iSecurity: The AI surveillance, A Smart Tracking System

Nishant Mhatre[1], Parag Ahire[2], Vaibhav Kunkerkar[3], Prashant Itankar[4]

[1,2,3]*Student, Department of Computer Engineering, DMCE, Maharashtra, India.*
[4]*Professor, Department of Computer Engineering, DMCE, Maharashtra, India.*

Abstract - *Crime all over the world is on a rise. Criminals committing crimes such as chain snatching, steeling vehicles can be located as they pass through a signal using CCTV cameras installed on signals. To catch such criminals, the police have to manually go through hours of CCTV footage to get mere glimpses of criminals. Our system, "iSecurity: The AI surveillance" aims to help the police by automating this process of going through hours of CCTV footage and catch criminals with ease. The system will use the CCTV footage to capture license plate numbers of vehicles and store their location in a database. This will avoid waste of time in going through hours of footage by providing location of the vehicles at various checkpoints with CCTVs installed. The system will display recent locations of a specific vehicle on a map which will help the police track vehicles easily and thus make catching criminals a less tedious process. The system is divided into three parts: video processing to obtain license plate numbers, storing the obtained license number plates in a database and displaying recent locations of a vehicle on a map.*

Key Words: Python, OpenALPR, google map API, CNN, Excel, Numpy, pandas.

1. INTRODUCTION

The license plate of a vehicle is a unique identity provided to all vehicles. It can also be used to track vehicles across locations with signals, tolls, etc. which are equipped with CCTV cameras. The license plate number of all the criminal owned vehicles or stolen vehicles captured in the cameras will be stored in the database along with all other vehicles that pass through such locations. The location's longitude and latitude values will be stored in the database. The license plate number is obtained by using image processing, which processes each frame of the footage individually to obtain all license plates present in the current frame. The location where the vehicle is spotted is stored against the license plate number in the database. To obtain the recent locations of a particular vehicle the database is searched for that particular license plate number. A simple GUI is provided to enter the license plate number of the vehicle to be tracked. If the entered number is found in the database, the longitude and latitude of the locations are obtained which in turn are plotted on a map and displayed in the web browser

2. LITERATURE SURVEY

License Plate Detection: The input at this stage is a car's image, and the output is the license plate of a car image. The license plate can be localized anywhere in the image. So, instead of processing pixel by pixel in the image, which leads to increase in the processing time, the license plate can be distinguished by its features such as color and shape of the license plate etc. The license plate's shape, color and other properties varies from country to country. The character schema in India is as follows, first two letters represent the state, and next two digits is corresponding state code for district within particular range followed by two letters and four digits. Thus, each vehicle has unique number represented by license plate characters [1]. Following are the manual methods by which extraction of the license plate can be done:

Step 1: Thresholding

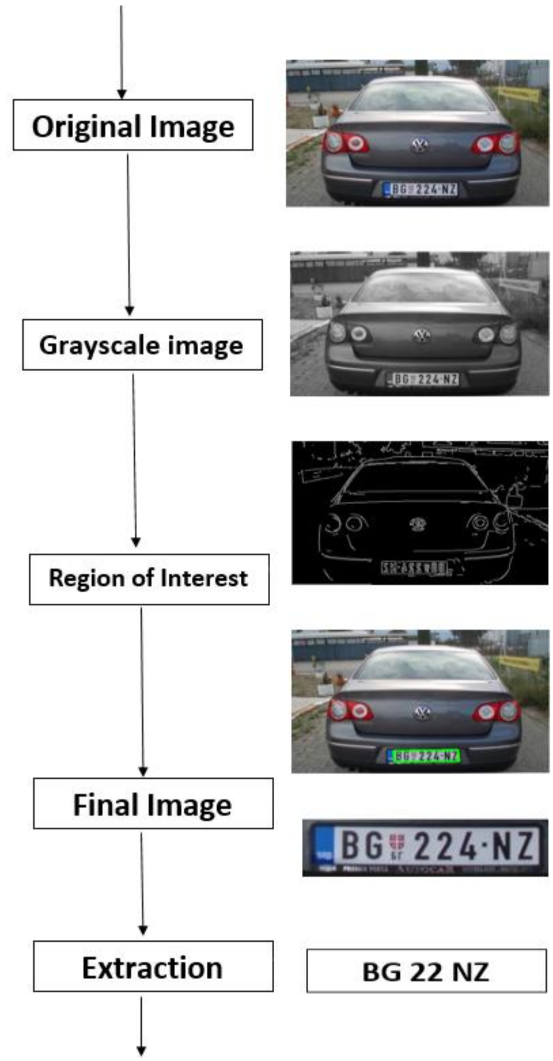
In the first stage the image is converted to grayscale. In a grayscale image, the pixels range between 0 & 255. Then it is converted to a binary image in which a pixel is either completely black or completely white.

Step 2: Filtering Contours

After thresholding the image, searching of contours is done with the size of 150px wide and 80px high (average image dimensions of the license plates). If these contour matches that size range, it is considered as a license plate [5]. If there are multiple matches, then the first contour is chosen.

Step 3: Region of Interest

Normally a license plate has a rectangular shape with a known aspect ratio, and by using various techniques license plate rectangles can be extracted by finding all possible combination of rectangles in the image. To find these rectangles edge detection methods are commonly used but by using contour method the results are more accurate. With the license plate contour, a green rectangle around it is drawn using OpenCV's rectangle function to indicate new ROI (region of interest) for analysis. The image is cropped to just show new ROI so that it is easy to extract characters from image.



Character Extraction: Most of the countries have specific colors for their license plates. Using color combination and unique characters, the numbers can be extracted. The cropped image with license plate is the input in this stage and the output is the text. For Reading/Extracting the characters there are 2 approaches:

1) Supervised Learning approach: This method makes use of a known dataset (called the training dataset) to make predictions. Supervised learning has two categories - classification and regression. Create 34 images of characters A-Z, 1-9 in a font that matched many of the fonts used in license plates. There are several classifiers which can be used. The best way is to use SVC (support vector classifiers) for this task because of its best performance. Using this, images are trained and characters are predicted from images.

2) The second approach is to use Google's Tesseract Optical Character Recognition which is open source library [5]. This library just takes an image and the Tesseract library attempts to return a string with what it thinks is the text.

Using the above approach, the accuracy of the results is around 60%. Because of this, the model was predicting wrong license plate numbers. So, to solve this issue we used ALPR API which performs same steps by using some advance techniques. By using this API, we achieved the accuracy for the India's license plate numbers of about 95-96%.

3. METHODOLOGY

The system is divided mainly in three parts: i) image processing of each frame to obtain license plate numbers. ii) Storing the location where the vehicle was in a central database. iii) Retrieving the recent locations of the vehicle and plotting it on a map. Many machine learning algorithms were needed for the system such as algorithms for object detection and character recognition. The complete system is thus developed using python programming language due to its vastly available libraries built for python, which also include machine learning algorithms which are implemented very efficiently.

3.1 Image processing

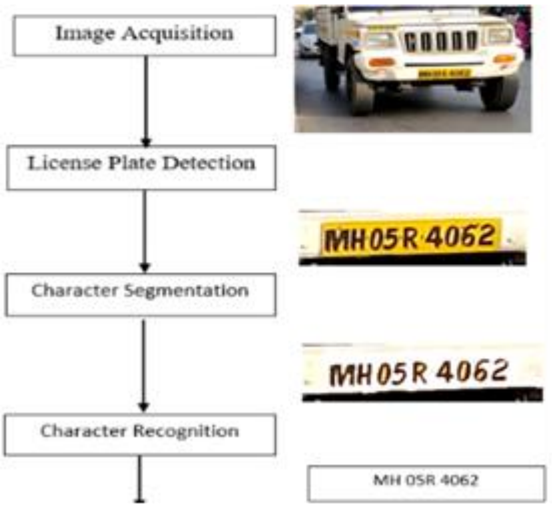
Open Automatic license plate recognition (ALPR) is an open source Automatic License Plate Recognition library written in C++ with bindings in C#, Python, Java, etc. The library analyses the images and videos to identify license plates [2]. The output is in the text format of any license plate characters. It plays an important role in real-life applications such as controlling car parking system, road traffic monitoring. Normally ALPR is used to recognize a vehicle's license plate number, detect car number plate and speed calculation from an image or video. It works on the principle of Local Binary Pattern. It has been trained by using various number of images of the vehicles to locate license plate of different countries such as India, US etc. OpenALPR is trained to recognize vehicle depending on country using the pattern of license plate. Our custom implementation of OpenALPR is referred to as OpenALPR in the remainder of this paper. For detection of a vehicle's license plate and to extract it uses various techniques such as image processing, pattern recognition, Etc. ALPR (Automatic License Plate Recognition) consists of 4 main important stages.

1] Image Acquisition: It is defined as the action of retrieving an image from some source such as camera of mobile, CCTV's, etc. Normally, this is considered as the first stage. In this the image captured and is completely unprocessed. After the image has been obtained, using various methods, processing can be applied to the image to perform many different vision tasks such as extraction, detection, etc.

2] License Plate Detection: This is the most important stage of the system. In this stage, the position of the license plate is detected. The input is the original image and the output is the cropped license plate.

3] Character Segmentation: In this, the characters of the license plate are mapped out and divided into individual images.

4] Character Recognition: The characters earlier segmented are identified here using Tesseract library. Humans can perceive the text on the image as text and can read. But computers won't understand this work. They need something more accurate and organized in a way they can understand.



In order to get more accurate results computerized Optical Character Recognition (OCR) comes into the picture. It recognizes car license plate from a camera, or hand-written documents that can be converted into a digital copy. Python - tesseract is an optical character recognition (OCR) tool for python. It recognizes and reads the text embedded in images. In this the image is first scanned and then converted into a bitmap, which is a matrix of black and white dots. The image is pre-processed where the brightness and contrast are adjusted so as to enhance the accuracy. The image is now divided for identifying the areas of interest such as where the text is located which helps in extraction process. The text is now broken down into lines, words and characters and now the software is able to match the characters through comparison and various detection algorithms. The final result of an image is in the text. These processes take more processing time to generate the output in text format. To get more accurate results, a completely free and open-source library (TESSERACT) can be used which takes less processing. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. The tesseract-ocr by default only supports tiff and bmp format [3].

3.2 Storing Data

The license plate numbers are thus obtained using OpenALPR are stored in a file in Json format. The reason Json is used was, ease of handling and storing data using python. The file is transferred to a server as a central database is maintained at the server. The server maintains a complete database of all the vehicle that have been captured by any of the cameras. To add new location of a particular vehicle or add a new vehicle to the database, the entire database is loaded into a dictionary in a python program and the dictionary is updated using the new data acquired from cameras. The data is then written back to the Json format database.

3.3 Retrieving the locations

The locations of the vehicles which is stored in a central database is retrieved to be displayed on the map. In our project we need to plot geocoordinates on map using python. There are plenty methods are available for this such as, 1. Gmplot library 2. Pygmaps library 3. Creating KML file of coordinates and combining file path with google map link.

Problem with Gmplot library is, there are few deprecated Google Maps API like Google Maps Engine and Google Maps Coordinate. Because of being in development phase, a lot of changes are happening. So, we decide to go with Pygmaps library. Pygmaps-extended is a Python wrapper for the Google Maps JavaScript API V3. It can be used to generate fully-interactive Google Maps from python as stand-alone HTML files.[4] To use pygmaps we just need to download library and put it in the same folder where code present and import it as usual. Pygmap library first make call to google map API and provide map for developers so that we can perform task on that. When user put car number plate to be shown, corresponding code will execute and result from central database is given to pygmap library for plotting purpose. Now library has list of latitude and longitude of car to be shown on map. First library will put marker on map and then plot line passing through each point. So, in this way we can plot path on map to get where car travel throughout a day.

4. CONCLUSION

iSecurity: The AI surveillance is a solution to the tracking problem faced by the police. The workload of the police will be majorly reduced by this system. It will also reduce time taken by the police to track a vehicle and get more time to serve justice to the criminals and concentrate on their work. It is also more visual in the way that it will also display the locations of the vehicles making it easy to know the exact location of the vehicles and also the path that they took to reach that location, which could also provide important clues as to how the criminals operate and how they think.

REFERENCES

[1] B. Sachin Prabhu; Subramaniam Kalambur; Dinkar Sitaram. "Recognition of Indian license plate number from live stream videos" IEEE 2017

[2]https://github.com/openalpr/openalpr/blob/master/README.md

[3] https://pypi.org/project/pytesseract/

[4]https://github.com/thearn/pygmaps extended/blob/master/README.md

[5]https://github.com/laddng/LiPlate/blob/master/README.md

[6]https://pdfs.semanticscholar.org/1a79/798226c044f50f9b7e80294067ac3397a28c.pdf