**ABSTRACT**

Automatic license plate recognition system aims at extracting the license plate from a vehicle and using it for various purposes. In this paper we do a systematic study of the existing ALPR systems, the basic algorithm used, the variations in the existing algorithm to improve the overall system.

We also present the list of applications where this system could be used, we elaborate one such application which is the criminal surveillance. The system which we are developing recognizes a six-digit license plate which could be also modified to detect various other types of license plates as well.

The algorithm mostly concentrates on localization of license plates and then go on to extract the characters by using morphological operations such as dilation, eroding the image, dilating, filtering etc. All these morphological operations leads to the efficiency of overall system.

**TABLE OF CONTENT**

| **Chapter No.** | | **Contents** | **Page**  **no** |
| --- | --- | --- | --- |
| 1 |  | Introduction |  |
|  | 1.1 | Problem Definition |  |
|  | 1.2 | Existing System |  |
|  | 1.3 | Proposed System |  |
| 2 |  | system Requirements |  |
|  | 2.1 | Hardware Requirements |  |
|  | 2.2 | software Requirements |  |
| 3 |  | System Design |  |
|  | 3.1 | Architecture |  |
|  | 3.2 | Use Case Diagram |  |
|  | 3.3 | Sequence Diagram |  |
| 4 |  | Implementation |  |
| 5 |  | Results |  |
| 6 |  | Conclusion |  |
|  |  | References |  |

## 1 INTRODUCTION

Vehicle plate detection and recognition appear in vast variety of applications, including travel time estimation, car counting on highways, traffic violations detection, and surveillance applications. Traffic monitoring cameras are mounted four to seven meters above the street level. Plate recognition range, where the cameras are able to capture the vehicles plates with sufficient resolution, starts from 20 to more than 50 meters away from the camera location. This range depends on the camera resolution and the lens mounted on the camera. At these heights and distances, vehicles plates are not as clearly visible as in other applications such as parking fee payment systems. High camera installation point causes some difficulties against the correct detection of vehicles plates. Vehicles with dirty plates make the situation even more complicated. On the other hand, number plate is the only trustworthy identity of a vehicle in Intelligent Transportation Systems (ITS) and correct vehicle identification depends highly on the accuracy of automatic number plate recognition (ANPR) systems.

A smart city can be well defined when all the assets are managed by latest technological solutions in such a manner that it includes every major and minor resource, like schools, govt., hospitals and most importantly its citizens. Smart city uses smart system to reduce human effort and to achieve accurate and efficiency in every factor. The goal of making any city smart is to provide better lifestyle to its citizens by offering them efficiency of services and enhanced performance. Taking all the factors in consideration systems use sensors, artificial intelligence and real time information. The major problem of today’s scenario is the manual data maintenance of information. Taking security into consideration till now the recording and maintenance of the vehicles at various places like at any organization, residential societies, industry etc. The people have to stop and write their vehicle details and for this purpose human effort is required, also the paper work is increased. To make all this tasks automatic these smart systems are introduced. One of the systems which would help the city in providing security factor is “Automatic Number Plate Recognition (ANPR)” which can have various application areas and is helpful in providing reduced human error and accuracy in maintaining the records of vehicles number plate. Using this system there is no need of hiring any human effort to maintain it and also it reduces the paper load. Making the automatic systems provide ease in achieving durability and stability. There are various applications imposing similar techniques

in different fields. One of the applications of this system is at National Borders where it aims at recognizing the vehicles number plate using Optical Character Recognition (OCR) by capturing the picture of number plate and processing it, which then store the number plate details into text format. Another area where it can be used is for Black list Vehicles, helps to recognize the vehicles through their number plates, which is then compared with the database if the data matches with database containing the information of Black listed vehicles, then an alarm is rang to notify to the Govt. authority. Parking Management is another asset for same type of systems that is used for parking areas for taking record of parked vehicles and also helps to generate receipt for respective vehicles. There are various countries which are using one of the applications of such system one of them is for tracking the traffic rules violated vehicles. It is developed to avoid the traffic signal violations. Any vehicle if violates any traffic signal the camera placed captures the image of its number plate and sends the detail to respective authority. This research paper details about one of the applications of ANPR for one of the aspect of security. Security is the major concern these days at every level. This project includes a camera which would be placed at the entrance gate as soon as any vehicles enters the organization the camera will capture the image of its number plate and sends it for Image Processing. There the data extraction from the image will take place and stores that data into the text format which further would get stored into the database. Database is consist of the registered vehicles if the extracted data gets matched with that of stored data it will notify the safe alarm else a warning will be generated. The timings of entering and exit of vehicles is recorded and also the count is maintained.

# Problem Definition:

The main focus in this research project is to experiment deeply with, and find alternative solutions to the image segmentation and character recognition problems within the License Plate Recognition framework. After that compare with database. Three main stages are identified in such applications.

First, it is necessary to locate and extract the license plate region from a larger scene image. Second, having a license plate region to work with, the alphanumeric characters in the plate

need to be extracted from the background. Third, deliver them to an OCR system for recognition. In order to identify a vehicle by reading its license plate successfully, it is obviously necessary to locate the plate in the scene image provided by some acquisition system (e.g. video or still camera).

Locating the region of interest helps in dramatically reducing both the computational expense and algorithm complexity. For example, a currently common 1024x768 resolution image contains a total of 786,432 pixels, while the region of interest (in this case a license plate) may account for only 10% of the image area. Also, the input to the following segmentation and recognition stages is simplified, resulting in easier algorithm design and shorter computation times.

#### Existing System

Existing system of ANPR is used to describe specially made CCTV cameras which are capable of dealing with challenges associated with capturing license plates such as car speeds, fog, night vision and reflective license plate materials.

Previous system combines Sobel edge detection operator and soft-threshold wavelet de-noising to do edge detection on images which include White Gaussian noises. In recent years, a lot of edge detection methods are proposed. The commonly used methods which combine mean de-noising and Sobel operator or median filtering. This method is mainly used on the images which includes White Gaussian noises.

Now a days user are getting information by entering the number plate. It is time taking process to get the proper result.

##### Major issues:

There are a number of possible issues that the software must be able to cope with. These include:

* + - Poor file resolution, usually because the plate is too far away but sometimes resulting from the use of a low-quality camera
    - Blurry images, particularly motion blur
    - Poor lighting and low contrast due to overexposure, reflection or shadows
    - An object obscuring (part of) the plate, quite often a tow bar, or dirt on the plate
    - Read license plates that are different at the front and the back because of towed trailers, campers, etc.
    - Vehicle lane change in the camera's angle of view during license plate reading
    - A different font, popular for vanity plates (some countries do not allow such plates, eliminating the problem)
    - Circumvention techniques
    - Lack of coordination between countries or states. Two cars from different countries or states can have the same number but different design of the plate.

While some of these problems can be corrected within the software, it is primarily left to the hardware side of the system to work out solutions to these difficulties. Increasing the height of the camera may avoid problems with objects (such as other vehicles) obscuring the plate but introduces and increases other problems, such as adjusting for the increased skew of the plate.

On some cars, tow bars may obscure one or two characters of the license plate. Bikes on bike racks can also obscure the number plate, though in some countries and jurisdictions, such as Victoria, Australia, "bike plates" are supposed to be fitted. Some small-scale systems allow for some errors in the license plate. When used for giving specific vehicles access to a barricaded area, the decision may be made to have an acceptable error rate of one character. This is because the likelihood of an unauthorized car having such a similar license plate is seen as quite small. However, this level of inaccuracy would not be acceptable in most applications of an ANPR system.

#### Proposed System

Automatic Number Plate Recognition (ANPR) is an image –processing technology. which is used to identify vehicles by their number plates using Optical Character Recognition.

In this section The process of automatic number plate recognition consists of four main stages:

(1) Preprocessing

(2) License plate localization

(3) Character segmentation

(4) Character recognition

After recognizing the number plate. It compares the vehicle number with existing data in database. This technology is used in various security and traffic application to find the thief and criminals.

**2 SYSTEM REQUIREMENTS**

#### Hardware Requirements:

* + - Core i5/i7 processor
    - At least 8 GB RAM
    - At least 60 GB of Usable Hard Disk Space

#### Software Requirements:

* + - Deep learning
    - Windows 10 Operating System.
    - Jupyter notebook

## SYSTEM DESIGN

#### Architecture Design

A system architecture diagram would be used to show the relationship between different components. Usually, they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them.

Input is vehicle image. First it detects the edges by using canny edge detection algorithm. IT localize the location in the image and extraction characters to interpret the pixels into numerically readable characters.

a processor and software application capable of performing optical character recognition (OCR) that transforms the image of the number plate into alphanumeric characters; an application software that compares the transformed numberplate characters from a database of number plates of interest to law enforcers; a graphical user interface to display the results of the OCR process, and an alert functionality to notify users when a number plate is matched. Finally compare with the database and get the vehicle details.

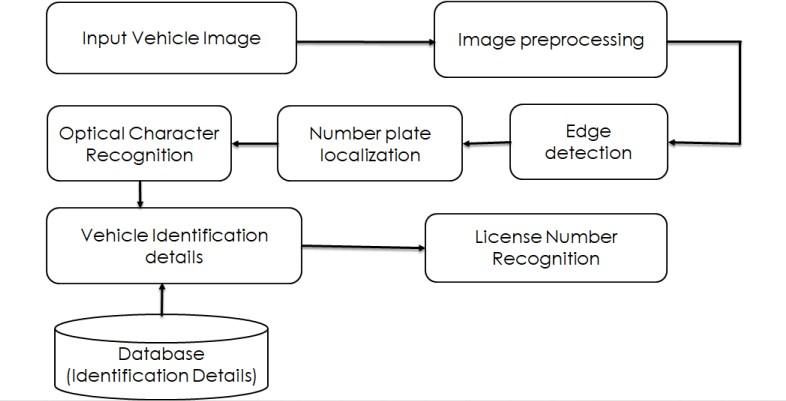


Fig 3.1 Architecture diagram

#### Use Case Diagram:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

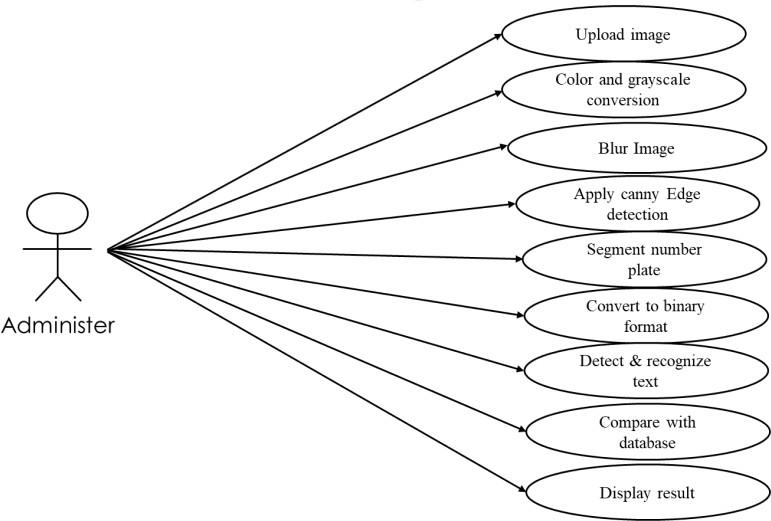


Fig 3.2 Use case diagram

#### Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time. first user gives the image as input. Preprocess the image detects the edges using canny edge detection, then segment the localized numberplate.

After that recognize the vehicle number by using Optical character recognizing (OCR)algorithm. Finally compare with database and gives the vehicle details.

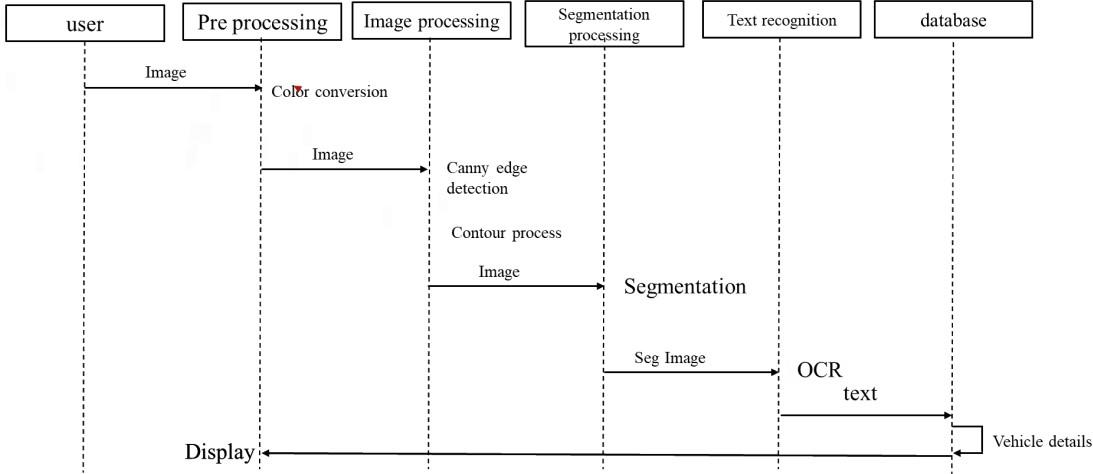


Fig 3.3 Sequence diagram

## IMPLEMENTATION

**Libraries:**

##### NumPy:

It is the most useful library if you are dealing with numbers in Python. NumPy guarantees great execution speed compared to standard Python libraries. It comes with a great number of built- in functions. NumPy is not part of the basic Python installation. We need to install it after the installation of Python in our system.

1. pip install NumPy
2. If you have installed Anaconda on your system, just run the following command to install NumPy.
   1. $ conda install NumPy importing NumPy import NumPy as n

**Matplotlib:**

Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive

environments across platforms (I Python shells, Jupyter notebook, web application servers, etc.).

##### Installation of Matplotlib

If you have Python and PIP already installed on a system, then installation of Matplotlib is very easy.

Install it using this command:

C:\Users\*Your Name*>pip install matplotlib

If this command fails, then use a python distribution that already has Matplotlib installed, like Anaconda, Spyder etc.

##### Import Matplotlib

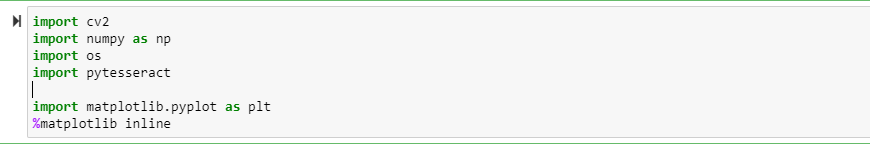
Once Matplotlib is installed, import it in your applications by adding the import module statement:

import matplotlib

Pytesseract:



**Importing libraries into the:**



### Preprocessing:

As mentioned before, the system of automatic number plate recognition faces many challenges. So, this step is essential to enhance the input image and making it more suitable for the next processing steps. The first step done in the preprocessing is to apply minimum filter to the image in order to enhance the dark values in the image by increasing their area. This is mainly done to make the characters and the plate edges bold, and to remove the effect of the light diagonal strips that appear in the characters and edges of the license plates. This process is followed by increasing saturation of the image to increase the separation between colors. Then the image is converted to grayscale (taking the luminance component of NTSC) . Then increasing the image contrast to separate the background from highlights.



### License plate localization:

In this stage, the location of the license plate is identified and the output of this stage will be a sub-image that contains only the license plate. This is done in two main steps.

First, canny edge detection is applied to the image. Then a threshold of 36 (This value is determined using trial and error) is applied, such that every edge with magnitude less than

36 is considered false edge and is set to 0. Then a vertical projection (projecting on the Y-axis) of the edge detected image is taken and smoothed using an average filter. It’s obvious that the characters of the plate along with the plate’s vertical edges will have very strong vertical edges. Moreover, these edges will sum up horizontally in the vertical projection and a strong peak will appear in the rows of the plate (This row will be called band). So, the approach is to take some number of peaks in the vertical projection and processing each of them individually in the next steps and when a successful band is found, the processing of the following bands is canceled. The reason behind taking more than one peak is that the image may contain objects (logos, road advertisement, etc..) that produce many vertical edges also these” false” edges may be centered in the same area so they will form a peak that may be stronger than the peak of the plate itself.

For each band, we take a sub-image referenced by this band and all subsequent processing will be applied on this sub-image. Now the problem is to cut the band image from the left and right to get a bounding rectangle over the license plate (Again, this rectangle doesn’t have to be tight on the plate). For this sake, a vertical Sobel edge detection is applied again, but the height is larger than the width of the filter, this is to decrease the effect of false edges and noise.



Fig 4.1.1 Edge detection

Color-based approaches are based on the principal that different countries have different colors on their number plates. Color level processing obtains the number plate by locating the specific colors on the number plate (Kawde, 2014).

Color-based information of license plates also plays an important role in license plates detection, where the unique color or color combination between the license plates and vehicle bodies are considered as the key feature to locate the license plates. This technique uses color features for license plate detection. However, this method is sensitive to the license plate color and brightness and needs much processing time (Kim, Jung & Hyung, 2003).



Fig 4.1.2 Number plate localization

### Number plate segmentation:

Locating a large bounding rectangle over the license plate. In this step a rectangle that contains the license plate is located (this rectangle may also has some extra parts from the four sides), and this rectangle is the input to the next step for further processing (removing the extra parts, character segmentation then recognition).

Determining the exact location of the license plate. Using the sub-image from the last step which contains the license plate with some extra parts (if any), the following processing is applied to this sub-image. The license plate may be skewed because of the angle of the camera while image acquisition process.



Fig 4.1.3 Number plate segmentation

### Character recognition:

This stage is meant for segmentation of the characters from the plate. The output of this stage is a set of monochrome images for each candidate character in plate. The first step in this stage is to convert the plate image to a binary image. This is done using adaptive threshold with a window of size 11 (This is selected using trial and error). Then a process of noise removal is applied. This is done by getting the connected components from the binary image based on the 8-neighbourhood using flood fill. For every component, we decide if it’s a noise or not based on the aspect ratio of the component and based on the number of pixels in that component. This is based on the fact that the characters of the plate have a certain range of aspect ratio and a certain range of number of pixels. After removing the noise components, a maximum filter is applied to make the effect of thinning the characters to

make sure that no two components are merged. This is followed by a horizontal projection, to detect the boundaries between the characters to be able to cut them individually. The peaks in this projection correspond to the gaps between the characters. So, we get all of these peaks and a rejection process is applied also, since a true plate has a fixed range of gaps between characters. So, any plate that has number of peaks that do not fit in that range, will be rejected. Also, there is a powerful rejection measure; it is the variance of the characters width (the variance of the spaces between peaks). After this the characters are cut according to the peaks of the previous projection. Then another set of measures are computed to reject the false characters that may still exist after the noise removal operation. These measures are aspect ratio, deviation from average height test, deviation from average contrast, deviation from average brightness, deviation from hue, deviation from average saturation. After rejecting false characters, if the number of characters is not located in a predefined range, then the plate is rejected. Otherwise, the processing is continued

The goal of this stage is to recognize and classify the binary images that contain characters received from the previous one. After this stage every character must have a label and an error factor, and this error factor if greater than a predefined value will be used to reject false characters accidently passed from the previous steps. For the sake of classification, some features must be collected from the characters. The feature we work with in this system is the chain code of the contour of the image after dividing it into four tracks then into four sectors.



Fig 4.1.4 Character recognition

Finally check the output number plate with already register license numbers in database. If it matched then it prints the number plate is matched and display the vehicle details. If output is not matched then it prints the number plate is not matched.



Fig 4.1.5 Database connection

##### Database:

Already registered vehicle numbers and details in database.

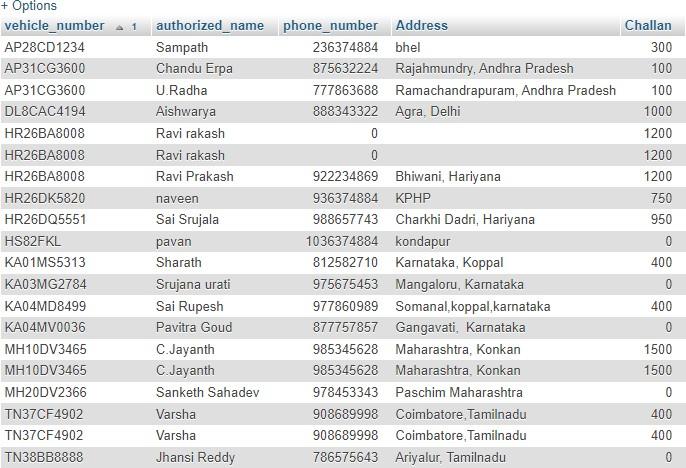


Fig 4.1.6 Database

## Proposed Algorithm:

**Canny edge detector**:

The **Canny edge detector** is an edge detection operator that uses a multi- stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a *computational theory of edge detection* explaining why the technique works.

The Canny filter is a multi-stage edge detector. It uses a filter based on the derivative of a Gaussian in order to compute the intensity of the gradients. The Gaussian reduces the effect of noise present in the image. Then, potential edges are thinned down to 1-pixel curves by removing non-maximum pixels of the gradient magnitude. Finally, edge pixels are kept or removed using hysteresis thresholding on the gradient magnitude.

The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

4.

**optical character recognition (OCR):**

optical character recognition, is one of the earliest addressed computer vision tasks, since in some aspects it does not require deep learning. Therefore there were different OCR implementations even before the deep learning boom in 2012, and some even dated back to 1914 (!).

This makes many people think the OCR challenge is **“solved**”, it is no longer challenging. Another belief which comes from similar sources is that OCR does not require **deep learning**, or in other words, using deep learning for OCR is an overkill.

Anyone who practices computer vision, or machine learning in general, knows that there is no such thing as a solved task, and this case is not different. On the contrary, OCR yields very- good results only on very specific use cases, but in general, it is still considered as challenging.

Additionally, it’s true there are good solutions for certain OCR tasks that do not require deep learning. However, to really step forward towards better, more general solutions, deep learning will be mandatory.

## RESULTS:

Output for matched number plate :

Fig 5.1.1 gray scale convertion



Fig 5.1.2 Edge detection and Number plate localization



Fig 5.1.3 Character recognition

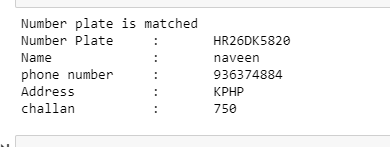


Fig 5.1.4 matched number plate details

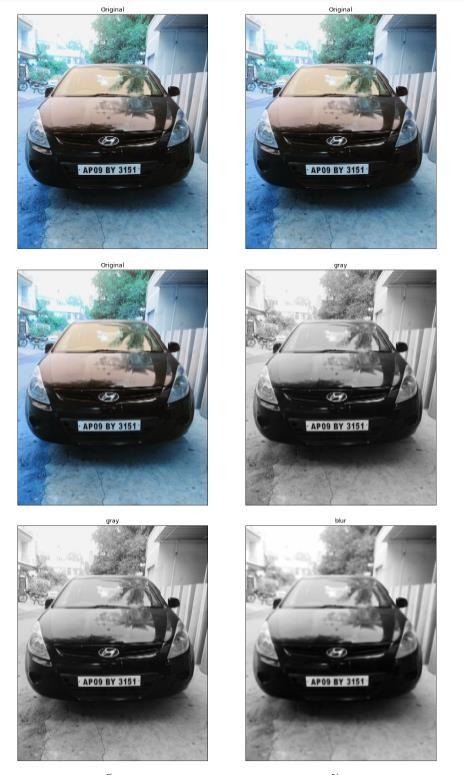
Output for not matched number plate:

Fig 5.1.5 gray scale convertion

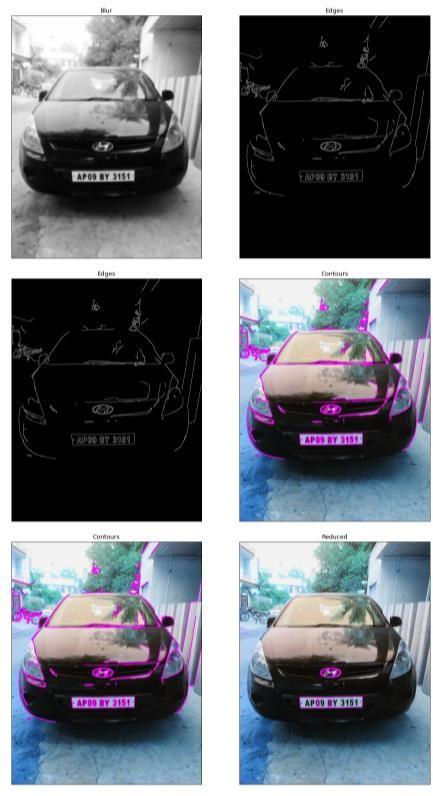


Fig 5.1.6 Edge detection and Number plate localization



Fig 5.1.7 Character recognition

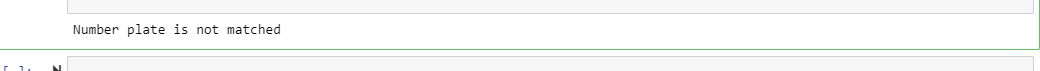


Fig 5.1.8 Unmatched number plate

## Applications:

1. Parking: The plate number is used to automatically enter pre-paid members and calculate parking fee for non-members
2. Access control: A gate automatically opens for authorized members in a secured area, thus replacing or assisting the security guard.
3. Tolling: The car number is used to calculate the travel fee in a toll- road
4. Border control: This installation covers the borders of the entire Country.
5. Traffic control: this finds the any vehicles going against to traffic rules or not following traffic rules.

## Conclusion:

This change will help in the progress of the nation. ANPR plays a very important role in detecting the security threat and effective law enforcement of the country. As India is going through a phase of modernization and infrastructure development, it is very important to implement a reliable and accurate systems. There is an immediate need of such kind of Automatic Number Plate Recognition system in India as there are problems of traffic, stealing cars etc. Government should take some interest in developing this system as this system is very economical and eco-friendly, if applied effectively.

## Future Enhancements:

1. To build an ANPR system to process live videos.
2. To build global ANPR solution & detect License plate with different languages.
3. To reduce processing time by reducing the number of images processing.
4. To build image-based recognition.
5. To detect non-standard Indian license plate.

## REFERENCES

1. Automatic Number Plate Recognition (2018) Abhishek Kashyap, B. Suresh, Anukul Patil, Saksham Sharma, Ankit Jaiswal. Electronics and Communication Engineering Department, Jaypee Institute of Information Technology, Noida-201304, India
2. Automatic Vehicle Number Plate Recognition Using Structured Elements Riazul Islam, Kazie Fatima Sharif and Satyen Biswas December 2015, Bandar Sunway, Malaysia
3. Automatic Number Plate Recognition System (ANPR): Chirag Patel, Dipti Shah, PhD. Atul Patel, PhD A Survey International Journal of Computer Applications (0975 – 8887) Volume 69– No.9, May 2013
4. **https://github.com/SyedUmaidAhmed/Vehicle-Plate-Recognition**
5. **https://www.pyimagesearch.com/2020/09/21/opencv-automatic-license-number- plate-recognition-anpr-with-python/**
6. **https://www.pyimagesearch.com/2020/09/21/opencv-automatic-license-number- plate-recognition-anpr-with-python/**