Vehicle Inspection System using ANPR

Aryman Srivastava
Electronics and
Communication Department
Indraprastha Institute of Information Technology Delhi
New Delhi, India
aryman20184@iiitd.ac.in

Dheeraj
Electronics and
Communication Department
Indraprastha Institute of Information Technology Delhi
New Delhi, India
dheeraj20194@iiitd.ac.in

Abstract- With an increase in the number of motor vehicles on roads especially in the developing countries, there is a huge demand for an effective affordable and efficient vehicle identification mechanism. There are also increased insecurity challenges including terrorism which call for increased surveillance. A security guard is typically involved in the continuing registration procedure for visitors, staff, and students entering academic buildings and parking lots. This process of writing is tedious and time consuming and is prone to inaccurate recordings. Furthermore, due to manual entry of data the process of searching the database for an older set of datasets is highly difficult. We propose to implement an ANPR system to aid in vehicle identification and registration, and to review a previously held data log. The automatic Number Plate Recognition System is an embedded system that acknowledges the vehicle number plate automatically. In this paper, we propose a technique for implementing an Automatic Number Plate Recognition System using Python, Computer Vision Library, and a simple client-server architecture. In this system, firstly, we will capture the image of a number plate, send it to the server via client-server architecture, process it and, then read the license plate of the image using an OCR. Our work can be applied in many fields, such as pay-slip, parking, 24/7 vehicle monitoring, etc.

Keywords—Automatic Number Plate Recognition (ANPR); Optical Character Recognition (OCR); Raspberry Pi; Computer Vision; Client-Server Architecture.

I. INTRODUCTION

The Data Logging of a vehicle can be done quickly by a person; however, with the Increasing number of vehicles in today's world, it is very difficult to keep records of the entire vehicle set manually. It is very tedious and requires a lot of manpower. It is also very time-consuming and cost-ineffective. Furthermore, tracking data in such a vast and old record stored manually on paper is tedious and a near-impossible task. A significant enhancement in today's technology can solve this problem. Creating an ANPR is the best way for this issue. In the last few years, ANPR, or License Plate Recognition (LPR), has grown its demand for vehicle inspection. An ANPR system works in four main steps, 1) Vehicle Image Capture, 2) Number Plate Detection, and 3) Optical Character Recognition. We further enhance this process by introducing data storage on the web server. The data stored on the web server will include the entry time and date of the vehicle, exit time and date of the vehicle, and an image of the picture of the car along with the driver.

Conventionally we started with our first step, transferring the data over the network. For the controller to send this data, a need for a simple client and server architecture arises. Since the client and server are needed to be in connection with each other every time, it is required that both client and server should be working every time. The next step is to detect the number plate from the image sent by the controller. For this step, we have implemented a Machine Learning Model from the TensorFlow library of Python to catch the region of interest. The third step is to provide character segmentation and check each character. The fourth and most crucial step is to recognize the character. The fourth step is implemented using an OCR.

Few research papers were read regarding the information and data for the systems centered around ANPR applications. Numerous methods were used to implement such systems, such as Artificial Neural Networks [1], Probabilistic Neural Networks [2], Optical Character Recognition [3], Sliding Concentric Window Window [4], Back-Propagation Neural Networks [5], and Support Vector Machines [6]. In this paper, the TensorFlow 2 detection model ZOO: SSd mobileNet V2 FPNLite model [7] is implemented along with Pandas Data frame for storing and retrieving the data and a Client-Server Program to transfer data between controller and server.

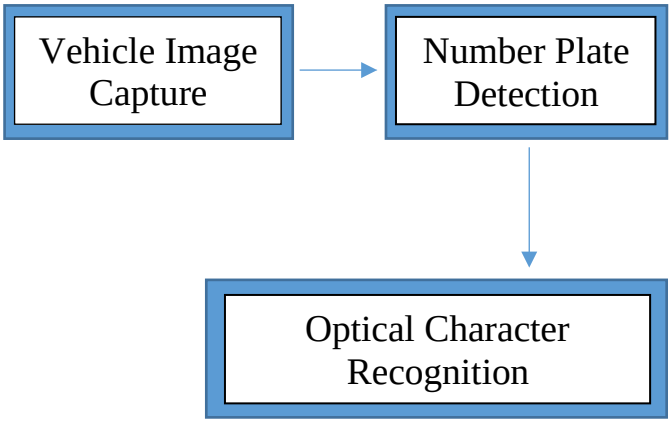

Fig.1. Block Diagram Of the Proposed System

Many works have been formulated on this topic; some of the methodologies that have been implemented are as follows: -

A. Automatic Number Plate Recognition by Abhishek Kashyap, B. Suresh, Nukul Patil, Saksham Sharma, Ankit Jaiswal [8]

In this paper, a template matching technique is used in implementing the ANPR system for the number plate recognition of vehicles. The objective of this paper is to recognize the vehicle's number plate by matching the template. They have sed many image processing techniques such as Binary Image Processing, Adaptive Thresholding, Contrast Extension, Median Filtering, Character Segmentation, Feature Extraction, to filter the region of interest from the rest of the image. The text extraction they

have been performed by using a template OCR. They have gotten an accuracy of 82.6 percent using the template OCR.

In our work we have simply performed OCR filtering to remove the extra texts that were written on license plate, getting us only license plate text. They also used many prototypes to check performance of their method.

B. *Automatic Number Plate Recognition System using Raspberry pi by Naveena Budda, K. Meenakshi, Padmavathi Kora, K. Swaraja, G.V. Subba Reddy [9]*

In this paper, the authors have implemented the ANPR system using Python and Open Computer Vision Library. They have mentioned different stages, which include Load Image from the file, Convert Image into Grayscale, Convert Image into Binary Format, Filter to Detect Edges in Images, Find the contours in the image, Detect the rectangle in the image, Recognize the license Plate based on Classification. The experimental setup they applied was able to give them desirable results.

In this work the authors have only detected the number plate of any image, however in our work we have also read text from the number plate. We have also used OpenCV to get the region with number plate however the detection has been done using an ML model.

C. *A Review Paper on Automatic Number Plate Recognition System Using Machine Learning Algorithms by Shraddha Shridhar Ghadage*

In this paper the author has proposed to use Machine Learning algorithms to detect the number plate. The author has proposed the use of Segmentation for the character isolation, which is then passed to Character Recognition system to give license plate as output. The author has made use of a sensor to detect the presence of a vehicle, if detected the ANPR system will activate and check in the database, if the data is present the gate will open otherwise the gate will not be opened and the parking management authority will be notified via GSM.

The author uses Segmentation as the Machine Learning Algorithm in this application of ANPR system. The review paper also talks about the future application of our ANPR system

II. Work Done

In this project, we have implemented a wireless ANPR system to detect the number plate of vehicle and store it in the server. The workflow of this project is shown below in the flowchart: -

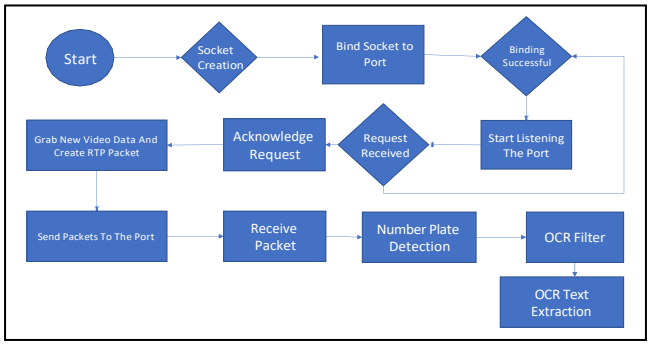

Fig.2. Flowchart of the Steps taken in this Project

A. *Client Server Architecture*

For our project to work it required to work wirelessly, hence since both the server and raspberry pi are connected to the same network, we implemented a simple client server architecture to send the images clicked by the raspberry pi of the car and send it over the network to the server. This is called streaming over network.

B. *Number Plate Detection*

We have implemented a number plate detection system using a generic code and a TensorFlow inbuilt model. The model that we are using contains optimum speed and accuracy in the present. Using this model, we have written a generic code using which we have trained our model based only on License Plate detection, however using our code, it can simply be used for other purposes as well, just by

Fig.3. Number Plate Detected with 82% Accuracy

changing the Machine Learning model, the dataset, or both. For example using the TensorFlow 2 detection model ZOO: SSd mobileNet V2 FPNLite model we have implemented an ANPR system, however changing this model to CenterNet HourGlass104 Keypoints 512x512, we can use the same code to point the keypoints on any images after detection of object.

C. *Optical Character Recognition (OCR)*

OCR is a technique used to detect text of images. We used EasyOCR to read the license plate of the image use it for vehicle logging. The Easy OCR is a simple library which gives details about the positioning of the text, and the text written in that region. However, a severe deficiency of this model is that it givens all kinds of text in the image. For Example, if there is a text written in the number plate unrelated to the license plate, it also provides the text of that region, thus giving us noise.

To solve this problem, we made an assumption that the license plate is takes maximum region of the detected image. Using this assumption, we performed OCR filtering, in which we selected a region with maximum area, and then performed text extraction from that maximum area region. Using this method, the accuracy of EasyOCR is also increased as well as we were able to reduce the noise in the output.

III. Result

The ANPR system was successfully implemented. The raspberry pi was sending continuous images to the server, and the server was able to detect the images from the image sent

by the raspberry pi, once the region of interest was detected by the number plate detector, we finally passed it through the OCR to get the number plate.

In order for the image to be passed to the OCR, we created a threshold of 80% accuracy, if the accuracy is above 80%, we were able to hold the number plate. The detector showed an average accuracy of 82-87% accuracy on the number plate detection. The OCR faced difficulties in recognizing some the similar character of the images such as 'G' and '8', 'S' and '8' of the plate, giving an average accuracy of 93%.

IV. Conclusion and Future Aspects

The system implementation was difficult to achieve, however with the model that we proposed, we were able to achieve results that were above our expectations. Due to unavailability of large amount of data our model was not able to achieve higher level accuracy. The OCR was also able to achieve desirable levels of results, given the short comings of our model. In the future the model can be upgraded very easily by just changing the type or name of the model that was imported in this project.

The future aspect of our system is huge. Just by changing the model that we are using; the same system can be used for detection of various objects. Our novelty lies in the implementation of our own client-server architecture, and the generified code, allowing us to improve our model on a large scale.

V. References

[1] https://i-proclaim.my/journals/index.php/abcjar/article/view/549/507

[2] https://www.researchgate.net/publication/257743677_A_New_License_Plate_Recognition_System_Based_on_Probabilistic_Neural_Networks/fulltext/026793560cf2946d9a21f87b/A-New-License-Plate-Recognition-System-Based-on-Probabilistic-Neural-Networks.pdf

[3] https://www.researchgate.net/profile/Muhammad-Asif-62/publication/224565257_Automatic_Number_Plate_Recognition_System_for_Vehicle_Identification_Using_Optical_Character_Recognition/links/0c96052c03a150264e000000/Automatic-Number-Plate-Recognition-System-for-Vehicle-Identification-Using-Optical-Character-Recognition.pdf

[4] https://www.researchgate.net/profile/Christos-Nikolaos-Anagnostopoulos/publication/4216505_Using_sliding_concentric_windows_for_license_plate_segmentation_and_processing/links/545c7d9a0cf25f65f56451c1/Using-sliding-concentric-windows-for-license-plate-segmentation-and-processing.pdf

[5] https://www.researchgate.net/profile/Harish-Rohil-2/publication/272863336_License_Plate_Recognition_System_using_Back_Propagation_Neural_Network/links/55668a6008aec22682ff1c68/License-Plate-Recognition-System-using-Back-Propagation-Neural-Network.pdf

[6] http://itfrindia.org/ICCIC/Vol2/9024ICCIC.pdf

[7] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

[8] https://ieeexplore.ieee.org/Xplore/cookiedetectresponse.jsp

[9] https://www.researchgate.net/profile/Padmavathi-Kora-2/publication/348920238_Automatic_Number_Plate_Recognition_System_using_Raspberry_Pi/links/6016cd1692851c2d4d0a68a2/Automatic-Number-Plate-Recognition-System-using-Raspberry-Pi.pdf

[10] https://medium.com/@nandacoumar/optical-character-recognition-ocr-image-opencv-pytesseract-and-easyocr-62603ca4357#:~:text=EasyOCR%20does%20the%20certain%20pre,algorithm%20to%20detect%20the%20text