International Journal of Future Generation Communication and Networking
Vol. 13, No. 1s, (2020), pp. 406-- 413

# Automatic Number Plate Detection System and Automating the Fine Generation Using YOLO-v3.

Prof. Rupali Hande[1], Simran Pandita[2], Gaurav Marwal[3], Gaurav Marwal[4], Sivanta Beera[5]

*1Asst. Professor, Department of Computer Engineering, VESIT ,Mumbai University, India.*
*2,3,4,5 Student, Department of Computer Engineering, VESIT ,Mumbai University, India.*

*[1]rupali.hande@ves.ac.in,[2]2017.simran.pandita@ves.ac.in,[3]2017.gaurav.marwal@ves.ac.in,[4]2017.nilesh.talreja@ves.ac.in, [5]2017.sivanta.beera@ves.ac.in*

***Abstract***

*Vehicles violating traffic rules must be charged with fines. Recognizing the vehicle number in the complex traffic conditions is difficult. An Android phone platform based automated number plate recognition system is proposed. The captured image is processed to get the optical characters. The built-in GPS module can be used to geo-tag images. The system that we are developing sends a trigger to the nearby traffic policeman as soon as the person breaks the signal. This is real time and the driver will be able to view where and when a person broke the rule. Previous methods for this, like R-CNN and its variations, used a pipeline to perform this task in multiple steps. This can be slow to run and also hard to optimize, because each individual component must be trained separately. This system is designed using YOLO-v3 and Django framework, and the application is developed using React-Native framework.*

***Keywords*:** *Number plate detection; fine generation; notification; object detection; optical character recognition; You Only Look Once- v3; django; react-native.*

## 1. Introduction

License Plate Recognition has been widely used in different domains like traffic control, security process, crime detection and automatic parking systems. In recent times LPR has taken a leap into technology easing the work of car detection. Besides all the improvements LPR has made in this modern era there still are problems related to its accuracy, efficiency and processing time. There are many challenges that affect the process of LPR like poor video quality and meteorological conditions which results in inaccurate results as output. At the same time many projects related to LPR have high processing time and are redundant in real-time execution for traffic monitoring and investigation To tackle this problem our project proposes a real time solution for license plate detection by first detecting the license plate and simultaneously converting the license plate using Optical Character Recognition to character data and sending alerts to all the traffic officials in the

ISSN: 2233-7857 IJFGCN

nearby signals. This will allow the authorities to act upon the problem in real time and the culprit will be caught on the next signal. Our project focuses on minimum processing time and providing real time solution. This will help the system to work more efficiently and hence will result in better working all together

## 2. Literature Survey

[10],[11],[14]YOLOv3 You Only Look Once. As the name suggests sends the whole image of the input only once through the network.YOLO is much faster than SSD and maintains a commendable accuracy in the same.YOLOv3 gives a better and faster result when running on a GPU.It divides the image into a 13*13 grid of cells. Input size of the image used in the experiments are generally of the size 416*416.Each cell is responsible for predicting a number of boxes in the Image.For each bounding box the algorithm also predicts the confidence that the bounding box encloses an object, and the probability of the enclosed object being a particular class. YOLOv3 is faster and more accurate than the previous versions of YOLO. YOLOv3 has different pre-trained versions to integrate with and one of them is the tiny yolo weights version which has less processing time as compared to others which is a perfect fit for our approach.[7],[8],[9] React-Native is a framework developed by facebook to build cross platform mobile apps. Results derived from this paper were that on comparing React-Native, Ionic, Native development, Reactnative gave the best results. It uses javascript as the main language and is based on the react library of javascript. Many applications in today's date including Netflix, Instagram etc, use this framework to develop their websites and mobile apps.[12] Legacy surveillance systems have a tendency to store camera images from different sources . On the other hand deep learning and convolutional neural networks gives an edge in the efficiency of the surveillance system in terms of speed and accuracy. Another advantage that it proposes is of the cost efficient way as compared to the legacy system.[21] Optical character recognition is the process of converting a conventional form of data visible to the eye into text for future use. The accuracy of the conversion can be increased by the use of pre processing methods like grayscale and bilateral filter. This eases the conversion of the data into text for the system and hence we acquire better results out of the input data.

## 3. Methodology

### 2.1 Algorithm used

Yolov3: Yolo abbreviates to You Only Look Once depicting its ability to detect objects and entities by using CNN (Convolutional Neural Network).Neural Network in YOLO uses weights trained by the user through annotated training data by using bounding boxes. Hence YOLO takes an image as input puts it through a Neural Network and gives the output in the image through bounding boxes.The input image is divided into S*S grid of cells.Each cell contributes to the object detection. Each cell predicts Bounding Boxes as well as Class probabilities. The prediction consists of 5 components (x,y,w,h,confidence).(x,y) represents the centre of the bounding box and (w,h) are the width and the height of the boxes.

ISSN: 2233-7857 IJFGCN

Confidence represents the Estimated Prediction Accuracy of the object.YOLO is extremely fast and accurate as compared to other algorithms and hence was our primary choice for this project.

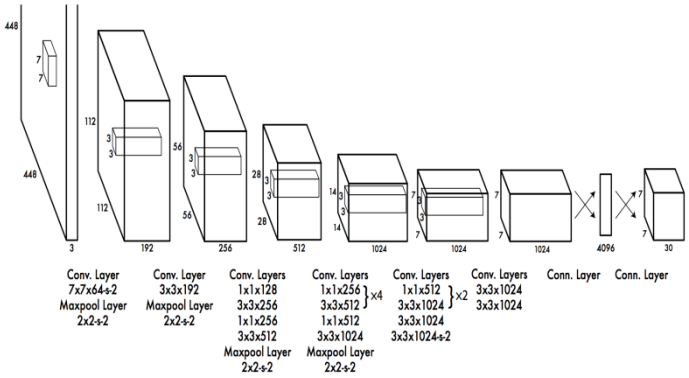



**Figure 1: YOLO Convolutional Neural Network**

**2.2 Implementation:**

Our team approached this problem in three descriptive phrases:
1)Object Detection
2)Image Conversion
3)Block Diagram

**2.2.1 Object Detection:**

In the first phase we perform Object Detection using Yolov3. We trained our model from a combination of exclusive manually annotated data and a pre existing weights file.The algorithm detects the bounding boxes of each license plate in each frame received from the camera.YOLO divides the input into n*n grids first.Once the threshold confidence is achieved in the detection then the frame is saved and we proceed to the next phase of Image conversion.The dataset used for training was annotated in a YOLOv3 friendly format as follows:

- <object-class> - integer number of object from 0 to (classes-1)
- <x> <y> <width> <height> - float values relative to width and height of image, it can be equal from (0.0 to 1.0]

**2.2.2 Image Conversion :**

For image conversion we've used the Optical Character Recognition approach while using Tesseract OCR.Tesseract OCR was released by HP and has a very high correct rate. To make the detection more accurate for the OCR we've applied certain pre processing steps before it such as:

**a) Grayscale:** OCR accuracy is increased drastically when preprocessing like grayscale is applied to the input image. Grayscale converts the diverse color values of the

ISSN: 2233-7857 IJFGCN

image into bitonal images that contain only shades of black and white.This also decreases the processing time and helps us in giving a quick, accurate real time output.

**b) Bilateral Filter:** Bilateral Filtering was used to remove the noise from edges in image smoothing the edges. Similar to gaussian blurring, bilateral filtering also uses a gaussian filter to find the gaussian weighted average in the neighborhood. However, it also takes pixel difference into account while blurring the nearby pixels.Thus, it ensures only those pixels with similar intensity to the central pixel are blurred, whereas the pixels with distinct pixel values are not blurred

### 2.2.3 Block Diagram:

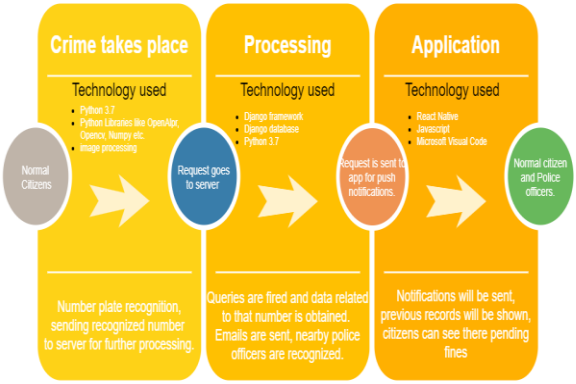



**Figure 2 – Block Diagram**

This project basically works in three stages. In the first stage, the camera captures the image, the processor attached to the camera(also known as a preprocessor) obtains the license plate and converts that image to text using OCR. After obtaining the license plate number, the processor sends its location and the number obtained to the server for the second stage. Now the server which is completely coded in the Django framework, receives that file sent by the preprocessor.

It checks license number in existing database, obtains the user information from the database and sends it to the third stage that is the end user App. A mobile application which will be present with each and every traffic policeman and normal citizens. Server will send this data to a policeman nearby that camera, that is the location where crime was committed, and this location was sent by the preprocessor. Also it sends mobile messages and E-mail to the person who has committed the crime. This was included because there may be users present who don't have the application installed in their mobile phones, so they will come to know about the crime committed and fine generated through E-mail and text message.

In the third stage, as soon as the app receives API requests from the server, it converts it into notification format and shows the notification to the policeman which are near the area of crime committed and to the person who has committed the crime. App was built using React-Native framework. In this type of framework the basic language is javascript and we build a

ISSN: 2233-7857 IJFGCN

cross platform mobile application using React library. This application also gives an extra edge to policemen, they can see the top criminals who haven't paid the fine in a long time. Hit list will be shown to policemen in which criminals are shown in descending order of fine unpaid. Normal citizens can see the amount of fine remaining that is to be paid. Main function of the app is to show the notification on the mobile to the policeman and the criminal while the crime is committed, to make the whole system real time.

### 2.3 System Workflow Diagram:

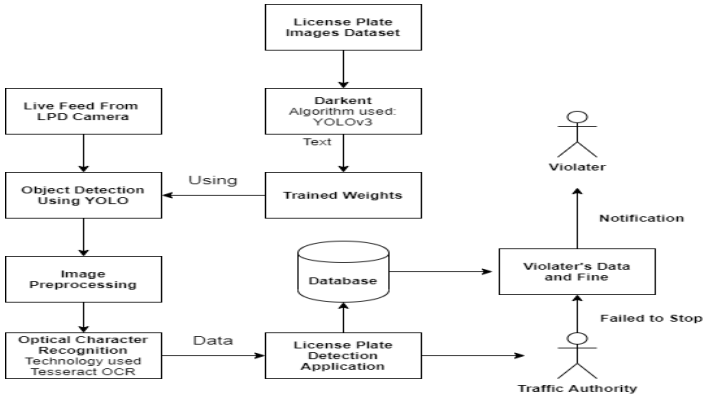
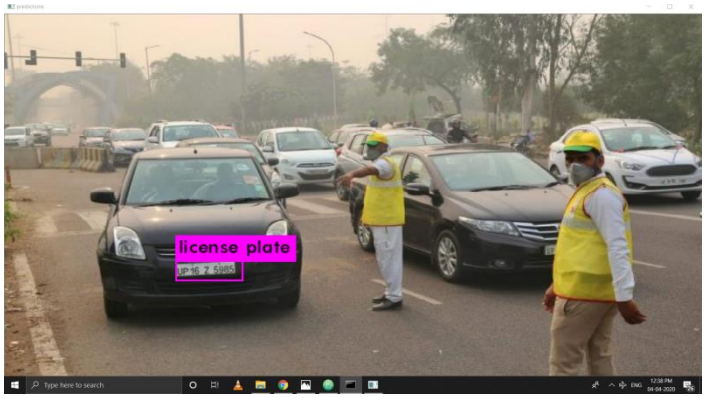



**Figure 3- System Workflow Diagram**

### 2.4 Results and Evaluation:



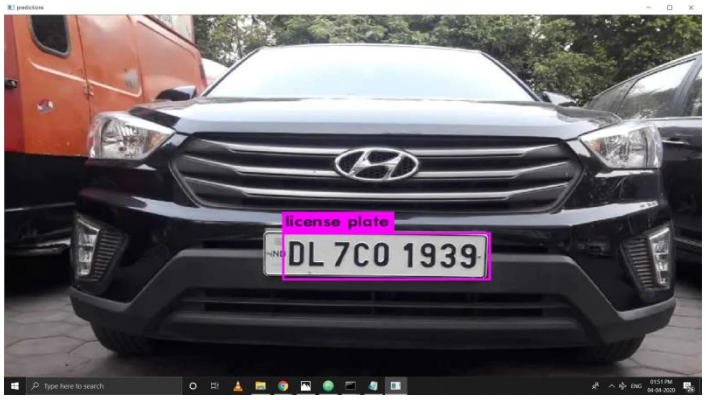



**Figure 4: License Plate Detection**

The generated images show the detection of the license plate in a clear pink bounding box. This result was gained due to the training of the tiny yolov3 weights with custom manually annotated data of local vehicles. The coordinates of this detection can be used to crop the license plate out from the image and send for conversion in Tesseract OCR.

| Dataset | Total Test Images | Correctly Identified |
|---|---|---|
| Vehicle Images | 10 | 9 |

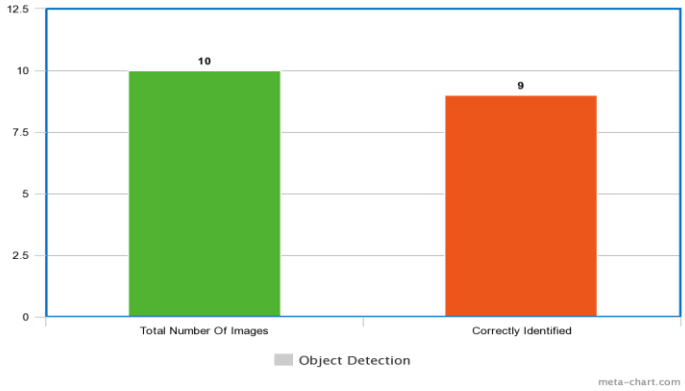



**Figure 5 : Bar Graph for test cases**

## 4. Conclusions

From this we conclude that our system has looked into details, the constraints of the existing system and we have managed to solve it to the utmost level. To decrease the number of traffic crimes and to ease the working of the traffic department our system provides a real time automated solution for the same. This model can be extended to detect car owners with pending challan at the traffic signal itself. With a real time working system a direct

ISSN: 2233-7857 IJFGCN

notification will be sent to all the authorities near the signal of commited crime and also to the person committing the crime. Our system provides the government with a feasible and quick processing solution to ease the working of the traffic department. Our system reduces the manual work that needs to be done as all the things are automated.

## 5. FUTURE SCOPE

**Face Detection:** In the future we hope to bring in face detection in our project. We would check the previous criminal records and any history of the person breaking the rule.

**Payment Gateway:** In the future, all the fines that have been registered to the person can be paid by the payment gateway that we chose to build.

## References

[1] A video-based traffic violation detection system-Publisher: IEEE - Year: 2013,Conference Location: Shenyang, China, Authors: Xiaoling Wang, Li-Min Meng , Biaobiao Zhang, Junjie Lu, K.-L. Du

[2] Fast Collective Activity Recognition Under Weak Supervision.-Publisher: IEEE - Year: 2015, Conference Location: China , Authors : Peizhen Zhang ; Yongyi Tang ; Jian-Fang Hu ; Wei-Shi Zheng

[3] Face detection and tracking: Using OpenCV.-Publisher: IEEE- Year : 2019, Conference Location : Coimbatore, India , Authors: Kruti Goyal ; Kartikey Agarwal ; Rishi Kumar

[4] Automatic Number Plate Recognition (ANPR)- Publisher: IEEE - Year: 2018 Conference Location : India , Authors: Chirag Patel, D Shah

[5] An efficient license plate recognition system using convolution neural networks. Publisher: IEEE- Year-2018, Conference Location: Chiba, Japan , Author : Cheng-Hung Lin , Yong-Sin Lin , Wei-Chen Liu

[6] Vehicle Activity analysis based on ANPR System - Publisher: IEEE- Year: 2014 , Conference Location: Milano, Italy , Authors : Yuyan Sun ; Xinyun Zhou ; Limin Sun ; Shuixian Chen

[7] Learning analysis of mobile JavaScript frameworks - Publisher: IEEE- Year: 2019 , Conference Location: Coimbra, Portugal, Portugal , Authors : Hugo Brito ; Álvaro Santos ; Jorge Bernardino ; Anabela Gomes

[8] Android Platform based determination of Fastest Cross-Platform framework- Publisher:IEEE Year-2018 , Conference Location : India , Authors : Faraz Ur Rehman Quazi, Nishant Sinha

[9] JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development Publisher:IEEE Year: 2018 , Conference Location: Caceres, Spain , Authors : Hugo Brito ; Anabela Gomes ; Álvaro Santos ; Jorge Bernardino

ISSN: 2233-7857 IJFGCN

[10] You Only Look Once: Unified, Real-Time Object Detection. Publisher: IEEE Year: 2016 , Conference Location: Las Vegas, NV, USA , Authors : Joseph Redmon ; Santosh Divvala ; Ross Girshick ; Ali Farhadi

[11] Object Detection Based on YOLO Network. Publisher: IEEE Year: 2018 , Conference Location : Chongqing, China, China , Authors : Chengji Liu ; Yufan Tao ; Jiawei Liang ; Kai Li ; Yihang Chen

[12] An Improved YOLOv3-based Neural Network for De-identification Technology. Publisher: IEEE Year:2019 , Conference Location : JeJu, Korea (South), Korea (South) , Authors : Ji-Hun Won ; Dong-Hyun Lee ; Kyung-Min Lee ; Chi-Ho Lin

[13] Automated Image Annotation based on YOLOv3. Publisher: IEEE Year: 2018 , Conference Location: Vilnius, Lithuania , Authors : Paulius Tumas ; Artūras Serackis

[14] Object Detection Based on Improved YOLOv3-tiny. Publisher: IEEE Year: 2019, Conference Location: Hangzhou, China, China , Authors : Hua Gong ; Hui Li ; Ke Xu ; Yong Zhang

[15] A Method of Optimizing Django Based on Greedy Strategy. Publisher: IEEE Year:2013 , Conference Location : Yangzhou, China , Authors : Jian Chou ; Lin Chen ; Hui Ding ; Jingxuan Tu ; Baowen Xu

[16] OpenCV compatible real time processor for background foreground identification. Publisher: IEEE Year: 2010 , Conference Location : Cairo, Egypt , Authors : M. Genovese ; E. Napoli ; N. Petra

[17] Documentation of Reactnative framework https://reactnative.dev/

[18] Documentation of React - A javaScript library https://reactjs.org/

[19] Documentation of Django https://docs.djangoproject.com/en/3.0/

[20] Documentation of YOLO https://yolocli.readthedocs.io/en/latest/

[21] Optical Character Recognition on images with colorful background, IEEE Year - 2018 , Conference Location : Berlin, Germany , Authors Matteo Brisinello ; Ratko Grbić ; Dejan Stefanovič ; Robert Pečkai-Kovač

ISSN: 2233-7857 IJFGCN