RV Educational Institutions ®
RV College of Engineering ®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

Go, change the world

DEPARTMENT OF
MASTER OF COMPUTER APPLICATIONS

Minor Project Report

On

Development of Vehicle Tracking System.

Submitted in partial fulfilment of the requirements for the III Semester MCA Academic Minor Project

20MCA36

By
Laxmikanth (1RV20MC036)
Roshith K R (1RV20MC086)

Under the Guidance
of
Dr. S. Anupama Kumar
Associate Professor
Department of Master of Computer Applications
RV College of Engineering®
Bengaluru – 560059

January 2022

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF

MASTER OF COMPUTER APPLICATIONS

Bengaluru– 560059

CERTIFICATE

This is to certify that the project entitled "**Development of vehicle tracking system"** submitted in partial fulfilment of Minor Project (20MCA36) of III Semester MCA is a result of the bonafide work carried out by Laxmikanth (1RV20MC036) and Roshith K R (1RV20MC086) during the Academic year 2021-22.

Dr. S. Anupama Kumar
Associate Professor
Department of MCA
RV College of Engineering®
Bengaluru-59

Dr. Andhe Dharani
Professor and Director
Department of MCA
RV College of Engineering®
Bengaluru-59

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF

MASTER OF COMPUTER APPLICATIONS

Bengaluru– 560059

DECLARATION

We, Laxmikanth (1RV20MC036) and Roshith K R (1RV20MC086), students of Third Semester MCA hereby declare that the Minor Project titled **"Development of vehicle tracking system"** has been carried out and completed successfully by us and is our original work.

Date of Submission:

Signature of the Student

Laxmikanth

USN: 1RV20MC036

Roshith K R

USN:1RV20MC086

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the success of any work would be incomplete unless one mentioned the names of people who made it possible whose constant guidance and encouragement served a beacon light and served our efforts with success.

We would like to express wholehearted gratitude to **Dr. K.N Subramanya**, Principle, R.V. College of Engineering for providing us an opportunity and support to complete the project successfully.

We are grateful to **Dr. Andhe Dharani**, Professor and Head of Department of MCA for encouraging us and put in place a healthy environment to carry out the project.

We would like to make a special mention of **Dr. K. S. Jasmine**, Associate Professor, Department of MCA, who had been a paying keen attention and providing us a systematic way in the conduct of my project.

We express our sincere thanks to our project guide **Dr. S. Anupama Kumar**, Associate Professor, Department of MCA, R.V. College of Engineering for her constant encouragement, support and guidance during the project work.

We are also thankful to the lab in-charge, staff and all faculty of the department for their help and support during the project.

On a moral personal note, our deepest appreciation and gratitude to our families who have been a foundation of inspiration and have provided unrelenting encouragement and support.

Laxmikanth (1RV20MC036)

Roshith K R (1RV20MC086)

Department of MCA

RV College of Engineering®

Bengaluru-59

ABSTRACT

Automatic Vehicle Tracking System recognizes the registration plate from vehicle image automatically and fetches the registration details of the vehicle. This system includes various operations such as loading pictures, localizing the number plate, character segmentation and character recognition. The main idea of this system is to design and develop effective image processing techniques and algorithms to localize the registration plate in the captured image, to divide the characters from that number plate and to identify each character of the segment by using the Open Computer Vision Library. This has been implemented in K-NN algorithm and python programming language. Many applications can be implemented by using this system, such as security, highway speed detection, violation of light, identification of handwritten text, discovery of stolen cars and automatic fee collection system.

The main modules are image processing, plate recognition, character recognition, user interface , database to store and fetch vehicle records. The major techniques used in the implementation of the designs are gray scale conversion, black and white image conversion, filling holes, border detection and image segmentation.

The automatic vehicle tracking system accurately localizes the registration plate on the input image.The character recognition algorithm is not accurate enough for the system to work with precision. Accuracy of the character recognition model can be enhanced further with deep learning techniques.

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Background

Traffic management is a serious issue confronted by the cities. Together with the increase in the number of vehicles, the connected problems have started to rise which includes violation of traffic rules, identifying the vehicle owner etc. As the number of rule violations increase, the fine collection also increases which results in a burden on Traffic police men. An efficient system that tracks all traffic violations and maintains the vehicle record is highly necessary.

Vehicles in each country have a unique identification number, which is written on its registration plate. This number distinguishes one vehicle from the other, which is useful especially when both are of same make and model. An automated system can be implemented to identify the registration plate of a vehicle and extract the characters from the region containing a registration plate. The registration plate number can be used to retrieve more information about the vehicle and its owner, which can be used for further processing. Such an automated system should be small in size, portable and be able to process data at sufficient rate.

In India, vehicle registration plates do not follow a standard language, font or size. Because of the variations in displaying the number plates, vehicle number plate extraction, segmentation and recognition are crucial. This system considers vehicle number plates which contain English characters and numbers only. The system works satisfactorily for wide variation of condition and different types of vehicle number plates. The system is implemented and executed in Python, OpenCV and performance is tested on real time images.

1.2 OpenCV

OpenCV [OpenCV] is an open source computer vision library available from http://SourceForge.net/projects/opencvlibrary. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages.[1]

OpenCV was designed for computational efficiency and with a strong focus on real- time applications. OpenCV is written in optimized C and can take advantage of multicore processors. If one desires further automatic optimization on Intel architectures [Intel], one can buy Intel's Integrated Performance Primitives (IPP) libraries [IPP], which consist of low-level optimized routines in many different algorithmic areas. OpenCV automatically uses the appropriate IPP library at runtime if that library is installed.[1]

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 users and an estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

1.3 Motivation

The main purpose of this project is to localize the registration plate, detect characters on it and fetch vehicle records on a vehicle image provided as an input. An efficient algorithm is developed to localize the registration plate in varying image quality. This algorithm localizes the registration plate from an image and provides it as an input to the Character extraction algorithm. The scope of this project is to localize the registration plate from the given vehicle image and display the vehicle records. This project can work

as a base for future improvements in the field of image processing, especially in registration plate extraction and character recognition.

CHAPTER 2: LITERATURE SURVEY

The comparative study of applications and methods on template matching is carried out . Template is primarily a sub-part of an object that is to be matched amongst entirely different objects.The techniques of template matching are flexible and generally easy to make use of, that makes it one amongst the most famous strategies of object localization.Template matching is carried out in versatile fields like image processing, signal processing, video compression and pattern recognition. The following template matching techniques are used Naive Template Matching, Image Correlation Matching, Sum of Absolute Difference, sum of square difference.[2]

The system with vehicle detection, tracking and license plate recognition system has been implemented. It consists of modules such as vehicle detection, license plate extraction and character recognition. Here, first foreground estimation is done by Gaussian mixture model, then a real time and robust method of license plate extraction based on block variance technique is proposed. License plate extraction is an important stage in license plate recognition for automated transport systems. The extracted license plates are segmented into individual characters by using a region-based approach. The recognition scheme combines adaptive iterative thresholding with a template matching algorithm.[3]

The paper presents a prosperous method to identify vehicle number plates. The proposed technique is built on morphological operations based on different structuring elements in order to maximally exclude non-interested regions and improve object area. This system has been experienced using a database of number plates and simulated results demonstrate major improvements as compared to other conventional systems.[4]

The paper's presents methodologies to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or areas around top government offices e.g. Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition

technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address, etc.[5]

The Automatic Number Plate Recognition System (ANPR) plays an important role in addressing these issues as its application ranges from parking admission to monitoring urban traffic and to tracking automobile thefts. In this paper, an attempt to review the various techniques and their usage. The ANPR system has been implemented using template Matching and its accuracy was found to be 80.8% for Indian number plates.[6]

The paper presents a novel method for real-time automatic license plate detection in high resolution videos. Although there have been extensive studies of license plate detection since the 1970s, the suggested approaches resulting from such studies have difficulties in processing high-resolution imagery in real-time. Paper proposes a novel cascade structure, the fastest classifier available, by rejecting false positives most efficiently. Furthermore, the paper focuses on training the classifier using the core patterns of various types of license plates, improving both the computation load and the accuracy of license plate detection. To show its superiority, the approach is compared with other state-of-the-art approaches. In addition, authors have collected 20,000 images including license plates from real traffic scenes for comprehensive experiments. The results show that the proposed approach significantly reduces the computational load in comparison to the other state-of-the-art approaches, with comparable performance accuracy.[7]

The paper proposes methodologies to recognize characters written on a vehicle license number plate. Method used is for the recognition of the characters from the license number plate and is based on template-matching. In this method, first the image of a car license number plate is taken as input, then pre-processing steps such as conversion to Gray-scale image, dilation, erosion, convolution is done to remove noise from the input image. Then each character in the number plate is segmented. Segmentation is done on the basis of connected components. Then after segmentation, recognition of characters is done by matching templates to the segmented characters. Matching is done on the basis of correlation between segmented characters and the templates in the database. In the last step, a text file shows the recognized number and the character from the input image.[8]

An algorithm for vehicle number plate extraction, character segmentation and recognition is presented. Database of the image consists of images with different size, background, illumination, camera angle, distance etc. The experimental results show that number plates are extracted faithfully based on vertical edge detection and connected component

algorithms, with a success rate of 85%. Character segmentation phase using connected component analysis and vertical projection analysis works well with the success rate of 80%. The success rate achieved for character recognition is 79.84%.[9]

From the following literature survey , various methodologies for vehicle registration plate localization such as edge detection based, pixel Threshold based segmentation are known which are efficient on high resolution images. In real time due to diverse effects the quality of vehicle images is not up to the mark but still the algorithm should be capable of localizing the number plate. Character recognition models are efficient up to 79.84% by component analysis and vertical projection analysis which has to be improved inorder to system function with precision.

2.1 Existing System

The vehicle registration plates appear in different types of fonts, either in a single row or double, different font sizes, and of varying character counts. Due to such complications even localizing or detecting the registration plates becomes a tedious process. In the existing system, foreground estimation is done by Gaussian mixture model then proposing a real time and robust method of license plate extraction based on block variance technique. Registration plate extraction is an important stage in registration plate recognition for automated vehicle tracking systems. The Extracted registration plates are segmented into individual characters by using a region-based approach. The recognition scheme includes adaptive iterative thresholding with a template matching algorithm.

2.2 Disadvantages of Existing System

- Registration plate localization doesn't work accurately on low quality images
- Noise reduction is not considered
- Image normalization is not considered
- Manual detection is expensive and labour intensive

2.3 Problem Statement

The system should necessarily localize and extract the registration plate region from a larger scene input image. Having extracted the registration plate region, then the

alphanumeric characters in the plate need to be extracted with high precision. Deliver the extracted characters to an Optical Character Recognition [OCR] system for recognition. In order to identify a vehicle automatically reading its license plate is highly essential. Locating the region of interest helps in dramatically reducing both the computational expense and algorithm complexity.

2.4 Proposed System

This project is on the development of the vehicle tracking system which automatically identifies the vehicle and displays the vehicle records . The proposed algorithm is based on image preprocessing , localization of registration plate, region based segmentation of plate characters and recognition of characters through OCR. Extraction of plates with high accuracy is a difficult task. In this project, a module on registration plate extraction method is developed . The medeth is basically based on the Edge Detection algorithm including four major stages, which are RGB to gray-scale conversion, Gaussian Blurring, morphological operations and extracting the accurate location of the license plate. Mean squared error method is used for recognition of characters.

2.4.1 Proposed Methodologies

- Read vehicle image
- Gaussian Blur
- RGB to grayscale conversion
- Sobel edge detection
- Otsu thresholding
- Morphological operations
- Localization
- Character segmentation
- Train the KNN model for character recognition
- Character recognition
- Fetch vehicle records

2.4.2 Advantages of Proposed System

- Accurate localization of registration plate
- Accurate segmentation of registration plate characters

2.4 Objectives

The main objective of the project is to experiment deeply and design algorithms for the image segmentation and character recognition problems within the Registration Plate Recognition framework. To develop a system in python which can perform detection as well as recognition of vehicle number plates and fetch respective records.

- Design algorithm to localize the registration plate on varying input image quality
- Train and test the KNN model to recognize characters on the registration plate
- Fetch and display the vehicle records from database

CHAPTER 3:

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Functional and nonfunctional requirements

3.1.1 Functional requirements

Functional requirement refers to the functionalities that are applicable to a system. The functional requirements of automatic vehicle tracking systems are stated below. The system must be able to:

1. Read images
2. Localize the number plate in the image
3. Segmentation of characters from localized plate
4. Recognise the characters
5. Fetch the vehicle records

3.1.2 Nonfunctional requirements

A non-functional requirement is about how the system must behave or how is the system's behavior. This also specifies the system's quality characteristics or quality attribute. In order to put this constraint upon the specific system behavior, the qualities goals of the designed system should go in these:

Execution qualities

- Functionality
- usability
- Security

- Effectiveness and efficiency

Evolution qualities

- Availability
- Manageability
- Reliability

3.2 User requirements

1. The ANPR system should read vehicles' images.

2. Quick detection of the vehicles' number plates.

3. Online optical character recognition (OCR) of the captured vehicles' number plates.

4. Providing a GUI operator control.

3.3 System requirements

The following is the suggested minimum system configuration to run the Vehicle Tracking System software:

- Intel i3 CPU or higher
- 4GB RAM or more
- Windows 7 (64 bit), Windows 8 (64 bit), Windows 10 (64 bit)

3.4 Software requirements

3.4.1 PYTHON 3.9.0

Python is a high level, interpreted and general purpose dynamic programming language that focuses on code readability. It has fewer steps when compared to Java and C. It was

founded in 1991 by developer Guido Van Rossum. It is used in many organizations as it supports multiple programming paradigms. It also performs automatic memory management.

CHAPTER 4: DESIGN

4.1 System architecture

System architecture is the conceptual model that defines the structure, behaviour and views of a system. The below figure is an architectural design for the Automatic Vehicle tracking system. Vehicle tracking system is a system that reads and processes an image that consists of a vehicle number plate as input and recognizes the number plate and produces vehicle owner details as output automatically.

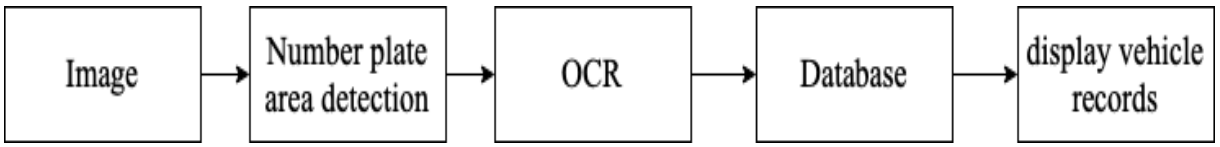


Fig.4.1 Automatic vehicle tracking System architecture

3.2 Activity diagram

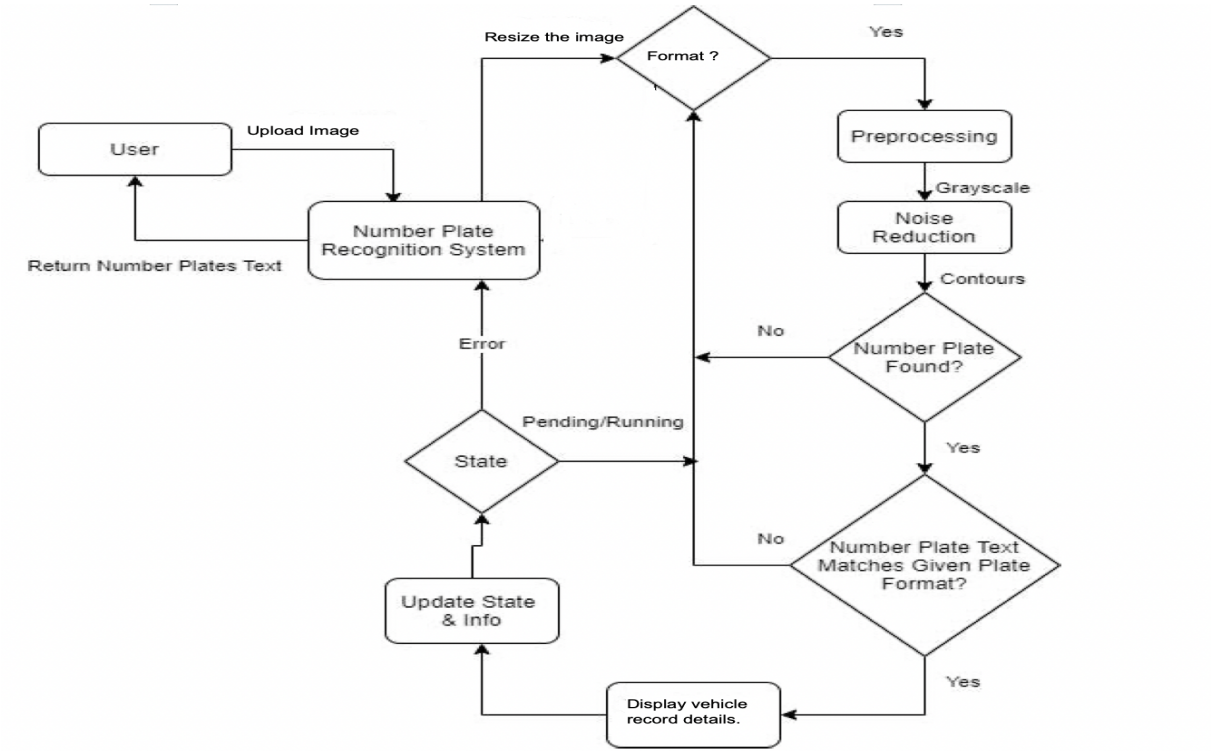


Fig.4.2 Activity diagram

4.3 Use case diagram

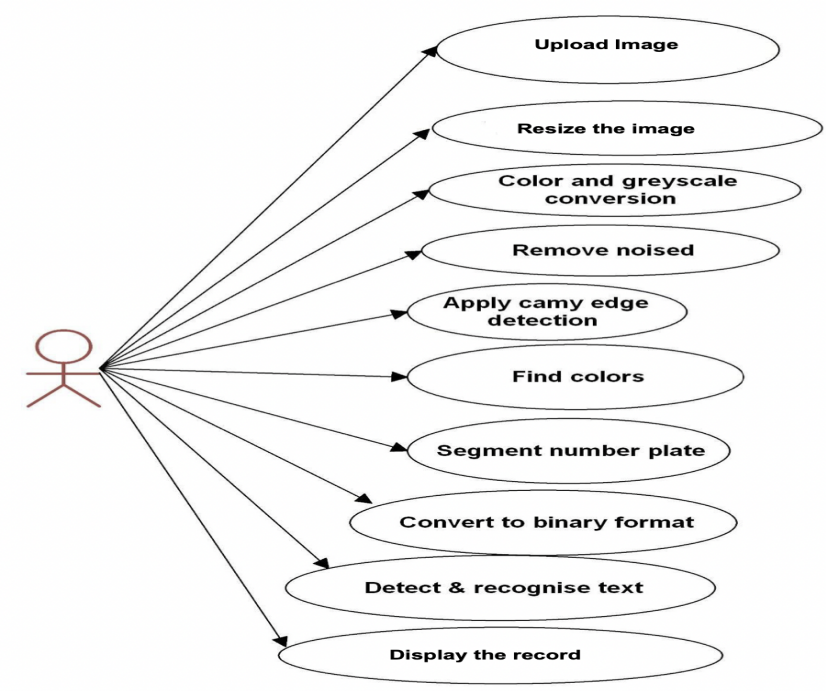


Fig.4.3 Use case diagram

4.4 Sequence diagram

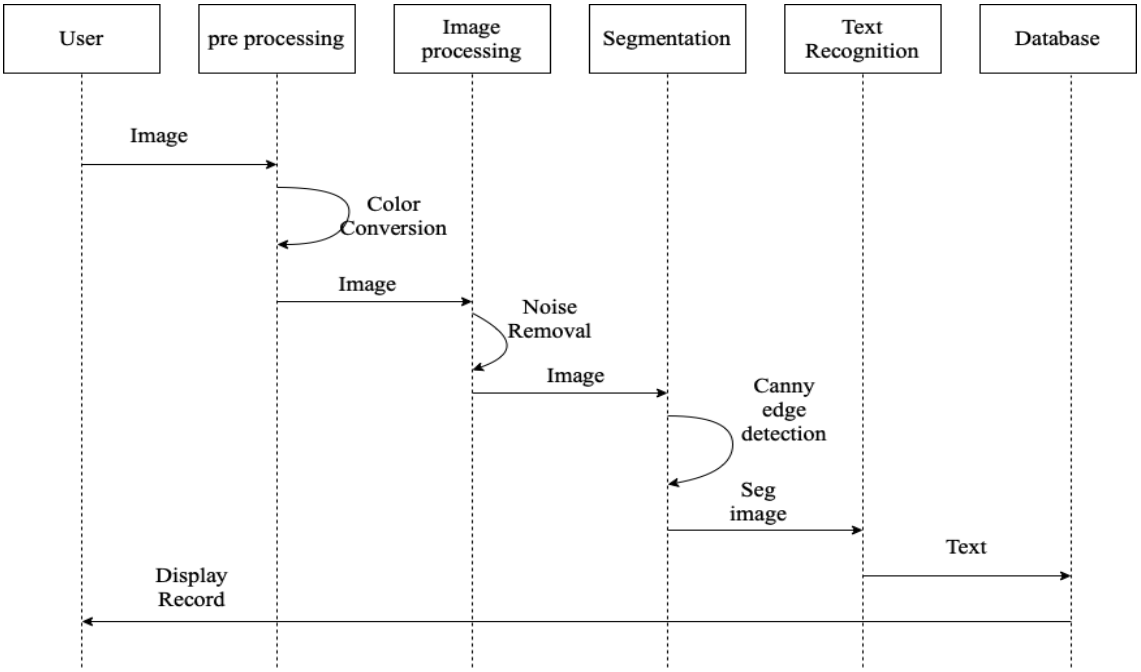


Fig.4.4 Sequence diagram

4.5 Class diagram

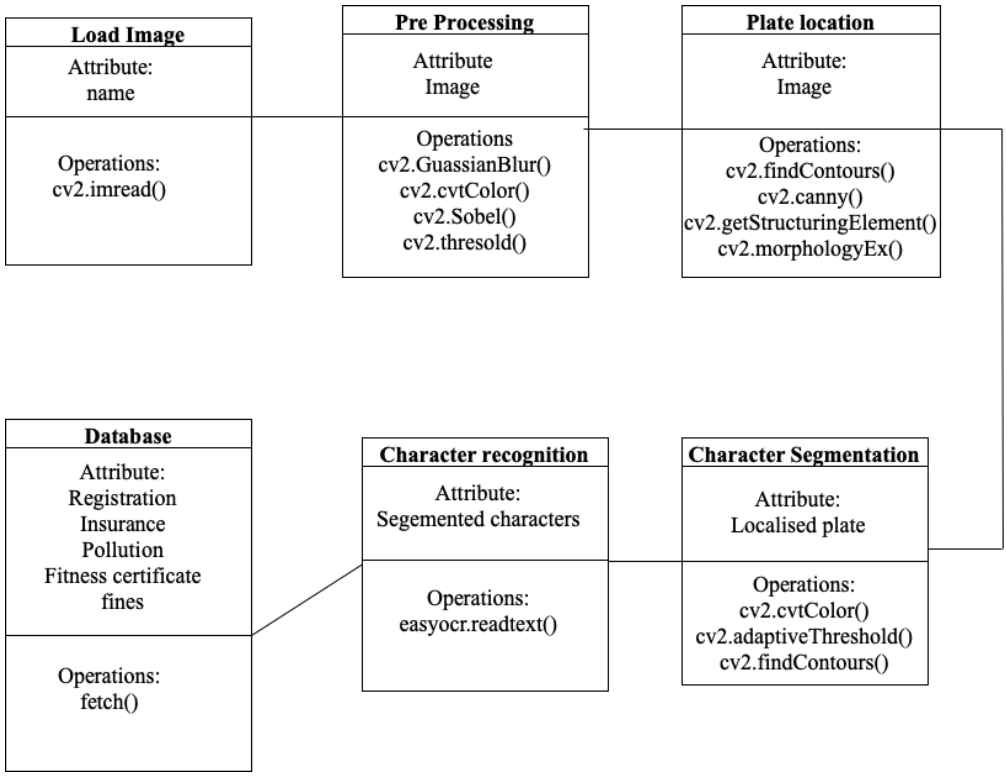


Fig.4.5 class diagram

CHAPTER 5 : IMPLEMENTATION

5.1 Platform selection

IDLE (short for integrated development environment or integrated development and learning environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language . It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk)

5.2 Modules

5.2.1 Preprocessing

Pre-processing steps involves the following methods

Gaussian blur: Gaussian blurring is highly effective in removing gaussian noise from the image. Image blurring is achieved by convolving the image with a low-pass filter kernel. It actually removes high frequency content (e.g: noise, edges) from the image resulting in edges being blurred. In this approach, instead of a box filter consisting of equal filter coefficients, a Gaussian kernel is used. It is done with the function, cv2.GaussianBlur().[13]

Fig.5.1 Input image

Fig.5.2 Gaussian Blur image

Gray scale conversion: It involves conversion of RGB image into Gray image. A gray-scale image is composed of different shades of grey color. A true color image can be

converted to a gray scale image by preserving the luminance (brightness) of the image.Here the RGB image is a combination of RED, BLUE AND GREEN colors. The grayscale image is obtained from the RGB image by combining 30% of RED, 60% of GREEN and 11% of BLUE. This gives the brightness information of the image. The resulting image will be two dimensional. The value 0 represents black and the value 255 represents white. The range will be between black and white values.

Fig.5.3 Gray Scale image

Sobel operator: It is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasizing edges. The Sobel edge detector is a gradient based method. It works with first order derivatives. It calculates the first derivatives of the image separately for the X and Y axes. The derivatives are only approximations (because the images are not continuous).

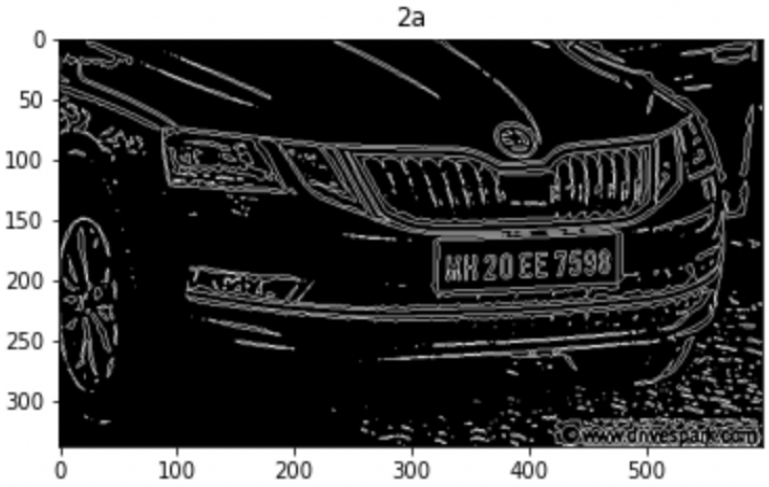


Fig.5.4 Sobel operator output

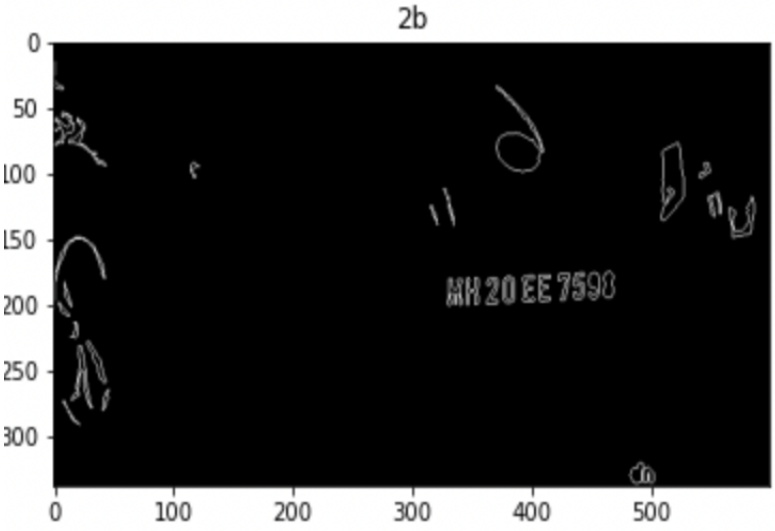


Fig.5.5 Plate extraction image

Thresholding: Converts gray scale image to binary image. If a pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black). The function used is cv2.threshold. First argument is the source image, which should be a grayscale image[4]

5.2.2 Plate detection

Plate localization is based on morphological operations and aspect ratio.

Morphological operation: Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. These techniques can be extended to grayscale images [13]. Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood. A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image. When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighborhood under the structuring element. The structuring element is said to fit the image if, for each of its pixels set to 1, the corresponding image pixel is also 1. Similarly, a structuring element is said to hit, or intersect, an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1.[4]

Localization: Based on the aspect ratio of the contours, license plates in the image are selected.[4]

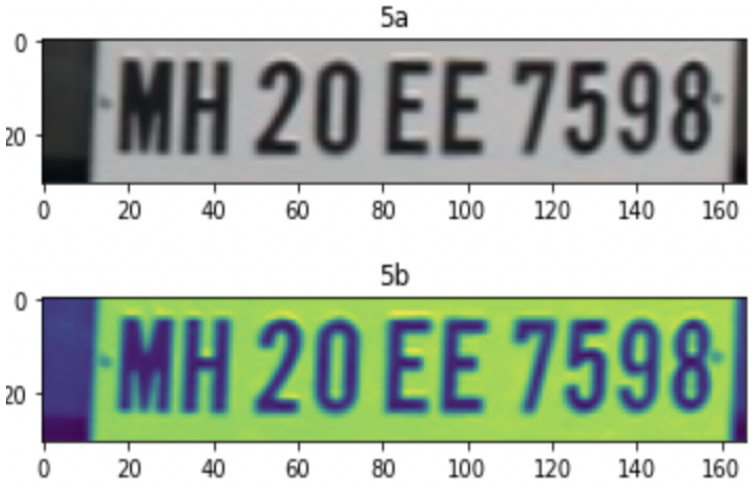
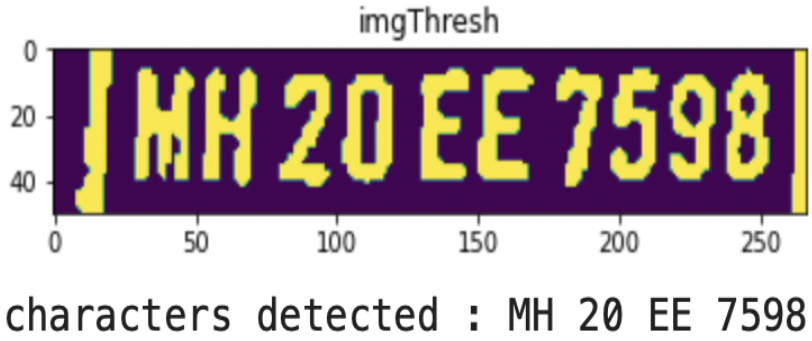


Fig.5.6 Plate localization

5.2.3 Character detection

In this work CHARS74K and ICDAR2003 datasets are used. The CHARS74K dataset contains a total of over 74K images which explains the name of the dataset. The ICDAR2003 dataset contains 249 images with 5370 characters and 1106 words [10]. Each dataset split into a number of training and testing portions [11]

This process involves training the KNN model with the above dataset. Each segmented character of the registration plate is fed into the model which will output the most nearest possible character. Finally the output is displayed on the monitor.[5]



characters detected : MH 20 EE 7598

Fig.5.7 Character detection

CHAPTER 6: SOFTWARE TESTING

6.1 Unit testing

Unit testing is a software testing method by which individual units of source code, sets one or more computer program modules together with associated control data, usage procedures and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could also be an individual method.

6.1.1 Test cases for plate localization

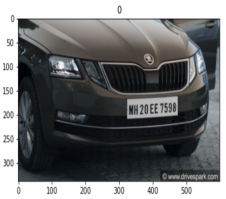
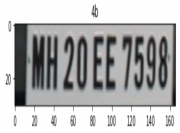
Test Case	Input	Expected output	Output	Status
Plate detection for front view of vehicle				Pass

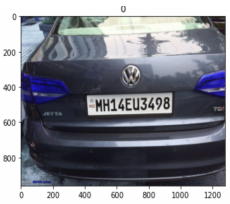
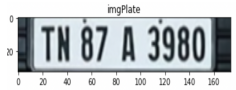
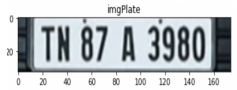
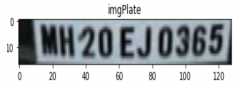
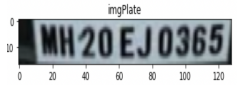
Plate detection for rear view of vehicle				Pass
Plate detection of rear view of vehicle				pass
Plate detection for front view of vehicle				Pass
Plate detection for front view of vehicle				Pass

Table.6.1.1 Test cases for plate detection

6.1.2 Test cases for character detection

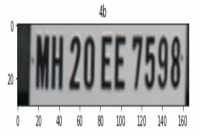
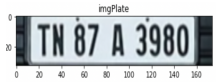
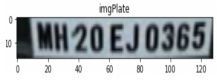
Test Case	Input	Expected output	Output	Status
Plate detection for front view of vehicle		MH 20 EE 7598	MH 20 EE 7598	Pass
Plate detection for rear view of vehicle		MH14EU3498	MH14EU3498	Pass
Plate detection of rear view of vehicle		MH12DE1433	MH11DE1433	Fail
Plate detection for front view of vehicle		TN 87 A 3980	TN 87 A 3980	Pass
Plate detection for front view of vehicle		MH 20 EJ 0365	nil	Fail

Table 6.1.2 Test cases for character detection

6.2 System testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

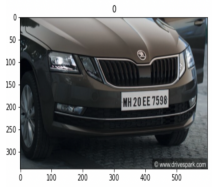
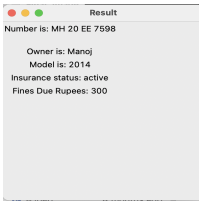
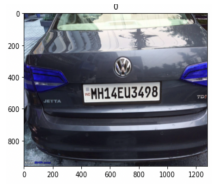
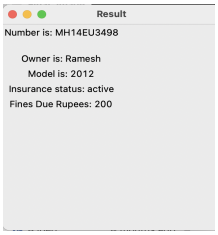
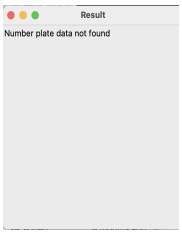
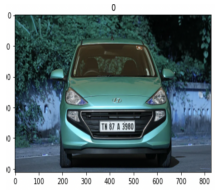
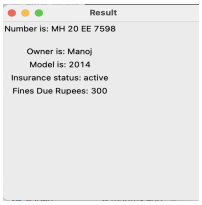
Test case	Input	Expected output	Output	Status
Plate detection for front view of vehicle		Registration details		Pass
Plate detection for front view of vehicle		Registration details		Pass
Plate detection for front view of vehicle		Registration details		Fail
Plate detection for front view of vehicle		Registration details		Pass

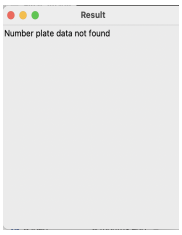
Plate detection for front view of vehicle		Registration details		Fail

Table 6.2 Test cases for system testing

CHAPTER 7 : RESULT AND DISCUSSION

The algorithm was tested using different license plates having various background conditions, light condition and pixel quality. The results for character recognition are shown in Table 5.1.2

For calculating the precision and recall values we need to know the concepts of true-positive, false-positive and false-negative.

- The positive class is confirmed by all the text in the output file, all the “recognized” text, correctly or incorrectly recognized.
- The true-negative items are the ones which are correctly rejected.
- The true-positive items are the ones correctly labeled as belonging to the positive class. So, in this case, the true-positive items are the correctly recognized characters by ANPR software. That will be the characters that are present in both the ground truth file and the program’s output file.
- The false-positives are the items incorrectly labeled as belonging to the positive class; in this case, all the incorrectly recognized characters.
- The false-negatives are the items that are not labeled as belonging to the positive class but should not have been. In this case, the false negatives are all the characters that are in the image but didn’t get recognized,i.e., all the characters in the ground truth file that are not present in the program's output file.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

$$Accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}$$

		Prediction outcome: positive	Prediction outcome: negative	
Actual value	positive	TP	FN	$TP + FN$
	negative	FP	TN	$FP + TN$
		$TP + FP$	$FN + TN$	

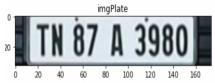
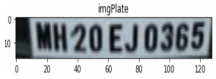
Actual Plate	Predicted Plate	Precision	Recall	Accuracy
	MH 20 EE 7598	10/(10+0)= 100%	10/(10+0)= 100%	10/(10+0+0+0)= 100%
	MH14EU3498	10/(10+0)= 100%	10/(10+0)= 100%	10/(10+0+0+0)= 100%
	MH11DE1433	9/(9+1)= 90%	9/(9+0)= 100%	9/(9+0+1+0)= 90%
	TN 87 A 3980	10/(10+0)= 100%	10/(10+0)= 100%	10/(10+0+0+0)= 100%
	nil	0/(0+0)= 0%	0%	0%
		78%	80%	78%

Table 7.1 Result calculations

CHAPTER 8:

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

Registration plate recognition applications are becoming increasingly complex in the Indian context with the phenomenal exponential growth in car, two-wheeler and auto Industries. Automatic vehicle tracking systems can be applied on toll collection, charging systems in parking spaces, management vehicles in parking spaces, and traffic monitoring, etc.,. Due to its wide applications Automatic vehicle tracking have posed new research tasks in Number plate recognition with newer dimensions. We have developed software for automatic license plate recognition with a GUI for user simplification. Character segmentation has been implemented on extracted number plates. Finally, segmented characters are recognized by using KNN algorithm.

8.2 Limitation

- Image should have proper light exposure
- Although plate localization accuracy is high, character recognition is not accurate

8.3 Future scope

The implementation of the proposed system can be extended for the recognition of registration plates of multiple vehicles in a single image frame. Android applications can be developed for the government traffic surveillance management systems. Accuray of character recognition can be improved using various deep learning algorithms like Convolutional neural networks (CNN).

REFERENCES

[1].Gary Bradski and Adrian Kaehler,"Learning OpenCV",O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[2].Paridhi Swaroop, Neelam Sharma, "An Overview of Various Template Matching Methodologies in Image Processing", International Journal of Computer Applications (0975 – 8887) Volume 153 – No 10, November 2016

[3].Lucky Kodwani & Sukadev Meher "Automatic License Plate Recognition in Real Time Videos using Visual Surveillance Techniques "ISSN (PRINT): 2320 – 8945, Volume -1, Issue -6, 2013.

[4].Riazul Islam, Kazi Fatima Sharif and Satyen Biswas, "Automatic Vehicle Number Plate Recognition Using Structured Elements", IEEE Conference on Systems, Process and Control December 2015, pp 44-48.

[5]Muhammad Tahir Qadri, Muhammad Asif "Automatic Number Plate Recognition System for Vehicle Identification using Optical Character Recognition" IEEE 2009.

[6]Aniruddh Puranic, Deepak K. T, Umadevi V "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching" International Journal of Computer Applications (0975 – 8887) Volume 134 – No.1, January 2016.

[7]Byung-Gil Han, Jong Taek Lee, Kil-Taek Lim, and Yunsu Chung "Real-Time License Plate Detection in High Resolution Videos Using Fastest Available Cascade Classifier and Core Patterns" ETRI Journal,Volume 37, Number 2, April 2015.

[8]Nighat Naaz Ansari, Ajay Kumar Singh "License Number Plate Recognition using Template Matching" International Journal of Computer Trends and Technology (IJCTT) – Volume 35 Number 4- May 2016.

[9]M. M. Shidore, S. P. Narote," Number Plate Recognition for Indian Vehicles" IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.2, Feb. 2011.

[10]T. E. d. Campos, B. R. Babu, and M. Varma, "Character Recognition In Natural Images," 2009.

[11] L. Neumann, and J. Matas, "Real-Time Scene Text Localization and Recognition," Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 3538-3545, 2012.

[12]Qadri, M.T., Asif, M. "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition", 17th April 2009

[13]Melba Lira D'souza, Brenda Meena D'souza, "Car Number Plate Recognition System",International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume - 3 Issue - 8 August, 2014 Page No. 7837-7840