

CHAPTER 1

INTRODUCTION

Chapter 1- Introduction

1.1 Basic Information

This projects about how we detect the number plate of different vehicles and storing them in the database. The ideology of the project had come up with the difficulties faced by the security to record the numbers of various vehicles at the gate way of the campus. Sometimes the user might not be able to record the data due to various inferences such as bad vision, Light factor, bad interpretation, and failure to record the data when there are multiple buses at an instance. This might not be considered as a serious issue but in case of failure of recording the data at gate ways where there is large scrutiny and high security it may lead to some serious security issues. So in order to avoid such circumstances we develop a module that detects the number plate of the vehicles that are passing through the gate. Within the module we shall employ an algorithm that detects the vehicles and captures the images of the number plate. Besides it also makes a record of incoming and outgoing of those particular vehicles with regardless of manual work and automatically stores them in the database. The KNN algorithm which means K-Nearest Neighbor algorithm is employed along with CNN which means Convolution Neural Network are used in this project.

Data associates with vehicles or vehicle numbers are recorded manually where there are chances of entering the wrong information into the records or even manipulation of data can also take place. With the help of this model we can avoids manipulations and also maintain the accurate data of the vehicles and also store this data with the help of database.

In cities, large scale car parking areas with hundreds/thousands of spaces are getting more and more common with the increasing vehicle population. The existing “pay and park” parking spaces are under-used, mismanaged and under-developed leading to an ineffective parking experience for the driver (parking consumer), and an unprofitable and ineffective scheme for the parking space owner. This deficiency in the existing parking infrastructure leads to accidents, thefts, mismanagement, losses, reduced transparency in the parking process, ineffective space management, etc. People now face problems parking their vehicles. The person parking in these parking-spaces has the concerns regarding the security, transparency and reliability of the existing parking infrastructure. The main idea of this project is to create a system that effectively

handles this deficiency in the parking infrastructure by using modern technologies like machine learning, scalable micro services, effective android interfaces, connected devices to increase the level of automation and efficiency.

1.2 Background and Motivation

The existing methods of parking management use human interaction in detecting and logging number plates. The existing methods mainly depend on human interaction to generate the bills, and it is time-consuming. They need to maintain data of all the vehicles by physically entering the information. This is error-prone and less effective than an automated solution. There have been a few solutions proposed and researched and prototyped by individuals and companies that have explored the using ANPR in real-time for parking management.

In addition to that, there have been some solutions being developed and tested in artificial environments for the same. Some drawbacks are lesser transparency in the parking system, and precious time wasted due to the inconvenient and ineffective methods at parking places and more consumption of fuel while idling or driving around the parking places, users have little knowledge about nearby parking spaces, the uncertainty due to the lack of connected infrastructure and a platform, proves to be less profitable to the parking space owners and the parking space consumers. We propose a solution that is a fully automated computer vision-based smart parking system and implement an interface for the driver providing real-time notifications regarding the status of his vehicle.

The solution we propose automates and guides the user in finding appropriate parking spaces using GPS based navigation. It automates the Vehicle entry and exit systems using machine learning techniques like Number plate recognition. It automates the billing, vehicle check-in and checks out processes, tracking vehicles, etc. It provides a connected solution consisting of devices like cameras that detect the vehicle entry and exit and send the data to the cloud. The solution we propose is decoupled, modularized, scalable, which improves the maintainability and future enhancement of the solution based on changing requirements. The automated parking management system is made up of 2 stations. One is at entry, and the other is at the exit at the parking places.

These stations are linked to the main processing center hosted in the cloud, which provides the various parking management functionalities like synchronous updates of the available slots, nearby parking spaces, automated check-in and check-out logs, automated billing, etc. Advantages of the proposed method include increased transparency in the system, automated processes reduce the latency of the parking process, if well implemented, users can find nearby parking spaces increasing the visibility and the profits of the parking space owner, digitalization of the generation of parking bills and automated billing provides a seamless, dynamic and effective parking process, providing systematic and efficient parking space management, additional features like space availability, tracking of vehicles inside parking area, analytics etc. provide extra benefits for the parking management infrastructure.

1.3 Digital Image Processing (DIP)

It refers to process real world images digitally by a computer. It is a broad topic, which includes studies in physics, mathematics, electrical engineering, computer science. It studies the conceptual foundations of the acquisition and deployment of images and in detail the theoretical and algorithmic processing as such. It also aims to improve the appearance of the images and make them more evident in certain details that you want to note. This chapter doesn't intend to provide a detailed explanation of digital image processing, but yes an overview of those concepts and methods more important for the realization of this project.

DIP Processes:

The capture or acquisition is the process through which a digital image is obtained using a capture device like a digital camera, video camera, scanner, satellite, etc... The preprocessing includes techniques such as noise reduction, contrast enhancement, enhancement of certain details, or features of the image. The description is the process that gets convenient features to differentiate one object from another type, such as: shape, size, area, etc... The segmentation is the process which divides an image into objects that are of interest to our study. The recognition identifies the objects, for example, a key, a screw, money, car, etc... The interpretation is the process that associates a meaning to a set of recognized objects (keys, screws, tools, etc...) and tries to emulate cognition.

1.4 Automatic Number Plate Recognition

Due to the mass integration of information technology in all aspects of modern life, there is a demand for information systems for data processing in respect of vehicles. These systems require data to be archived or by a human or by a special team which is able to recognize vehicles by their license plates in real-time environment and reflect the facts of reality in the information system. Therefore, several techniques have been developed recognition and recognition systems are license plates used today in many applications. In most cases, vehicles are identified by their license plate numbers, which are easily readable by humans but not machines. For machines, a registration number plate is just a dark spot that is within a region of an image with a certain intensity and luminosity. Because of this, it is necessary to design a robust mathematical system able to perceive and extract what we want from the captured image.

These functions are implemented or mathematical patterns in what is called "ANPR Systems" (Automatic Numbers Plate Recognition) and mean a transformation between the real environment is perceived and information systems need to store and manage all that information. The design of these systems is one of the areas of research in areas such as Artificial Intelligence, Computer Vision, Pattern Recognition and Neural Networks. Systems of automatic recognition of license plates are sets of hardware and software to process a signal that is converted into a graphical representation such as static images or sequences of them and recognize the characters in the plate.

The basic hardware of these systems is a camera, an image processor, an event logger memory and a storage unit and communication. In our project we have relied on images of cars in which we can see their license plate. The license plate recognition systems have two main points: The quality of license plate recognition software with recognition algorithms used and the quality of imaging technology, including camera and lighting. Elements to consider: maximum recognition accuracy, achieve faster processing speed, handling as many types of plates, manage the broadest range of image qualities and achieve maximum distortion tolerance of input data. Ideally, for extreme conditions and with serious problems of normal visibility, would have special cameras ready for such an activity, such as infrared cameras that are much better to address these goals and achieve better results. This is because the infrared illumination causes reflection of light on the license plate is made of special material which causes a different light in that area of the

image relative to the rest of it, causing it to be easier to detect. There are five main algorithms that the software needed to identify a license plate:

1. Location license plate, responsible for finding and isolating the plate in the image. It should be located and extracted from the image for further processing.
2. After the number plate is located and extracted from the image, it can be transformed into a standard format for brightness and contrast.
3. Orientation and plate size, offset angles that make the plate look "crooked" and adjust the size.
4. Segmentation of individual characters is present in plate.
5. Optical Character Recognition (OCR) for each image we segmented individual character. The output of the recognition of each character is processed as ASCII code associated with the image of the character. By recognizing all successive images of the characters are completely read the license plate.

The basic process takes place in the Optical Character Recognition is to convert the text on an image to a text file that can be edited and used as such by any other program or application that needs it. Assuming a perfect image, namely, an image with only two gray levels, recognition of these characters will perform basically in comparison with patterns or templates that contain all possible characters.

However, the actual images are not perfect; therefore Optical Character Recognition encounters several problems:

- The device that obtains the image can enter gray levels that do not belong to the original image.
- The resolution of these devices can introduce noise into the image, affecting the pixels to be processed.
- Connecting two or more characters in common pixels can also produce errors. The complexity of each of these subdivisions of the program determines the accuracy of the system. ANRP software must be able to face different potential difficulties, including:

- Poor image resolution, often because the plate is too far, or is the result of using a low quality camera.
- Blurred images, including motion blur and very often in mobile units.
- Poor lighting and low contrast due to overexposure, reflection or shadow.
- An object obscuring (part of) the license plate.
- Evasion techniques.
- Blurred images difficult to OCR; ANPR systems should have high shutter speeds to avoid motion blur.

CHAPTER 2

LITERATURE SURVEY

Chapter 2- Literature Survey

2.1 Survey

We undertook a comprehensive literature survey which covered the following papers and articles: K.M. Sajjad[1] discusses about using real-time embedded systems for number plate detection. The author has also emphasized on using open source software and tools like open computer vision library for development of effective ALPR System. The author also briefly states different applications of such a ALPR software ranging from urban traffic control to parking admission.

According to Sergey Zherzdev and Alexey Gruzdev[2], Automatic License Plate Recognition is a challenging and important task which has various applications in the field of mobility solutions like traffic management, digital security surveillance, vehicle recognition, parking management of big cities. This paper tackles the License Plate Recognition problem and introduces the LPRNet , a Deep Convolutional Neural Network algorithm, designed to work without pre-segmentation and the consequent recognition of characters. The authors further discuss techniques like pruning and quantization by which the overall efficiency and processing can be improved.

Yann Lecun, Patrick Haffner, Leon Bottou and Yoshua Bengio[3] gives a detailed description and implementation of Object recognition using convolutional neural networks. It gives a detailed analysis of topics like local receptive fields, shared weights, spatial sub-sampling etc. The paper also briefly discusses LeNet-5 architecture used in character classification. It also gives a comparison between Convolutional Neural Networks and other neural networks like Support Vector machines etc. Their paper also gives a detailed analysis of gradient descent-based learning in complex systems. The paper also talks about Graph Transformer networks and their application in object detection.

Joseph Redmon and Ali Farhadi[4] focus on detailed analysis on the working of the yolov3 architecture and its various components used in object detection in their paper. It also compares the performance of yolov3 architecture with its predecessors, R-CNN, RetinaNet, Resnet and other architectures. It gives a brief about the bounding box predictions and its implementation. It

also explains about feature extraction and Prediction across scales. Also, the advantages and disadvantages of Yolo architecture over other architectures is mentioned.

2.2 Related work

The methods discussed in preceding sections are common methods for plate detection. Apart from these methods, various literature discussed method for plate detection. As most of the methods discussed in these literatures use more than one approach, it is not possible to do category wise discussion. Different number plate segmentation algorithms are discussed below. In [5], for faster detection of region of interest (ROI) a technique called sliding concentric window (SCW) is developed. It is a two step method contains two concentric windows moving from upper left corner of the image. Then statistical measurements in both windows were calculated based on the segmentation rule which says that if the ratio of the mean or median in the two windows exceeds a threshold, which is set by the, then the central pixel of the windows is considered to belong to an ROI. The two windows stop sliding after the whole image is scanned. The threshold value can be decided based on trial and error basis. The connected component analysis is also used to have overall success rate of 96%. The experiment was carried out on Pentium IV at 3.0 GHz with 512-MB RAM and took 111ms of processing time for number plate segmentation.

Another SCW based system is presented in [8] for locating Korean number plate. After applying SCW on vehicle image authors used HSI color model for color verification and then tilt was corrected by using least square fitting with perpendicular offsets (LSFPO). The distance between camera and vehicle varies from 3 to 7 meters. A cascade framework was used in [33] for developing fast algorithm for real time vehicle number plate detection. In this framework a compact frame detection module is used to segment number plate. This module contains three steps: First - Generation of Plate Region Candidates which is used to reject non plate regions by using gradient features. Second – Extraction of complex plate regions which contains three steps to identify plate region and reject non plate regions. Third – plate verification is used to make sure that no non plate regions are extracted in preceding steps. The experiment was carried out on 3-GHz Intel Pentium 4 personal computer.

To detect multi-style number plate a configurable method is proposed in [17]. For detecting different style of number plates, a user can configure the algorithm by changing parameter value in the number plate detection algorithm. The authors define four parameters mainly:

- Plate rotation angle- to rotate number at certain angle plate if it is skewed which is shown in fig. 2(a).
- Character line number – to determine whether characters are spanned in more than one line or column as shown in fig. 2(b). The algorithm works for maximum three lines.
- Recognition models – to determine whether number plate contains alphabets only, alphabets and digits or alphabet, digits and symbols.
- Character formats – To classify the number plate characters based on their type. For example, Symbols can be represented as S, Alphabets can be represented as A and digits can be represented as D. So the number he algorithm was executed on Pentium IV 3.0GHz. To locate Indian number plate, a feature based number plate localization is proposed in [34]. The authors use Otsu's method to convert gray scale images into binary images. It is a seven-step procedure to extract number plate without any background image from vehicle image.

In [15] a feature salient method is used to extract vehicle number plate by using salient features like shape, texture and color. The authors used Hough transform (HT) to detect vertical and horizontal lines from rectangular vehicle number plate and then processed it by converting red, green, blue (RGB) to hue-intensity-saturation(HIS). Finally, the number plate is segmented. This algorithm is executed on Pentium-IV 2.26-GHz PC with 1 GB RAM using MATLAB.

An Improved bernsen algorithm is used in [19] for license plate location. This algorithm is used for the conditions like uneven illumination and particularly for shadow removal. The authors used local Otsu, global Otsu, and differential local threshold binary methods for good accuracy. By using this algorithm, shadow was removed and license plate was successfully detected, which was not possible with the traditional bernsen algorithm. The experiment was carried out on Windows XP Operating system Intel Core 1.8 GHz central processing unit and 1.5 GB RAM. The algorithm was developed using Visual C++.

To locate Chinese number plate Hui Wu and Bing Li [35] proposed a method to find horizontal and vertical difference to find exact rectangle with vehicle number. The Authors converted vehicle image into gray scale and then applied automatic binarization using MATLAB. Any further detail regarding number plate detection algorithm is not mentioned in this paper. The authors claim to have average recognition rate of 0.8s. To extract license plate characters in Indian condition.

Ch.Jaya Lakshmi et al. [16] proposed a novel approach which is based on texture characteristics and wavelets [36]. The authors also used morphological operation [37] for better performance in complicated background. Sobel mask is used to detect vertical edges. The system was implemented using MATLAB. A Sobel edge detector operator is also used in [38].

To detect license plate from CCTV footage, M.S.Sarfraz et al.[39] proposed a novel approach for efficient localization of license plates in video sequence and the use of a revised version of an existing technique for tracking and recognition. The authors proposed a novel solution for adjusting varying camera distance and diverse lighting conditions. License plate detection is a four step procedure including finding contours and connected components, selection of rectangle region based on size and aspect ratio, initial learning for adaptive camera distance/height, localization based on histogram, gradient processing, and nearest mean classifier. After processing these steps final detection result is forwarded for tracking.

In [6], canny edge detector operator was applied to find out the transition points. As per H.Erdinc Kocer et al a license plate contains white background and black character normally. The Canny edge detector uses a filter, which is then based on Gaussian smoothing's first derivative to eliminate the noise. Then in the next step, the edge strength is calculated by considering the gradient of the image. The canny edge detector operator used 3 X 3 matrix to accomplish this task. Based on this information transition points region is determined. The edge map is used to find transition points between black and white colors. The further technical details of this algorithm are not mentioned. The vehicle images were captures from CCD camera.

For detecting number plates of different countries Ankush Roy et al. [7] presented improved segmentation algorithm. The number plate segmentation algorithm is a four step procedure

including median filtering, adaptive thresholding, component labeling and region growing and segmentation and normalization to remove noise, for binarization of image, to label the pixel according to color value and to segment the plate of 15 X 15 pixel size. The authors used Otsu's method for image binarization in the adaptive thresholding process. The overall success rate of system is mentioned but success rate of number plate detection rate is not mentioned in this paper.

In [14], global edge features and local Haar-like features are proposed for real-time traffic video. License plate detection is accomplished by moving a scanning window around the vehicle image. The scanning windows is categorized a license plate region and non license plate region based on the pre-defined classifier. In the training phase, six cascade classifier layers are constructed for future processing. In the testing phase, local Haar-like features and global features are extracted. Haar-like features are the digital image features generally used for object segmentation. These features are generally collection of functions to find number of rectangles covering adjacent image regions. Global features include edge density and edge density variable. These features are calculated by using fix size of sample image i.e. 48 X 16 which is scaled in training phase. The experiment was carried out on a PC with Pentium 2.8 GHz CPU. The average processing time for segmentation was 0.204s. Another edge based number plate segmentation algorithm is presented in [38].

A weighted statistical method is applied in [18]. Before processing further, authors converted 24 bit color image into gray scale image. In the weighted statistical method, a 2D image matrix of N rows and M columns is prepared. Then weighting operation is applied to the modified image matrix prepared after adding weights. As per Zhigang Zhanga et al. standard license plate length and breadth proportion is 3.14:1. More implementation details regarding this method are not mentioned in this paper. Similar method is proposed in [40].

An inductive learning RULE-3 based system is proposed in [20]. RULE-3 is a simple algorithm containing several steps for extracting objects having certain attributes. Recognition of number plate contains four steps such as finding edges which are contained in letter, Recognizing the letter using an extracted set of rules, Applying previous steps to all characters contained in the number plate being processed and Recognize number plate by bringing all characters used together.

In [23], fuzzy-based algorithm is applied. To extract license plate region a four step method is implemented. In the first step noise is eliminated from the input image. Edge detection is used in second step of find rectangle area of candidate region. In the third step, based on size, histogram and other information invalid rectangle areas are discarded. In the last step geometric rectification is used to obtain license plate candidate region. As these steps need some addition processing, authors used fuzzy-based algorithm containing several steps to extract license plate with more accuracy. The system was developed on TI DM642 600 MHz/32 MB RAM with C language under CCStudio V3.1 environment. The overall average processing time was ~418.81ms. approach is presented in [11]. The PNN algorithm works on gray scale image. Bottom-Hat filtering is used to enhance the potential plate regions. To separate the object of interest from background a thresholding is employed for binarization of the gray level image. Because of varying lighting conditions, brightness levels may vary and some adaptation is necessary.

To perform it Otsu's Thresholding technique is used as it is adaptive in nature. Each segment of the binary image is labeled according to color of each segment to enable classification. The plate extraction is done calculating the Column Sum Vector (CSV) and its local minima. The algorithm was executed on Intel® Core™2 Duo Processor CPU P8400 (2.26GHz, 2267 MHz). The plate recognition processing time was 0.1s.

In [24], a robust and reliable SIFT based method is used to describe local feature of a number plate. As per Morteza Zahedi et al., the set of features extracted from the training images must be robust to changes in image scale, noise and illumination in order to perform reliable recognition. The more details regarding license plate recognition are not mentioned in this paper.

In [28], a trichromatic imaging and color-discrete characteristic approach is used to locate license plate. As per Xing Yang et al. number plates from 105 countries are composed of 11 combinations such as : (1) cyan-black; (2)cyan-white; (3) black-red; (4) black-white; (5) blue-white; (6) white-red; (7) white-green; (8) yellow-black; (9) yellow-blue; (10) yellow-green; and (11) yellow-red. Based on this information, authors divided six groups of RGB

components. Then authors derived several trichromatic functions and a binarized image is obtained. In the next step, de-noising and searching are used to locate license plate finally. The system was Intel Pentium 4, 2.4 GHz, and 1-GB memory. The overall average processing time was 57ms.

2.3 Analysis

Researchers have conducted lot of work in character recognition, including character tracing, capturing, classifying. Number plate recognition is a system to identify the number plate of a vehicle based on the details already available at the admin end. Researchers present a few problem statements where concept of character recognition and number plate detection can be used. These are as follows: In e-challan concept of road transport ministry, the cameras on the road can detect high speed cars exceeding the speed limit and their details based on the image captured can be extracted using the number plate. The number plate helps in identifying the vehicle and to produce an e-challan thus reducing the patrolling of police officers. In accident detection system, the OpenCV can be used to read the data of accident prone areas based on machine learning concepts and can notify the authority if any miss happening took place.

The immense lion's share of the in most recent couple of years, ALPR has been one of the helpful methodologies for vehicle observation. It tends to be applied at number of open spots for satisfying a portion of the reasons like traffic wellbeing authorization, programmed cost content assortment, vehicle leave framework and Automatic vehicle leaving framework which is our principle center. ALPR calculations are commonly separated in four stages: Vehicle picture catch, Number plate discovery Character division and, Character acknowledgment. The initial step for example to catch picture of vehicle looks simple however it is very urgent assignment as it is hard to catch picture of moving vehicle continuously in such a way, that none of the segment of vehicle particularly the vehicle number plate ought to be missed.

Directly number plate discovery and acknowledgment preparing time is under 50ms in numerous frameworks. These frameworks follow various ways to deal with find vehicle number plate from vehicle and afterward to separate vehicle number from that picture. A large portion of the number plate limitation calculations combine a few strategies, bringing about long computational

(and in like manner significant execution) time (this might be decreased by applying less and easier calculations).

The outcomes are profoundly subject to the picture quality, since the unwavering quality of the methods seriously debases on account of intricate, uproarious pictures that contain a ton of subtleties. It can likewise be utilized to distinguish and forestall a wide scope of crimes and for security control of profoundly limited zones like military zones or territory around top government workplaces. The framework is computationally economical contrast with the other ANPR systems.

The precision and accuracy of the framework/setup depends on the technology and computational calculations used. From the other works based on image recognition and shape recognition and from researchers data it can be inferred that the lightweight the calculation is the more fast the outcome will be, and the more accurate the calculations and datasets learning is, the more accurate the output will be.

CHAPTER 3

DESIGN AND METHOD

Chapter 3

Design and Method

3.1 Design

Recognizing Car License Plate is a very important task for a camera surveillance-based security system. We can extract the license plate from an image using some computer vision techniques and then we can use Optical Character Recognition to recognize the license number. Here I will guide you through the whole procedure of this task.

Approach:

- Find all the contours in the image.
- Find the bounding rectangle of every contour.
- Compare and validate the sides ratio and area of every bounding rectangle with an average license plate.
- Apply image segmentation in the image inside validated contour to find characters in it.
- Recognize characters using an OCR.

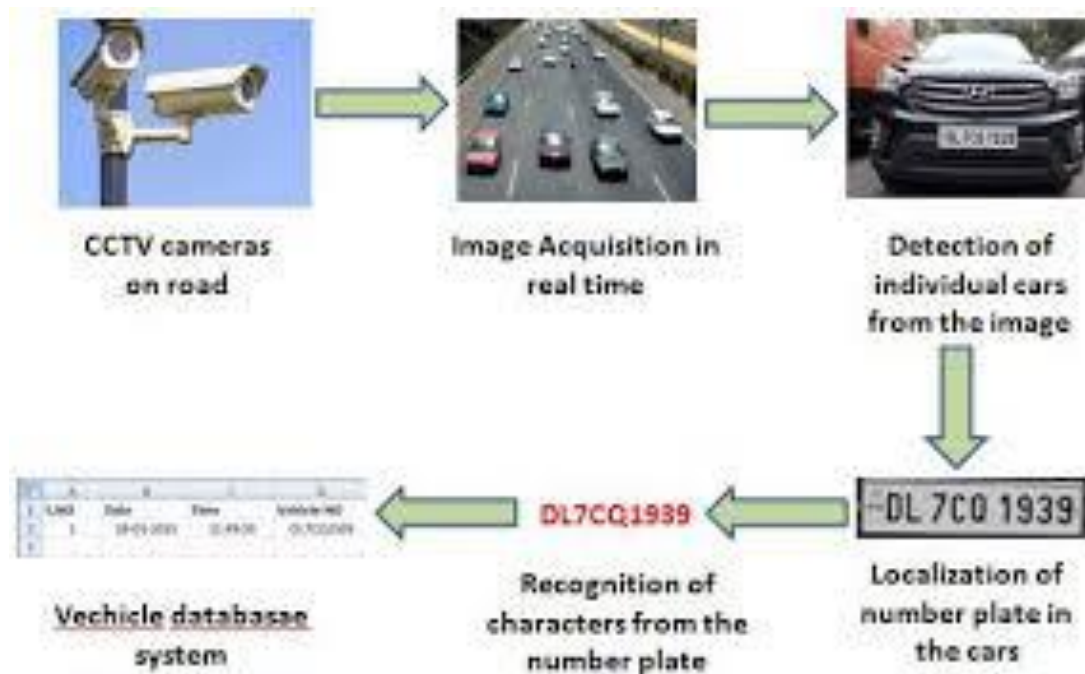


Fig 3.1 System Architecture

Automatic License/Number Plate Recognition systems come in all shapes and sizes:

ANPR performed in controlled lighting conditions with predictable license plate types can use basic image processing techniques.

More advanced ANPR systems utilize dedicated object detectors, such as HOG + Linear SVM, Faster R-CNN, SSDs, and YOLO, to localize license plates in images.

State-of-the-art ANPR software utilizes Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) to aid in better OCR'ing of the text from the license plates themselves.

And even more advanced ANPR systems use specialized neural network architectures to pre-process and clean images before they are OCR'd, thereby improving ANPR accuracy.

Automatic License/Number Plate Recognition is further complicated by the fact that it may need to operate in real time.

For example, suppose an ANPR system is mounted on a toll road. It needs to be able to detect the license plate of each car passing by, OCR the characters on the plate, and then store this information in a database so the owner of the vehicle can be billed for the toll.

Several compounding factors make ANPR incredibly challenging, including finding a dataset you can use to train a custom ANPR model! Large, robust ANPR datasets that are used to train state-of-the-art models are closely guarded and rarely (if ever) released publicly:

These datasets contain sensitive identifying information related to the vehicle, driver, and location.

ANPR datasets are tedious to curate, requiring an incredible investment of time and staff hours to annotate.

ANPR contracts with local and federal governments tend to be highly competitive. Because of that, it's often not the trained model that is valuable, but instead the dataset that a given company has curated.

For that reason, you'll see ANPR companies acquired not for their ANPR system but for the data itself!

In this tutorial we'll be building a basic Automatic License/Number Plate Recognition system. By the end of this guide, you'll have a template/starting point to use when building your own ANPR projects.

3.2 Methodology

In this section, we tend to shall study the methods and ways of implementing the framework. We tend to study the technologies and algorithms used in achieving the objective. In section A we tend to shall provide information regarding the used algorithms and technologies. In section B we can justify our projected system

. • **OpenCV:**OpenCV can also be known as Open Computer Vision. It helps in detecting and capturing of data with the help of a camera.OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

k-NN:In pattern recognition, the k-nearest neighbours algorithm (k-NN) is a non-parametric method used for classification and regression. [1] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether kNN is used for classification or regression. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. K-nearest neighbours (k-NN) algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

3.2 Number Plate Detection

Most of the number plate detection algorithms fall in more than one category based on different techniques. To detect vehicle number plate following factors should be considered:

- (1). Plate size: a plate can be of different size in a vehicle image.
- (2). Plate location: a plate can be located anywhere in the vehicle.
- (3). Plate background: A plate can have different background colors based on vehicle type. For example a government vehicle number plate might have different background than other public vehicles.
- (4). Screw: A plate may have screw and that could be considered as a character.

A number plate can be extracted by using image segmentation method. There are numerous image segmentation methods available in various literatures. In most of the methods image binarization is used. Some authors use Otsu's method for image binarization to convert color image to gray scale image. Some plate segmentation algorithms are based on color segmentation. A study of license plate location based on color segmentation is discussed in [22]. In the following sections common number plate extraction methods are explained, which is followed by detailed discussion of image segmentation techniques adopted in various literature of ANPR or LPR.

3.2.1 Image Binarization

Image binarization is a process to convert an image to black and white. In this method, certain threshold is chosen to classify certain pixels as black and certain pixels as white. But the main problem is how to choose correct threshold value for particular image. Sometimes it becomes very difficult or impossible to select optimal threshold value. Adaptive Thresholding can be used to overcome this problem. A threshold can be selected by user manually or it can be selected by an algorithm automatically which is known as automatic thresholding.

3.2.2 Edge Detection

Edge detection is fundamental method for feature detection or feature extraction. In general case the result of applying edge detection of algorithm is an object boundary with connected curves. It becomes very difficult to apply this method to complex images as it might result with object boundary with not connected curves. Different edge detection algorithm / operators such as Canny, Canny-Derliche, Differential, Sobel, Prewitt and Roberts Cross are used for edge detection.

3.2.3 SEGMENTATION

The input variable is the output from Location Plate, 'ImgPlate', corresponding to the image of the gray-scale plate (only plate), and returns the number of characters 'Objects', and a matrix images size [100 100 N] 'ImgChar', that contains the subpictures of the characters found for subsequent recognition. The procedure can be summarized in five stages: Apply the method of Otsu to work with a binary image. This procedure selects the optimal level for thresholding depending on the intensity levels of each image. Eliminate the inclination of the binary image using features of higher object found in the image.

we note this orientation and we apply rotation with this angle in the full picture. Remove impurities larger and smaller than the measurements of a character through the characteristics of each region and morphological operations until there are only eight objects (maximum characters in a plate). Divide those characters that are together because of previous operations or conditions of the image and original registration. In this loop must be especially careful because dividing the object, the number of them grows, which must be extended 'ImgChar' ($100 * 100 * N$) in the right place to show and recognize the characters in order and not to alter the sequence of registration. Finally, remove impurities that were created by segmenting characters, not to return to the next function an array of erroneous images.

3.2.4 Character Recognition

Recognition is final section. In the section presents the methods that are used to classify stored characters and then recognize the individual characters. The classification is based on the extracted features. Statistical pattern recognition approach for recognition but their technique

found to be inefficient. This approach is based on the probabilistic model and uses statistical pattern recognition approach. In this 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. We used a Convolutional Neural Network to classify the characters from the number into one of the 35 classes. The 35 classes correspond to the alphanumeric characters (A-Z 0-9) with the character 'o' and the number 0 merged into one class. The model was trained on a dataset comprising about 60000 images (about 1800 images per class). The model was evaluated on a test set consisting of about 2000 images.

CHAPTER 4

PROJECT DESCRIPTION

Chapter 4- Project Description

4.1 Proposed Work

The main objective of the system is to recognize the number plate. Here recognition refers to capturing the number plate and identifying the characters and numerical present on the plate. In general, we find number plates in white and yellow colors where characters are printed in black. The whole recognition process follows as per the block diagram shown in Fig. 4.1

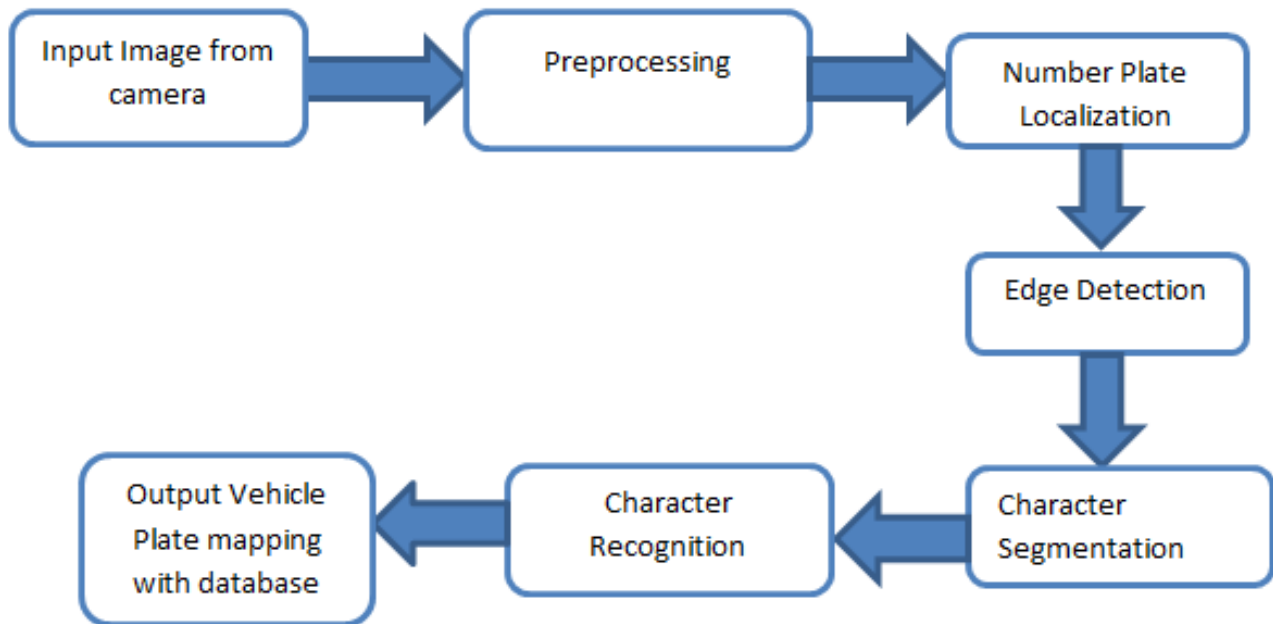


Fig 4.1 Block Diagram

The most challenging task is to detect the number plates of the moving vehicles. if there are multiple vehicles at an instance there are chances where a human eye may not capture the precise data so there is a prospect of entering incorrect data into the log book. In order to avoid such situation we need to develop the algorithms so as to record and store the precise information in to the database.

Here we are about to implement this model with in the campus to recognize the vehicles passing through the gate. In a day there may be approximately hundreds of vehicles that enter and exit through the gate and all this information is logged in the Google sheet or excels sheet.

1. Vehicle image capture

In order to capture the images of the vehicles we must employ high resolution LPR cameras which are designed to recognize them at the gate way and takes an image of the vehicle as shown below.



Fig 4.1.1 Image capture

2. Preprocessing

Here, the captured image will undergo few steps such as contrast enhancement, noise reduction, resizing the image. Initially the captured image is in RGB mode and this has to be converted into a grey scale image.

3. Plate Detection

In order to detect the license plate from the image that is captured we need to get the accurate location of the plate and also the plate may be of different shapes but majorly it is found in rectangular shape .

Here image segmentation plays a key role so as to detect the plate from the image. The segmented image is shown below.



Fig 4.1.2 Plate Detection

5. Character Recognition

An automatic license plate recognition system must recognize alphanumeric characters. The character image is compared with the training set and the best similarity is measured and according to this recognized character is displayed.



Fig 4.1.3 Number Detection

We used a Convolutional Neural Network to classify the characters from the number into one of the 35 classes. The 35 classes correspond to the alphanumeric characters (A-Z 0-9) with the character 'o' and the number 0 merged into one class. The model was trained on a dataset comprising about 60000 images (about 1800 images per class). The model was evaluated on a test set consisting of about 2000 images. The accuracy for this model (i.e. the character classifier) was found to be about 95.66%.

4.2 Working Modules

Finding the closest matching number plate from a set of registered number plates. In order to improve the overall accuracy of the system, we used a string matching algorithm to find the closest number plate id. After the prediction module returns the predicted string, the string is compared with a list of existing (registered) number plate strings. We used the Levenshtein distance as a measure to find the similarity between strings. The Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other.

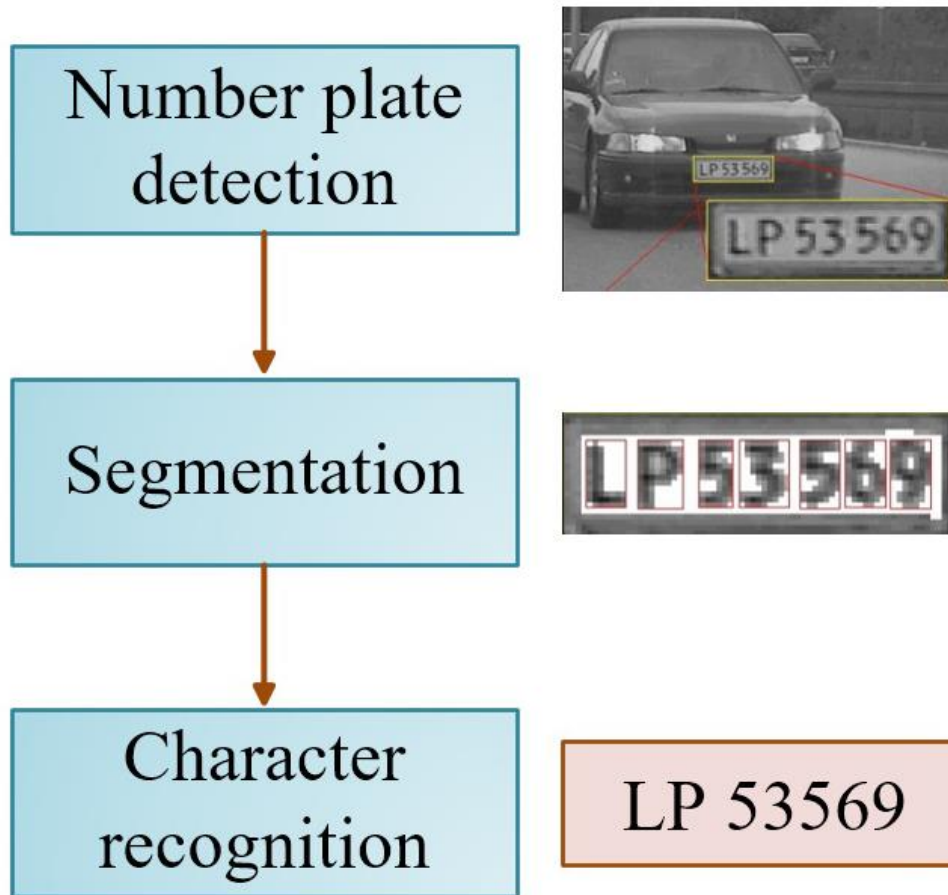


Fig 4.1.4 System Flow

4.3 Data Flow

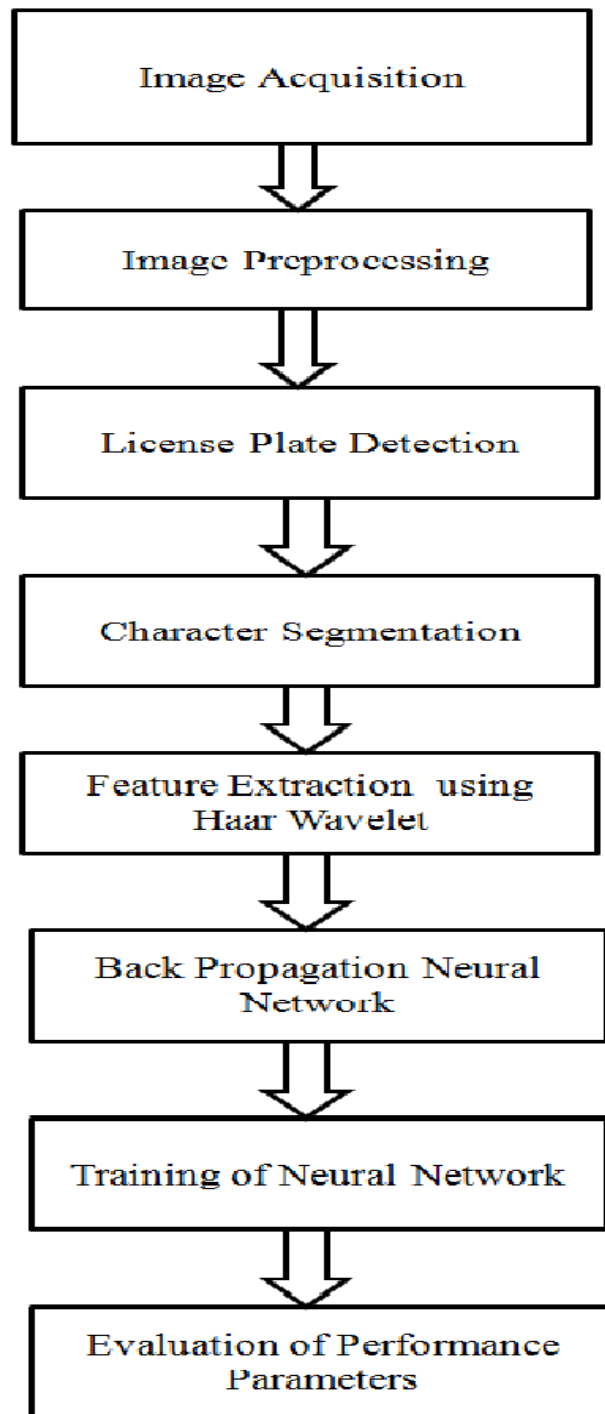


Fig 4.2 Diagram/Flow Charts

CHAPTER 5

PROJECT IMPLEMENTATION

Chapter 5- Project Implementation

5.1 Module Implementation

In Automatic Number Plate Recognition we used two main algorithms KNN, CNN. There are lot of algorithms used in the concept of machine learning. But while coming to our Number plate recognition. So as to detect the characters, alphanumeric and for calculating each frames. we need some mathematical computation. So for mathematical computation and detection of characters we use KNN and CNN Algorithms. Here KNN is K-Nearest Neighbour so in the detection of number plate each and every dimension is stored in the form of frames. KNN and CNN are the different types of neural network consists of 3 layers. That is 1. Input layer, 2. Hidden layer, 3. output layer

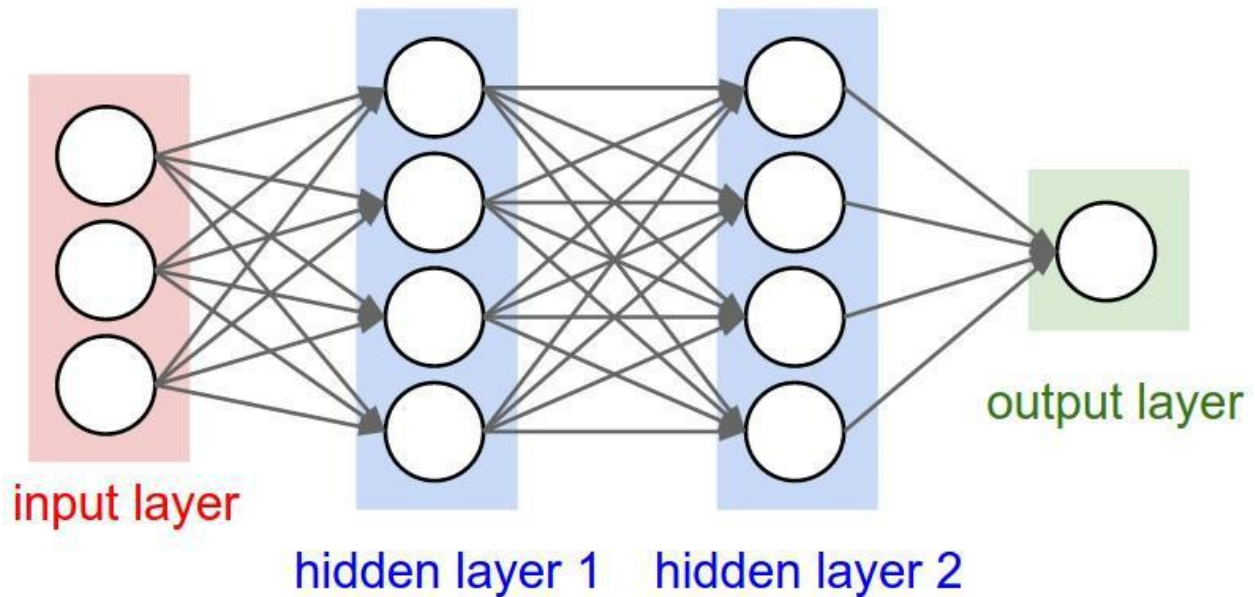


Fig 5.1 CNN layer

KNN is also a type of neural network it is also have comprise of all 3 layers. In these every layers is consisting of nodes and these nodes are interconnected to one by one. And these nodes consists the particular weights. So when an image is detected it forms different interconnected images or pixels or nodes. So when we use KNN it takes the data what we need later it trains and then tests. Takes the data which is near to it, which in term we define it as nearest neighbour. So while taking the data it misses out some part of the data. Then we do not get the accurate results. and for further information simplification in the detection of an image we used an another

algorithm and a type of neural network CNN where it can be like it is the multiplication of data. Here in this data all the nodes are interconnected to each by each layer. So that when we give 'm' layers it finally reduces to 'm-1' layers



Fig 5.1.1 m- Layer

By each node get to know the edge detection, of an image, so that each layer has its own nodes. Suppose if we consider about 28x28 pixels it reduces to (0-9) layers. So each nodes is given its weight. when we want to get an image out we need to give input to the nodes and it neat go to the hidden layers and then to the output layers. Each node weights must be given that it could multiply the layers of every nodes.

The process continues till the we get the perceptive and Nonperceptive goals. So by this we can even convert a blur image in plane image by using Kernal convolution. So that these are converted like Networks and these are called convolution networks. Where every data of an image is stored and hence, the number plate date is divided into frames and then it is used for the display of Automatic number plate Recognition. So it takes every data into consideration and detects the required image.

So for the alphanumeric and for calculating each frames. we need some mathematical computation. So for mathematical computation and detection of characters we use KNN and CNN Algorithms. Here KNN is K-Nearest Neighbour so in the detection of number plate each and every dimension is considered.

In this section, we will talk about the means which were actualized while doing the examination. We will be performing testing on different number plates to get the accuracy of the system.

1. We have collected various number plate data in our database.

2. The first step of scanning includes the capturing of number plate and its data.
3. For refining of number plate and its data we have used the implemented feature of OpenCV.
4. The data of number plate is extracted in multiple steps before it reaches the final result. It have different steps including converting image in grey scale and then converting it to negative and then process of removing of noise is done to get the clear data at the end.
5. After the number plate data is extracted, the resultant data is compared with the data stored in the database.
6. The comparison takes place character-wise.
7. At the last stage, if the extracted data is matched with any instance of database, then it's a success in granting certain permission.
8. The extracted data stored in temporary is removed and a system log file is updated based on the current event.

5.2 Screenshot



Fig 5.2 Final Image

5.3 Applications

In the parking, the recognition of license plates is used to calculate the duration in which the car has been parked. When a vehicle arrives at the entrance to the parking, the registration number is automatically recognized and stored in the database. When the vehicle leaves the parking later and reaches the door, the registration number of the plate is recognized again and compared to the first stored in the database.

The time difference is used to calculate the cost of parking. This technology is used in some companies to grant access only to authorized personnel vehicles. In some countries these systems are installed recognition throughout the city area to detect and monitor traffic.

Each vehicle is registered in a central database and, for example, can be compared to a blacklist of stolen vehicles or congestion control access to the city during peak hours.

They can also be used to:

- Border crossing
- Service stations to keep track of drivers who leave the station without making payment
- A marketing tool to track usage patterns.

CHAPTER 6

TOOLS USED

Chapter 6- Tool Used

6.1 Python

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed. Computer Vision Is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos so the idea is to automate tasks that the human visual systems can do so a computer should be able to recognize that a face of a human being is being detected.



Fig. 6.1 Python

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy highlights code readability, and its syntax allows programmers to express concepts in less lines of code than would be possible in languages such as C++ or Java.

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy highlights code readability, and its syntax allows programmers to express concepts in less lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

6.2 Visual Studio Code

It is an open-source framework used to build the dynamic application. Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring. And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive

debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.



Fig. 6.2 Visual Studio Code

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

6.3 OpenCV

Open CV (Open Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD- licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be employed for the detection and recognition of faces, identification of objects, extraction of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following eye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc. It includes C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

Open CV mainly involves real-time vision applications taking advantage of MMX and SSE instructions when available. A full-featured CUDA and Open CL interfaces are being progressively developed. There are over 500 algorithms and about 10 times functions that form or back those algorithms. Open CV is written inherently in C++ and has a template interface that works harmoniously with STL containers.

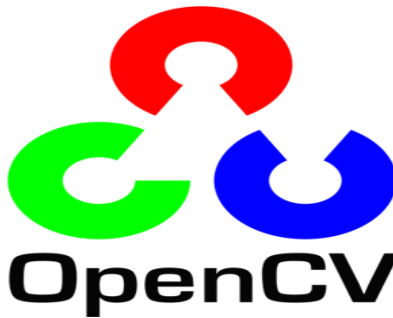


Fig. 6.3 OpenCV

Is a library that is used for computer vision it was first developed in the year 1999 at Intel by Gary Brad sky and the first release came out in 2000. It is supposed to a wide variety of programming languages such as C++ Python Java X vector and also supports different platforms including Windows Linux etc. In OpenCV, all the images are converted into NumPy arrays and make it easier to integrate them with other libraries that use NumPy. Which, also all the images will be defined into a matrix. OpenCV will read it as a NumPy array so basically python stores the images as NumPy array to a matrix of numbers so if it's a colored image it will be a 3d matrix and if it's a grayscale image it will be a 2d matrix.

OpenCV (Open Source Computer vision) is permitted for both scholastic and commercial use. It is a library of programming functions mainly aimed at real-time computer vision.

OpenCV's application has wide areas which includes 2D and 3D feature toolkits, Ego motion estimation, Facial recognition system, Gesture recognition, Motion understanding, Object identification Segmentation and recognition and Motion tracking.

OpenCV is written in C++ and its primary interface is in C++, however it despite retains a less comprehensive though extensive older C interface. OpenCV contains libraries of pre-defined functions supportive in image processing.

CHAPTER 7

RESULT

Chapter 7- Result

The below figure8, shows python model training to detect and recognize license number plate. The below output is achieved by running the python program which trains the model using the training dataset and saves the model by using python libraries. Then a predictor file is run with a test image which is not included in the training dataset and gets recognized and for better visibility of detection the license plate area is marked by a red border.

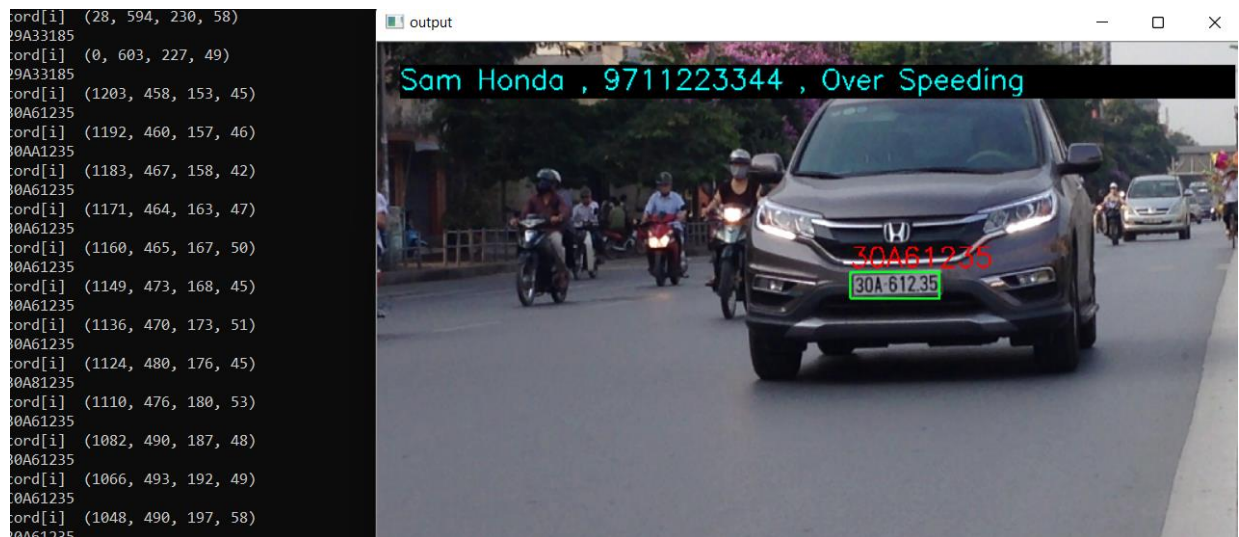


Fig 7.1 Result 1

The Number plate recognition API performs as expected but lacks the accuracy due to the quality of the available datasets used. The Number plate recognition API performs below par in unnatural or custom lighting conditions, unconventional number plate fonts, misplaced orientations. The accuracy of the Number plate recognition API can be vastly improved by improving the quality of the dataset and quantity of samples in the dataset. The micro service can be further scaled up in order to increase the response time and parallelize the processing of the image.

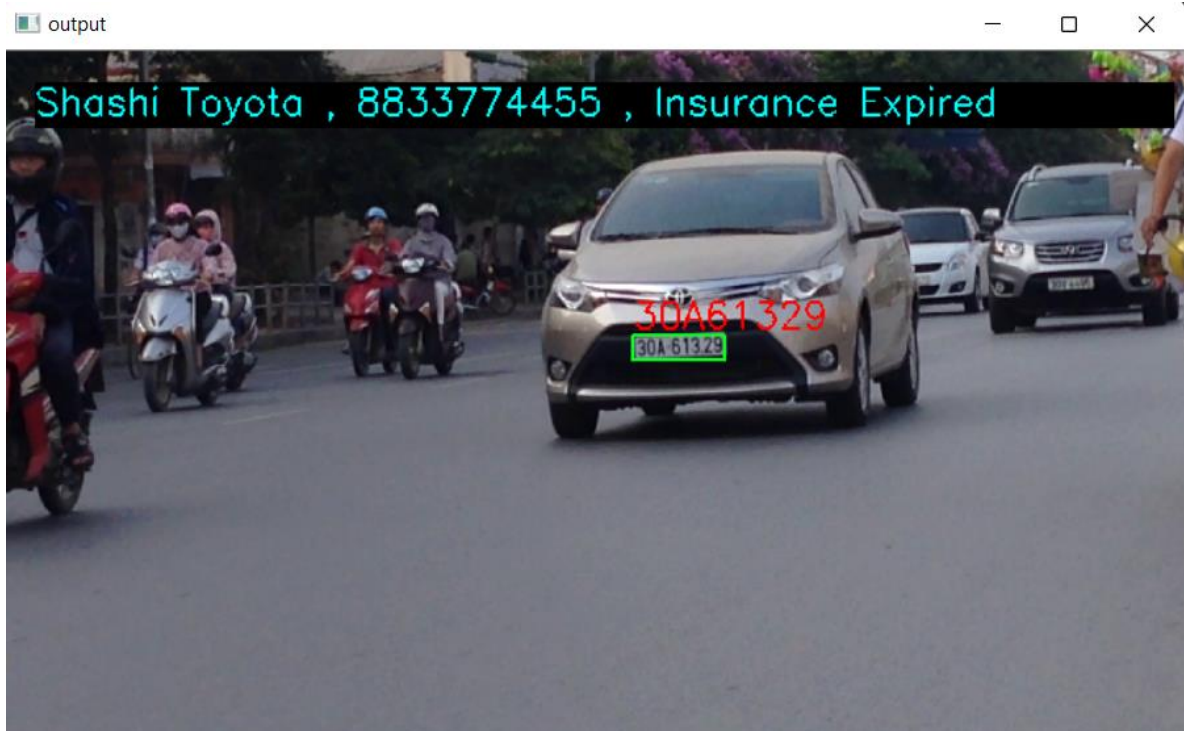


Fig 6.2 Result 2

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Chapter 8- Conclusion and Future Scope

Conclusion:

This program can be used for variety of works where a license number is the primary way of recognizing a vehicle. Current application is completely usable for little uses as parking area or traffic line use. The procedure of vehicle number plate acknowledgment requires high precision when we are taking a shot at an extremely bustling street or stopping which may not be conceivable physically as an individual has a tendency to get exhausted because of dull nature of the employment and they can't monitor the vehicles when there are different vehicles are going in a brief timeframe. A comparable exertion has been made in this work to build up a precise and programmed number plate acknowledgment framework The setup has been tried for vehicles containing distinctive number plates from various states. We get a general effectiveness of 98% for this framework. In spite of the fact that this precision is not satisfactory all in all, but rather still the framework can be utilized for vehicle distinguishing proof.

Future Scope:

Implementing for the detecting of number plate for multiple vehicles in a single video. Live implementation processing of the project with the digital advanced cameras.

In future we can include face acknowledgment and connection the entire framework with criminal database and recognize passing criminal outlaws.

CHAPTER 9

REFERENCES

Chapter 9 -References

- [1] You-Shyang Chen and Ching-Hsue Cheng, "A Delphi-based rough sets fusion model for extracting payment rules of vehicle license tax in the government sector," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2161-2174, 2010.
- [2] Anton Satria Prabuwono and Ariff Idris, "A Study of Car Park Control System Using Optical Character Recognition ," in *International Conference on Computer and Electrical Engineering*, 2008, pp. 866-870.
- [3] A Albiol, L Sanchis, and J.M Mossi, "Detection of Parked Vehicles Using Spatiotemporal Maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1277-1291, 2011.
- [4] Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas, *License Plate Recognition From Still Images and Video Sequences: A Survey*, vol. 9, no. 3, pp. 377-391, 2008.
- [5] Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Vassili Loumos, and Eleftherios Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," pp. 377-392, 2006.
- [6] H. Erdinc Kocer and K. Kursat Cevik, "Artificial neural networks based vehicle license plate recognition," *Procedia Computer Science*, vol. 3, pp. 1033-1037, 2011.
- [7] A Roy and D.P Ghoshal, "Number Plate Recognition for use in different countries using an improved segmenation," in *2nd National Conference on Emerging Trends and Applications in Computer Science(NCETACS)*, 2011, pp. 1-5.
- [8] Kaushik Deb, Ibrahim Kahn, Anik Saha, and Kang-Hyun Jo, "An Efficeint Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network," *Procedia Technology*, vol. 4, pp. 812-819, 2012.
- [9] Lucjan Janowski et al., "Quality assessment for a visual and automatic license plate recognition," *Multimedia Tools and Applications Springer US*, pp. 1-18, 2012.
- [10] Yifan Zhu, Han Huang, Zhenyu Xu, Yiyu He, and Shiqiu Liu, "Chinese-style Plate Recognition Based on Artificaial Neural Network and Statistics," *Procedia Engineering*, vol. 15, pp. 3556-3561, 2011.

- [11] Fikriye Öztürk and Figen Özen, "A New License Plate Recognition System Based on Probabilistic Neural Networks," *Procedia Technology*, vol. 1, pp. 124-128, 2012.
- [12] Jian Liang, D Dementhon, and D Doermann, "Geometric Rectification of Camera-Captured Document Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 3, pp. 591-605, 2008.
- [13] Xin Fan and Guoliang Fan, "Graphical Models for Joint Segmentation and Recognition of License Plate Characters," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 10-13, 2009.
- [14] Lihong Zheng, Xiangjian He, Bijan Samali, and Laurence T. Yang, "An algorithm for accuracy enhancement of license recognition," *Journal of Computer and System Sciences*, , 2012.
- [15] Zhen-Xue Chen, Cheng-Yun Liu, Fa-Liang Chang, and Guo-You Wang, "Automatic License-Plate Location and Recognition Based on Feature Saliance," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3781-3785, 2009.
- [16] Ch.Jaya Lakshmi, Dr.A.Jhansi Rani, Dr.K.Sri Ramakrishna, and M. KantiKiran, "A Novel Approach for Indian License Recognition System," *International Journal of Advanced Engineering Sciences and Technologies*, vol. 6, no. 1, pp. 10-14, 2011.
- [17] Jianbin Jiao, Qixiang Ye, and Qingming Huang, "A configurabe method for multi-style license plate recognition," *Pattern Recognition*, vol. 42, no. 3, pp. 358-369, 2009.
- [18] Zhigang Zhang and Cong Wang, "The Reseach of Vehicle Plate Recogniton Technical Based on BP Neural Network," *AASRI Procedia*, vol. 1, pp. 74-81, 2012.
- [19] Ying Wen et al., "An Algorithm for License Plate recognition Applied to Intelligent Transportation System," *IEEE Transactions of Intelligent Transportation Systems*, pp. 1-16, 2011.
- [20] Mehmet Sabih Aksoy and Ahmet Kürsat Türker Gültekin Çağıl, "Number-plate recognition using inductive learning," *Robotics and Autonomous Systems*, vol. 33, no. 2-3, pp. 149-153, 2000.
- [21] Wenjing Jia, Huaifeng Zhang, and Xiangjian He, "Region-based license plate detection," *Journal of Network and Computer Applications*, vol. 30, no. 4, pp. 1324-1333, November 2007.
- [22] Yang Yang, Xuhui Gao, and Guowei Yang, "Study the Method of Vehicle License Locating Based on Color Segmentation," *Procedia Engineering* , vol. 15, pp. 1324-1329, 2011.

- [23] Feng Wang et al., "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 1007-1020, May 2008.
- [24] Morteza Zahedi and Seyed Mahdi Salehi, "License plate recognition system based on SIFT features," *Procedia Computer Science*, vol. 3, pp. 998-1002, 2011.
- [25] Mei-Sen Pan, Jun-Biao Yan, and Zheng-Hong Xiao, "Vehicle license plate character segmentation ," *Intenational Journal of Automation and Computing*, pp. 425-432, 2008.
- [26] Kaushik Deb, Andrey Vavilin, Jung-Won Kim, and Kang-Hyun Jo, "Vehicle license plate tilt correction based on the straight line fitting method and minimizing variance of coordinates of projection point," *International Journal of Control, Automation and Systems*., pp. 975-984, 2010.
- [27] Francisco Moraes Oliveira-Neto, Lee D. Han, and Myong K Jeong, "Online license plate matching procedures using license-plate recognition machine and new weighted edit distance," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 306-320, April 2012.
- [28] Xing Yang, Xiao-Li Hao, and Gang Zhao, "License plate location based on trichromatic imaging and color-discrete characteristic," *Optik- International Journal for Light and Electron Optics*, vol. 123, no. 16, pp. 1486-1491, August 2012.
- [29] Cynthia Lum, Julie Hibdon, Breanne Cave, Christopher S. Koper, and Linda Merola, "License plate reader(LRP) police patrols in crime hot spots: an experimental evaluation in two adjacent jurisdictionss," *Journal of Experimel Criminology*, Springer Netherlands, , pp. 321-345, 2011.
- [30] K.V. Suresh, G. Mahesh Kumar, and A.N. Rajagopalan, "Superresolution of license plates in real traffic videos," *IEEE Trans. Intell. Transp. Syst*, vol. 8, no. 2, pp. 321-331, 2007.
- [31] Yushuang Tian, Kim-Hui Yap, and Yu He, "Vehicle license plate super-resolution using soft learning prior," *Multimedia Tools and Applications*, Springer US, pp. 519-535, 2012.
- [32] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.
- [33] Shen-Zheng Wang and Hsi-Jian Lee, "A cascade framework for real-time statistical plate recognition system," *IEEE Trans. Inf. Forensics security*, vol. 2, no. 2, pp. 267-282, 2007.

- [34] Prathamesh Kulkarni, Ashish Khatri, Prateek Banga, and Kushal Shah, "Automatic Number Plate Recognition (ANPR)," in RADIOELEKTRONIKA. 19th International Conference, 2009.
- [35] Hui Wu and Bing Li, "License Plate Recognition System," in International Conference on Multimedia Technology (ICMT), 2011, pp. 5425-5427.
- [36] Abdulkar Sengur and Yanhui Guo, "Color texture image segmentation based on neutrosophic set and wavelet transformation ," Computer Vision and Image Understanding, vol. 115, no. 8, pp. 1134-1144, August 2011.
- [37] Jiann-Jone Chen, Chun-Rong Su, W.E.L Grimson, Jun-Lin Liu, and De-Hui Shiue, "Object Segmentation of Database Images by Dual Multiscale Morphological Reconstructions and Retrieval Applications," IEEE Transactions on Image Processing, vol. 21, no. 2, pp. 828-843, 2012.
- [38] Mahmood Ashoori Lalimi, Sedigheh Ghofrani, and Des McLernon, "A vehicle license plate detection method using region and edge based methods," Computers & Electrical Engineering, November 2012.
- [39] M. S. Sarfraz et al., "Real-Time automatic license plate recognition for CCTV forensic applications," Journal of Real-Time Image Processing- Springer Berlin/Heidelberg, 2011.
- [40] Rongbao Chen and Yunfei Luo, "An Improved License Plate Location Method Based On Edge Detection," Physics Procedia, vol. 24, pp. 1350-1356, 2012.
- [41] T Naito, T Tsukada, K Kozuka, and S yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2309-2319, 2000