International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 5, July-August 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Real Time Number Plate Recognition

P. Harini[1], K. Harish[1], V. Gnanaprakash[2]

[1]Student, [2]Assistant Professor,
[1,2]Bannari Amman Institute of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

In the era of fast growing technologies, there is a huge demand among people for a comfortable lifestyle and travelling. In the last few years, the number of vehicles on road has been explosively grown. With the increasing growth in the vehicular sector every day, the need to track individual vehicle becomes a very challenging task. The proposed project suggests an automated way of tracking the fast moving vehicles with the help of the surveillance cameras on the road. One of the tedious step towards this project is to process the CCTV footage in the real time background. To overcome this strategy, an efficient deep learning framework particularly meant for object detection named YOLO (You Only Look Once) is used. The proposed work consists of four main steps. The first process is to convert the video footage into images and detect the car from each of the frames. The next step is to detect the license plate from the detected cars. Reading and recognizing the number plate characters from the detected number plates form the final step. To make the process of training the deep learning model easier, this system uses ImageAI library. It provides very efficient and easy to use classes to perform image recognition tasks. This system uses a customized dataset of 1000 images specifically containing Tamil Nadu license plate images. These images are taken under distinct light conditions and contradictory angles. The proposed method is tested in real time environment and achieved 97% accuracy for car detection, 98% accuracy for number plate localization and 90% accuracy for character recognition.

KEYWORDS: *CCTV footage, object detection, car detection, YOLO, number plate localization, ImageAI, image recognition, customized dataset, character recognition*

How to cite this paper: P. Harini | K. Harish | V. Gnanaprakash "Real Time Number Plate Recognition" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-5, August 2020, pp.715-720, URL: www.ijtsrd.com/papers/ijtsrd31872.pdf

IJTSRD31872

I. INTRODUCTION

A. Overview of the project

The drastic increase in the vehicular traffic on the roadways stimulates a huge demand in the technology for traffic monitoring and management. In this scenario, manual tracking of vehicles running fast on the road is practically not feasible. There will be wastage of man power and time. Even if it is operated manually, that will reflect huge difficulties and enormous errors. There are already available solutions for tracking the vehicles and number plates using machine learning algorithms. But in real time, these algorithms literally fail due to its complexity for processing in real time background. Hence there is an instantaneous necessity to develop an automatic system that will help tracking the vehicles by tracing their number plates in a most efficient way.

Besides playing an important role in vehicle tracking, Automatic Number Plate Recognition (ANPR) also plays an inevitable role in systems like parking management systems, toll payment processing systems etc., and several systems where authorization is a much needed. It greatly helps security officers to save their time by automating the process. In recent decades, computer vision technology has taken great strides on several real world issues. In earlier days vehicle number plates were identified using template matching techniques by identifying the width, height, contour area etc., Now several deep learning models trained over an enormous amount of data is widely used in Number Plate Recognition.

In this project, two CNN models are used. Hence two datasets consisting of car images and number plate images are required. For training the car images, Stanford cars dataset from the internet is used. For number plate images, customized dataset was created with the help of internet source and by taking pictures of the cars around. Once the data is obtained, it must be split into train and test images and annotated to the machine readable form. To train the model to detect the vehicles, the system uses YOLO (You Only Look Once) algorithm. The live CCTV footage is converted into frames. The frames are passed through YOLO algorithm to detect the cars in it. The detected cars are stored in separate images in a folder. These images are checked for number plate. The detected number plate will be cropped and stored in another folder. The characters in these number plates are recognized using OCR (Optical Character Recognition). The extracted text is then copied to an excel sheet with the time, date and vehicle number. This system tends to give a higher accuracy than the existing systems and has an additional advantage of being implemented in real time.

B. Scope of the Project

In recent years, Automatic Number Plate Recognition takes a vast progress in the applications of machine vision technology. It is expected to come up with more advancement in future. It is able to support a wide range of community with its efficient and useful applications like parking management system, toll fee collecting system, vehicle fine collecting system, on-road vehicle tracking system etc., A large amount of population reside in cities in which they need a secure parking space to avoid the access of unauthorized persons. ANPR also helps collecting the fee in tolls by identifying the vehicle's number plate. Vehicle tracking also helps with fine collection for those who are trying to violate the traffic and road rules. It also aids in maintaining a database of the moving vehicles on road.

This project creates an efficient base system for all these applications. This ANPR is a very complex task. At the moving speed of the vehicles, the images captured must be sufficient enough in brightness, intensity and clarity to be processed further. Furthermore, the angle at which the images are taken forms a major part. The most complex part is that every nation has a unique standard for printing number plates. But in India, there is no standard for this. It may have variations in the font, spaces, letters and numbers. The main scope of the project is to detect and recognize Indian number plates with a higher accuracy.

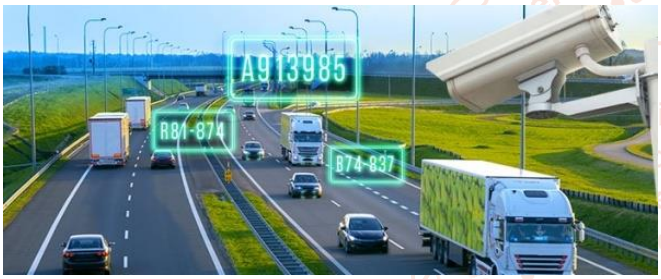

Fig 1 ANPR in real time

C. Objective of the Project

Automatic Number Plate Recognition nowadays plays an inevitable role in several applications. Though the ANPR uses several algorithms it still fails in the case of accuracy and implementation in real time. It can be overcome by implementing it using deep learning techniques. Deep learning is a vast field of Artificial intelligence (AI) which uses neural nets to learn from a huge amount of data. It is a subset of Machine Learning which uses multiple layers to get high level features from a raw input. Deep Learning is now used in almost all the real time applications. Unlike other algorithms, it shows a high level of accuracy and minimum acceptable errors. This system uses Convolutional Neural network (CNN) to detect the cars and number plate. The main aim of the system is to design a deep learning model that could read the number plate of the fast moving vehicles on the road using the surveillance camera and save the extracted number plate data on a excel sheet. Also, this system uses ImageAI library to make the process easier and also efficient. It also uses several image processing techniques to preprocess the obtained frames. To annotate the images, an efficient annotation tool called LabelImg is used. The whole system is implemented using python programming language.

II. LITERATURE SURVEY

Several studies and work has been carried out earlier. In [2], a deep learning model is created to recognize the number plate using the Turkish dataset made by them. They used Tensorflow framework with the Keras deep learning library. They collected 34, 58 images of which the smearing algorithm is applied using a MATLAB program. 75% of the images were used for training, 25% for testing and 5% for validation. Since the images were taken from the real time background, they carried out several image processing techniques like median blur smoothening, Adaptive Gaussian thresholding and morphological transformations. After these preparations, the CNN model is trained using the images. The image features extracted from CNN are applied to LSTM network followed by the decryption algorithm. By this method, they achieved an overall accuracy of 96.36% for plates, 99.43% for numbers, 99.05% for letters 99.31% for all the characters.

In [3], an automatic number plate recognition system using machine learning approach is developed. They got the input from an Infrared camera followed by contrast enhancement and noise reduction as preprocessing steps. Then they localized the number plate in the image by finding out the Region of Interest (RoI). After that, contour tracing is applied to get the salient features of the image. Then Canny's edge detection is done to find out the edges of the characters in the number plate. Finally, segmentation is applied to separate the characters. The individual characters are recognized using pattern matching by Artificial Neural Networks (ANN).The whole system was developed using MATLAB software. The author describes this system as a cost efficient and accurate system.

A survey was carried out [4] on various methodologies used in implementing Automatic Number Plate Recognition (ANPR). The authors took nearly 78 reference papers and evaluated their accuracy results. The basic steps in ANPR include vehicle Image capture, number plate detection, Character segmentation and Character recognition. For number plate detection the factors such as plate size, plate location, plate background and screw must be considered. The maximum accuracy for plate detection was achieved by Canny's edge detection as per the survey. Character segmentation can be implemented using image binarization, CCA (Connected Component Analysis), vertical and horizontal projection which produces better results. It is followed by character recognition which is usually done by Artificial Neural Networks, template matching or Optical Character Recognition (OCR) techniques. The maximum accuracy for Character recognition was achieved by tesseract OCR with 98.7% results.

In [5], a review based on Automatic Number Plate Recognition (ANPR) is carried out . The proposed system consists of a camera module, sensor, control unit, GSM and an embedded server. It tries to block the unauthorized vehicles by comparing the vehicle database saved already. The images captured from the camera are converted to gray scale and enhanced by adjusting the histogram. The edges are detected using Sobel's edge detection method. Then morphological image processing is done. After that, the segmentation is done on the edge detected image. Finally, the characters are recognized using machine learning approach.

In [6], a plate recognition system using deep learning approach is developed. They developed an OCR system with a customized dataset. The dataset was made artificially by

taking some images from the internet and adding noises and backgrounds to those images. For background, SUN database and Stand ford database are used. For number plate detection YOLO (You Only Look Once), an object detection framework is used. For character recognition, Convolutional Neural Network (CNN) is used. The output layer of the CNN consist of 7.62 neurons for 7 characters. 10-fold-cross validation is applied at the output to find the accuracy. The overall accuracy for the CNN plate detector is 98.5% and for the CNN character recognition is 96.8%. This OCR based system gives an overall system efficiency of 94%.

In [7], a number plate recognition system is created using Convolutional Neural Network. The images taken from a camera is preprocessed by converting the RGB image to gray scale, noise removal and binarization. Then the license plate is extracted by using Connected Component method depending upon the properties such as major axis length, minor axis length, area and bounding box etc., The characters in the extracted license plate is segmented using horizontal and vertical scanning. Finally, The characters are recognized using Convolutional neural network (CNN). The dataset used to train the CNN consists of 1000 images for each 36 characters. Out of 36,000 images, 30,000 samples are used as training data and 6000 for testing data. They used descent algorithm to minimize cross-entropy with a learning rate of 0.5. The overall accuracy obtained was 97%.

In [9], an automatic system for license plate recognition is developed. The system was divided into three categories: license plate detection, character segmentation and character recognition. In license plate detection, the input image is converted to HSV image and applied with several filters. Then CNN is applied over the images to detect the plates. In character segmentation, the images ia again pre-processed. The second CNN model is applied to segment the characters in the image. Finally, soft max layer is applied to predicate the class proabilities. The dataset used is of two classes: positive class and negative class. The first dataset is LP dataset and non-LP dataset. The second dataset is for characters and non-characters. The overall accuracy achieved is 94.8%.

III. EXISTING WORK

In the existing systems, the detection algorithms have severe drawbacks in number plate localization. They need large computational power and the results are highly dependent on the image quality and the reliability degrades with the high noisy pictures. Band pass filters are generally used for finding the intensity variation in the horizontal and vertical rows of the number plate, which leads to several independent regions and it causes false results. The surroundings are also included in this technique hence window filtering should be used to remove the surroundings, to complete this process it takes huge power and time.

A morphological technique is the collection of non linear operations related to the shape in an image. The structuring element is centered on pixels of image and mathematical operations are carried out with its neighborhood pixels. It enhances the bright regions surrounded by dark regions, this algorithm gives 83% accurate output. It does not work for white cars having white background and failed for different illuminating environments.

Edge processing is the technique where histograms of the edges are plotted. These edges are processed in both horizontal and vertical directions, and the major disadvantage is the boundary clipping effect and it reduces the efficiency level. OpenCV with tesseract OCR engine correctly detects the number plate and extract the characters, but the process is very slow. It takes more time for edge and feature detection. To overcome the computation time inefficiency and also to improve the efficiency, neural networks are used.

YOLO object detection algorithm is developed using convolution neural network. CNN increases the efficiency and by increasing the hidden layers the system learns well about the relation between input and output layers. The main drawback in YOLO is, it works well for the live video detection. But for processing every frame the process is very tedious. In neural network architecture the code complexity is very high and GPU are required for processing neural networks, but it gives high efficiency when the system is trained for more number of layers. To reduce the complexity of code and also to reduce the processing speed ImageAI framework is used and it produces high efficiency than other methods discussed earlier.

IV. PROPOSED WORK

The number of vehicles on road has exponentially increased which are the major reasons for traffic congestion and violation. In order to reduce the violations and also to automate the traffic management, ANPR (Automatic Number Plate Recognition) has been developed. There are various ANPR techniques carried out in India and their efficiency is very less. The proposed system focuses to optimize and improve the efficiency of ANPR. Initially the system was trained with YOLO (You Only Look Once) object detection algorithm. The algorithm works well for the detection but for recognition it does not perform well. ImageAI framework which performed well for detection and recognition techniques. The system is trained using NVIDIA Jetson Nano for car and number plate recognition and the software is built using python.

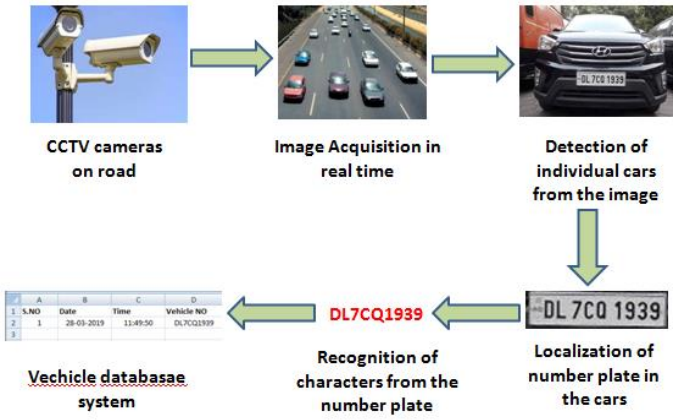

Fig 2 Block diagram of the proposed work

A. Image Acquisition

The system is developed to assist the algorithm in all dynamic environments and aids the system in number plate recognition. Camera is the significant component for any ANPR system, this system is built for accessing all the cameras using their IP address. Image acquisition can be done from CCTV footage of Traffic control, Stolen cars tracking, Border crossings, Parking system cameras. The

videos are converted into frame by frame and every frame is processed for the detection of the vehicle.

B. Vehicle Detection

Vehicle Detection is the first step after the image acquisition. The system has to be trained to identify cars in the frame.

C. Dataset Preparation

For training, Stanford car dataset from the Kaggle is used. The dataset consists of 16,185 images of 196 classes. It is split into training and testing .The training data consist of 8144 images and for testing 8041 images are taken. First step is to create two folders named train and validation; the train folder should consist of two subfolders annotations and images, similarly for the validation. In the image folder we will have all the train and test images while the annotations folder consists of XML file. The XML (Extensible Markup Language) is used to define a set of rules for encoding documents in a format that is both machine and human readable. The XML file is obtained using LABELIMG tool.

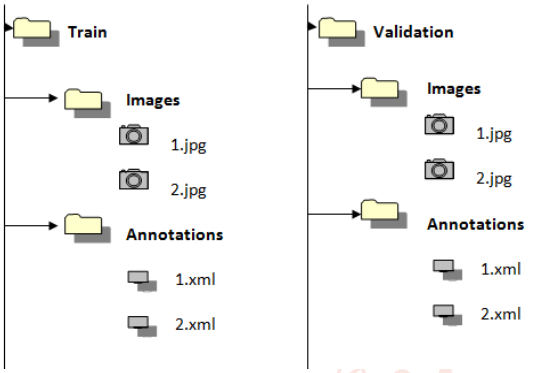

Fig 3 Organization of the dataset

D. Training the model

Next step is training the images which is done using the NVIDIA Jetson Nano all the images in the train folder are trained. It understands all the features of the car and generates a JSON file. It also generates 5 different detection models in H5 format. After completion of training the next step is testing, the input for the testing is H5 models and the JSON file. For every model it gives MAP(Mean Average Precision) score in the evaluation process, finally it gives model which has the highest MAP score that H5 file should be used as weights file in detection step. The final step is detection of cars and the input is JSON and H5 file and the car obtained in each and every frame is stored in the separate folder. In this IMAGEAI framework we can able to detect all types of vehicles our system requires only car, hence trained custom objects only for car.

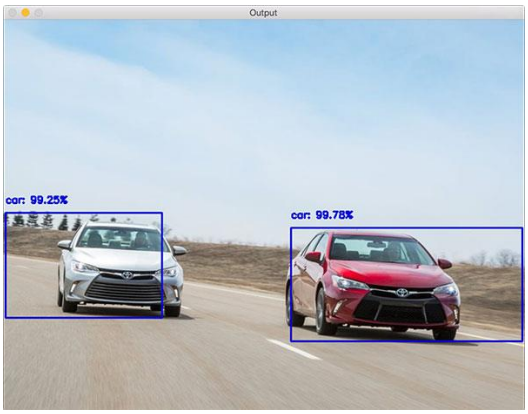

Fig 4 Cars detected in a frame

E. Number Plate Localization

Number plate Localization is the step where the number plate from the larger scene is bound. The location of the number plate is identified and the output will be a sub image contains only the number plate. The dataset consists of 1000 images of number plates and it is split into training and testing. For training, 800 images are taken and annotated using LabelImg similar to the process done in vehicle detection. After training, the testing is done. After that detection is done. All the detected number plates are stored in separate folder for Character Recognition.

Fig 5 Number plate detected in a car image

F. Character Segmentation

The number plate images are converted to gray scale. After that, bilateral filter is applied on the grayscale image. A bilateral filter is an edge preserving, non - linear, noise reduction and smoothening filter. Each pixel is replaced with the weighted intensity values of the neighborhood pixels. Specifically, it will preserve the edges while removing the noises in the image. After that, Canny's edge detection is applied The process of edge detection involves five steps:

- Gaussian filter is applied to remove the noises in the image
- To find the intensity gradients in the image
- Non-maximum suppression is applied to get rid of spurious response from edge detection
- Double threshold is applied to determine the potential edges
- Suppress the other edges that are not connected to strong edges by hysteresis

After edge detection, contours are traced. The extracted segmented characters are sent as an input for character recognition.

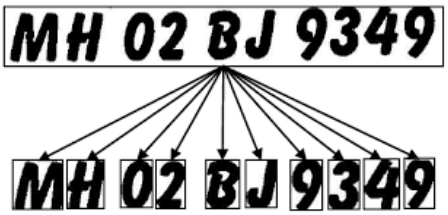

Fig 6 Character Segmentation

G. Optical Character Recognition

Optical Character Recognition (OCR) is the conversion of images of handwritten or printed text into a machine text.

There are several OCR engines. This system uses Tesseract - OCR engine. This can be downloaded in Anaconda using git. The engine path must be specified and added. The segmented characters are given as input to OCR. The OCR will recognize those characters. The extracted data is stored in a data file or an excel sheet.

	A	B	C	D	E
1	S.No	Date	Time	Vehicle Number	
2	1	28-03-2020	08:30:42	TN36AE5899	
3	2	28-03-2020	09:42:51	TN42JI3200	
4	3	28-03-2020	09:50:01	DL7CQ1399	
5	4	28-03-2020	10:48:54	KL90ST6655	
6					
7					

Fig 7 Data Entry in excel sheet

V. RESULT AND CONCLUSION

The results obtained by implementing the proposed work is discussed here. Initially, the dataset is collected. The data is annotated using LabelImg tool. The original image files in the dataset and the .xml files are saved in the separate folders in same directory for both the test and train images. The input from the CCTV live footage is converted into frames. The CNN model is trained to detect the cars from image. The number of epochs are set by the users. The system will get trained using the power of GPU until all the epochs are completed.

```
Epoch 1/100
1/25 [>.............................] - ETA: 52s - loss: 2.3026 - acc: 0.2500
2/25 [=>............................] - ETA: 41s - loss: 2.3027 - acc: 0.1250
3/25 [==>...........................] - ETA: 37s - loss: 2.2961 - acc: 0.1667
4/25 [===>..........................] - ETA: 36s - loss: 2.2980 - acc: 0.1250
5/25 [=====>........................] - ETA: 33s - loss: 2.3178 - acc: 0.1000
6/25 [======>.......................] - ETA: 31s - loss: 2.3214 - acc: 0.0833
7/25 [=======>......................] - ETA: 30s - loss: 2.3202 - acc: 0.0714
8/25 [========>.....................] - ETA: 29s - loss: 2.3207 - acc: 0.0625
9/25 [=========>....................] - ETA: 27s - loss: 2.3191 - acc: 0.0556
10/25 [==========>...................] - ETA: 25s - loss: 2.3167 - acc: 0.0750
11/25 [===========>..................] - ETA: 23s - loss: 2.3162 - acc: 0.0682
12/25 [============>.................] - ETA: 21s - loss: 2.3143 - acc: 0.0833
13/25 [==============>...............] - ETA: 20s - loss: 2.3135 - acc: 0.0769
14/25 [===============>..............] - ETA: 18s - loss: 2.3132 - acc: 0.0714
15/25 [================>.............] - ETA: 16s - loss: 2.3128 - acc: 0.0667
16/25 [=================>............] - ETA: 15s - loss: 2.3121 - acc: 0.0781
17/25 [==================>...........] - ETA: 13s - loss: 2.3116 - acc: 0.0735
18/25 [===================>..........] - ETA: 12s - loss: 2.3114 - acc: 0.0694
19/25 [====================>.........] - ETA: 10s - loss: 2.3112 - acc: 0.0658
20/25 [=====================>........] - ETA: 8s - loss: 2.3109 - acc: 0.0625
21/25 [======================>.....] - ETA: 7s - loss: 2.3107 - acc: 0.0595
22/25 [=======================>....] - ETA: 5s - loss: 2.3104 - acc: 0.0568
23/25 [========================>...] - ETA: 3s - loss: 2.3101 - acc: 0.0543
24/25 [=========================>..] - ETA: 1s - loss: 2.3097 - acc: 0.0625Epoch 00000:

25/25 [==============================] - 51s - loss: 2.3095 - acc: 0.0600 - val_loss: 2.3
```

Fig 8 Training the model

After the training is completed, a. json file is created. The models are also created in a separate folder. Next step is to validate the model using the validation samples. After the validation, mAP (mean Average Precision) score of each model is analyzed. Average precision is one of the famous metrics to measure the accuracy of deep learning models. It computes the average Precision value for recall value over 0 to 1. The general meaning for average precision (AP) is finding the area under the precision -recall curve. After validating the model the predictions for vehicles detection can be done.

mean Average Precision(A.P) = $\int_0^1 p(r)dr$

After that, the model can be trained to detect the number plate in the car images using the cars dataset. A JSON file and H5 files are generated and stored in folder.

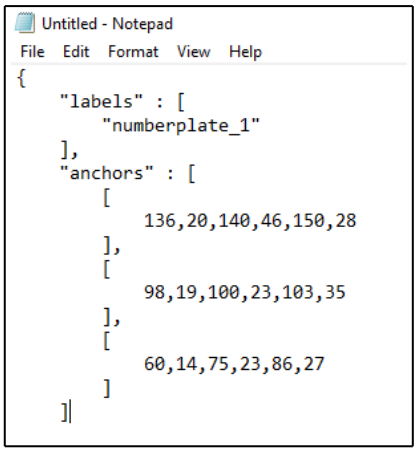

Fig 9 JSON file of number plate

Finally, the model is trained to detect both the cars and number plates. The frames obtained from the input video footage is stored in a folder. The cars from those images will be detected and stored in a separate folder. From that, the number plates are detected and again stored in a separate folder.

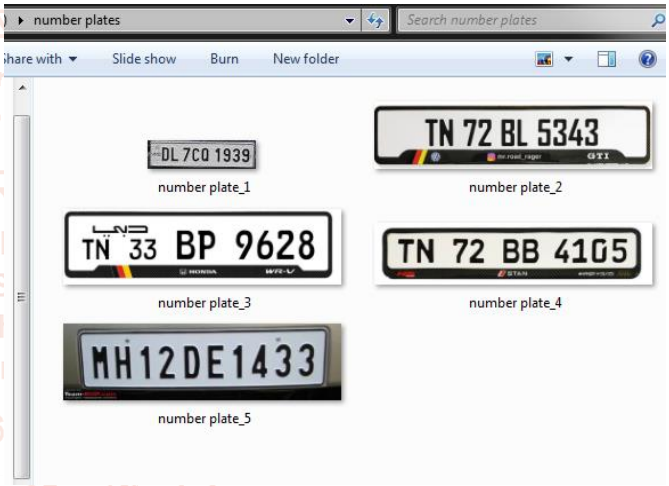

Fig 10 Detected number plates in a folder

The detected number plates are then processed for character segmentation. The segmented character are then recognized using OCR(Optical Character Recognition). The extracted character in a number plates are entered in a excel sheet with the date and time of the vehicle entry. The accuracy obtained by the system is shown below:

Models	Accuracy
Car detection	98.5%
Number plate detection	97%
Optical Character Recogniton	96.7%

Table no 1. Accuracy and the models

VI. CONCLUSION

The implementation of ANPR system for various applications were achieved using IMAGEAI framework and NVIDIA Jetson Nano hardware has been used for training. At first, morphological processing was employed for license plate localization and the efficiency is analyzed, the algorithm poorly performed under bright illuminated environments. This is followed by the implementation of edge detection algorithms and the efficiency of localization is increased. Finally the IMAGEAI framework is used which is very much

efficient than YOLO object detection. All the architecture of neural network can be implemented in IMAGEAI, the system is also trained for detection of cars and number plate localization. The parameters of neural network were optimized to enhance the performance of neural network and using this framework the code complexity is also reduced. The system was tested in a dynamic environment where it automatically detected the vehicles. The plate localization and character recognition was performed successfully than existing systems.

VII. FUTURE SCOPE

In recent years, digitization and IP based communication networks have allowed Automatic Number Plate Recognition (ANPR) to achieve greater utility. The ANPR market will grow by around 30 percent in the next few years. Electronic identification of vehicles using DSRC onboard units will always require some complementary, non-intrusive technique such as ANPR. This technique can be implemented in parking, access control, motorway road tolling, border control, journey time measurement, law enforcement. In future this system can be improved with difficulties such as poor file resolution, because the plate is too far away but sometimes resulting from the use of a low quality camera, poor lighting and low contrast due to overexposure and reflection. The algorithm is developed to capture images from a large distance and also have the ability to predict multiple license plates when multiple vehicles are present on single frame. The efficiency of detection of number plate and OCR (Optical Character Recognition)output can be increased.

References

[1] İrfan Kılıç, Galip Aydin, "Turkish Vehicle License Plate Recognition Using Deep Learning", International Conference on Artificial Intelligence and Data Processing (IDAP), 1-5. 10.1109/IDAP.2018.8620744, 2018.

[2] J. V. Bagade, Sukanya Kamble, Kushal Pardeshi, Bhushan Punjabi, Rajpratap Singh, "Automatic Number Plate Recognition System: Machine Learning Approach", IOSR Journal of Computer Engineering (IOSR-JCE) ISSN: 2278-0661, ISBN: 2278-8727, PP: 34-39.

[3] Atul Patel, Chirag Patel, Dipti Shah, " Automatic Number Plate Recognition System (ANPR): A Survey", International Journal of Computer Applications (0975 – 8887) Volume 69– No.9, May 2013.

[4] Shraddha S. Ghadage, Sagar R. Khedkar, "A Review Paper on Automatic Number Plate Recognition System using Machine Learning Algorithms", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 8 Issue 12, December-2019.

[5] Marko Arsenović, Srđan Sladojević, Andraš Anderla, Darko Stefanović, "Deep Learning Driven Plates Recognition System, XVII International Scientific Conference on Industrial Systems (IS'17)" Novi Sad, Serbia, October 4. – 6. 2017.

[6] Shrutika Saunshi, Vishal Sahani, Juhi Patil, Abhishek Yadav, Dr. Sheetal Rathi, "License Plate Recognition Using Convolutional Neural Network", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727.

[7] Selmi, Zied & Ben Halima, Mohamed & Alimi, Adel, "Deep Learning System for Automatic License Plate Detection and Recognition", 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) 1132-1138. 10.1109/ICDAR.2017.187, 2017.