ISSN No:-2456-2165

License Plate Detection, Recognition and Speed Estimation of Vehicles

Shivangi Gupta[1],
[1]Student,
Dept. of ECE, Jaypee Institute of Information Technology,
Noida, India

Dhruv Kumar Sharma[2],
[2]Student,
Dept. of ECE, Jaypee Institute of Information Technology,
Noida, India

Mandeep Singh Narula[3]
[3]Assistant Professor,
Dept. of ECE, Jaypee Institute of Information Technology,
Noida, India

Abstract:- The goal of this research is to develop a model for detecting and recognizing number plates as well as detecting speed of vehicles. Maintaining security in the neighborhood is a tough task in the current times. To address this issue, a surveillance system that uses license plate recognition to monitor the visiting vehicles and speed detection to monitor the over speeding vehicles is being presented as the solution. We have compared different model's performance for detecting and recognizing license plates. The best approach has been chosen and implemented to get results.

***Keywords*:** *Vehicle Detection, Number Plate Detection, recognition, Speed Estimation, Yolov5, Tesseract*.

I. INTRODUCTION

As the need for monitoring and tracking of vehicles on roads is increasing, we aim to develop a surveillance system that will detect the number plates of vehicles and their speed to enhance the safety and security of roads. In the current times, the amount of traffic on the road is increasing steadily every year. Increase in the number of vehicles on road make it important to monitor the vehicle number plates and speed to ensure the safety and security on roads. As a result, deep learning models can be used to detect and track a vehicle using its license plate and also estimate its speed for safety purposes.

➢ *Related Work*

An algorithm for license plate detection based on top-hat transform and wavelet transform has been presented [1]. The approach followed the following steps: the vertical gradient of the gray image is calculated and the region of license plate is distinguished by using top-hat transformation, then the horizontal projection of gradient variance is obtained and multiresolution features of candidates are extracted by using wavelet transformation. Lastly, the license plate's precise vertical position is recognised and recovered from the image. 360-degree photos from various landscapes and environments were used in experiments. The system immediately identified the license plate region, and the success percentage for successfully detecting license plates is 98.61%.

The paper explained the use of wavelet analysis for license plate location and suggested a type of location method based on the unique texture character of the given license plate [2]. The authors tried to improve the algorithm for low-contrast images for accurate detection of the license plate.

The paper examines the effectiveness of employing license plate recognition to obtain all vehicle-related information [3]. To raise the standard of the vehicle pictures fusion technique is applied to extract the number plate, separate the characters that are present on the number plate, and then use an ANN to determine the characters on the number plate. The experiment concluded that even in the presence of noise with a 50% density, a neural network can properly identify the characters on a license plate with a probability of 95%.

The author has evaluated the efficacy of three methods for detecting license plates in India [4]. In order to test their accuracy on both still photos and live stream videos, OpenALPR, k-NN, and CNN based approaches are chosen and customized for the local environment, and used.

Chomtip Pornpanomchai and Nuchakarn Anawatmongkonhave proposed a study to assess the efficacy of a simple algorithm used to locate license plates in a frame of video [5].

The research aims to identify a license plate on the front or back of a vehicle. The application allows users to snap screenshots from the video file, which will then result in the screenshot. The application will also pinpoint the location of the license plate in that screenshot. The application stores the image it captures as a jpg file.

A method for detecting license plates using a Support Vector Machine classifier with Histogram of Oriented Gradients features is discussed [6]. The system runs window searches at various scales, analyzes the HOG feature using an SVM, and finds the bounding boxes of those features using the Mean Shift technique. The lengthy scanning procedure is sped up using edge information.

The results on license plate detection indicate that this method is mostly unaffected by changes in lighting, license plate patterns, camera angle, and background. Alignment is then done on the plate candidates after license plate detection. It is suggested to segment the license plates. The ideal segmentation—where the characters are all segmented but not broken up—is what this model looks for. Finally, another SVM classifier including raw features, vertical and horizontal scanning features, recognises the characters. The results indicated that 99% of the digits were correctly identified, while 95% of the letters were.

Mangala A.G and Dr. Balasubramani R (2019) reviewed various methods for speed detection of vehicle detection. Various methods used included doing background image subtraction, object tracking, motion vector methodology, absolute, and centroid method. They used MATLAB to process the data [7].

The author introduces a new approach called Speed Detection Camera System (SDCS), a Radar substitute [8]. It can calculate a vehicle's speed by using a variety of image processing techniques on a video stream that is recorded from a single camera. The limitations of radar are solved by providing a cheaper and better solution with equal or greater precision.

A non-intrusive video-based method for assessing vehicle speed on city streets is suggested by the author in [9].The technique effectively locates vehicle license plates in moving image areas by integrating a novel text detector to an upgraded motion detector. Later, selected features on the license plate regions are tracked over numerous frames and perspective distortion is adjusted. Vehicle speed is determined by comparing the tracked features' trajectories to established outside observations. For more than 96.0% of the instances, the speeds were inside the [-3, +2] km/h limit specified by regulatory bodies in many countries, with an average error of -0.5 km/h. There was precision of 0.93 and recall of 0.87.

The author states the use of the Haar cascade classifier for object detection as a possibility [10]. In this work, fake drivers in the flow of traffic are found and tracked using image processing techniques. To find automobiles, the Haar Cascade Classifier is employed. The vehicle recognised in the image's axis coordinates are assessed, and an effort is made to pinpoint the vehicle using these axis coordinates. The system's accuracy rate reaches a very high value. The Haar cascade classifier is a reasonable choice for object detection, according to this study.

- *Problem Statement*

The need for monitoring and tracking vehicles is increasing due to safety and security issues. The objective of this study is to develop a dependable and effective surveillance system that will give authorities the information they need in real-time to ensure public safety and efficient traffic management.

- *Proposed Solution*

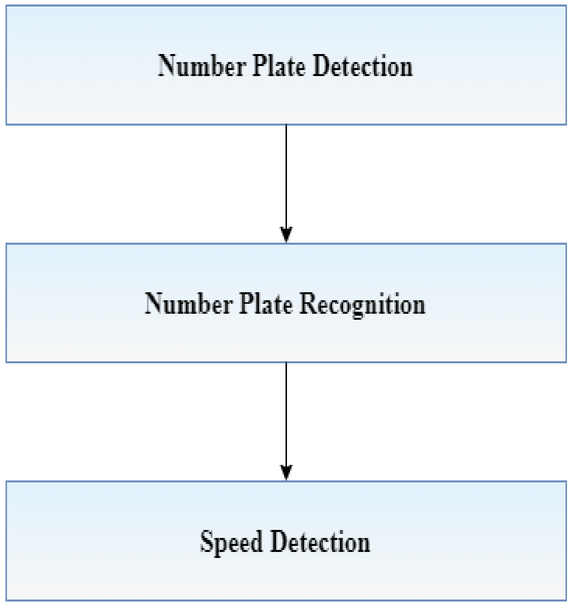

Fig 1 Proposed Solution

- *Number Plate Detection*

For the purpose of number plate detection, we used two approaches:

- ✓ Tensorflow
- ✓ Yolov5

Our dataset consisted of images of number plates of cars. The images had number plates of various colors and in different orientations.

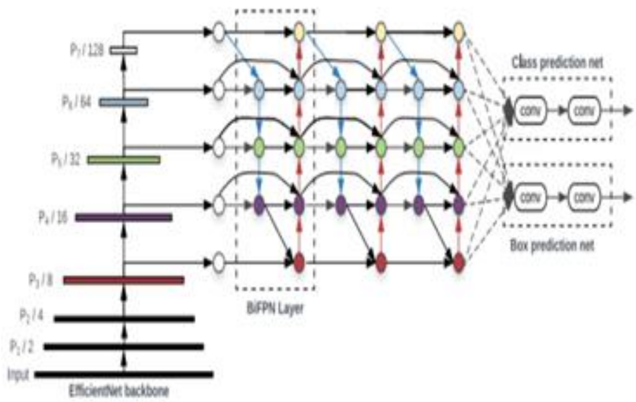

Fig 2 Object Detector Anatomy from [11]

In our first approach, we used tensorflow object detection api to detect, locate, and trace an object from an image or a video. The api was used to locate the number plate in an image using different car images and annotations to verify the results.

In our second approach, we used YOLO (You Only Look Once) [17] for object detection. A use case for which YOLO is intended involves the creation of features from input photos for object detection. Following that, a

ISSN No:-2456-2165

prediction system is given these features in order to create boxes around objects and determine their classes.

The YOLO model was the first object detector to include class labels into the end-to-end differentiable network process of predicting bounding boxes. There are three key components to the YOLO network.

- *Backbone:* A convolutional neural network that gathers and creates visual features at various granularities.
- *Neck:* A number of layers that integrate and mix visual features before sending them on to prediction.
- *Head:* Consumes neck-related features and performs class and box prediction procedures.

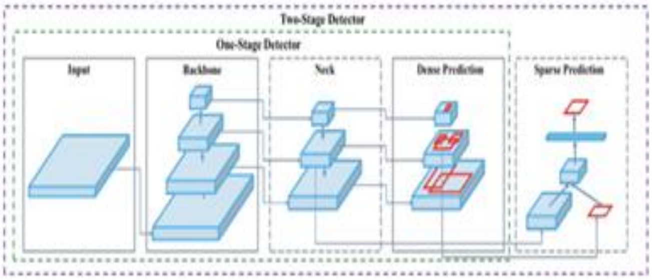

Fig 3 Representation of YOLO Process from [12]

We fine-tuned the object detection model to detect a type of object over different types of objects and then fine-tuned it on a dataset of number plates so that it would detect number plates in videos and images.

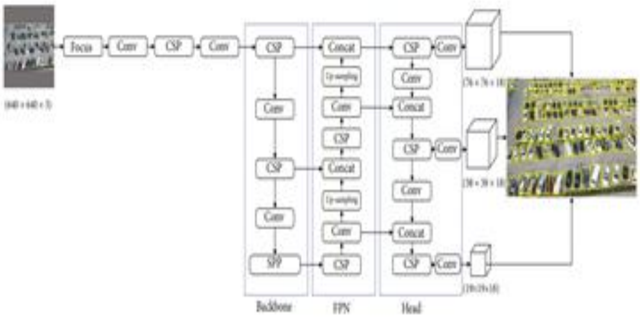

Fig 4 The Structure of YOLOv5 from [13]

We used YOLOv5 for detection of the license plates. It is responsible for the bounding boxes around the object to be detected, here the object being license plates. The YOLOv5 model does not have a particular feature like accuracy. The performance can be measured using the confidence of every object that is being detected in the frame. IOU is another metric to evaluate its performance which compares the true bounding box to the predicted bounding box.

- *Number Plate Recognition*

In image processing, the Hough Transform is a feature extraction method used to locate instances of objects within a particular class of forms. We have used it for skew correction of the license plates. When a license plate is captured, its orientation may vary and there might be skewness that will make it difficult to recognize the characters in the number plate. Hough transform solves this issue, making recognizing characters much easier and accurate.

We used two methods for optical character recognition to read the characters on the license plates:

- ✓ EasyOCR
- ✓ Tesseract

Developers of computer vision software can easily perform optical character recognition using the Python tool known as EasyOCR. Jaided AI, a business that specializes in optical character recognition services, developed and maintains the EasyOCR package. The PyTorch library and Python are used to implement EasyOCR.

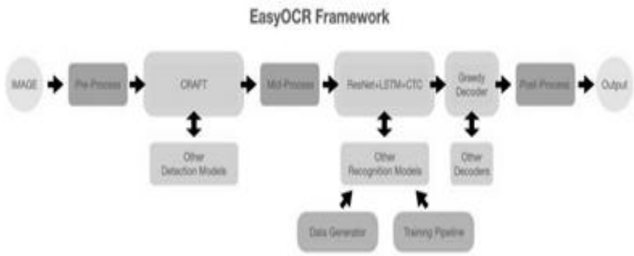

Fig 5 EasyOCR Pipeline from [14]

EasyOCR is a simple light-weight package that performs simple, straightforward OCR. It performs great when using a clean document or image. Camera captured images tend to have some distortions hence it was a good but not the best technology for optical character recognition.

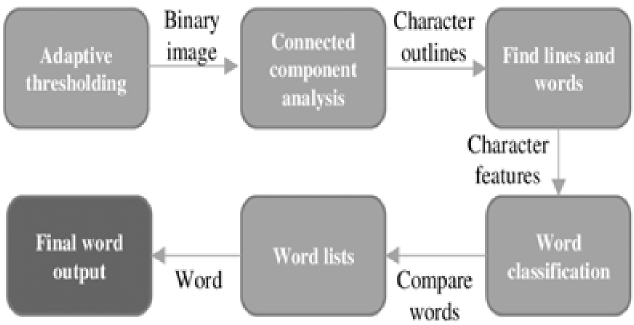

Fig 6 Tesseract OCR Engine Architecture from [15]

Tesseract is an open-source optical character recognition (OCR) engine. It is acknowledged as one of the most well-known, reliable and free-of-cost open-source OCR engines. It is currently one of the best technologies to implement character recognition. It is lightweight and easy to use.

We used tesseract OCR for optical character recognition and it performed well in reading number plates. Even in video captures, it was able to extract the correct number plate. It took different frames from the video and applied the OCR to extract number plates even on moving vehicles. The text read from the license plate was stored in a csv to create a database of all the vehicles captured by the camera.

- *Speed Estimation*

We have used open cv to track the speeds of various cars on multi lane roadways. For every nth frame, it makes use of the Haar Cascade Classifier to find automobiles. For faster processing, it removes superfluous portions from the image.

The process involves setting up two reference lines that have been specified for vehicle entry and exit. Any vehicle in any lane that crosses the entrance point has its time and location recorded. Utilizing centroid tracking techniques, tracking is performed. The moment the vehicles pass the exit line, time is taken. The projected speed of the vehicle is based on the time difference. Correlation tracker from the dlib library has been used for tracking vehicles on the road.

- *Implementation using Raspberry Pi*

The Raspberry Pi is a low-cost, "credit card-sized computer" that plugs into any display device. It is a very user-friendly tool, capable of everything a desktop computer is supposed to be able to accomplish [16].

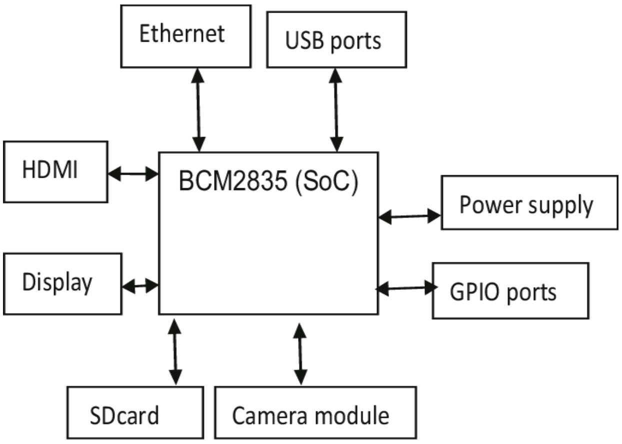

Fig 7 Raspberry Pi Hardware Architecture from [18]

The footage is then sent from the camera mounted on the Raspberry Pi to the client device, which is our own server, over the internet. Every image in the video is taken individually and saved as a JPEG file before being added to a buffer. Because the video is handled on the computer frame- by-frame and by many threads, the streaming of videos is susceptible to asynchronization issues. As a result, the frame will be processed ahead of the others if one thread has less work to do than the others. Race conditions may result from this, which will cause the frames to be out of sequence and the video to not be correctly stitched.

We need to employ a video buffer to hold some frames for transmission in order to get around this issue. Keeping this buffer guarantees that the frames from the Raspberry Pi server reach the client in the proper order and at the proper time, even if there will be a slight delay between the captured video and the streamed video.

Additionally, the video is compressed to reduce network traffic and to make it simpler for our computers to read and decode since raw video is more challenging for computers to process due to its larger size and incompatible file formats.

II. RESULTS

We have designed a system to detect license plates and speed with good accuracy. It detects the license plate using YOLOv5 and reading the characters of the number plate is done using Tesseract OCR. The performance of the system is as expected.

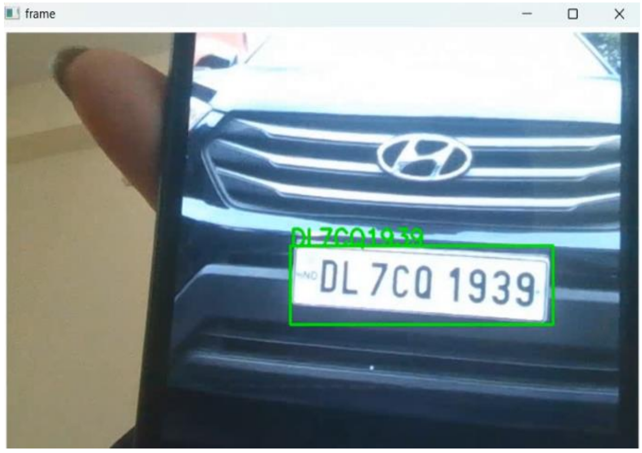

Fig 8 License Plate Detection and Recognition

Fig 9 Speed Estimation with Speed Limit 30kmph

III. CONCLUSION

The main purpose of this project is to create an inexpensive surveillance system using readily available IOT devices employing license plate detection and speed estimation for a secure and safe surrounding.

In this work, we present a method for detecting and recognising the number plate details on different license plates. We used an object detection model (YOLO) to detect and extract the license plate for recognition. Then we used an optical recognition technique, which is in charge of segmenting and recognising the number plate details The data is then stored in a csv file to maintain record for tracking vehicles for security purposes. We have also used speed detection techniques to detect speed violations. A speed limit can be set manually to detect the cars that violate the speed regulations. Hence, the system serves its purpose to ensure safety in the area.

REFERENCES

[1]. Feng Yang and Fan Yang, "Detecting license plate based on top-hat transform and wavelet transform," 2008 International Conference on Audio, Language and Image Processing, Shanghai, 2008, pp. 998-1003;

[2]. Ye Yunyang, Chen Wenming, Hu Junda and Zou Wenbin, "Research on localization method of vehicle license plate based on wavelet analysis," 2010 International Conference on Networking and Digital Society, Wenzhou, China, 2010, pp. 672-675;

[3]. Koval, Vasyl & Turchenko, Volodymyr & Kochan, V. & Sachenko, A. & Markowsky, George. (2003). Smart license plate recognition system based on image processing using neural networks. 123 - 127. 10.1109/IDAACS.2003.1249531.

[4]. B. S. Prabhu, S. Kalambur and D. Sitaram, "Recognition of Indian license plate number from live stream videos," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 2359-2365;

[5]. C. Pornpanomchai and N. Anawatmongkon, "Thai License Plate Detection from a Video Frame," 2009 WRI Global Congress on Intelligent Systems, Xiamen, China, 2009, pp. 533-537;

[6]. Xin Li, Dr. Zhihai He.(2010).Vehicle License Plate Detection and Recognition, University of Missouri.

[7]. Ramasamy, Balasubramani. (2017). A Review on Vehicle Speed Detection using Image Processing. 4. 23-28.

[8]. Ibrahim, Osman & ElGendy, Hazem & Elshafee, Ahmed. (2011). Speed Detection Camera System using Image Processing Techniques on Video Streams. International Journal of Computer and Electrical Engineering. 3. 771. 10.7763/IJCEE.

[9]. D. C. Luvizon, B. T. Nassu and R. Minetto, "A Video-Based System for Vehicle Speed Measurement in Urban Roadways," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 6, pp. 1393-1404, June 2017, doi: 10.1109/TITS.2016.2606369.

[10]. Luvizon, Diogo & Nassu, Bogdan & Minetto, Rodrigo. (2016). A Video-Based System for Vehicle Speed Measurement in Urban Roadways. IEEE Transactions on Intelligent Transportation Systems. PP. 1-12. 10.1109/TITS.2016.2606369.

[11]. Gowsikraja P, Thevakumaresh T, Raveena M, Santhiya.J, Vaishali.A.R.R. (2022). Object Detection Uing Haar Cascade Machine Learning Algorithm.

[12]. Tan, Mingxing & Pang, Ruoming & Le, Quoc. (2019). EfficientDet: Scalable and Efficient Object Detection.

[13]. Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

[14]. Z. Chen, L. Cao and Wang ,YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images.

[15]. GitHub-JaidedAI/EasyOCR: Ready-to-Use OCR with 80+ Supported Languages and all Popular Writing Scripts Including Latin, Chinese, Arabic, Devanagari, Cyrillic, etc.Available online: https://github.com/JaidedAI/EasyOCR.

[16]. Gjoreski, Martin & Zajkovski, Gorjan & Bogatinov, Aleksandar & Madjarov, Gjorgji & Gjorgjevikj, Dejan & Gjoreski, Hristijan. (2014). Optical character recognition applied on receipts printed in macedonian language. 10.13140/2.1.1632.4489.

[17]. Khalil, Faisal. (2015). Raspberry Pi Personal Cloud Storage. 10.13140/RG.2.2.18628.88960.

[18]. You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick , Ali Farhadi.

[19]. Pathinettampadian, Karthikeyan & Sivamurugesan, Vasuki & Karthik, K. & Murugesan, Sakthivel. (2018). Design and Development of Laplacian Pyramid Combined with Bilateral Filtering Based Image Denoising: Second International Conference, ICSCS 2018, Kollam, India, April 19–20, 2018, Revised Selected Papers. 10.1007/978-981-13-1936-5_24.