VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"Automatic Number Plate Recognition System"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

KARTHIK HOSUR (1CR16CS067)

LOHITH JAGANATHAN (1CR16CS076)

MANJUNATH RAMESH (1CR16CS083)

NIKITA PARIKH (1CR16CS105)

Under the Guidance of,

Mr. Kartheek G.C.R

Asst. Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Automatic Number Plate Recognition System"** carried out by **Mr. Karthik Hosur**, USN **1CR16CS067**, **Mr. Lohith Jaganathan**, USN **1CR16CS076**, **Mr. Manjunath Ramesh**, USN **1CR16CS083, Ms. Nikita Parikh**, USN **1CR16CS105,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Prof. Kartheek G.C.R	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**Automatic Number Plate Recognition System**" has been successfully completed under the guidance of Prof. Kartheek G.C.R, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bengaluru

Date: 7th June,2020

Team members:

KARTHIK HOSUR (1CR16CS067) _____

LOHITH JAGANATHAN(1CR16CS076) _____

MANJUNATH RAMESH (1CR16CS083) _____

NIKITA PARIKH(1CR16CS105) _____

ABSTRACT

Automatic number-plate recognition is a technology that uses optical character recognition on images to read vehicle registration plates using OpenCV and Tesseract OCR Engine. It can be used on existing closed-circuit television, road-rule enforcement cameras, or cameras specifically designed for the task. Using Selenium web driver, number plate recognized is parsed to the government website vahan.nic.in along with the solved captcha and the vehicle details can be accessed for further Inference and analysis. The crawled information is converted to structured and unstructured data and stored in Firebase and MySQL for data analysis and live dashboard. Through the dashboard the notification triggers can be set if a vehicle defaults any of the rules , an SMS will be sent to the mobile phone of the authority. Tested on 1500 Indian Number Plates gave us a success rate of 64% which is better than the current existing systems. As well as, successfully retrieve vehicle information from secure government website with a success rate of 75%.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Mr. Kartheek G.C.R. , Assistant Professor,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANPR	**Automatic Number Plate Recognition**
LP	**License Plate**
SVM	**Support Vector Machine**

Chapter 1

INTRODUCTION

Automatic Number Plate Recognition or ANPR is a technology that uses pattern recognition to 'read' vehicle number plates. In simple terms ANPR cameras 'photograph' the number plates of the vehicles that pass them. This 'photograph' is then fed in a computer system to find out details about the vehicle itself. ANPR consists of cameras linked to a computer. As a vehicle passes, ANPR 'reads' Vehicle Registration Marks – more commonly known as number plates - from digital images, captured through cameras located either in a mobile unit, in-built in traffic vehicles or via Closed Circuit Television (CCTV). The digital image is converted into data, which is processed through the ANPR system. We proposed a method mainly based on edge detection, OCR operation and Finding Rectangles in a Vehicle Image.

Owning a vehicle today is not merely a symbol of luxury but has become a necessity. However, considering vehicles, any catastrophic situation can take place. Therefore there is always an urgent need to arrange appropriate measures to increase the safety, security as well as monitor the vehicles to avoid any mishap. It would help us in the situations such as: Instantaneously obtain vehicle details using image processing. Allowing an agency to detect the location of its vehicles. Automatically notify the user if there are traffic violations registered to the vehicle. One such measure is the use of a vehicle tracking system using the GPS (Global Position System). Such a tracking system includes a mechanized device that is equipped in a vehicle. Using software present at an operational base, it helps track the location of the vehicle. This base station is used for monitoring purposes. It is accompanied by maps such as Google maps, Here maps, Bing maps etc for the representation of the location.

ANPR can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver. Systems commonly use infrared lighting to allow the camera to take the picture at any time of the day. A powerful flash is included in at least one version of the intersection monitoring cameras, serving both to illuminate the picture

and to make the offender aware of his or her mistake. ANPR technology tends to be region-specific, owing to plate variation from place to place.

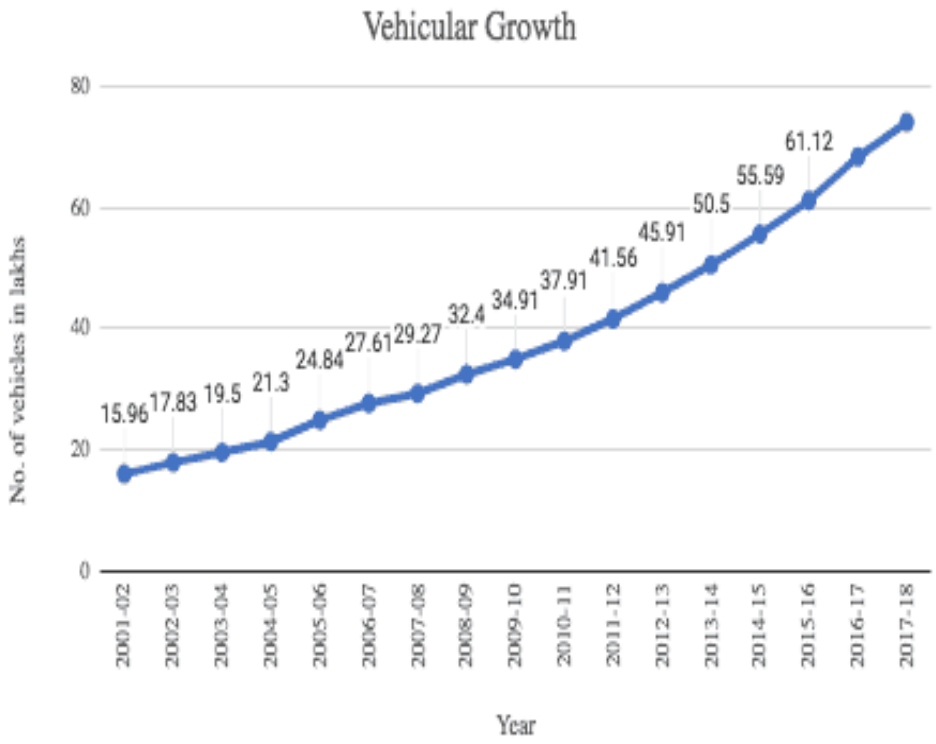

Fig 1.1 Vehicle growth in India

The acquisition of digital image usually suffers from undesirable camera shakes and due to unstable random camera motions. Hence image enhancement algorithms are required to remove these unwanted camera shakes. . Python is used as the main programming language.

We have connected to the http://vahan.nic.in with the ANPR system to extract all the vehicle and owner details.We extract the information and save the data in JSON format for further processing and analysis.

1.1 Existing system:

1.1.1 Online ANPR framework: In an online ALPR framework, the limitation and elucidation of tags occur promptly from the approaching video outlines, enabling Real-time tracking through the surveill ance camera.

Example :OpenALPR CloudWatch

1.1.2 Offline ANPR framework: A logged off ALPR framework, interestingly, catches the shovel, dumper number plate pictures and stores them in a concentrated information server for further preparation, i.e. for translation of vehicle number plate

Example :OpenALPR Library

Looking at the works of other countries pushes and inspire us to try to solve the challenges that we face in our country and also motivates us to use ANPR in all facets.

United States

Mobile ANPR use is widespread among US law enforcement agencies at the city, county, state and federal level. According to a 2012 report by the Police Executive Research Forum, approximately 71% of all US police departments use some form of ANPR. Mobile ANPR is becoming a significant component of municipal predictive policing strategies and intelligence gathering, as well as for recovery of stolen vehicles, identification of wanted felons, and revenue collection from individuals who are delinquent on city or state taxes or fines, or monitoring for "Amber Alerts".

United Kingdom

The Home Office states the purpose of automatic number-plate recognition in the United Kingdom is to help detect, deter and disrupt criminality including tackling organised crime groups and terrorists. Vehicle movements are recorded by a network of nearly 8000 cameras capturing between 25 and 30 million ANPR ‘read’ records daily. These records are stored for up to two years in the National ANPR Data Centre, which can be accessed, analysed and used as evidence as part of investigations by UK law enforcement agencies.

Saudi Arabia

Vehicle registration plates in Saudi Arabia use white background, but several vehicle types may have a different background. United States diplomatic plates have the letters 'USD', which in Arabic reads 'DSU' when read from right to left in the direction of Arabic script. There are only 17 Arabic letters used on the registration plates.A Challenge for plates recognition in Saudi Arabia is the size of the digits. Some plates use both Eastern Arabic numerals and the 'Western Arabic' equivalents. A research with source code is available for APNR Arabic digits.

Turkey

The system has been used with two cameras per lane, one for plate recognition, one for speed detection. Now the system has been widened to network all the registration number cameras together, and enforcing average speed over preset distances. Some arteries have 70 km/h (43 mph) limit, and some 50 km/h (31 mph), and photo evidence with date-time details are posted to registration address if speed violation is detected. As of 2012, the fine for exceeding the speed limit for more than 30% is approximately US$175.

Canada

The police service in Ontario uses automatic licence-plate recognition software[21] to nab drivers behind the wheels of vehicles with Ontario number plates..

1.1.3 Challenges in the existing system:

In the created nations the qualities of the vehicle number plate are entirely kept up. For instance, the measure of the plate, shade of the plate, text style face/size/shade of every character, dispersing between ensuing characters ,the quantity of lines in the vehicle number plate, script and so on are kept up particularly. A portion of the pictures of the standard tags utilized as a part of created nations. In most academic institutions and car parks, the ongoing car park entry registration process for visitors, staff or students entering the institution involves a security guard having to confirm membership details by checking for membership sticker on the windscreen of the vehicle or by checking the driver's identification card. This process of writing is tedious and time consuming and is prone to inaccurate recordings, furthermore the backup and sharing of this vehicle information is difficult because the data is hard copy.

A city like Bangalore has multiple apartment complexes and societies, most of them also verify by checking for membership sticker on the windscreen of the vehicle. If a stranger or unknown vehicle enters, they are required to register which is time consuming. Most complexes even consider it unsafe as once a vehicle enters it is hard to track the movement of the members of the vehicle. Security issues are the main drawback with many cars being stolen, especially when they are left at parking lots even if for a few hours, it is hard to keep a record of all the vehicles entering/exiting at peak usage times.

Thus keeping in mind these drawbacks of the traditional system we aim to get a step ahead and address each of them individually when building our solution.

Automatic license plate recognition has two major technological requirements:

1. The quality of the license plate recognition algorithms.

2. The quality of the image acquisition (camera and the illumination conditions) The better algorithms are:

2.1. Higher is the recognition accuracy.

2.2 Faster is the processing speed.

2.3. Wider is the range of picture quality it can be used on.

3. Varying Indian Number Plate Formats

By and large, one LPR program can read plates from one specific nation just .This is on the grounds that the geometrical structure of the plate and introduction, text styles, and grammar were imperative parts of the LPR system. Without the earlier information of the plate geometry (character distribution, character spacing, plate color, dimension ratios etc.), the algorithm may not even find the plate in the captured image.

1.2 Proposed system:

Automatic Number Plate Recognition using an efficient OCR engine like Pytesseract along with major and vast libraries of OpenCV for image processing. As we have seen so far ANPR covers as a solution to most of the problems we have posed. We would like to dig a bit deeper now and highlight the scope of the project and the extent to which we can push the boundaries.The main issue that is usually recognized when it comes to number plate detection is the noise that is added to the image in the process of capturing the image or due to the environment around, taking that into consideration we can say that using our system, we can implement it in all environments, be it rain or even in the dark.Usually when any new system is proposed to possible clients, their main concern is the addition of new features into their existing system. Keeping this in mind we can say for sure that our system can be integrated to the pre-existing infrastructure of most clients.

Using a web crawler, number plate recognized is parsed to the government website vahan.nic.in along with the solved captcha and the vehicle details can be accessed for further Inference and analysis. Also showcase the vulnerabilities in the security of the government websites and privacy issues in government website. Also provide analytics and solution on the extracted data.

1.2.1 Advantages of the proposed system:

- To perform successful and efficient preprocessing on the raw RGB image
- To exploit the high performance and effectiveness of OpenCV and Pytesseract framework to detect and recognize LP of vehicles,to improve our system reliability.
- To correctly determine the number plate based on Indian Number plate Standards
- To Successfully extract the information from Government vehicle information database
- To Show the security vulnerabilities on vahan.nic.in

1.3 Market Value of the ANPR System

According to the new market research report "Automatic Number Plate Recognition (ANPR) System Market by Type (Fixed, Mobile, Portable), Component (ANPR Cameras, Software, Frame Grabbers, Triggers), Application (Traffic Management, Law Enforcement, Electronic Toll

Collection, Parking Management), and Geography - Global Forecast to 2023", the automatic number plate recognition (ANPR) system market in 2016 was valued at USD 1.78 Billion and is expected to reach USD 3.57 Billion by 2023, at a CAGR of 9.74% between 2017 and 2023.

Factors that are driving this market include the infrastructure growth in emerging economies, increasing allocation of funds by various governments on intelligent transport systems (ITS), deployment of camera technologies in security and surveillance, and traffic enforcement application, and the growing usage of video analytics technology for intelligent monitoring of vehicles.

Europe was the largest market for ANPR systems in 2016. The large market in this region can be attributed to the high adoption of intelligent transportation systems for traffic management, tolling management, law/police enforcement, and other applications. The market in Europe has been segmented into Germany, the UK, France and the Rest of Europe. Some major companies offering ANPR systems in Europe include ARH Inc. (Hungary), Digital Recognition Systems Ltd. (UK), NDI Recognition Systems Ltd. (UK), and Q-Free ASA (Norway).

The key players in the market include Kapsch TrafficCom AG (Austria), Conduent, Inc. (US), QFree ASA (Norway), Siemens AG (Germany), Genetec Inc. (Canada) Neology, Inc. (US), Bosch Security Systems GmbH (Germany), Tattile srl (Italy), TagMaster North America, Inc. (US), NDI Recognition Systems Ltd. (UK), Euro Car Parks Limited (UK), Quercus Technologies, S.L. (Spain) Vigilant Solutions, Inc. (US), Elsag North America, LLC (US), ARH Inc. (Hungary), Digital Recognition System Ltd. (UK), Beltech BV (Netherlands), ANPR International Ltd. (UK), HTS (New York), FF Group (Cyprus), and so on.

This report categorizes the global ANPR system market on the basis of type, component, application, and geography. The report describes the drivers, restraints, opportunities, and challenges for the growth of this market.

1.4 Project Report Methodology:

The total report is composed in the following way:

Chapter 2: Literature Survey- The literature survey is an effective tool to research about the current developments in the related field and their drawbacks, in order to incorporate a better mechanism in the proposed system.

Chapter 3: Theoretical Background – This contains information about the underlying technologies and algorithms that have been made use of in the architecture and design of the proposed system.

Chapter 4: System Requirement Specification – This chapter deals with the various functional and non functional requirements that are to be implemented in the proposed system. It also encloses both hardware and software requirement specifications that are needed to run the software.

Chapter 5: System Design – This chapter deals with the architecture and design of the system. Various diagrams have been included that describe the working of the system in a way that can be understood better.

Chapter 6: Implementation – The algorithms that form the proposed system have been listed, along with important modules of code.

Chapter 7: Results and testing – This chapter discusses the obtained results and the outcome of the testing process.

Chapter 8: Conclusion and future work – This chapter concludes the discussion of the work done so far. Future enhancements that can be made to the work are also discussed in this chapter.

References – This chapter lists the various websites and books that have been referred to, in the making of this project and the project report.

Chapter 2

LITERATURE SURVEY

2.1 INTRODUCTION:

Literature survey is the process in which a complete and comprehensive review is conducted encompassing both the published and unpublished work from other alternative sources of information. This review is conducted in the domains of specific interest to the person or researcher. Further, the results of this process are documented. This entire process comes in aid of the researcher to address the important and relevant aspects of the research that had not been addressed prior to the conduction of this research. Therefore it can be understood that the conduction of literature survey is necessary for the process of gathering secondary data for the research which might prove to be extremely helpful in the research and also designing the architecture of the project. There can be multiple reasons behind the purpose of conducting literature survey.

2.2 PAPER 1:

Title:

Amninder Kaur, Sonika Jindal ,Richa Jindal "License Plate Recognition Using Support Vector Machine (SVM)" Dept. Of Computer Science, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 7.

Context:

ANPR is a mass surveillance system that captures the image of vehicles and recognizes their license number. In this paper, A system is proposed that incorporates to successfully locate and read Indian vehicle number plates in digital images by using SVM. In this proposed model pre-processing and number plate localization is performed by using —Otsu's methods and —feature based localization methods respectively. It gives reliability and time optimization. Finally, the character reorganization performs using the Support Vector Machine.

In this paper, another algorithm to number recognition is proposed. This technique uses a Support Vector Machine (SVM) to train character samples and obtain the rules that are used to recognize

the numbers on number plates. SVM is forcefully competing with many methods for pattern classification. An SVM is a supervised learning technique first discussed by Vatnik.

SVM takes Statistical Learning Theory (SLT) as its theoretical foundation, and the structural risk minimization as its optimal object to realize the best generalization. They are based on some simple ideas and provide a clear intuition of what learning from examples is all about. More importantly, they possess the feature of high performance in practical applications. From the 1960s to the present, SVMs have become more and more important in the field of pattern recognition.

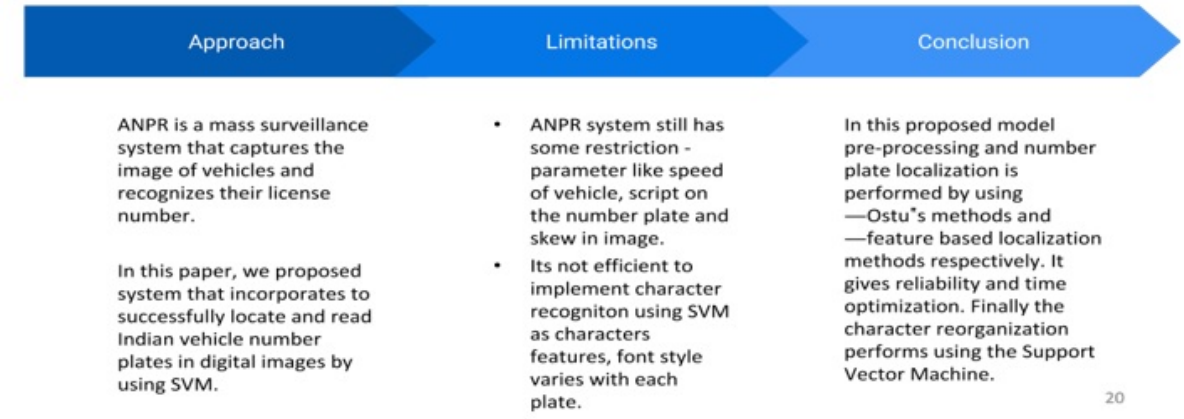

Fig 2.1 Literature Survey on SVM

2.3 PAPER 2:

Title:

ANISH LAZRUS,SIDDHARTHA CHOUBEY,SINHA G.R.,"AN EFFICIENT METHOD OF VEHICLE NUMBER PLATE DETECTION AND RECOGNITION" Department of Computer Science, International Journal of Machine Intelligence, Volume 3, Issue 3.

Context:

The images of various vehicles have been acquired manually and converted into grayscale images. Then the Wiener2 filter is used to remove noise present in the plates. The segmentation of grayscale image generated by finding edges using Sobel filter for smoothing image is used to

reduce the number of connected components and then Bilateral filter is used to calculate the connected component. Finally, a single character is detected.

However, sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots, which has been captured clearly Due to rapidly increase in number of vehicles across the world's big cities, vehicle number plate recognition system has become one of the most important digital image processing systems to be used. This system will solve so many problems for these city facilities which are hard to be controlled by humans 24 hours.

S. N.	Research papers	Real time data	Images correctly detected	character known	Results
1	Kok Kiaw T et al. (2003)	60	49	50	83 %
2	F.Martin et al (2002)	75	67	66	88 %
3	**Proposed Method**	50	46	49	**98 %**

Fig 2.2 Performance Matrix

Overall the vehicle license plate recognition software had been successfully designed and developed to recognize the 38 different characters using correlation in two dimensions.

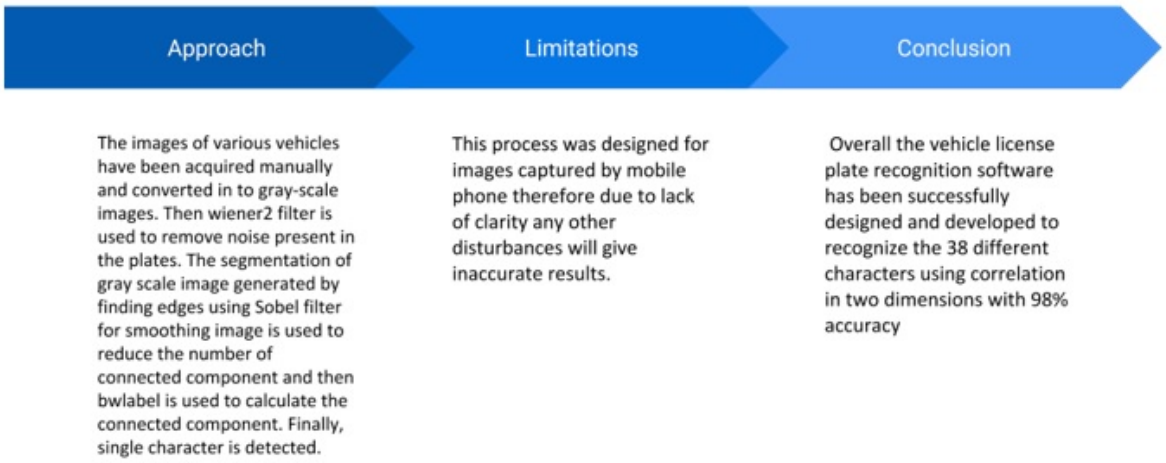

Fig 2.3 Literature Survey on Vehicle Number Plate Recognition

2.4 PAPER 3:

Title:

Abhay Singh, Anand Kumar Gupta ,Anmol Singh, Anuj Gupta ,Sherish Johri, "VEHICLE NUMBER PLATE DETECTION USING IMAGE PROCESSING", Department of IT, Volume: 05 Issue: 03 | Mar-2018

Context:

In this technology we will be working on CCTV footage or input images given. The CCTV footage must be clear to extract the Vehicle number from the image taken as Input. These input images are converted to grayscale and characters are segmented and recognised using OCR. There are some conditions for this software to work:

1) Vehicle plates should be white and according to the rules given by the government of India.
2) Image should be of appropriate brightness and contrast: In this, a software is designed which detects the vehicle number plate number using MATLAB.

In this technique we will be performing several methods step by step to find the vehicle number. Then using that vehicle number found we will be comparing that number from our database.

Fig 2.4 Literature Survey on images extract from CCTV

2.5 PAPER 4:

Title:

Ganesh R. Jadhav, Kailash J. Karande, "Automatic Vehicle Number Plate Recognition for Vehicle Parking Management System", IISTE, Vol.5, No.11, 2014.

Context:

This paper discusses a method for the vehicle number plate recognition from the image using mathematical morphological operations. The main objective is to use different morphological operations in such a way that the number plate of vehicles can be identified accurately.

This is based on various operations such as image enhancement, morphological transformation, edge detection and extraction of number plates from vehicle images. After this segmentation is applied to recognize the characters present on the number plate using template matching. This algorithm can recognize the number plate quickly and accurately from the vehicle's image.

The goal of the research is to investigate the possibility to create a comprehensive system for Indian vehicle identification based on the license plate recognition. In that case no additional hardware, such as e.g. transmitters, mounted on a vehicle, and responders will be required. The system performs well on various types of vehicle license plate images.

The recognition errors of letters and numbers mainly occur in some of the characters with the very similar main structures but some detailed differences, such as B and 8, O and 0, S and 5

You have to resize each character based on standardized size in this method, which becomes an added step in pre-processing and increases time needed.

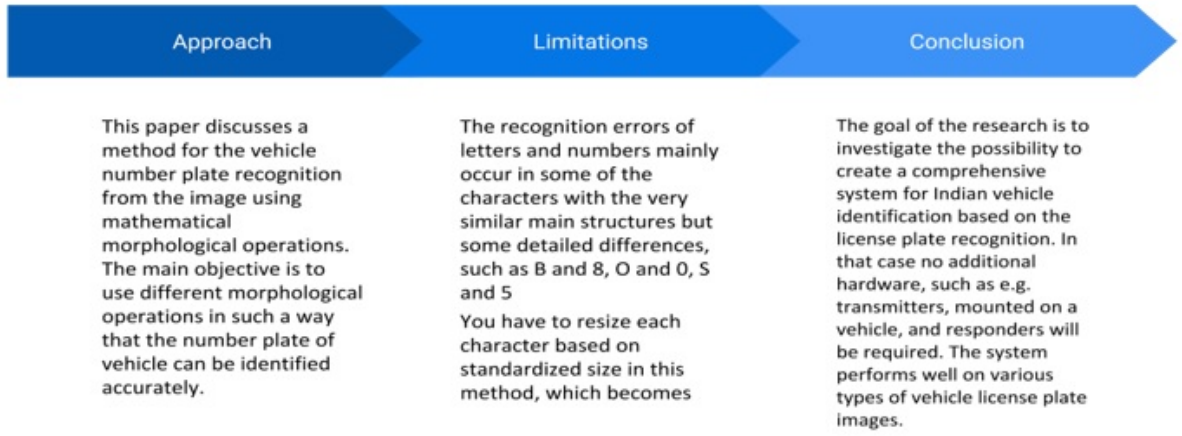

Fig 2.5 Literature Survey on Vehicle Parking Management System

2.6 PAPER 5:

Title:

Mutua Simon Mandi ,Bernard Shibwabo, Kaibiru Mutua Raphael, ”An Automatic Number Plate Recognition System for Car Park Management”, International Journal of Computer Applications, Volume 175 – No.7, October 2017

Context:

It proposes the adoption of a mobile based software solution that has ANPR capabilities to aid in vehicle identification and vehicle registration. The software that was developed adopted an object-oriented analysis and design methodology and implements Optical Character Recognition. (OCR) using the mobile device camera to detect and capture the vehicle number plate.

The proposed system turned out to be efficient when it came to implementation of automatic number plate recognition system for car park management, using Optical Character Reader (OCR) on a mobile device.

Successful implementations of ANPR systems have resulted in faster and easier vehicle identification. This has also resulted in faster and easier search and retrieval of vehicle information mostly done by law enforcers in identifying vehicles that are uninsured, stolen, or driven by someone without a license or prohibited from driving. It was recorded that the system required 1/5th of the original time that was needed by the manual system. Requires an efficient Local Area Network Systems need to be integrated to be efficient.

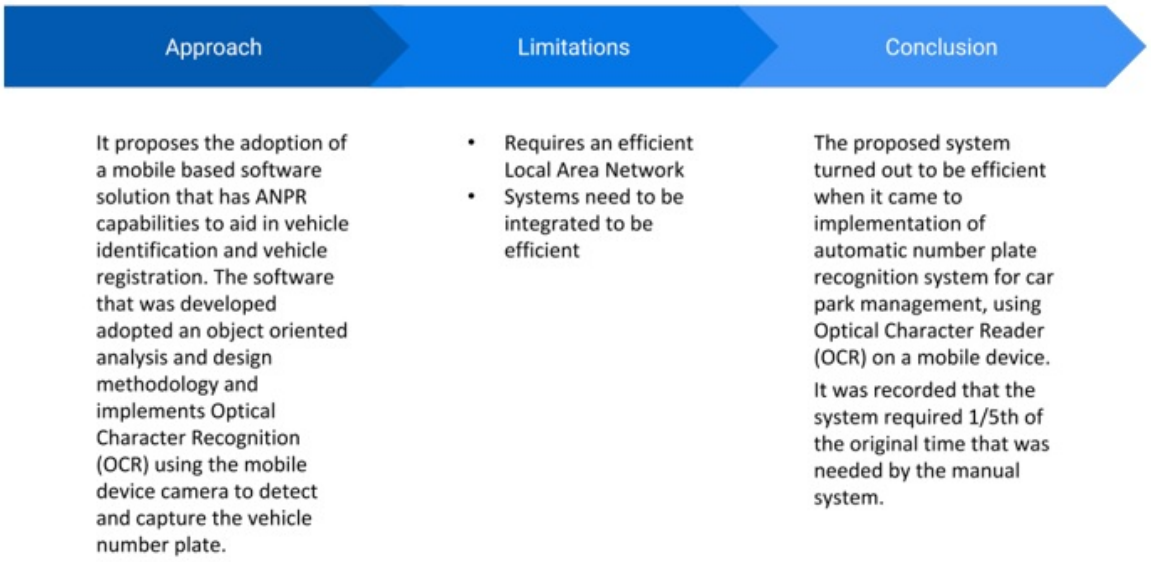

Fig 2.6 Literature Survey on Car Parking System

Chapter 3

THEORETICAL BACKGROUND

3.1 Digital Image Processing

Signal processing is a discipline in electrical engineering and in mathematics that deals with analysis and processing of analog and digital signals , and deals with storing , filtering , and other operations on signals. These signals include transmission signals , sound or voice signals , image signals , and other signals e.t.c.

Out of all these signals , the field that deals with the type of signals for which the input is an image and the output is also an image is done in image processing. As it name suggests, it deals with the processing on images.The digital image processing deals with developing a digital system that performs operations on an digital image.

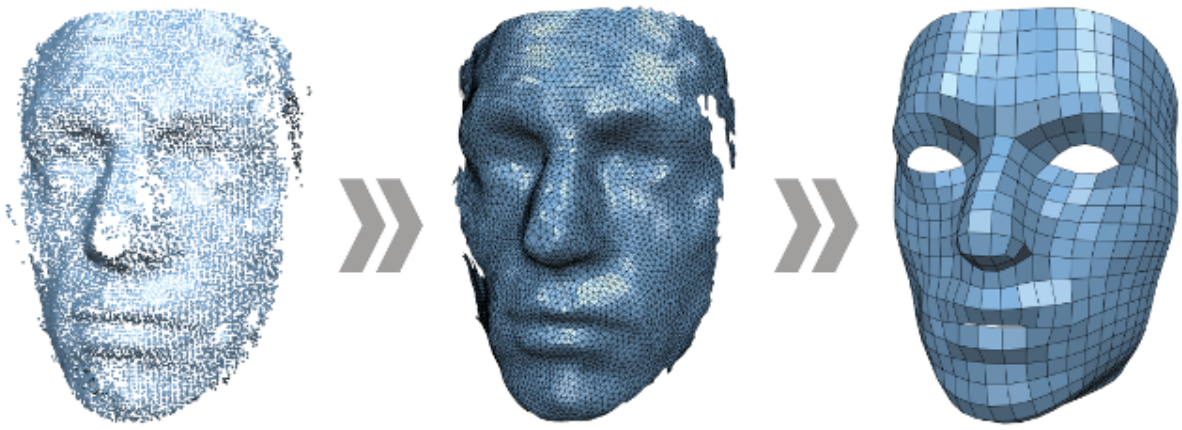

Fig 3.1 Digital Image Processing

Since capturing an image from a camera is a physical process. The sunlight is used as a source of energy. A sensor array is used for the acquisition of the image. So when the sunlight falls upon the object, then the amount of light reflected by that object is sensed by the sensors, and a continuous voltage signal is generated by the amount of sensed data. In order to create a digital image , we need to convert this data into a digital form. This results in an two dimensional array or matrix of numbers which are nothing but a digital image.

3.1.1 Steps in Digital Image Processing:

- Image Preprocessing: It is usually necessary to preprocess remote sensing data prior to its analysis because image data recorded by sensors contain errors which degrade quality of the image and cause the image to appear noise, blurred and distorted. The errors creep into during data acquisition process. Most common types of errors are geometric and radiometric errors. All these errors are corrected using suitable mathematical models at the time of preprocessing.
- Image Enhancement: Image enhancement is carried out to improve the appearance of certain image features to assist in human interpretation and analysis. You should note that image enhancement is different from image preprocessing step. Image enhancement step highlights image features for interpreter whereas image preprocessing step reconstructs a relatively better image from an originally imperfect/degraded image.
- Image Transformations: Image transformations are operations similar in concept to those for image enhancement. However, unlike image enhancement operations which are normally applied only to a single channel of data at a time, image transformations usually involve algebric operations of multi-layer images. Algebric operations such as subtraction, addition, multiplication, division, alogarithms, exponentials and trigonometric functions are applied to transform the original images into new images which display better or highlight certain features in the image. We shall discuss about all the four digital image processing steps in the next block, i.e. Block 4 of MGY-002. 71 Ground Truth Data Collection.
- Thematic Information :Extraction It includes all the processes used for extracting thematic information from images. Image classification is one such process which categorises pixels in an image into some thematic classes such as land cover classes based on spectral signatures. Image classification procedures are further categorised into supervised, unsupervised and hybrid depending upon the level of human intervention in the process of classification.

3.1.2 Advantages of Digital Image Processing:

1. Digital images can be processed by digital computers.

2. Important features such as edges can be extracted from images which can be used in industry.
3. Images can be given more sharpness and better visual appearance.
4. Minor errors can be rectified.
5. Image sizes can be increased or decreased.
6. Images can be compressed and decompressed for faster image transfer over the network.
7. Images can be automatically sorted depending on the contents they have.
8. unrecognizable features can be made prominent.
9. Images can be smoothened.
10. It allows robots to have vision.
11. It allows industries to remove defective products from the production line.
12. It allows weather forecasting.
13. It is used to analyse cells and their composition.
14. It is used to analyse medical images.

3.1.3 Problems associated with Digital Image Processing:

1. It is very costly depending on the system used, the number of detectors purchased.\
2. Time consuming
3. Lack of qualified professional

3.2 Optical Character Recognition

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo).

Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Fig 3.2 Optical Character Recognition

Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs. Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

3.2.1 Techniques of OCR

- **Pre-processing**

  OCR software often "pre-processes" images to improve the chances of successful recognition. Techniques include:

  - De-skew – If the document was not aligned properly when scanned, it may need to be tilted a few degrees clockwise or counterclockwise in order to make lines of text perfectly horizontal or vertical.
  - Despeckle – remove positive and negative spots, smoothing edges
  - Line removal – Cleans up non-glyph boxes and lines
  - Layout analysis or "zoning" – Identifies columns, paragraphs, captions, etc. as distinct blocks. Especially important in multi-column layouts and tables.

- **Text Recognition**

  Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as "pattern matching", "pattern recognition", or "image correlation". This relies on the input glyph being correctly isolated from the rest of the image, and on the stored glyph being in a similar font and at the same scale. This technique works best with typewritten text and does not work well when new fonts are encountered. This is the technique the early physical photocell-based OCR implemented, rather directly.

- **Post-processing**

  OCR accuracy can be increased if the output is constrained by a lexicon – a list of words that are allowed to occur in a document.[15] This might be, for example, all the words in the English language, or a more technical lexicon for a specific field. This technique can be problematic if the document contains words not in the lexicon, like proper nouns. Tesseract uses its dictionary to influence the character segmentation step, for improved accuracy. The output stream may be a plain text stream or file of characters, but more sophisticated OCR systems can preserve the original layout of the page and produce, for example, an annotated PDF that includes both the original image of the page and a searchable textual representation.

3.2.2 Uses of OCR

- Data entry for business documents, e.g. check, passport, invoice, bank statement and receipt
- Automatic number plate recognition
- In airports, for passport recognition and information extraction
- Automatic insurance documents key information extraction
- Traffic sign recognition
- Extracting business card information into a contact list
- More quickly make textual versions of printed documents, e.g. book scanning for Project Gutenberg
- Make electronic images of printed documents searchable, e.g. Google Books
- Converting handwriting in real time to control a computer (pen computing)
- Defeating CAPTCHA anti-bot systems, though these are specifically designed to prevent OCR. The purpose can also be to test the robustness of CAPTCHA anti-bot systems.
- Assistive technology for blind and visually impaired users

- Writing the instructions for vehicles by identifying CAD images in a database that are appropriate to the vehicle design as it changes in real time.
- Making scanned documents searchable by converting them to searchable PDFs.

3.3 Web Crawler

Web crawlers go by many names, including spiders, robots, and bots, and these descriptive names sum up what they do — they crawl across the World Wide Web to index pages for search engines. Search engines use web crawler programs as their helpers to browse the Internet for pages before storing that page data to use in future searches. WebDriver is a remote-control interface that enables introspection and control of user agents. It provides a platform- and language-neutral wire protocol as a way for out-of-process programs to remotely instruct the behaviour of web browsers. To have the ability to write instruction sets that can be run interchangeably in many browsers on different platforms is critical to deliver a consistent experience to users.

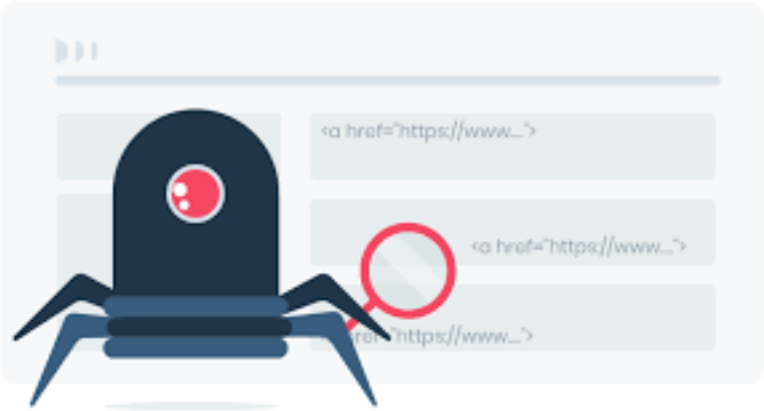

Fig 3.3 Web Crawler

The different libraries for Web Crawlers are beautiful soup and selenium web driver

1. Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.
2. Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others.

A browser-driver then executes these scripts on a browser-instance on your device.

WebDriver is an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to web pages, user input, JavaScript execution, and more. ChromeDriver is a standalone server that implements the W3C WebDriver standard. ChromeDriver is available for Chrome on Android and Chrome on Desktop (Mac, Linux, Windows and ChromeOS).

3.4 Database

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Computer databases typically contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers.

In a relational database, digital information about a specific customer is organized into rows, columns and tables which are indexed to make it easier to find relevant information through SQL or NoSQL queries. In contrast, a graph database uses nodes and edges to define relationships between data entries and queries require a special semantic search syntax. As of this writing, SPARQL is the only semantic query language that is approved by the World Wide Web Consortium (W3C).

Typically, the database manager provides users with the ability to control read/write access, specify report generation and analyze usage. Some databases offer ACID (atomicity, consistency, isolation and durability) compliance to guarantee that data is consistent and that transactions are complete.

3.4.1 Types of databases

Databases have evolved since their inception in the 1960s, beginning with hierarchical and network databases, through the 1980s with object-oriented databases, and today with SQL and NoSQL databases and cloud databases.

In one view, databases can be classified according to content type: bibliographic, full text, numeric and images. In computing, databases are sometimes classified according to their organizational approach. There are many different kinds of databases, ranging from the most prevalent approach, the relational database, to a distributed database, cloud database, graph database or NoSQL database.

Fig 3.4 Database

Relational database

A relational database, invented by E.F. Codd at IBM in 1970, is a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways.Relational databases are made up of a set of tables with data that fits into a predefined category. Each table has at least one data category in a column, and each row has a certain data instance for the categories which are defined in the columns.The Structured Query Language (SQL) is the standard user and application program interface for a relational database. Relational databases are easy to extend, and a new data category can be added after the original database creation without requiring that you modify all the existing applications.

Distributed database

A distributed database is a database in which portions of the database are stored in multiple physical locations, and in which processing is dispersed or replicated among different points in a network.Distributed databases can be homogeneous or heterogeneous. All the physical locations in a homogeneous distributed database system have the same underlying hardware and run the same operating systems and database applications. The hardware, operating systems or database applications in a heterogeneous distributed database may be different at each of the locations.

Cloud database

A cloud database is a database that has been optimized or built for a virtualized environment, either in a hybrid cloud, public cloud or private cloud. Cloud databases provide benefits such as the ability to pay for storage capacity and bandwidth on a per-use basis, and they provide scalability on demand, along with high availability.A cloud database also gives enterprises the opportunity to support business applications in a software-as-a-service deployment.

NoSQL database

NoSQL databases are useful for large sets of distributed data.NoSQL databases are effective for big data performance issues that relational databases aren't built to solve. They are most effective when an organization must analyze large chunks of unstructured data or data that's stored across multiple virtual servers in the cloud.

Object-oriented database

Items created using object-oriented programming languages are often stored in relational databases, but object-oriented databases are well-suited for those items.An object-oriented database is organized around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value.

Graph database

A graph-oriented database, or graph database, is a type of NoSQL database that uses graph theory to store, map and query relationships. Graph databases are basically collections of nodes and edges, where each node represents an entity, and each edge represents a connection between nodes.Graph databases are growing in popularity for analyzing interconnections. For example, companies might use a graph database to mine data about customers from social media.Graph databases often employ SPARQL, a declarative programming language and protocol for graph database analytics. SPARQL has the capability to perform all the analytics that SQL can perform, plus it can be used for semantic analysis, the examination of relationships. This makes it useful for performing analytics on data sets that have both structured and unstructured data. SPARQL allows users to perform analytics on information stored in a relational database, as well as friend-of-a-friend (FOAF) relationships, PageRank and shortest path.

3.4.2 Firebase database

The Firebase Realtime Database is a cloud-hosted database. ... When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Firebase is based on a data structure used by the NoSQL database is vastly different from those used in a relational database. Some operations are faster in NoSQL than relational databasesFirebase RealTime Database with Operations in Android with Examples. Firebase Realtime Database is a Cloud hosted database, i.e. it runs on a cloud and access to the user is provided as a service. It stores data in JSON (Javascript Object Notation) format, a format to store or transport data.

One of the main problems with it, is limited querying capabilities. Realtime database provides no way to filter capabilities, because the whole DB is a huge JSON file, which makes it difficult to make complex queries.

3.4.3 MySQL database

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system

SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

3.5 Graphical User Interface

The graphical user interface, developed in the late 1970s by the Xerox Palo Alto research laboratory and deployed commercially in Apple's Macintosh and Microsoft's Windows operating systems, was designed as a response to the problem of inefficient usability in early, text-based command-line interfaces for the average user.

Graphical user interfaces would become the standard of user-centered design in software application programming, providing users the capability to intuitively operate computers and other electronic devices through the direct manipulation of graphical icons such as buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Many modern graphical user interfaces feature touchscreen and voice-command interaction capabilities.

Graphical user interface design principles conform to the model–view–controller software pattern, which separates internal representations of information from the manner in which information is presented to the user, resulting in a platform where users are shown which functions are possible rather than requiring the input of command codes. Users interact with information by manipulating visual widgets, which are designed to respond in accordance with the type of data they hold and support the actions necessary to complete the user's task.

Fig 3.5 Graphical User Interface

The appearance, or "skin," of an operating system or application software may be redesigned at will due to the nature of graphical user interfaces being independent from application functions. Applications typically implement their own unique graphical user interface display elements in addition to graphical user interface elements already present on the existing operating system. A typical graphical user interface also includes standard formats for representing graphics and text, making it possible to share data between applications running under common graphical user interface design software.

Graphical user interface testing refers to the systematic process of generating test cases in order to evaluate the functionality of the system and its design elements. Graphical user interface testing tools, which are either manual or automated and typically implemented by third-party operators, are available under a variety of licenses and are supported by a variety of platforms. Popular examples include: Tricentis Tosca, Squish GUI Tester, Unified Functional Testing (UFT), Maveryx, Appium, and eggPlant Functional.

3.5.1 Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

The foundational element of a Tkinter GUI is the window. Windows are the containers in which all other GUI elements live. These other GUI elements, such as text boxes, labels, and buttons, are known as widgets. Widgets are contained inside of windows.

3.5.2 PyQt

PyQt is one of the most popular Python bindings for the Qt cross-platform C++ framework. PyQt developed by Riverbank Computing Limited. Qt itself is developed as part of the Qt Project. PyQt provides bindings for Qt 4 and Qt 5. PyQt is distributed under a choice of licences: GPL version 3 or a commercial license.

PyQt is available in two editions: PyQt4 which will build against Qt 4.x and 5.x and PyQt5 which will only build against 5.x. Both editions can be built for Python 2 and 3. PyQt contains over 620 classes that cover graphical user interfaces, XML handling, network communication, SQL databases, Web browsing and other technologies available in Qt.

3.6 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented

reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDAand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.7 Tesseract Engine

An open-source OCR engine that has gained popularity among OCR developers. Even though it can be painful to implement and modify sometimes, there weren't too many free and powerful OCR alternatives on the market for the longest time. Tesseract began as a Ph.D. research project in HP Labs, Bristol. It gained popularity and was developed by HP between 1984 and 1994. In 2005 HP released Tesseract as an open-source software. Since 2006 it is developed by Google.

Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API to extract printed text from images. It supports a wide variety of languages. Tesseract doesn't have a built-in GUI, but there are several available from the 3rdParty page. Tesseract is compatible with many programming languages and

frameworks through wrappers that can be found here. It can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line.

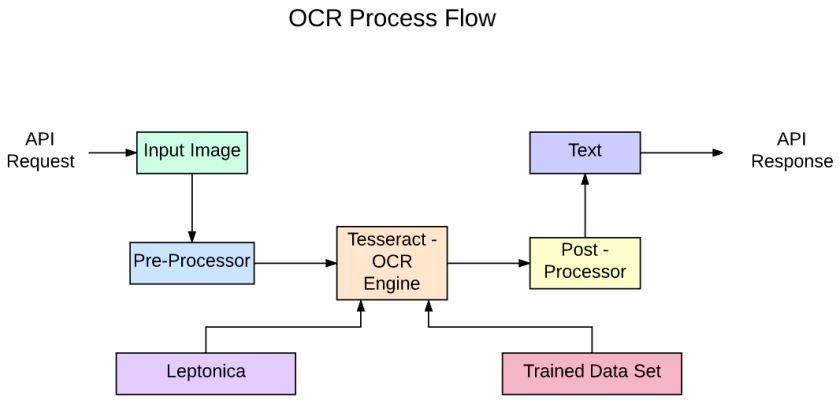

Fig 3.7 Tesseract Architecture

LSTMs are great at learning sequences but slow down a lot when the number of states is too large. There are empirical results that suggest it is better to ask an LSTM to learn a long sequence than a short sequence of many classes. Tesseract developed from OCRopus model in Python which was a fork of a LSMT in C++, called CLSTM. CLSTM is an implementation of the LSTM recurrent neural network model in C++, using the Eigen library for numerical computations.

Legacy Tesseract 3.x was dependant on the multi-stage process where we can differentiate steps:

- Word finding
- Line finding

Character classificationWord finding was done by organizing text lines into blobs, and the lines and regions are analyzed for fixed pitch or proportional text. Text lines are broken into words differently according to the kind of character spacing. Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in turn. Each word that is

satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page.

Modernization of the Tesseract tool was an effort on code cleaning and adding a new LSTM model. The input image is processed in boxes (rectangle) line by line feeding into the LSTM model and giving output. In the image below we can visualize how it works.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

4.1 INTRODUCTION:

This chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirement of this dissertation.

A SRS document describes all data, functional and behavioral requirements of the software under production or development. SRS is a fundamental document, which forms the foundation of the software development process. It is the complete description of the behavior of a system to be developed. Requirement Analysis discusses the conditions to be met for a new or altered product. Requirement Analysis is critical to the success to a development project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The SRS functions as a blueprint for completing a project. The goal of preparing the SRS document is to:

- Facilitate communication between the customer, analyst, system developers, maintainers.
- To form a foundation for the design phase.
- Support system testing facilities.
- Controlling the evolution of the system.

4.2 FUNCTIONAL REQUIREMENTS

Functional Requirement defines in detail how the system must respond to the various kinds of input that is given to the system. It also talks about the expected system behavior under certain conditions. In this system following are the functional requirements:

- **1. Mobility:** The device identifying the License plate should be movable to capture the LP on the go
- **2. Convenience:** The system will make it efficient to access for vehicles and prevent congestion at entry and exit points
- **3. User-Interface:** The system shall provide an easy-to-use user-interface.
- **4. Transparency:** Users should be able to possess a general knowledge and understanding of the ALPR process.
- **5. Flexibility:** The system shall be flexible in that it allows a variety of formats to ingrate the scanning of characters to
- **6. Support for Disabled:** The system shall cater to the needs of physically challenged voters (e.g. blind voters).
- **7. Accuracy:** The system shall accurately convert the image to characters
- **8. Uniqueness:** The system is trained to handle unique and varying types of number plates
- **9. Documentation and Assurance:** The design, implementation, and testing procedures must be well documented so that the confidence is ensured.
- **10. Cost-effectiveness:** Should be affordable and efficient.

4.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system

properties. **Non-functional requirements** are requirements that are not specifically concerned with the functionality of a system. They normally place restrictions on the product being developed and the development process. Non-functional requirements may be regarded as parameters of functionality in that they determine how quickly, how accurately, how reliably, how securely, etc., functions must operate. Some of the ALPR non-functional requirements are as follows:

• The system may issue a receipt to remove any papers printed and make it a green initiative

• The system must be working at 100% peak efficiency

• When checking the database for errors, a 100% scan of the data is required, rather than selecting a sample set

• A process must be devised to support normal precinct business hours

• The system should provide documentation to inform users of system functionality and any change to the system

• The system should provide friendly graphical Interface to ensure ease of use when end users utilize system functionality

4.4 HARDWARE REQUIREMENTS:

· Processor	:	Intel I5 2.1 Ghz.
· Storage	:	100 GB.
· RAM	:	4 GB

4.5 SOFTWARE REQUIREMENTS:

• **Platform:** Windows/Linux/macOS

• **Language used:** Python

•**Technologies used:** PyTesseract, OpenCV, Selenium, Chrome Driver, Tkinter, Pyrebase,MySQL connector

Pytesseract : Library to use the Tesseract-OCR. Tesseract is an optical character recognition engine for various operating systems. Tesseract is considered to be one of the most accurate open-source OCR engines available.

OpenCV : OpenCV stands for *Open Source Computer Vision*. It is an open source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

Selenium : It is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium tool is usually referred to as Selenium Testing. Selenium Software is not just a single tool but a suite of software, each piece catering to different testing needs of an organization.

Pyrebase - is a Python interface to Firebase's REST API. In layman's terms, it allows you to use Python to manipulate your Firebase database.

Chrome Driver - WebDriver is an open source tool for automated testing of web apps across many browsers. It provides capabilities for navigating to web pages, user input, JavaScript execution, and more. ChromeDriver is a standalone server that implements the W3C WebDriver standard. ChromeDriver is available for Chrome on Android and Chrome on Desktop.

Tkinter - is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Chapter 5

SYSTEM DESIGN

This section elaborately describes the architecture of a Number plate recognition system and an information extractor . The proposed system detects the vehicle number plate and extracts the vehicle number plate in characters. In this module of vehicle number plate recognition, the input image is preprocessed and the characters in the image are segmented. The characters that are segmented are cropped and recognized. The recognized characters are then returned as a string. The next module involves a web crawler which fetches the information of the vehicle. The information fetched is converted to JSON format and stored in a database or displayed on a dashboard for further processing.

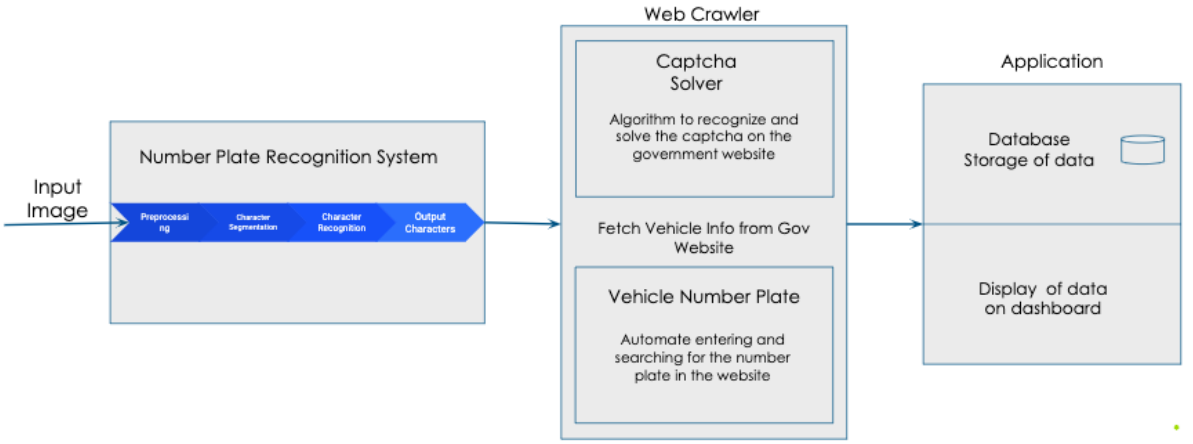

Figure 5.1 System Architecture

The basic architecture of the system is demonstrated in Fig 5.1. The system has the following set of modules:

- **Number Plate Recognition System**: The input image is captured by the system through a camera feed or an image file. The image is pre-processed by converting the image to a grayscale image and then bilateral filter is applied to smoothen the image further. The edges in the image are detected using the canny edge detection algorithm. Once the canny edges

are detected. Contours with 4 edges are ranked . The contour with the highest rank is considered as the number plate region and it is cropped. Pytesseract is an OCR which performs character recognition on this region of the image.

- **Web Crawler**: The web driver opens the website Vahan.nic.in and captures the captcha. Once the captcha is captured, it is recognized and converted to a string. The string is then solved as a mathematical equation and then it is sent back to the captcha input field. The vehicle number plate is also sent the input field of the registration number.
- **Application**: The data of the vehicle extracted is converted to JSON format and stored MySQL database and Firebase database. The data is then read by a Tkinter application and displayed to the user.

  The Following information can be collected:

  1. Registering Authority
  2. Registration No
  3. Registration Date
  4. Chassis No
  5. Engine No
  6. Owner Name
  7. Vehicle Class
  8. Fuel
  9. Fitness/REGN Upto
  10. MV Tax upto
  11. Insurance Upto
  12. Maker/Model
  13. PUCC Upto
  14. Emission norms
  15. RC Status
  16. Finance

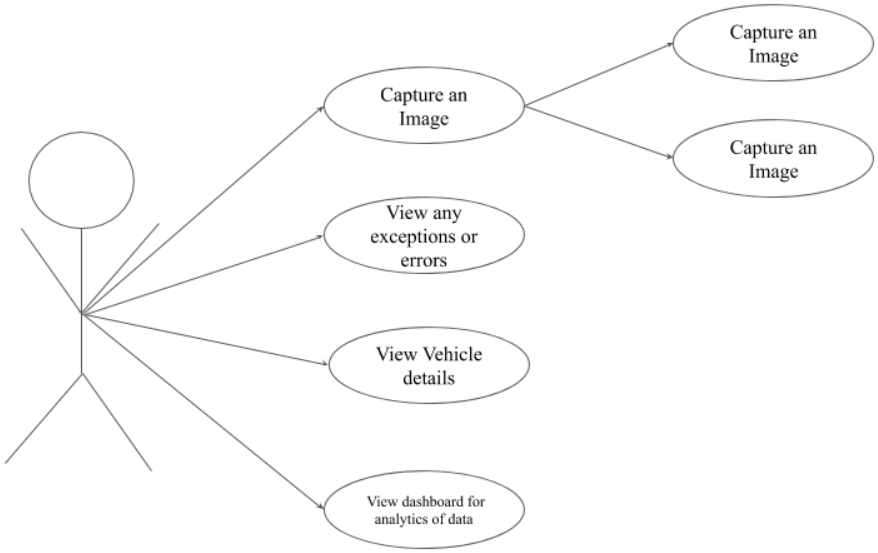

Figure 5.2 Use Case diagram – User

Fig 5.2 shows the use case diagram of the User. The main function of the User is to upload or capture an image and then view the details and analyse the vehicle information that is extracted.

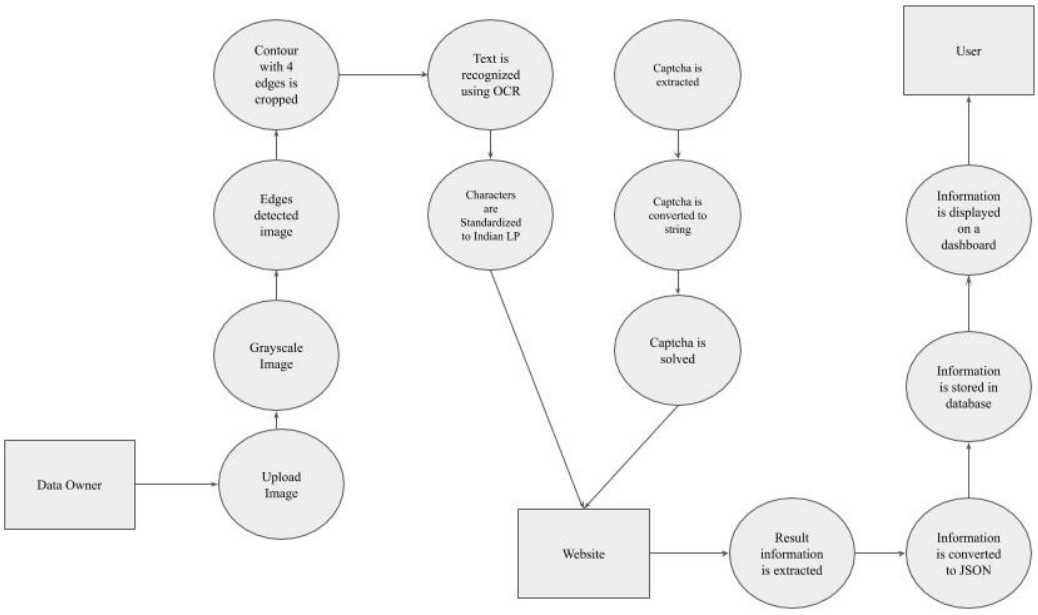

Figure 5.3 Data Flow Diagram of the Proposed System

Chapter 6

IMPLEMENTATION

This section contains the algorithms that are used by the system. The system makes use of these algorithms in order to satisfy the various functional and non functional requirements of the proposed scheme.

6.1 ALGORTIHMS:

6.1.1 Algorithm to Recognize the Number Plates

The sequence of processes associated with Number plate recognition is given below. The file upload process is initiated by the data owner entity.
Input: Uploading the image file from camera
Output: Vehicle number plate in characters

1) Read the original image or Capture the image

2) Resize the image

3) Convert it to grayscale.

4) Apply Bilateral Filter. *What is a bilateral filter* ? A bilateral filter is a non-linear, edge preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.

5) Identify and store the Canny edges. *What are Canny edges* ? The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

6) Find the contours in from the edges detected and sort the top 30 contours.

7) Get the perimeter of each contour and select those with 4 corners.

8) Mask all other parts of the image and show the final image.

9) Read the text using Tesseract OCR

10) Standardize the text to Indian vehicle number plate format

11. Stop

On upload of the image file to the system, the number plate recognition system performs its functions to provide the output.

6.1.2 Algorithm to Extract Vehicle Information from Vahan.nic.in

The sequence of processes associated with extracting the vehicle information from vahan.nic.in . The information is extracted from the website

Input:Vehicle Number plate characters

Output: Vehicle information

1. Initialize the Selenium web driver

1. Open the (vahan.nic.in) url in the web driver
2. Find the Captcha Element
3. Captcha Element is then converted to string
4. Captcha is solved based on the written rules
5. Captcha Solution is determined
6. The Vehicle Number and Captcha Solution are sent to their input field using Keys function
7. The Search button is pressed using the click function in Selenium
8. The Vehicle details are then extracted as text
9. The text is converted into JSON data and stored in a database

6.2 CODE:

6.2.1 Code to capture the image of the vehicle number plate

```
class App:
    def __init__(self, window, window_title, video_source=0):
        self.window = window
        self.window.title(window_title)
        self.video_source = video_source

        # open video source (by default this will try to open the computer webcam)
        self.vid = MyVideoCapture(self.video_source)

        # Create a canvas that can fit the above video source size
        self.canvas = tkinter.Canvas(window, width = self.vid.width, height = self.vid.height)
        self.canvas.pack()

        # Button that lets the user take a snapshot
        self.btn_snapshot=tkinter.Button(window, text="Snapshot", width=50,
command=self.snapshot)
        self.btn_snapshot.pack(anchor=tkinter.CENTER, expand=True)

        # After it is called once, the update method will be automatically called every delay
milliseconds
        self.delay = 15
        self.update()

        self.window.mainloop()



    def snapshot(self):
        # Get a frame from the video source
        ret, frame = self.vid.get_frame()
        if ret:
            cv2.imwrite("vehicleplate.jpg", cv2.cvtColor(frame, cv2.COLOR_RGB2BGR))

    def update(self):
        # Get a frame from the video source
```

```
        ret, frame = self.vid.get_frame()

        if ret:
            self.photo = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(frame))
            self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)

        self.window.after(self.delay, self.update)


class MyVideoCapture:
    def __init__(self, video_source=0):
        # Open the video source
        self.vid = cv2.VideoCapture(video_source)
        if not self.vid.isOpened():
            raise ValueError("Unable to open video source", video_source)

        # Get video source width and height
        self.width = self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)
        self.height = self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)

    def get_frame(self):
        if self.vid.isOpened():
            ret, frame = self.vid.read()
            if ret:
                # Return a boolean success flag and the current frame converted to BGR
                return (ret, cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
            else:
                return (ret, None)
        else:
            return (ret, None)
         # Release the video source when the object is destroyed
    def __del__(self):
        if self.vid.isOpened():
            self.vid.release()

    # Create a window and pass it to the Application object
App(tkinter.Tk(), "Tkinter and OpenCV")
cv2.destroyAllWindows
```

6.2.2 Code to recognize the vehicle number plate

```
img= cv2.imwrite("filename.jpg", frame)
   print(img)
   # 8. shutdown the camera
   video.release()
   cv2.destroyAllWindows

   img = cv2.imread("filename.jpg")
   #Using imutils to resize the image.
   img = imutils.resize(img, width=500)

   cv2.imshow("Original Image", img)
   cv2.waitKey(0)

   #Convert from colored to Grayscale.
   gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
   cv2.imshow("Preprocess 1 - Grayscale Conversion", gray_img)       #Show modification.
   cv2.waitKey(0)

   #Applying Bilateral Filter on the grayscale image.

   gray_img = cv2.bilateralFilter(gray_img, 11, 17, 17)
   cv2.imshow("Preprocess 2 - Bilateral Filter", gray_img)
   cv2.waitKey(0)


   c_edge = cv2.Canny(gray_img, 170, 200)
   cv2.imshow("Preprocess 3 - Canny Edges", c_edge)
   cv2.waitKey(0)

   #Finding contours based on edges detected.
   cnt, new = cv2.findContours(c_edge, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
   #Storing the top 30 edges based on priority
   cnt = sorted(cnt, key = cv2.contourArea, reverse = True)[:30]
   NumberPlateCount = None

   im2 = img.copy()
   cv2.drawContours(im2, cnt, -1, (0,255,0), 3)
```

```
    cv2.imshow("Top 30 Contours", im2)         #Show the top 30 contours.
    cv2.waitKey(0)

    count = 0
    for c in cnt:
        perimeter = cv2.arcLength(c, True)      #Getting perimeter of each contour
        approx = cv2.approxPolyDP(c, 0.02 * perimeter, True)
        if len(approx) == 4:            #Selecting the contour with 4 corners/sides.
            NumberPlateCount = approx
            break

    #Masking all other parts, other than the number plate.
    masked = np.zeros(gray_img.shape,np.uint8)
    new_image = cv2.drawContours(masked,[NumberPlateCount],0,255,-1)
    new_image = cv2.bitwise_and(img,img,mask=masked)
    cv2.imshow("4 - Final_Image",new_image)
   # path ="C:/Users/nikit/Downloads/ANPR-FinalYear/Implementing-Basic-ANPR-
master/Final_images"

  # cv2.imwrite(os.path.join(path , 'test16.jpg'), new_image)
    cv2.waitKey(0)


    configr = ('-l eng --oem 1 --psm 3')

    #Running Tesseract-OCR on final image.
    text_no = pytesseract.image_to_string(new_image, config=configr)

    #The extracted data is stored in a data file.
    data = {'Date': [time.asctime(time.localtime(time.time()))],
        'Vehicle_number': [text_no]}

    df = pd.DataFrame(data, columns = ['Date', 'Vehicle_number'])
    df.to_csv('Dataset_VehicleNo.csv')

    #Printing the recognized text as output.
    print(text_no)
    cv2.waitKey(0)
```

6.2.3 Code to extract information from vahan.nic.in

```
driver = webdriver.Chrome('./chromedriver')
counter = 0
while counter<1:
    counter = counter + 1
    regions = ['in'] # Change to your country
    with open("vehicleplate.jpg", 'rb') as fp:
        response = requests.post(
            'https://api.platerecognizer.com/v1/plate-reader/',
            data=dict(regions=regions),  # Optional
            files=dict(upload=fp),
            headers={'Authorization': 'Token 541c8820aad4d36e3a80a71ede91658445ec47f1'})
    try:
        plate = response.json()
        plate_no = str(plate['results'][0]['plate'])   #Reg_no is captured
        plate_no = plate_no.upper()

    except:
        plate_no = "NOT Recognized"

    s = str(plate_no)

    if re.match(r'^[A-Z]{2}[0-9]{1}[A-Z]{2}', s, flags=0):
        s = s[:2]+'0'+s[2:]

    elif re.match(r'^[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{3}$', s, flags=0):
        s = s[:6]+'0'+s[6:]
    elif re.match(r'^[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{2}$', s, flags=0):
        s = s[:6]+'00'+s[6:]
    elif re.match(r'^[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{1}$', s, flags=0):
        s = s[:6]+'000'+s[6:]
    plate_no = s
    print("Plate No. : "+str(plate_no)+"\n")

    driver.get("https://vahan.nic.in/nrservices/faces/user/searchstatus.xhtml")
    #driver.minimize_window()
    captcha = str(driver.find_element_by_id("capatcha").text)

    print("Captcha : "+str(captcha)+"\n")
```

```
    os.remove("vehicleplate.jpg")
    captcha = captcha.split()

    if captcha[0].isdigit():
        equation = str("".join(captcha))
        captcha_result = eval(equation)
    else:
        if str(captcha[3]) == "greater":
            captcha_result = max(int(captcha[4]),int(captcha[6]))
        else:
            captcha_result = min(int(captcha[4]),int(captcha[6]))

    reg_no_search_bar =driver.find_element_by_id('regn_no1_exact')
    captcha_search_bar =driver.find_element_by_id('txt_ALPHA_NUMERIC')

    #Regno generator


    reg_no_search_bar.send_keys(plate_no)
    captcha_search_bar.send_keys(captcha_result)


    print("Captcha Result : "+str(captcha_result)+"\n")


    time.sleep(1)
    search_button = driver.find_element_by_id('page-
wrapper').find_element_by_tag_name('button')

    search_button.click()
    time.sleep(3)

    try:
        if driver.find_element_by_id('resultPanel').text:
            status = "Success"
            print("Find Information : "+status+"\n")

        rc_details = driver.find_element_by_id('rcDetailsPanel').text
        rc_details = rc_details.split("\n",1)[1];
```

```
    rc_details = rc_details.replace("1. ","")
    rc_details = rc_details.replace("Authority: ","Authority:\n")
    rc_details = rc_details.split("\n")
    rc_details = [s.replace(':',") for s in rc_details]
    rc_details = [s.replace(' / ',") for s in rc_details]
    rc_details = [s.replace('/',") for s in rc_details]
```

6.2.4 Code to store information in database

```
headers = []

for i in range(len(rc_details)):

    if i%2 == 0:
        headers.append(rc_details[i])

print(headers)
def Convert(lst):
    res_dct = {lst[i]: lst[i + 1] for i in range(0,n-1,2)}
    return res_dct

rc_details_dict = Convert(rc_details)
rc_details_json = json.dumps(rc_details_dict)
print(rc_details_json)

rc_details_json_obj = json.loads(rc_details_json)
config = {
"apiKey": "AIzaSyBW5g22CfUqejW2O4F_DinPCHFhfc4vJSU",
"authDomain": "anpr-15ebf.firebaseapp.com",
"databaseURL": "https://anpr-15ebf.firebaseio.com",
"projectId": "anpr-15ebf",
"storageBucket": "anpr-15ebf.appspot.com",
"messagingSenderId": "114786024077",
"appId": "1:114786024077:web:656f6b7f97c51dad9200c4",
"measurementId": "G-0MGT6CYTEY"
}

firebase = pyrebase.initialize_app(config)
```

```
    db = firebase.database()
    data = {
            "Registering Authority":rc_details_json_obj["Registering Authority"],
            "Registration No":rc_details_json_obj["Registration No"],
            "Registration Date":rc_details_json_obj["Registration Date"],
            "Chassis No": rc_details_json_obj["Chassis No"],
            "Engine No":rc_details_json_obj["Engine No"],
            "Owner Name":rc_details_json_obj["Owner Name"],
            "Vehicle Class":rc_details_json_obj["Vehicle Class"],
            "Fuel": rc_details_json_obj["Fuel"],
            "MakerModel": rc_details_json_obj["MakerModel"],
            "FitnessREGN Upto": rc_details_json_obj["FitnessREGN Upto"],
            "MV Tax upto":rc_details_json_obj["MV Tax upto"],
            "Insurance Upto":rc_details_json_obj["Insurance Upto"],
            "PUCC Upto": rc_details_json_obj["PUCC Upto"],
            "Emission norms": rc_details_json_obj["Emission norms"],
            "RC Status": rc_details_json_obj["RC Status"]
          }


    with open("sample.json", "a") as outfile:
       json.dump(data, outfile)
    data_id = rc_details_json_obj["Registration No"]

    db.child("users").child(data_id).push(data)
    print("Data added to real time database ")

driver.quit()
```

Chapter 7

RESULTS AND TESTING

Testing is the process of checking whether the actual results obtained meet the expected results and to ensure that the software system is free from any kind of defects. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

Testing is an important phase in the development life cycle of the product. Testing performs a very critical role for quality assurance and ensuring the reliability of the software. Each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing that it is supposed to do. Testing is the final verification and validation activity within the development environment.

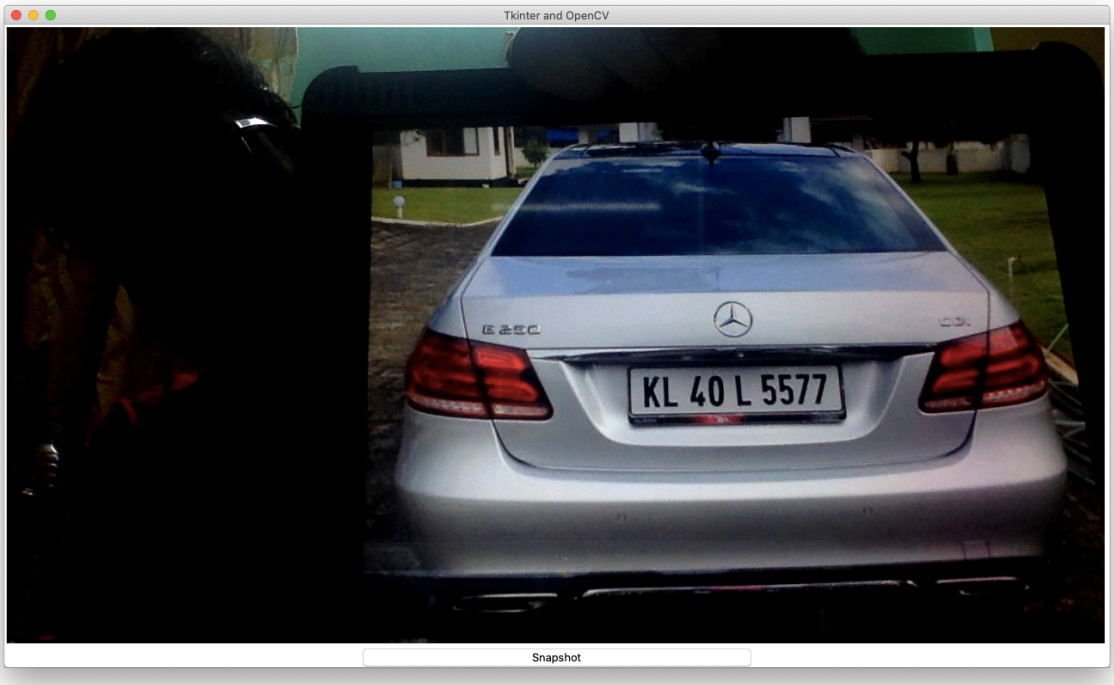

Figure 7.1 Window to capture the Number Plate

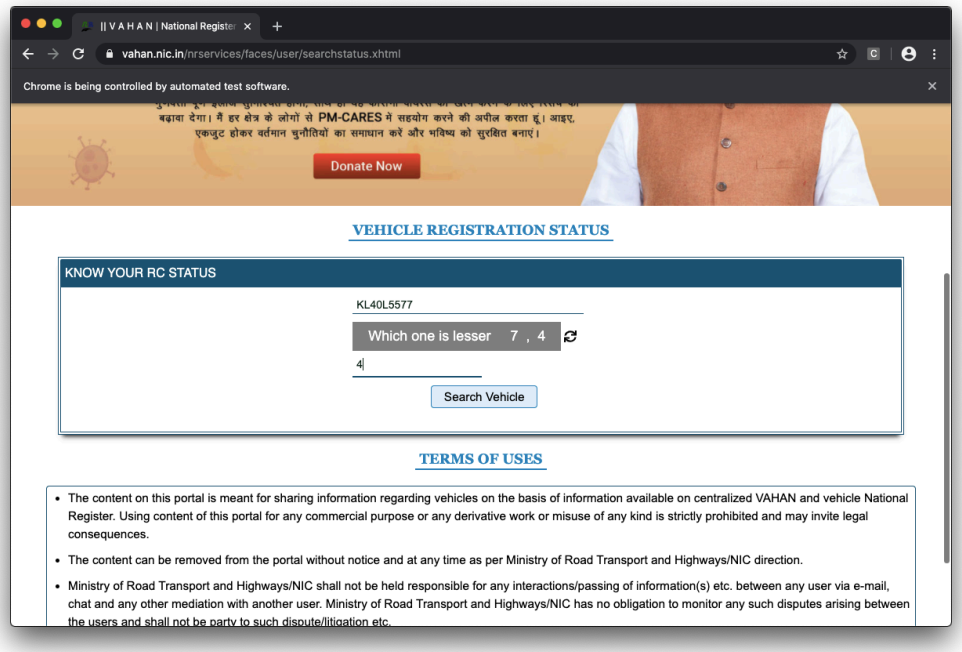

Figure 7.2 Web Crawling Vahan.nic.in

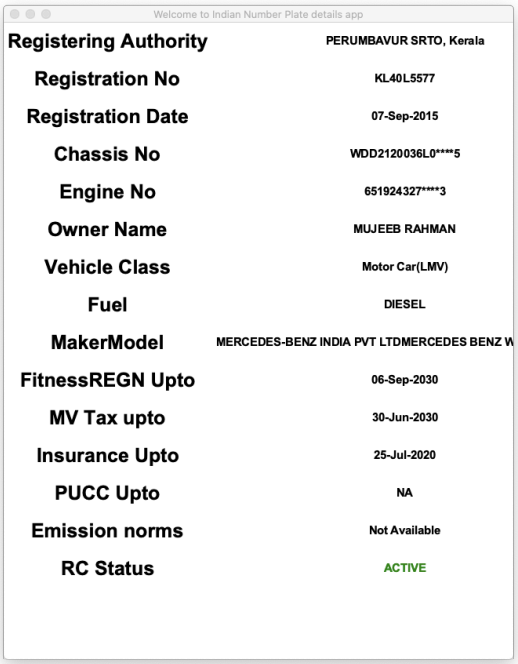

Figure 7.3 Data extracted is displayed in a TKINTER window

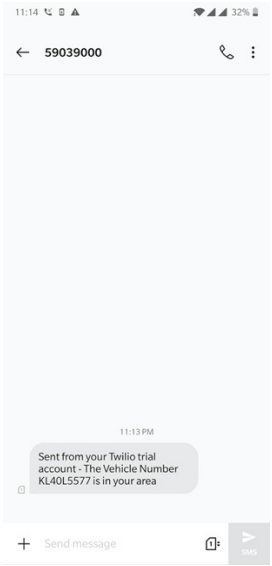

Figure 7.4 Message Notifications when an Invalid Car is visible

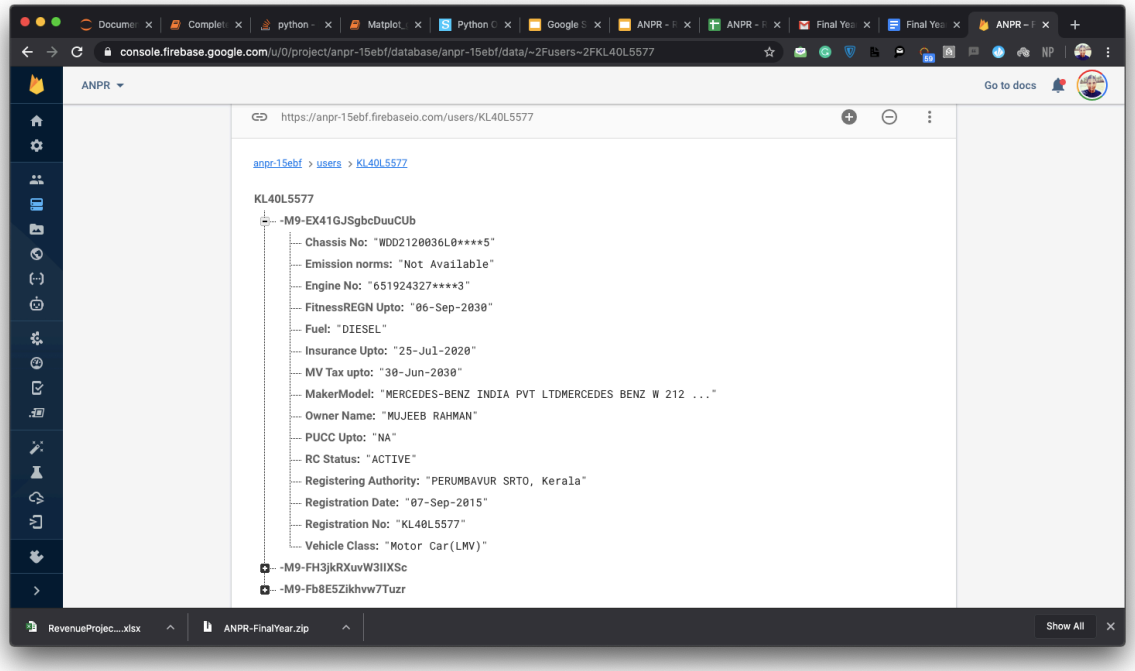

Figure 7.5 Data Stored in Firebase Database in JSON

The system was successfully tested on 1500 Indian number plates where we found that, Out of the 1500 Number plates, 948 number plates could be recognized perfectly. Whereas it failed for the rest of the 552 case. Failure occured in differentiating characters like M and N, P and R , 8 and B .

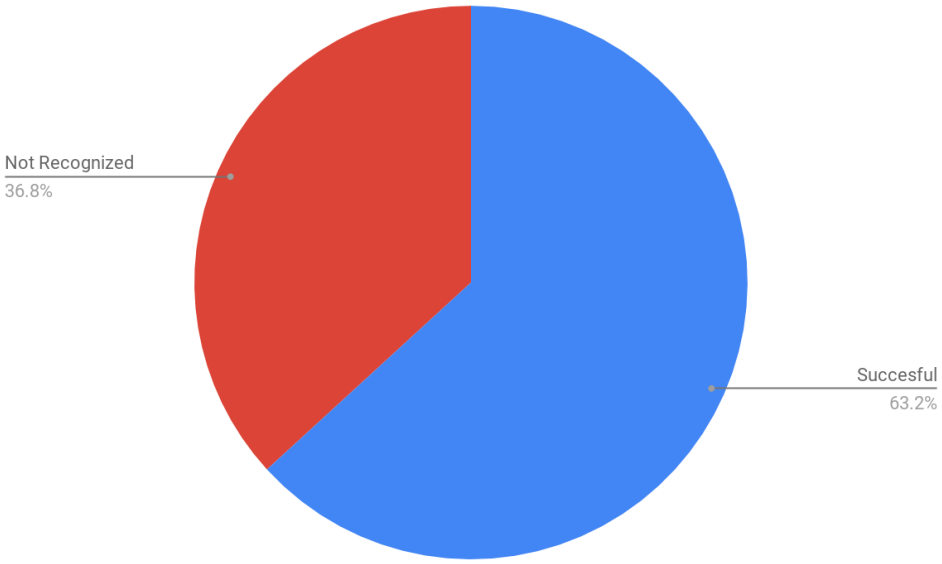

Figure 7.6 Test Analysis 1

Out of the 948 Successful Number plate recognitions. 715 number plates had records in Vahan.nic.in. Rest 233 number plates can be deemed as fake or a record doesn't exist with the government. 73 Number plates were to be found with inactive RC or do not have the permission to be used

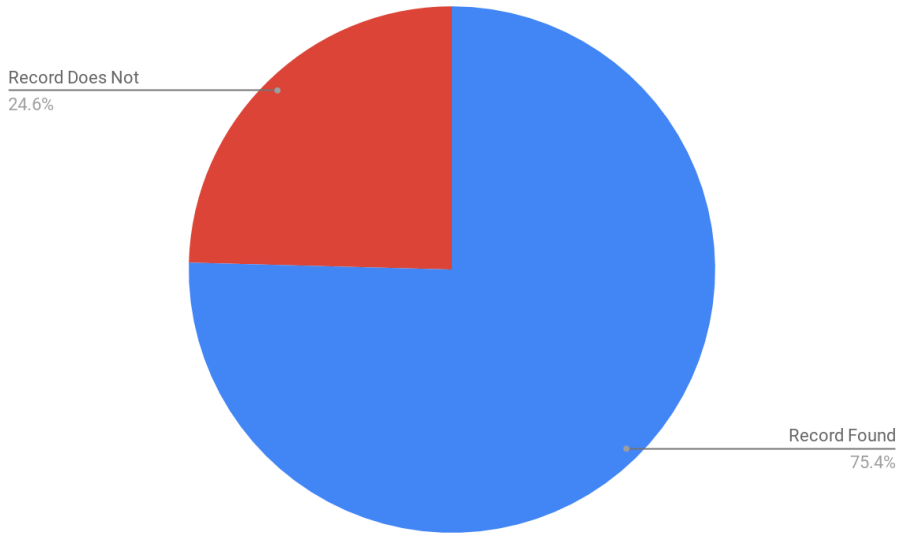

Figure 7.7 Test Analysis 2

The success rate in evaluating the captcha was 99%. The algorithm efficiently determined solution at all the given test cases. It also shows the vulnerability in government websites.

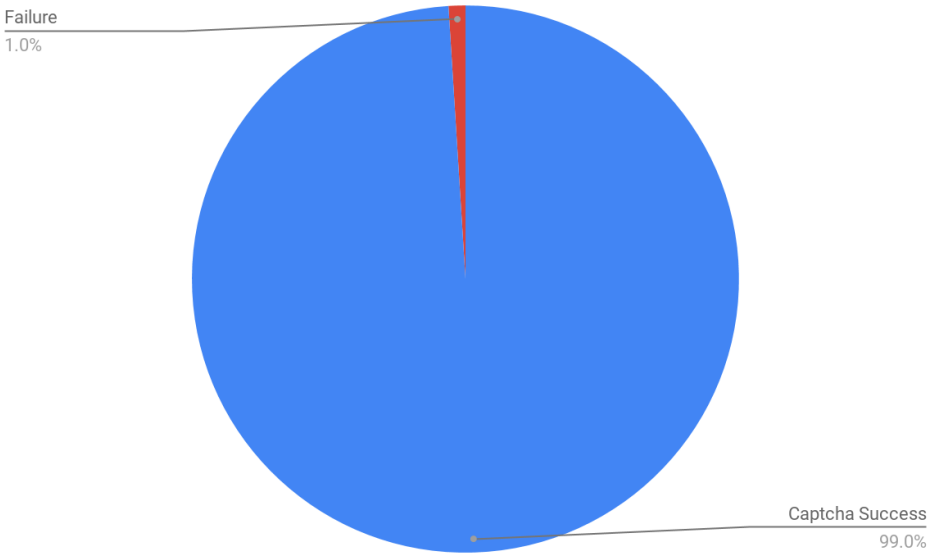

Figure 7.8 Test Analysis 3

Chapter 8

CONCLUSION AND FUTURE WORK

Through this project it is possible to recognise Vehicle registration numbers through digital image processing. From this system we have effortlessly obtained the various results such as

- Whether the vehicle which is registered is blacklisted or not.
- This also enables one single user to effectively monitor the traffic, and can easily locate the traffic violated vehicle.
- The data can be easily stored and transferred which makes the system more efficient.

The system has been designed using a modular approach which allows easy upgrading and/or substituting of various sub-modules thus making it potentially suitable for a large range of vision applications. The performances of the system makes it a valid choice among its competitors especially in those situations when the cost of the application has to be maintained at reasonable levels. Furthermore, the modular architecture makes it extremely flexible and versatile.

The earlier methodologies which have been implemented have not been as accurate and efficient as the designed Recognition system , this is because of the implementation of digital Image Processing which gives an accuracy of 90% under normal conditions

This Project is based on automatic vehicle license plate recognition, in which it is observed that the existing techniques don't pay much attention towards improving the system's efficiency in terms of its power consumption. As the objective in our proposed design is to reduce power consumption of the system, with the successful implementation of the same it will play a very important role in traffic management and security systems such as automobile theft prevention, parking lot management etc. implementations of the software algorithm have shown promising results.

The system can be made more robust if high precision cameras can be used to increase overall accuracy if this system is implemented in real time applications. Also a sensor can be designed to allow the camera to capture the image only when required to save power.

REFERENCES

[1] Amninder Kaur, Sonika Jindal ,Richa Jindal "License Plate Recognition Using Support Vector Machine (SVM)" Dept. Of Computer Science, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 7.

[2] ANISH LAZRUS,SIDDHARTHA CHOUBEY,SINHA G.R.,"AN EFFICIENT METHOD OF VEHICLE NUMBER PLATE DETECTION AND RECOGNITION" Department of Computer Science, International Journal of Machine Intelligence, Volume 3, Issue 3.

[3] Abhay Singh, Anand Kumar Gupta ,Anmol Singh, Anuj Gupta ,Sherish Johri, "VEHICLE NUMBER PLATE DETECTION USING IMAGE PROCESSING", Department of IT, Volume: 05 Issue: 03 | Mar-2018

[4] Ganesh R. Jadhav, Kailash J. Karande, "Automatic Vehicle Number Plate Recognition for Vehicle Parking Management System", IISTE, Vol.5, No.11, 2014.

[5] Mutua Simon Mandi ,Bernard Shibwabo, Kaibiru Mutua Raphael, "An Automatic Number Plate Recognition System for Car Park Management", International Journal of Computer Applications, Volume 175 – No.7, October 2017

[6] https://en.wikipedia.org/wiki/Automatic_number-plate_recognition