Automatic Number Plate Recognition

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Information Technology

by
Saksham Agrawal
Aditya Gangwar
Sridhar Addagatla
Ram Lakhan Meena
Rohit Ranjan

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD
November, 2017

UNDERTAKING

I declare that the work presented in this report titled "*Automatic Number Plate Recognition*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Information Technology***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

November, 2017
Allahabad

(Saksham Agrawal
Aditya Gangwar
Sridhar Addagatla
Ram Lakhan Meena
Rohit Ranjan)

CERTIFICATE

Certified that the work contained in the report titled "*Automatic Number Plate Recognition*", by *Saksham Agrawal*
Aditya Gangwar
Sridhar Addagatla
Ram Lakhan Meena
Rohit Ranjan, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Suneeta Agarwal)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

November, 2017

Preface

This project has been undertaken by the team as a part of the B.Tech curriculum of Information Technology.
Through this project we have hoped to achieve a nuanced insight into the world of Image Processing and Machine Learning.
We also hope our theoretical report along with the accompanying implementations find relevance and importance in the field of Computer Science.

Acknowledgements

We would like to take this opportunity to acknowledge and appreciate the efforts of the people who have helped us during our research and documenting this thesis. First and foremost, we would like to express our deep-felt gratitude towards our thesis adviser - Dr. Suneeta Agarwal for her excellent guidance and invaluable support. Her invaluable and continuous exposure, not only to research but also to other aspects of life have helped in constructive development and shaping of our ideologies and understanding.

We would also like to thank all the teachers of the Computer Science and Engineering Department, who have helped us develop a deep love and passion for this stream.

Contents

0.1 Abstract

A license plate recognition system is divided into four steps: Pre-processing, license plate identification from the vehicle image, license plate character segmentation, and finally license plate character recognition. We have utilized various properties of license plate for identification(like its rectangular shape) and also filtered the candidate plates using some metrics(area of license plate, position of the plate, angle with the horizontal of the plate) which have been found after experimentation with images. For segmentation connected component analysis is done and then character recognition is carried out using tesseract.

Chapter 1

Introduction

Intelligent Transportation Systems (ITSs) have a powerful impact on human beings lives nowadays. The main aim of Intelligent Transportation System is to exploit the transportation mobility and safety to improve the productivity by the use of emerging technologies and is built up of sixteen various types of technology based systems which are categorized into intelligent vehicle systems and infrastructure systems. Among them, Automated License Plate Recognition (LPR) Systems play an important role in various real time applications including the following such as: electronic payment systems (toll payment and parking fee payment), traffic monitoring systems, border crossing control systems, identification of stolen vehicles, petrol station forecourt surveillance systems, red light, violation systems, vehicle tracking systems, speed enforcement control systems, ticketing vehicles without the human control, policing, security and customer identification systems. License Plate numbers uniquely identifies a particular vehicle. Each country has their own license plate format which differs in their sizes and colors. So there is a necessity for them to develop the License Plate Recognition system suitable for the vehicle License Plate format. Hence the Intelligent Transport Systems heavily depend on the robust Automated License Plate Recognition Systems (ALPR). Cropping a License Plate manually with a mouse from the input image is a very simple method, but when it comes to do the same for any real time or automatic system, difficulty may arise and will not be suitable for such systems. Therefore, a License Plate Recognition

(LPR) System consists of the following FOUR stages and the basic block diagram of the system is as shown in Fig 1.

- Acquisition from digital camera.
- License Plate Localization (LPL: Detecting and verifying the location of the License Plates).
- Character Segmentation (Segmenting the characters from the License Plates).
- Character Recognition (Recognizing the segmented characters from the License Plates)

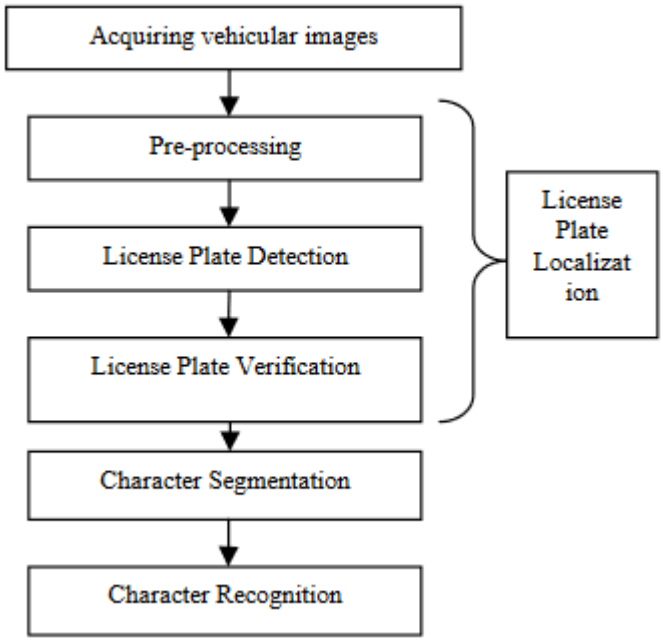


Figure 1: Basic block diagram of License Plate Recognition System

1.1 Motivation

An automated model for most of the activities is efficient in terms of time and labour. Breaking traffic rules or entry to a restricted zone is surely monitored manually, but a 24-hour sharp surveillance is only possible if we can automate the detection of defaulters looking at the images. And so, our project aims identifying the license plate number directly from the image of the vehicle. This project can be used as a service for theft detection system. The identified license plate could be used to check whether the corresponding car is stolen or not by making a query on a central database.

Chapter 2

Related Work

Automatic recognition of license plate is an essential stage in intelligent traffic system and many methods have been developed for the construction of License Plate Recognition (LPR) system. Real time LPR plays a major role in automatic monitoring of traffic rules and maintaining law enforcement on public roads [1]. The automatic identification of vehicles by the contents of their license plates is important in private transport applications.

The steps involved in recognition of a license plate are Image acquisition, Candidate region extraction, Segmentation, and Recognition. There is a lot of literature in this area but some of the related work is as follows: Naito et. al. [5] developed a sensing system, which uses two CCDs (Charge Coupled Devices) and a prism to capture the image. Hontani et. al. [3] proposed a method for extracting characters without prior knowledge of their position and size in the image. Cowell et. al. [2] discussed the recognition of individual Arabic and Latin characters. Their approach identifies the characters based on the number of black pixel rows and columns of the character and comparison of those values to a set of templates or signatures in the database. Mei Yu et. al. [7] and Naito et. al. [5] used template matching.

Chapter 3

Tools and Libraries Used

3.1 Python

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum. An interpreted language, Python has a design philosophy which emphasizes code readability and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

3.2 Open Source Computer Vision

Open Source Computer Vision, also known as OpenCV, is a real time computer vision library with many image processing functions developed by Intel. OpenCV is written in C++ and its primary interface is in C++. There are bindings in Python, Java and MATLAB. This API provides functions to perform various complex Image processing operations. We used the python interface of OpenCV.

Chapter 4

Proposed Work

4.1 Pre-processing

4.1.1 Introduction

The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing. Following pre-processing techniques are implemented in the current work with corresponding images are shown for the illustration.

Figure 2: Original Image.

4.1.2 Conversion Of Color Image To Gray Image

The color pictures contain thousands of bits information, which when processed directly, reduces the execution speed of processing system. Therefore, colored images are converted to grey scale before processing, unless the detailed color information is needed in the further steps.

For example, a pixel of 24-bit BMP images takes three bytes in memory, which contains red, green, blue color information. We denote red, green and blue with R, G and B The grey value is showed by g. We can use following formula to turn a image into gray image:

$$g = (0.3R + 0.59G + 0.11B)/3$$

Figure 3: Conversion Of Color Image To Gray Image.

4.1.3 Bilateral filtering

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.
Bilateral filtering is highly effective in noise removal while keeping edges sharp.

Figure 4: Noise Removed Image.

4.1.4 Histogram equalization

It is a method that improves the contrast in an image, in order to stretch out the intensity range. This method usually increases the global contrast of images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values

Figure 5: After Histogram equalization.

4.1.5 Morphology and Image subtraction

Erosion is used to remove irrelevant details from binary image. Structured removal of image region boundary pixels, Opening is the combination of erosion-dilation. Structured filling of image region boundary pixels closing is the combination of dilation-erosion. Morphological methods were used to achieve the detection of License Plate. This obtained image is then subtracted from the above equalized image to remove noise.

Figure 6: Image after applying Morphological Opening

Figure 7: Image after subtracting Fig 5 from Fig 4

4.1.6 Otsu Thresholding

Thresholding is the process of reduction of gray image into binary image. Otsu thresholding is used to automatically perform clustering-based image thresholding. The algorithm assumes that the image contains two classes of pixels following bimodal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal.

Figure 8: Image after Otsu Thresholding

4.1.7 Canny Edge Detection

In order to find contours we need to find edges first. This can be done efficiently with canny. Canny edge detection is known as optimal detector which is used to detect wide range of edges in an image. This algorithm aims to satisfy three main criteria:

- **Low error rate:** Meaning a good detection of only existent edges
- **Good localization:** The distance between edge pixels detected and real edge pixels have to be minimized
- **Minimal response:** Only one detector response per edge

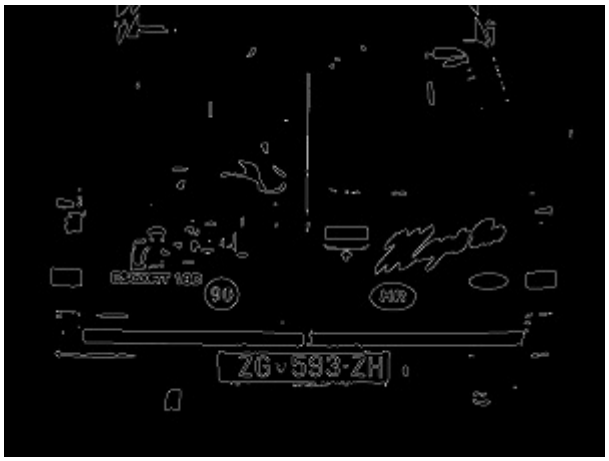


Figure 9: Image after applying Canny Edge-Detection.

4.1.8 Dilation

Edges detected by canny edge detection are strengthen with the help of dilation. We used 3x3 rectangular kernel for the dilation.

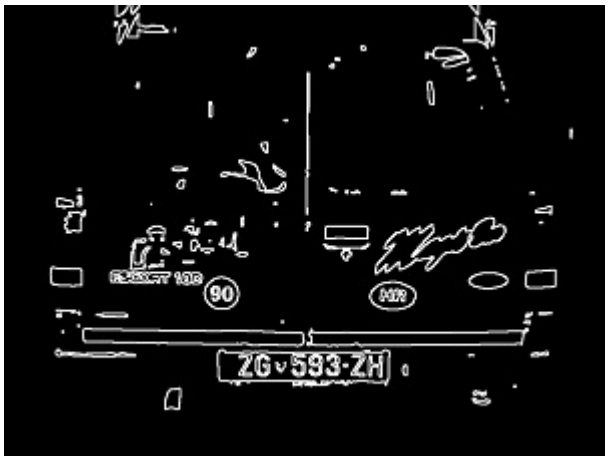


Figure 10: Image after applying Dilation.

4.2 License Plate Localization

Any license plate must be composed of contours. So system finds contours in the image that it got after above operations. Minimum area rectangle for each contour is calculated and this rectangle is passed through following heuristics to minimize number of possible plates.

4.2.1 Finding all possible license plates

The following heuristics are applied on each minimum area rectangle to be able to find possible rectangles for plate.

- **Aspect Ratio:** Although standard aspect ratio should be between 2 to 4(Standard U.S license plate 12x6 inches or 11.44x5.44) But there are many cars having aspect ratio of license plate between 1 to 10. So, we used aspect ratio between 1 to 10.

$$AspectRatio = Width/Height$$

- **Skewness of License plate:** License plate can be skewed based on the angle at which picture is taken. We are not considering License plates which are at angle more than 20°.

- **Area of License plate:** As height increases proportionally with width aspect ratio remains constant but area increases. So, we are only considering rectangles with area between 4000 pixel^2 to 25000 pixel^2.

- **Position of Plate:** The license plate is generally located in the lower half of the image so we are considering the candidate regions which are located in the lower-half of image.

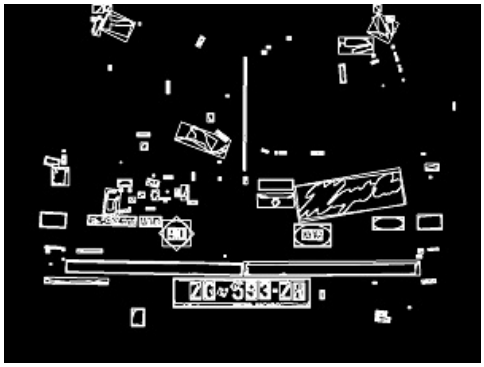


Figure 11: All Min Area Rectangles Found

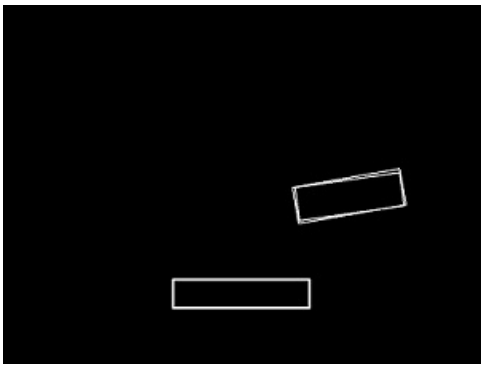


Figure 12: Possible Min Area Rectangles for Plate detection

4.2.2 Finding License plate

Now we have some possible candidates for the License plate. We find license plate using the property of license plate that it contains characters.
As each character is having certain properties such as aspect ratio, area, height and width. We use these properties to find characters in a plate and we choose the plate which is having maximum number of characters.

- **aspect ratio:** Aspect ratio of any standard character should be less than or equal to 1.
- **Area:** We used area of character between 50 pixel^2 and (length of the plate/2)*(width of the plate/4) pixel ^{2}.

Figure 13: Final Image.

4.3 Operations on detected License plate

4.3.1 Introduction

The focus of this part is on the last two phases of the license plate recognition system, i.e., license plate character segmentation and recognition. Here, the character segmentation is defined as the minimal binary region bounded-box adjusted/surrounding the characters. The automation of both character binarization and segmentation is a very complex issue because the image quality degradation. During image acquisition, factors as the diversity of plate formats, the illumination of the environment, superimposed characters, and presence of some noise in the plate can degrade the image. Once the character are precisely detected and segmented it is expected that the recognition step is performed effectively since the extracted features for classification would discriminate better each character. We have tried various approaches for segmentation which include horizontal and vertical projections, adaptive horizontal projection and finally connected components analysis. The most effective one that was used after was connected component analysis.

4.3.2 Pre-processing

The first step of pre-processing is to convert the plate image from above process to a gray level image. Second step is to convert the gray level image to a binary image i.e. thresholding. Thresholding is of three types global thresholding, adaptive thresholding, otsu thresholding. Global Thresholding or simple thresholding is just selecting a value threshold, if a pixel value is less than threshold then assign it black else make it white. Global thresholding fails when image has varying lightning conditions for various areas of image. Adaptive method here takes information of the neighbouring regions to calculate a threshold value for classifying the pixel values. Since we are taking different values for different areas so it gives us better results for images with varying illuminations. In our work we have calculated weighted mean by taking a neighbourhood of 21 and a constant value of 10 which is to be subtracted from the mean obtained. Otsu thresholding is used to automatically perform clustering-based image thresholding. The algorithm assumes that the image contains two classes of

pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal.

Figure 14: Adaptive Thresholding

4.3.3 Segmentation

We have tried different methods for segmentation which include horizontal and vertical projection, adaptive horizontal projection and connected component analysis. In horizontal projection, the value of a projection bin is the sum of the black pixels along a particular line in horizontal direction. When all values along all lines in the horizontal direction are computed, the horizontal projection is obtained. After this lower and upper bound are for plate region is calculated. The part of image above lower bound and the part below the upper bound are eliminated and we get the plate region. For lower bound the first row from top with horizontal projection zero is calculated, for upper bound the first row from bottom with horizontal projection zero is calculated. The main disadvantage with this technique was many times it was observed that no lowerbound and upperbound were found and complete image was returned. We then calculated a threshold value which is the average of horizontal projection over all rows, now the first row from top with horizontal projection

between threshold/3 and threshold is considered for lower bound and similarly for upper bound. Although this method was good but the problem with this was that it also resulted in cutting above part of characters in some plates. Thus it led to loss of some vital information in license plates.
In Vertical projection, instead of taking sum along lines in horizontal lines it is taken along the vertical direction. After this the area between two vertical lines having projection zero is cut and is considered a character. The main disadvantage with this method is that in license plates there are some unwanted symbols which are not license plate characters and they are also cut, thus it depends on the character recognition system to eliminate the invalid characters. In many cases it was found that if there are some black pixels left at top because of wrong horizontal projection then this method is unable to cut characters.
In our work to overcome the above problems we used connected component analysis to find the candidate characters and then classify them based on some heuristics to be a character or not.

4.3.4 Connected Component Analysis

1. Start from the first pixel in the image. Set current label to 1 and push it into the Queue. Go to (2).
2. If the Queue is empty then Go to(4) else pop the pixel from Que and Go to(3).
3. Look at the neighbours (based on any type of connectivity) of current pixel. If a neighbour is a foreground pixel and is not already labelled, add it to the queue then Go to(2)
4. Go to (2) for the next foreground and unlabelled pixel in the image and increment current label by 1. If all are labelled then end.

Algorithm 1: Connected Components [9]

4.3.5 Filtering Components

For each components found using above algorithm we find its contours. For each contour the contour with maximum area is considered to be a candidate for character. Now we find the bounding box for the contour and then it needs fulfill below conditions to be considered a character:

- **Aspect Ratio** The aspect ratio for character should be less than 1 as the rectangle enclosing the character will always more height as compared to the width.
- **Solidity** It is defined as the ratio of contour area and bounding rectangular area. All contours with solidity less than 0.14 were considered to be a possible character.
- **Height-Ratio** It is the ratio of height of bounding box of character and height of the image. The contours with height ratio between 0.2 and 0.95 were considered as possible character.
- **Connected Components Analysis** If a contour passes all above filters then we calculate mask for that character and segment that character out by doing bit-wise and with the thresholded image. Now the resultant image has only that component in the image. We find the number of connected components inside the resultant image. If the number of connected components is exactly 2 then only it is considered to be a character else it is ignored. Here connected components should be exactly 2 as character itself will be one component and other will be the background.

Figure 15: All bounding Rectangles

Figure 16: Mask after CCA

Figure 17: All characters

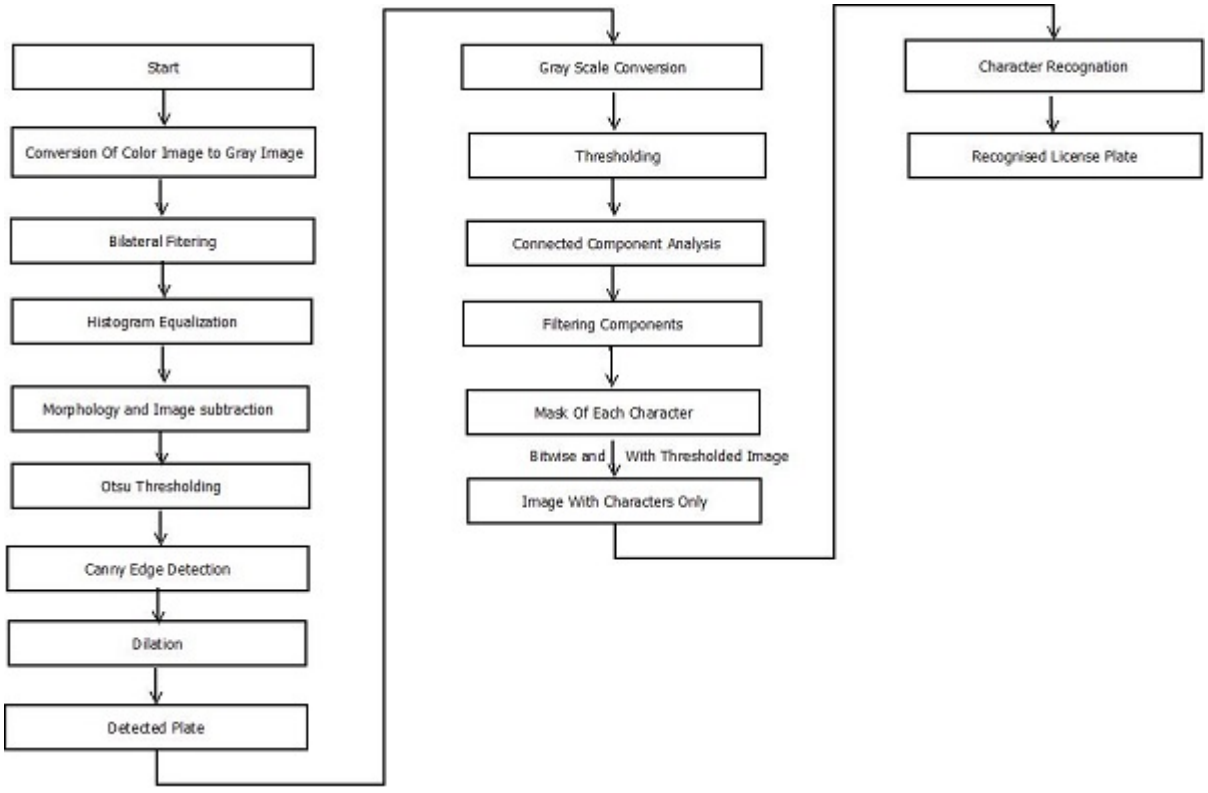
4.4 Character Recognition

After getting the segmented plate we carry out character recognition. For character recognition we have used **tesseract**- an open source optical character recognition engine. Here we get characters which are identified from the image by tesseract. License plates contains only uppercase english letters and numbers so we have eliminated all other invalid characters which might be identified due to noise present in image.

Chapter 5

Project Flow Chart and Results Analysis

5.1 Project Flow Chart



5.2 Result Analysis

The proposed system is tested on a data set of 228 images. Accuracy of detection of License plate is 77.63% (177/228). Almost all good detected plates are segmented well by our method and its accuracy is 75.877%. Thus the real cap in this system is the detection of license plate.

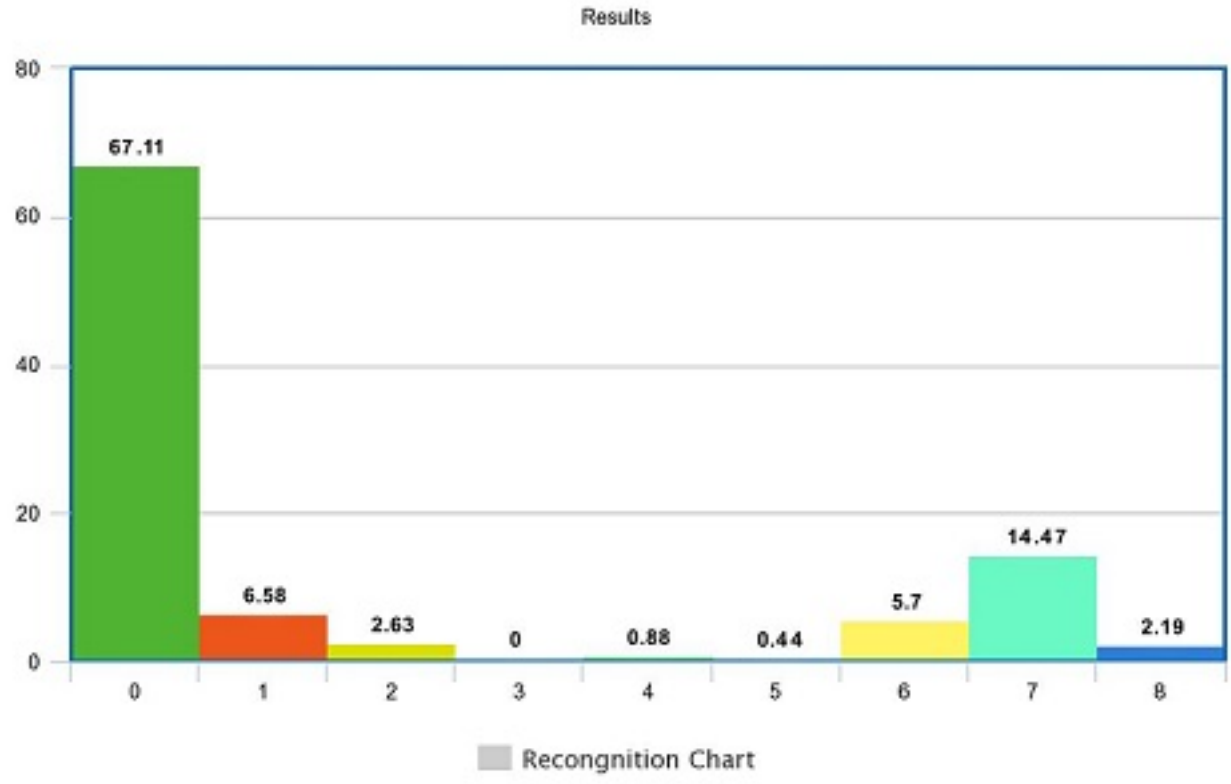


Figure 18: Above bar-chart depicts the percentage of data-set of car images on y-axis and the number of characters not recognized in comparison to original plate on x-axis.

Chapter 6

Conclusion and Future Work

In this work, a new method for license plate recognition of car images based is presented. The proposed method considers minimum area rectangles in the car image and filters these candidate regions based on some properties like aspect ratio, area and angle with horizontal to find the license plate. Connected component analysis is then used for segmenting the characters inside it. Finally, optical character recognition is carried out using tesseract. This method is computationally efficient as we have used machine learning only for character recognition part and not for all stages of the project. The heuristics were found after analysis of many plate features.
One of the future application can be a theft detection system wherein we can have a central database which will be accessible from a central server. Whenever a person lodges a complaint related to car theft, this complaint with relevant license number of car can be entered in the database by making a query to the server and then all the toll-booths can use above work to automatically recognize the license plate which then can be used by a client side application to check whether the car is stolen or not by making a query to the same central server.

References

[1] D. G. Bailey, D. Irecki, , B.K. Lim, and L. Yang, Test bed for number plate recognition applications, Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications (DELTA 02). IEEE Computer Society. 2002.

[2] J. Cowell, and F. Hussain, A fast recognition system fur isolated Arabic characters, Proceedings Sixth International Conference on Information and visualisation, IEEE Computer Society London, England, pp. 650-654,2002.

[3] H. Hontani, and T. Kogta Character extraction method without prior knowledge on size and information, Proceedings of the IEEE international Vehicle Electronics Conference (IVECOI). pp. 67-12.2001.

[4]Wenjing Jia, Xiangjian He and Massimo Piccardi, Automatic License Plate Recognition: a Review

[5] C. Nieuwoudt, and R. van Heerden, Automatic number plate segmentation and recognition, In Seventh annual South African workshop on Pottern Recognition, lAPR pp. 88-93, 1996

[6]https://opencv-python-tutroals.readthedocs.io/en/latest/

[7] M., Yu, and Y.D. Kim, An approach to Korean license plate recognition based on vertical edge matching, IEEE international Conference on Systems, Man, and Cybernetics, vol. 4, pp. 2975-2980, 2000.

[8] Chih-Hai Fana, Yu-Hang Peng,Vehicle License Plate Recognition System Design, Chung Hua Journal of Science and Engineering, Vol. 7, No. 2, pp. 47-52 (2009)

[9] Tetsuo Asano,In-place Algorithm for Connected Components Labeling, Journal of Pattern Recognition Research 1 (2010) 10-22, School of Information Science,JAIST