview. A Review of Various Approaches Used in Automatic

License Plate Detection Systems. These approaches can be broadly categorized into two categories: Techniques for automatic car number plate detection (Adaptive image segmentation, Edge detection, Hough transform, Blob detection, CCA or connected component analysis, Mathematical morphology) and techniques for character recognition (Artificial neural network, Template matching, Deep Learning, Adaptive binarization technique). This paper also discusses advantages and disadvantages of various approaches falling under these categories. Finally, the paper concludes with a comparative analysis of all the approaches reviewed in both categories.

- ANPR is a system that uses Optical Character Recognition (OCR) to recognize different characters from sources such as surveillance cameras or other cameras. It is important to position the camera at the right angle to capture the best and most accurate image [10]. OCR analyzes each brand separately. OCR is a method used to convert written or printed text and information from various sources, such as documents and images, into a text format using an optical character recognition (OCR) decoder. It is a complex process that involves several algorithmic steps, including loading the image, character recognition, personalizing them on the page, removing blur, and creating a final editable shape.

- Automatic License Plate Recognition (ALPR) is a computer system that automatically recognizes a license plate from a captured image of a vehicle. Their system [1] involves image capture, localization of number plates, segmentation of characters, and optical character recognition (OCR) of alphanumeric characters. Their system aims to develop efficient image processing techniques and algorithms using the Open Computer Vision (OpenCV) Library. This system has been implemented on a single board computer called BeagleBone Black, with the support of OpenCV and the Python programming language.

- High-resolution images are crucial for digital image processing and analysis, particularly in the field of forensic science. Common security surveillance videos are often low in clarity and degraded, which can significantly impact image interpretation and analysis. Super-resolution methods are utilized to generate high-resolution images from low-resolution images, with output validation based on Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) values. They proposed a combination method [11] which were evaluated against other typical resampling methods and the sparse representation method. The experimental results had shown that the sparse representation method has the highest PSNR mean value, but their proposed combinational method is comparable to it as the difference in means is too small.

- This paper [12] provides a technique for the registration of images that can be detected at the time of supervision. In this paper recent and classic image registration methods, which involve overlaying images of the same scene taken at different times, from different viewpoints, and/or by different sensors. The reviewed approaches are classified based on their nature and four basic steps: feature detection, feature matching, mapping function design, and image transformation and resampling.

- This article [13] proposes a segmentation scheme for Indian license plates using connected component analysis. It includes preprocessing operations, connected component analysis using a labelling algorithm, aspect ratio analysis, and pixel count analysis. The method extracts characters from license plates using specific coordinates, and the results are promising even under challenging conditions. This method is easy to implement and effective for character recognition.

- This paper [14] presents a method for detecting and identifying vehicle number plates, with a focus on authorized and unauthorized vehicles. It employs a simple morphological operation and Sobel edge detection technique, which simplifies the segmentation of letters and numbers by using the bounding box method. The template matching approach is then used for the recognition of numbers and characters. The focus is on accurately locating the number plate region in order to segment all numbers and letters and identify each number individually.

- This article [15] presents the following methods: first, preprocessing is done using a specific method where the color input is converted to a grayscale image. This grayscale image is then distributed across the number of pixels in the image. Next, edge detection is performed using a pitcher edge detector to identify edges in the image and minimize interruptions. Then, the input image is expanded using a morphological operator to increase the thickness of the numbers, making them easier to identify. Segmentation is performed on an enlarged image. Segmentation uses a pattern-matching algorithm, such as OCR. Finally, there are the numbers that can be adjusted to enhance the desired image in the output.

- In this article [16], the image was initially captured using a Pi camera. The video serves as the input, and the resulting images are saved in color JPEG format. There is a noise in the system. Gray processing and median filtering are used to remove the noise. Grayscale processing is used to convert the image to grayscale and apply median filtering to eliminate noise. The license plate area is identified by detecting the edges of the rectangular plate using the crop function. The box and edge detectors are used to detect the edges of the vehicle. After removal, segmentation is performed. Segmentation is used to separate the characters on the license plate. Text recognition is used to recognize different characters and numbers. Once detected, the characters are displayed as text in the output.

- In this paper [17], the author focuses on edge detection and mathematical morphology for plate location. The original color image was changed to grayscale. They applied morphology operations such as erosion, dilation, opening operation, and closing operation. They collected 120 images as a dataset. 109 images were predicted correctly, while 11 images failed, resulting in an accuracy rate of 90.833%.

- Mutua Simon Mandi and his colleagues proposed introducing a mobile phone software solution with Automatic Number Plate Recognition (ANPR) capabilities to assist in vehicle identification and registration [18]. The developed software adopts object analysis and design methodology and applies optical character recognition. (OCR) uses the mobile device's camera to recognize and record the vehicle license plate.

- Vehicle plate recognition is a widely used image processing technique for identifying vehicle license plates. It has various applications in marketing, traffic control, law enforcement, and travel. This study [19] proposes a canny edge detection method for a plate recognition System in practical situations, such as environmental or meteorological conditions. Image processing tools scan the plate area, resize it, and convert it to grayscale before filtering to remove small objects. The details of the image are controlled by the standard deviation of the Gaussian filter.

- This paper [20] introduces a color texture-based method for object detection in images, with a specific focus on a system for localizing vehicle license plates. The technique utilizes a support vector machine (SVM) to analyze the color and textural properties of license plates (LPs), eliminating the need for an external feature extraction module. The LP regions are identified using a continuously adaptive mean shift algorithm (CAM Shift). This combination results in robust and efficient low-power (LP) detection, reducing the time-consuming analysis of color textures for pixels that are less relevant.

- The paper [21] presents a method for recognizing vehicle license plates (VLP) using a combination of the Hough transform and contour algorithm. This method optimizes speed and accuracy by enhancing the algorithm's performance through the application of the Hough transform to contour images. The algorithm can be used on VLP images taken from various distances and inclined angles between ±30° from the camera, particularly in images with multiple VLPs. This approach is useful for VLP recognition systems and traffic management systems.

- License plate recognition is a computerized system that detects a license plate from a digital image. In this proposed work [22], the system involves various operations, including capturing images, locating the actual license plate, dividing letters, and optical character recognition. The key idea is to enhance and broaden image transformation techniques. The algorithm uses OpenCV to find the license plate by segmenting each character. The implementation has been completed using the K-Nearest Neighbor algorithm with the aid of the Python programming language.

Chapter 3

3.0 BACKGROUND STUDY

3.1 Digital Image Processing

The term "digital image processing" refers to using a digital computer to process digital images. To obtain an improved appearance or extract meaningful information, we can also state that the usage of computer algorithms is necessary. Use employment of algorithms and mathematical models in digital images enables the processing and analysis of digital images. Digital image processing aims to enhance image quality, extract valuable information from photographs, and automate operations using images [23].

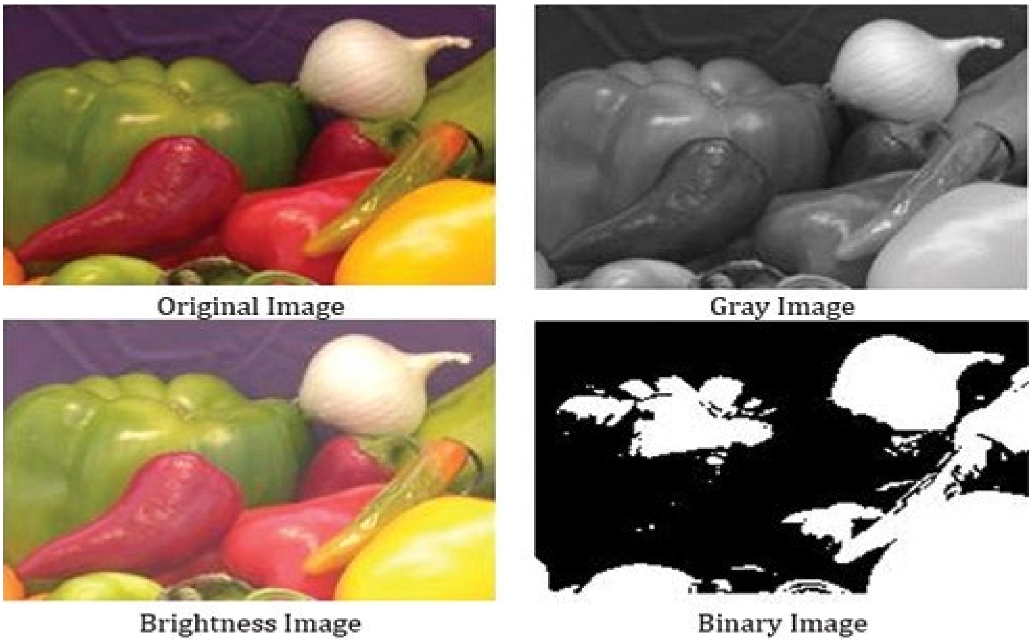

Figure 2: Digital Image Processing [24]

3.2 Computer Vision

Computer vision is a branch of computer science that aims to give computers the ability to recognize and comprehend people and objects in pictures and movies. Computer vision, like other forms of AI, aims to carry out and automate tasks that mimic human abilities [25].

3.3 Colour image to grayscale conversion

RGB to grayscale conversion is a process based on light perception and the human visual system. RGB images are represented as combinations of three primary colours, with each pixel having three colour channels with intensity values ranging from 0 to 255. Grayscale images are represented in a single channel, with each pixel having a unique intensity value that represents the brightness of the pixel. The grayscale colour space ranges from 0 (black) to 255 (white), with values in between representing different shades of grey. The theoretical concept of RGB grayscale conversion involves assigning an appropriate intensity value to each pixel. A common approach is to calculate the weighted sum of the three colour channels.

The formula is:

Gray = 0.299 * Red + 0.587 * Green + 0.114 * Blue.

The coefficients 0.299, 0.587, and 0.114 represent the relative contributions of red, green, and blue to the total light. These coefficients are based on the sensitivity of the human eye to certain hues. Our eyes are most sensitive to green light, followed by red and blue, so it receives the most emphasis. After applying this formula to each pixel of an RGB image, you obtain a grayscale image in which each pixel possesses a single intensity value that represents its brightness [26]. This process effectively removes the colour information from the original image and retains only the light information, resulting in a black-and-white image.

Figure 3: RGB to Grayscale conversion [27]

3.4 Noise reduction in image using filter

Denoising images with filters is a fundamental concept in image processing. It involves applying various filters to an image to remove or reduce unwanted noise while preserving the important details of the image. Different filters use various mathematical operations to reduce noise. Some noise reductions filters are: Smoothing filters (low-pass filters), Bilateral filter, Wiener filter, Non-local means filter, Wavelet denoising, Total Variation Denoising, Mean filter, Median filter, Gaussian filter [28].

- **Bilateral filter**

A bilateral filter is used to preserve edges while reducing noise and smoothing images. This article discusses a method that utilizes a median filter, while the other one explains a strategy that employs an averaging filter. However, since they blur out everything, whether it is noise or an edge, these convolutions often result in the loss of important edge information. The non-linear bilateral filter was developed to address this issue. The bilateral filter is an advanced technology that reduces noise while preserving edges. It takes into account both the spatial distance between pixels and the similarity of pixel values. This filter utilizes a weighted average, with the weights determined by both spatial and intensity variances. This reduces noise and keeps sharp edges intact [29].

Figure 4: Noise reduction using bilateral filter [30]

3.5 Edge detection techniques

Detecting abrupt variations in pixel intensity, which often correlate to object borders or significant image features, is a fundamental concept in image processing known as edge detection. Edge detection techniques aim to identify and emphasize these edges for subsequent analysis or processing. Some edge detection techniques are: Gradient-Based Methods (Sobel operator, Prewitt operator, Scharr Operator), Canny Edge Detection, Laplacian of Gaussian (LoG), Canny-Deriche edge detection, Marr-Hildreth edge detection, Edge detection using Wavelets, Structured edge detection [31].

- **Canny Edge Detection**

The Canny edge detector, which is an edge detection operator, utilizes a multi-stage method to identify various edges in images. In addition, Canny developed a computational theory of edge detection that explains how and why the method functions [32].

a. Smoothing: Use a Gaussian filter to reduce image noise.
b. Gradient calculation: Calculate the magnitude and direction of the gradient.
c. Non-Maximum Suppression: Suppress non-maximal gradients to get thin edges.
d. Double Threshold: Use two thresholds to classify edge pixels as strong, weak, or no edges.
e. Edge Tracking Using Hysteresis: Link weak edge pixels to strong pixels based on connectivity.

Figure 5: Edge detection [33]

3.6 OpenCV (Open-Source Computer Vision Library)

OpenCV (Open-Source Computer Vision Library) is a popular open-source computer vision and image processing library that provides a wide range of tools and functions for processing images, videos and computer vision related tasks. It is written primarily in C++, but provides bindings to several programming languages, including Python [34]. Here is an overview of the main functions and features of the Python OpenCV library:

1. View and upload images and videos.
2. Basic image processing
3. Image filtering and processing
4. Color space conversion
5. Object detection
6. Feature identification and association
7. Image transformation and perspective correction
8. Drawing on images
9. Video capture and processing
10. Integrating deep learning

These are just a few examples of what you can do with OpenCV in Python. The library is quite comprehensive and widely used in many fields, including computer vision, robotics, machine learning and more. The Open-Source Computer Vision Library is a common platform and a set of software functions for real-time applications. The open CV library contains several algorithms more than 500 optimized algorithms. It supports many of the improved algorithms for computer vision and automated learning, which are spread daily. Open CV has been developed with a major emphasis on real-time applications in order to achieve computational efficiency.

3.7 EasyOCR

Text recognition is another term for optical character recognition (OCR). An OCR program extracts and reuses data from scanned documents, camera images, and image-only PDFs. The original content can be accessed and edited using OCR software, which isolates letters in the image, converts them into words, and then forms sentences from the words. Furthermore, it eliminates the need for manual data entry [35].

EasyOCR is a Python library for Optical Character Recognition (OCR). OCR is a technology that enables computers to recognize and extract text from images or scanned documents. EasyOCR is built on PyTorch's deep learning framework and is designed to be user-friendly and seamlessly integrated into various applications. Here are some basic features and information about EasyOCR [36].

1. Language support: EasyOCR supports a wide range of OCR languages, making it useful for multilingual applications.
2. Model availability: EasyOCR offers pre-trained models trained on large datasets to recognize text in different languages and styles.

3. Usage: To use EasyOCR, you need to install the library using pip:

pip install easyocr

Then, you can import the library and use its functions to perform OCR on images:

import easyocr

reader = easyocr.Reader([‘en’ , ‘es’])

results = reader.readtext(‘image.jpg’)

4. Customization: EasyOCR allows you to customize some aspects of the recognition process, such as specifying the languages used for recognition and setting the confidence threshold of the model.

5. Applications: EasyOCR is useful for many different tasks, such as text extraction from scanned documents, photos, and even video feeds for real-time OCR.

6. Performance: EasyOCR's accuracy and performance may vary depending on input image quality, text complexity, and other factors. Highly stylized or noisy text may be less accurate.

7. Community: EasyOCR has gained popularity in the Python community due to its ease of use and relatively good performance. But like any technology, it's important to test it in your use case to make sure it meets your requirements.

3.8 Imutils

Imutils is a Python library that provides a set of handy functions to simplify and simplify common image manipulation tasks, especially when working with OpenCV. It aims to enhance OpenCV's image manipulation capabilities by offering straightforward functions for performing operations such as resizing, rotating, displaying, and other commonly used tasks [37]. The library is designed to make code more concise and legible, which is especially beneficial for computer vision projects.

1. Resizing images: The library has a resize() function that allows you to easily resize images while preserving their aspect ratio. This feature can be very useful when preparing images for further processing.
2. Displaying Images: The library includes the imshow() function, which simplifies displaying images using OpenCV's imshow() function. It manages window resizing and keyboard event handling.
3. Edge detection: imutils provides functions to detect the edge of images using methods such as Canny edge detection.
4. Contours: Features are available to simplify sketching operations, such as sorting outlines by size, calculating the center of an outline, and drawing outlines on images.
5. Installation: pip install imutils

Chapter 4

4.0 PROPOSED METHOD

This section describes the process of the license plate detection and recognition system. The proposed system has two parts. The first part detects and separates the vehicle license plate. In the second part, the vehicle license plate recognition module works using Optical Character Recognition (OCR). The figure 6 shows the block diagram of the proposed approach. This figure shows that the input image is first read by the system and then converted from RGB to grayscale. Remove the noise from the image by applying a bilateral filter. The detection of the image's edges is performed using the Canny Edge Detection technique. Find contours in the image. Next, plate detection is performed using contour analysis. Extract the detected license plate from the entire image. After that, character recognition is performed by applying OCR. Recognized characters are displayed in the output.

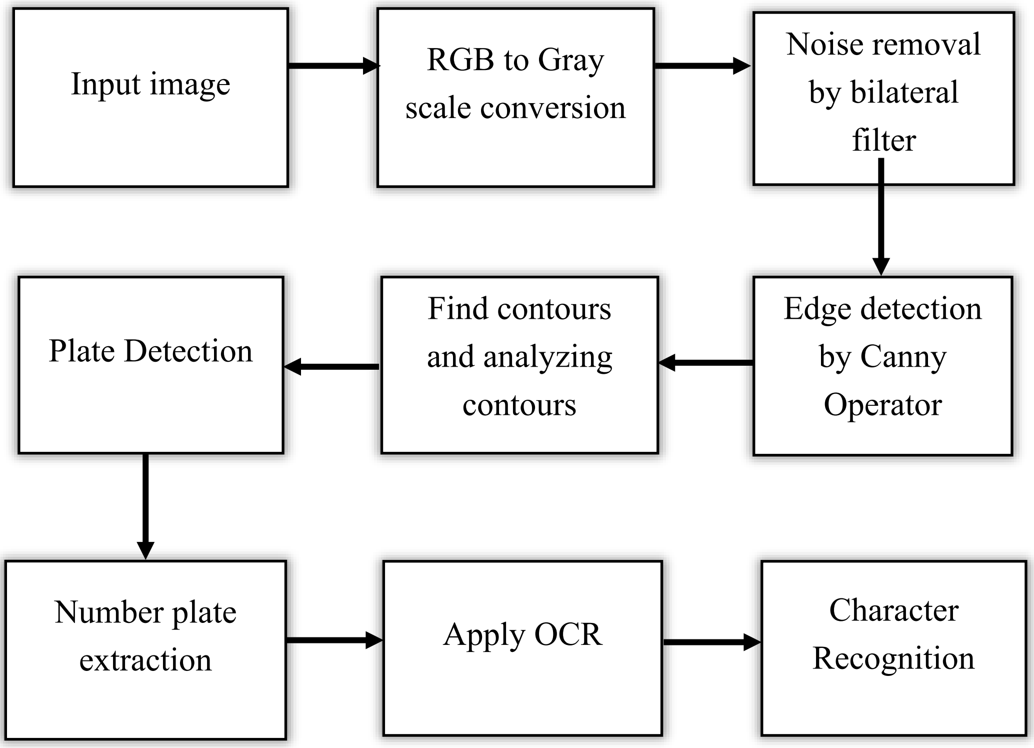

Figure 6: Block diagram of the proposed approach

4.1 Dataset

In this research work, we utilized a well-known dataset from Kaggle [38]. There are 200 images of vehicles. We have utilized the good images and identified the specific region where the license plate is detected. The recognition of the number from the number plate has been completed.

4.2 Algorithm

The steps involved in the license plate recognition process are described below.

Input: Load image file from camera.

Output: Vehicle number plate with letters.

a) Read the original image or the image captured by the camera as shown in Figure 7. If that is a video camera (CCTV), extract the image from the video.

Figure 7: Read the image

b) The second step is to convert the color image (RGB) to a grayscale image. If the image is in color, then we need to convert it to grayscale. In Figure 8, we can see the conversion of an image from RGB to grayscale.

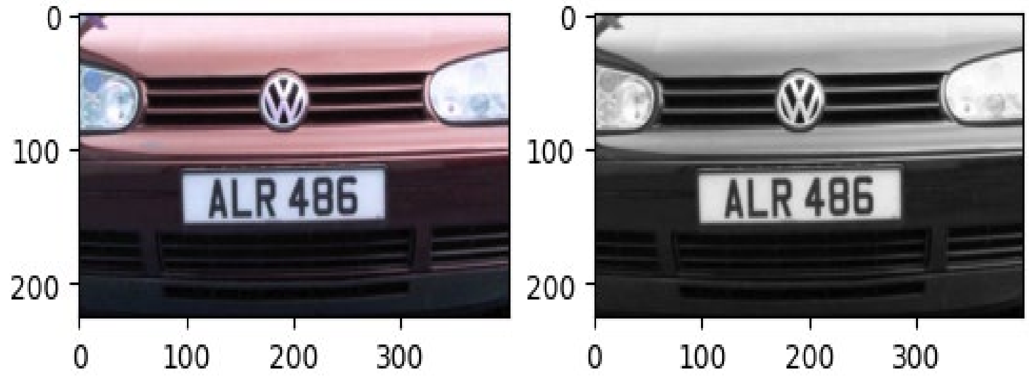

Figure 8: RGB to Grayscale conversion

c) The third step is to apply a bilateral filter for noise removal. The bilateral filter is a non-linear, edge-preserving, and noise-reducing image smoothing filter. Each pixel's intensity is changed to a weighted average of the intensity values of neighbouring pixels as shown in Figure 9.

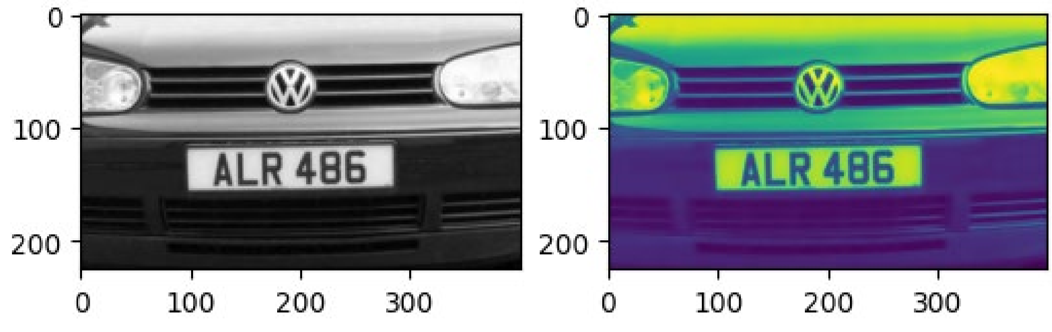

Figure 9: Noise reduction using bilateral filter

d) The next step is to find edges. For this, we have used the Canny edge detection technique. The Canny edge detector is an algorithm that is used to detect various types of edges in images which is shown in Figure 10.

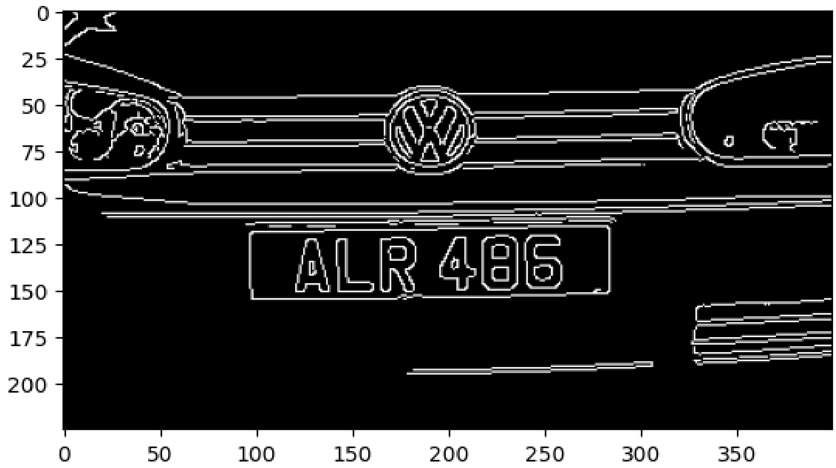

Figure 10: Edge detection using Canny's algorithm

e) The next step is to find the contour and then analyze it. We have extracted the contours from the image that were processed using the Canny edge detection algorithm. Here, we have used a binary image to detect contours. Then, we carefully observe the contours and arrange them in descending order based on their areas. The ten largest contours are retained. This is done to filter out smaller noise contours and focus on the most significant ones. By employing the trial-and-error approach, we adjusted the contours to ten in order to achieve a satisfactory outcome. Then we detect the quadrilateral shape from the filtered contours. For this, we approximate the shape of each contour. If a contour has exactly four vertices, it is considered a quadrilateral, and the coordinates of the vertices are saved.

f) The next step is to detect the license plate and crop it from the main image. A masking operation is performed for this purpose. A mask image is generated based on the observed quadrilateral vertices. The bitwise AND operation is used to extract the region of interest from the original image by applying a mask. The region of the detected license plate is shown in Figure 11.

Figure 11: Mask Image

g) The next step is to extract the license plate from the main image. For this, we cropped the region of interest from the original image. In Figure 12, the extracted license plate is shown.

Figure 12: License Plate

h) The next step is to recognize the characters from the license plate image which is shown in Figure 13. For this, we use the Optical Character Recognition (OCR) technique. In this research work, the EasyOCR library has been used.

```
[([[8, 12], [152, 12], [152, 36], [8, 36]], 'TN 99F2378', 0.9347226731961)]
```

Figure 13: Result of EasyOCR

i) Finally, we have found the number plate information and displayed it on the output device as shown in Figure 14.

Figure 14: Information about the detected license plate

Chapter 5

5.0 EXPERIMENTAL RESULTS

5.1 Experimental Setup

In this research work, we used Python (version 3.10). The experiment is conducted in a notebook environment. But this experiment can be done in any integrated development environment (IDE) with Python 3.10. The necessary libraries for this task include OpenCV, EasyOCR, and Imutils. They need to be installed in the system. It would be helpful if the system had an NVIDIA graphics card and the CUDA library installed.

5.2 Result

We conducted the experiment using 125 vehicle images. The step-by-step result of the output is showing below.

a) The first step is to load the test image into the system. In Figure 15, we can see that the test image has been loaded into the system.

Figure 15: Load test image

b) The next step involves preprocessing the images, which includes converting them to grayscale and applying a bilateral filter. In Figure 16, we can see that the test image has been converted to a grayscale image. The image is then filtered using a bilateral filter.

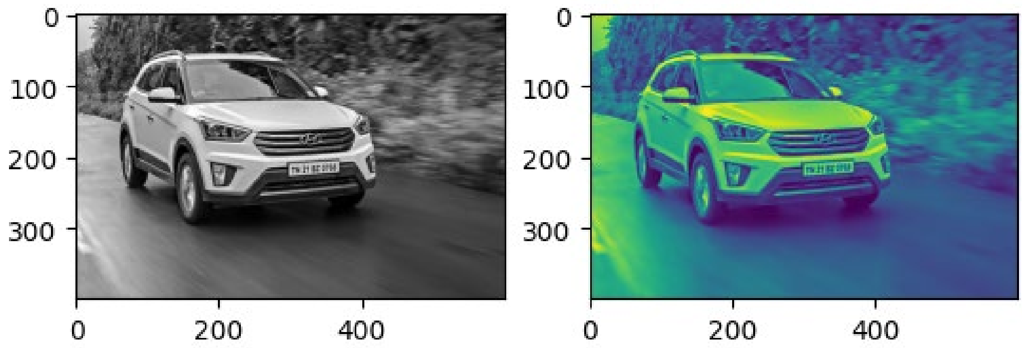

Figure 16: Pre-processing step

c) The third step is to detect the edges of the test image using the Canny edge detection algorithm, as shown in Figure 17. This edge detection technique yields satisfactory results in obtaining edges.

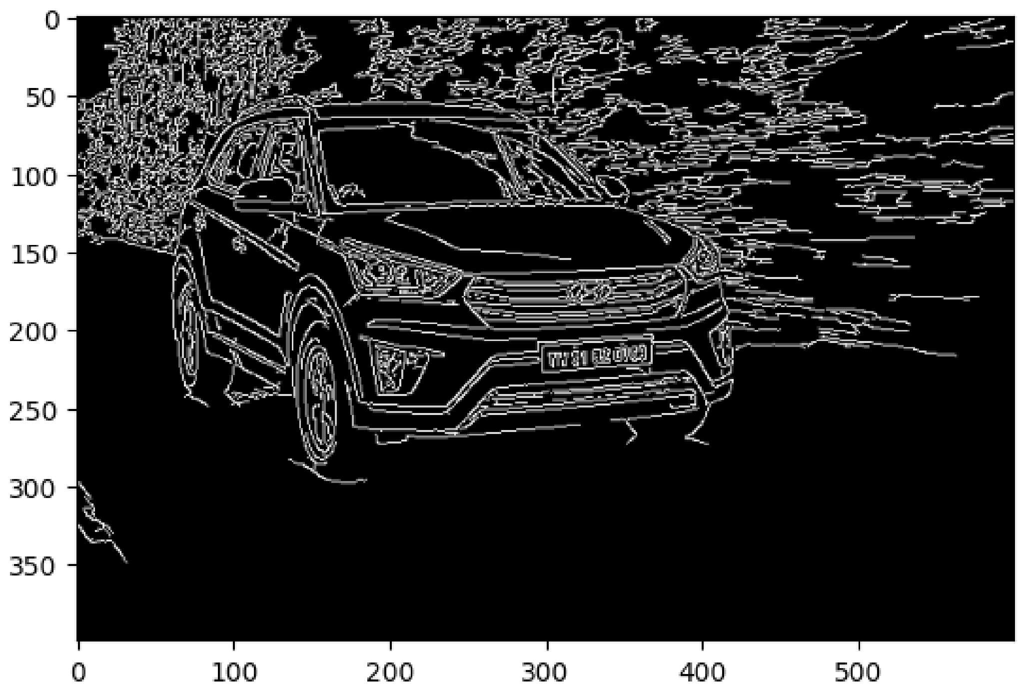

Figure 17: Result of edge detection

d) The fourth step is to detect the license plate area and apply a mask to it. The license plate detected in the test image, after applying a mask, is shown in Figure 18.

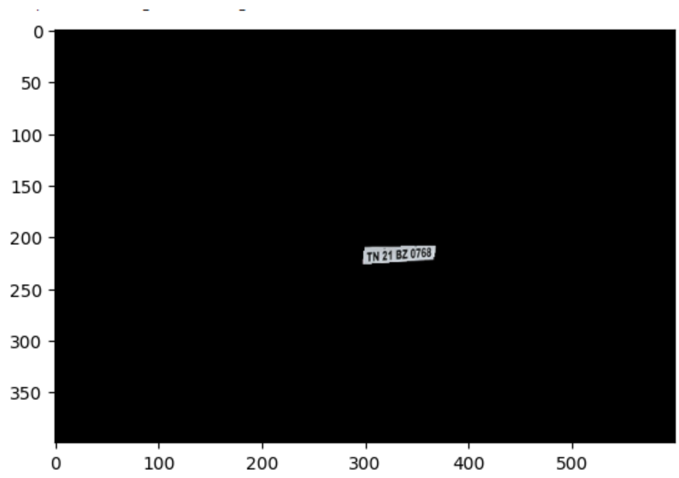

Figure 18: Detected License plate with masking

e) The fifth step involves cropping the detected area of the license plate from the entire test image, as shown in Figure 19.

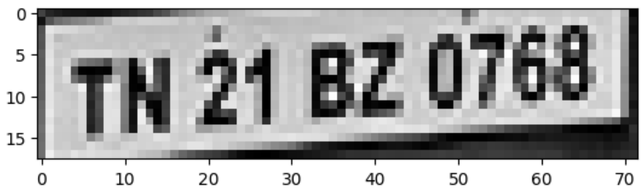

Figure 19: Crop license plate from image

f) The final step demonstrates that our system has successfully detected and recognized the license plate. After detecting the license plate area, recognition is performed using OCR, as shown in Figure 20.

Figure 20: Output result of ALPR

5.3 COMPARATIVE ANALYSIS

In this research work, we have designed an Automatic License Plate Recognition (ALPR) system to enhance accuracy. We have employed contour analysis in the license plate detection algorithm. This research work has achieved an algorithm detection accuracy of 93.60% and a recognition accuracy of 95.06%. This is slightly greater than the existing systems [6] [23]. The accuracies of these systems are shown in Table 1. The processing time of this system is 2 to 3 seconds. So, the research work also optimized the processing time of this system.

Table 1: Accuracy analysis

Name of the Operation		**Total No. of Samples**	**No. of Successful Samples**	**Failure Samples**	**Success Ratio**
Present research work	Detection of license plate	125	117	8	93.60%
	Character Recognition	81	77	4	95.06%
Previous research work [6]	Detection of license plate	15	14	1	93.34%
	Character Recognition	12	11	1	91.67%
Previous research work [23]	Detection of license plate	100	92	8	92%
	Character Recognition	88	83	5	94.30%

Chapter 6

6.0 CONCLUSION AND FUTURE SCOPE

In this research work, we were able to identify vehicle numbers using the proposed license plate detection and recognition algorithm. We have demonstrated how ANPR can be implemented using open-source computer vision libraries such as OpenCV and EasyOCR. Using our proposed method, we have successfully achieved a 93.60% accuracy rate in the detection algorithm and a 95.06% accuracy rate in the character recognition algorithm even in challenging lighting and weather conditions. Additionally, we have optimized the processing time of our system.

This research work focuses on an automatic number plate recognition system for vehicles. The system is built in a modular way, which makes it simple to upgrade and/or replace different sub-modules. This feature potentially makes it suitable for a variety of vision applications. In the future, we can create a method that directly retrieves information from government parivahan website about the owner of a license plate or car. It will be very helpful for security purposes. The system can become more robust if high-resolution cameras are used to enhance overall accuracy when implementing this system in real-time applications. The sensor can also be deployed in a way that the camera only takes pictures, when necessary, in order to conserve energy. This work can be extended to include more functionality, as mentioned in the following.

- ✓ Whether the registered vehicle is blacklisted or not.
- ✓ In this way, even an individual user can effectively monitor traffic and easily identify the location of a vehicle that is involved in a traffic violation.

REFERENCES

[1] S. Uma and M. Sharmila, “Implementation of License Plate Recognition System in ARM Cortex A8 Board,” in International Journal of Computational Engineering & Management, vol. 19, no. 3, pp. 9-13, May 2016, ISSN: 2230-7893.

[2] C. Gurav, V. Kamble, R. Gurav, P. N. Sakhalkar, and S. Mohite, “A Review Paper on Vehicle Number Plate Recognition,” International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 4, Apr. 2019, ISSN: 2278-0181.

[3] CEIC Data. “India Motor Vehicle Sales: FY: Year to Date: Quarterly: DM: Commercial Vehicle (CV),” [Online]. Available: https://www.ceicdata.com/en/india/automobile-sales-society-of-indian-automobile-manufacturers-siam-domestic/motor-vehicle-sales-fy-ytd-quarterly-dm-commercial-vehicle-cv. [Accessed: September 20, 2022].

[4] N. Abirami, Dr. J. S. Leena, and Jasmine, “Accurate vehicle number plate recognition and Real-time Identification using Raspberry Pi,”in International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 4, pp. 569-573, 2018.

[5] A. Menon and B. Omman, “Detection and Recognition of Multiple License Plates from Still Images,” inInternational Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 2018.

[6] M.Samantaray, A. K. Biswal, D. Singh, D. Samanta, M. Karuppiah, and N. P. Joseph, “Optical Character Recognition (OCR) based Vehicle's License Plate Recognition System Using Python and OpenCV,” in5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2021.

[7] L. Lubna, N. Mufti, and S.A.A. Shah, "Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms," Sensors (Basel), vol. 21, no. 9, pp. 3028 - 3030, April 2021.

[8] S. Singh, “Automatic Car License Plate Detection System: A Review,” Materials Science and Engineering, 2021.

[9] S. Ghulyani, D. Jain, P. Singh, S. Joshi, and A. Ahlawat, “A Data Entry Optical Character Recognition Tool Using Convolutional Neural Networks,”in IEEE IAS Global Conference on Emerging Technologies (GlobConET), pp. 721-728, May 2022.

[10] M. Karabatak and T. Mustafa, “Performance Comparison of Classifiers on Reduced Phishing Website Dataset,” in 2018 6th International Symposium on Digital Forensics and Security (ISDFS) pp. 1-5, March 2018.

[11] J. Jameson, S. N. H. Sheikh Abdullah, N. N. A. Nik Ghazali, and N. A. Zamani, “Multiple Frames Combination Versus Single Frame Super Resolution Methods for CCTV Forensic Interpretation,” Journal of Information Assurance & Security, vol. 8, no. 5, 2013.

[12] B.Zitova and J. Flusser, “Image Registration Methods: A Survey,” Image and Vision Computing, vol. 21, no. 11, pp. 977-1000, 2003.

[13] N. Vishwanath, S. Somasundaram, M. R. Ravi, and N. K. Nallaperumal, “Connected Component Analysis for Indian License Plate Infra-Red and Color Image Character Segmentation,” in 2012 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-4, IEEE, 2012.

[14] R. Bhat and B. Mehandia, “Recognition of Vehicle Number Plate Using MATLAB,” International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, vol. 2, no. 8, pp. 1899-1903, 2014.

[15] V. Laksmi Priya and K. Perumal, “Detecting the Car Number Plate Using Segmentation,” International Journal of Engineering and Computer Science, vol. 3, no. 10, pp. 8823-8829, October 2014.

[16] R. A. Chavhan, “Real-Time Vehicle License Plate Recognition,” presented at the 6th International Conference on Recent Trends in Engineering & Technology (ICRTET-2018), Nashik, 2018.

[17] A. Wang, X. Liu, Y. Han, and C. Qi, “License Plate Location Algorithm Based on Edge Detection and Morphology,” in 7th International Forum on Strategic Technology (IFOST), 2012.

[18] M. S. Mandi, B. Shibwabo, and K. M. Raphael, “An Automatic Number Plate Recognition System for Car Park Management,” International Journal of Computer Applications, vol. 175, no. 7, October 2017.

[19] A. Mousa, “Canny Edge-Detection Based Vehicle Plate Recognition,”International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 5, no. 3, pp. 1-8, 2012.

[20] K. I. Kim, K. Jung, and J. H. Kim, “Color Texture-Based Object Detection: An Application to License Plate Localization,” in Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines, pp. 293-309, 2002.

[21] T. D. Duan, D. A. Duc, and T. L. H. Du, “Combining Hough Transform and Contour Algorithm for Detecting Vehicles' License Plates,” in Proceedings of the IEEE International Symposium on Intelligent Multimedia, Video, and Speech Processing, pp. 747-750, 2004.

[22] P. L. Kumari, R. Tharuni, I. V. S. Sai Vasanth, and M. Vinay Kumar, “Automatic License Plate Detection using KNN and Convolutional Neural Network,” in 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1634-1639, 2022.

[23] GeeksforGeeks, “Digital Image Processing Basics,” [Online]. Available:https://www.geeksforgeeks.org/digital-image-processing-basics/.[Accessed: January 12, 2023].

[24] "Digital Image Processing Using MATLAB," in "Title of the Book or Collection,"IntechOpen, 2017. [Online]. Available: https://www.intechopen.com/chapters/51312.[Accessed: January 12, 2023].

[25] Microsoft, "What is Computer Vision?" Microsoft Azure, [Online]. Available: https://azure.microsoft.com/en-in/resources/cloud-computing-dictionary/what-is-computer-vision/#:~:text=Computer%20vision%20is%20a%20field,tasks%20that%20replicate%20human%20capabilities.[Accessed: January 14, 2023].

[26] "How to Convert RGB Image to Grayscale,"Baeldung, [Online]. Available: https://www.baeldung.com/cs/convert-rgb-to-grayscale. [Accessed: October 20, 2022].

[27] "Convert RGB Image to Grayscale Image Using OpenCV,"Lindevs, [Online]. Available: https://lindevs.com/convert-rgb-image-to-grayscale-image-using-opencv.[Accessed: October 20, 2022].

[28] GeeksforGeeks, "What are Different Types of Denoising Filters in MATLAB?" [Online]. Available: https://www.geeksforgeeks.org/what-are-different-types-of-denoising-filters-in-matlab/. [Accessed: October 21, 2022].

[29] GeeksforGeeks, "Python Bilateral Filtering," [Online]. Available: https://www.geeksforgeeks.org/python-bilateral-filtering/.[Accessed: October 21, 2022].

[30] Wikipedia, “Bilateral Filter,” [Online]. Available: https://en.wikipedia.org/wiki/Bilateral_filter.[Accessed: October 21, 2022].

[31] TutorialsPoint, “Concept Of Edge Detection,” [Online]. Available:https://www.tutorialspoint.com/dip/concept_of_edge_detection.htm.[Accessed: October 24, 2022].

[32] “Canny Edge Detection Step by Step In Python - Computer Vision,” Towards Data Science, [Online]. Available: https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123.[Accessed: October 24, 2022].

[33] MathWorks, “What Is Edge Detection?” [Online]. Available: https://in.mathworks.com/discovery/edge-detection.html.[Accessed: October 24, 2022].

[34] OpenCV,“About OpenCV,” [Online]. Available: https://opencv.org/about/.[Accessed: January 25, 2023].

[35] IBM, “What Is Optical Character Recognition (OCR)?” [Online]. Available: https://www.ibm.com/blog/optical-character-recognition/.[Accessed: February 2, 2023].

[36] “Hands-On Tutorial On EasyOCR For Scene Text Detection In Images,” Analytics India Magazine, [Online]. Available: https://analyticsindiamag.com/hands-on-tutorial-on-easyocr-for-scene-text-detection-in-images/.[Accessed: February 2, 2023].

[37] GitHub, "PyImage Search/imutils," [Online]. Available: https://github.com/PyImageSearch/imutils.[Accessed: February 4, 2023].

[38] Kaggle, "Car License Plate Detection," [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/car-plate-detection.[Accessed: November 10, 2022].

APPENDIX

Python Code –

```
!pip install easyocr

!pip install imutils

!pip install opencv-python


import cv2

from matplotlib import pyplot as plt

import numpy as np

import imutils

import easyocr


image_path = 'Car/images/Cars332.png'

img = cv2.imread(image_path)

if img is None:

    print(f"Failed to load the image at path: {image_path}")


# Display the original image

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

plt.title('Original Image')

plt.show()
```

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) # Noise reduction
edged = cv2.Canny(bfilter, 30, 200) # Edge detection

fig = plt.figure()
subplot1 = fig.add_subplot(1, 2, 1)
subplot1.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
subplot2 = fig.add_subplot(1, 2, 2)
subplot2.imshow(bfilter)

plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
plt.title('Edged Image')
plt.show()

keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE,
                cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
location = None
```

```
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break

location     #Give the Coordinate

mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
plt.title('Isolated Object')
plt.show()

(x, y) = np.where(mask == 255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
plt.title('Cropped Image')
plt.show()


reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result     #Give the result OCR


text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=font, fontScale=1, color=(0, 255, 0), thickness=2, lineType=cv2.LINE_AA)
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0, 255, 0), 3)


plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))
plt.title('Final Result')
plt.show()
```