YOLO based License Plate Detection using CNN

Ankith C Kowshik[1], Anil Kumar B[1], B P Gagan Deepa[1], Anoop S[1], Asha T[2]
[1, 2] Department of Computer Science and Engineering, Bangalore Institute of Technology
Karnataka, India

ABSTRACT: *This research paper is based on you only look once (YOLO) object detection algorithm that detects and recognizes Indian license plates on conventional environments. Traffic control and management has been rising problems in urban conditions and several attempts have been already made to mitigate it using different methods. The safety as well as smooth flow of traffic is very important. Currently such a problem lacks an elegant solution. By using state-of-the-art object detection technique combined with the configured convolutional neural network (CNN) built using Tensor flow core open source libraries, real-time detection and recognition has been achieved. Although license plate detection as well as recognition had been widely adopted by enormous number of countries around the world for surveillance purposes it remains as a significant challenge in India, where the size of the number plates on Indian vehicles are not fixed and the CCTV used are not of high resolution. This paper tries to solve this problem by using a version of YOLOv2 (improved version of YOLO). The results of this achieved acceptable real time detection up to mean average of 9.8 FPS when running in Nvidia GT 940M graphics card.*

Keywords: Convolution, Neural Network, Object Detection, YOLO, Tesseract, License Plate Recognition

Received: 18 August 2019, Revised 29 November 2019, Accepted 6 December 2019

© 2020 DLINE. All Rights Reserved

1. Introduction

In recent years the number of vehicles on the road in India has increased exponentially. In 2017 the Indian Ministry of Statistics and Programme Implementation reported a total of 210023289 vehicles being registered in 2015 alone, the problems caused by the increase in number of vehicles on the road is only getting worse. Among these problems is the problem of traffic management. The pervasiveness of automobiles makes their management a tedious task and proves to be challenging for any manually operated system. This challenge has led to the development of automated systems. Automatic license plate detection can be used to monitor and alleviate the burden of traffic management.

The concept of automatic license plate detection is not a new one and has been frequently addressed in many studies. The solutions however are still not robust enough to handle real world scenarios. The current solutions often require ideal conditions such as optimal lighting and camera placement without any obstructions such as rain or snow.

In the recent years with the advancements made in computer vision technology though the development of deep learning techniques and open source libraries such as Tensor flow the problem of automatic license plate detection can be tackled in a novel way.

A convolutional neural network (CNN) is often used in the field of object detection. It is a type of deep neural network that is best suited for applications involving visual imagery. These design of these type of neural networks were inspired by the connectivity pattern between neurons in the visual cortex of animals. CNNs have the advantage over traditional image processing algorithms that they use relatively little pre-processing.

You Only Look Once (YOLO) is an innovative approach to object detection. Classifiers or localizers were reused in previous object detection methods. In the YOLO model only a single neural network is used over the whole image. The network then divides the images into different regions and their corresponding bounding boxes as well as their probabilities are predicted.

2. Related Work

The task of automatic license plate recognition was first tackled by the Police Scientific Development Branch in Britain in 1976. This system was unreliable due to a large number of factors such as poor camera quality and lack of sufficient processing capability. In the 1990s with developments in software and hardware technology this task was more frequently dealt with, having varying degrees of success.

The work of Shyang-Lih Chang [3] provided a major leap in the detection and extraction of license plate features. The proposed method included two modules. The first is used to locate the license plate and is based on fuzzy disciplines. The second module identifies the characters and other features of the license plate. This methodology provided an overall success rate of 93.7%. Yet real time detection was still a while away as it only worked when images were provided as input.

L. Xie [4] proposed a model that is able to perform the detection of the license plates of multiple license plates simultaneously. This model was based on the previously mentioned YOLO algorithm. The model was successfully able to detect license plates in real world traffic scenarios. The authors optimized the YOLO algorithm for the detection of license plates by adding in a rotational component to the images in order to compensate for images that were captured in different angles.

A complete end to end system of license plate detection was proposed Rayson Laroca [5] provided a solution for the detection as well as extraction of license plates in a real world scenario.

In order to achieve better results during the detection phase, Gee-Sern Hsu [6] split the process of license plate detection into three major categories based on different parameters. The three categories were access control, law enforcement and road patrol. Based on the three categories, images were then split into these categories based on parameters such as pan, tilt, distance, size, etc. This allowed the authors to more accurately train the neural networks based on the application of their license plate detection model.

3. Implementation

3.1. Dataset Information

One of our goals was to develop a model that could be used under a variety of conditions and even in sub optimal conditions. Thus multiple sources had to be considered. The data set was collected from three major sources. A considerable number of the images were obtained from Google images by means of a web scraper. The web scraper would download a specified number of images from Google images based on the keyword that is provided. This method has the drawback that much of the images got this way are not suitable for our objective and hence have to be filtered manually and the suitable images picked. Using this method, we were able to collect 476 images. The second source of our images was the AOLP data set provided by National Taiwan University of Science and Technology's Artificial Vision Lab. This data set consisted of previously filtered and sorted images based on their application specific to license plate detection in different scenarios. From this data set another 1147 images were procured for training. Our final source of images was manually captured images of vehicles on the street throughout various times of the day. 268 images were captured in this way. With a total of 1891 images we were able to build a comprehensive data set that contained images of vehicles captured under different lighting conditions and angles. The data set was then separated into two groups consisting of testing and training data. The split of the data set was random with 85% of the images being used to training and the rest for testing. This also gives us an opportunity to evaluate our methods against other competing models using the same training and testing data.

3.2. Convolutional Neural Networks

A Convolutional Neural Network(CNN) is arguably among the most popularly used artificial neural networks for image analysis. A CNN is a specialized neural network for pattern detection. A CNN operates much like any other neural network in that it is made up of neurons with learn-able weights and biases. Each neuron receives multiple inputs and then takes the weighted sum over them. This result is then passed onto an activation function that results with an output. Just like in any neural network a loss function is applied over the whole network.

A CNN differs from a standard multi-layer perceptron through the use of multiple hidden layers called convolutional layers. These convolutional layers operate over volumes such as matrices of data from multi - channeled images, etc., where as a standard neural network would take a vector as input. These convolutional layers perform the operation of convolution on the input matrix, thus acting as a filter to reduce the image into only the features that are deemed important. Initially the filters are assigned random numbers but through training the values are learned. Through the use of multiple convolutional layers more complex features can be extracted from the image.

One of the consequences of having a convolutional layer is that a sparse matrix is generated as output. This a pooling layer and activation function often accompany a convolutional layer to reduce the number of parameters in the whole system and make computation more efficient. The final fully connected layer of the network is the layer where the classification is made.

3.3. You Only Look Once (YOLO)

You Only Look Once (YOLO) is a novel approach to object detection. Previous works on object detection modify the previously used classifiers to perform the detection. YOLO on the other hand takes a different approach. The problem is tackled by using regression to spatially separate bounding boxes and their corresponding class probabilities. This enables us to make predictions on the class as well their corresponding probabilities in a single pass.

The YOLO system splits the input image up into a $M X M$ grid. Each cell of the grid responsible for detecting the object is the grid cell into which the center of the object falls. N bounding boxes are predicted by each grid cell as well as a confidence score for each of the predictions. The confidence score represents the confidence that the box contains an object accuracy of the prediction that the model has made. The bounding box is defined by 4 values: x, y, w, h which are the center coordinates of the box relative to the boundaries of the grid cell and the width and height of the object respectively.

The grid cells also predict the conditional class probability given that the grid has an object in it. The value of the conditional class probability is given by the following equation.

$$Pr\,(Class_i \mid Object) * Pr\,(Object) * IOU_{pred}^{truth} = Pr\,(Class_i) * IOU_{pred}^{truth} \quad (1)$$

The above score encodes both the probability of that class which has a presence in that box as well as how well the predicted box encloses the object. Finally, the class with the scores above the set threshold are considered and the results displayed.

Using YOLO along with our convolutional neural network we can train the network to simultaneously detect and predict the location of the detected license plates. The frame containing the plate is then extracted from the video and the plate is cropped out and stored. Traditional optical character recognition techniques are used to read the information on the license plate.

3.4. Optical Character Recognition

The tesseract optical character recognition (OCR) engine is a popular open source (OCR) engine and our choice for extraction of details from the detected license plate. Tesseract uses a two step approach. The first stage is the data stage for character recognition and the second stage fulfills any letters it wasn't insured in by letters that can match the word or sentence context. The initial step of Tesseract is an analysis of the connected components. The outlines of these components are stored and nested into groups called blobs. Lines of text are then generated. These lines are then broken down into words and if the analysis of words cannot be done directly, the words are broken down into their corresponding characters and analyzed.

We use the tesseract OCR engine to detect the characters on the license plate that was extracted from each video frame. The output returned by tesseract are then stored in a text file along with the image in a specified folder.

4. Results

The task of license plate detection in a real world environment requires for fast processing of the input data which in our case was a 720p video feed and high accuracy. The following results is a comparison of our model with similar competing models that are commonly used for the application of license plate detection which have been given the same input data. The threshold for each of the models was set at 60% to be considered for further processing.

4.1. Performance

All our tests were run on a computer that had been configured with an Nvidia GT 940M graphics card, Intel Core i7 8550U and 8GB of RAM. The performance was evaluated by calculating the average frames per second (FPS) achieved for a given particular set of videos. The set of videos consisted of five separate videos captured in a real traffic environment for a period of ten minutes each. The set of videos was chosen to include a variety of scenarios. The scenarios even cover sub optimal conditions such as cloudy days and poor camera placement. As per our testing, we were able to achieve an average frame rate of 9.8 FPS while other competing methodologies trailed behind by approximately 25 percent. The following graph provides a detailed description of the FPS achieved in each of the videos has been shown in figure 1.

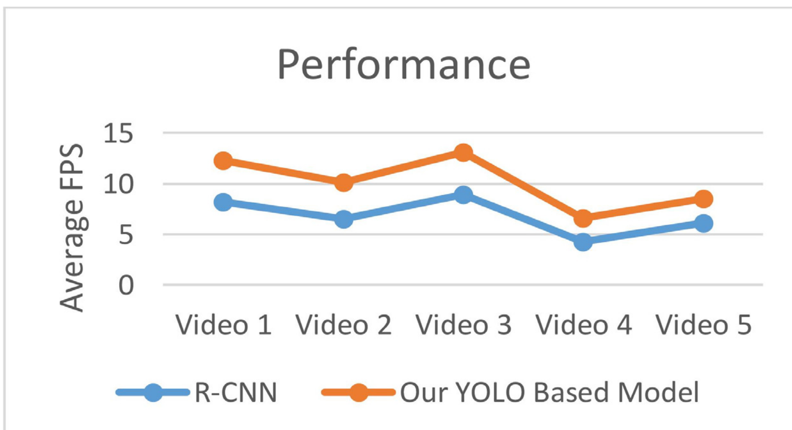

Figure 1. Performance Results

4.2. Precision

A confusion matrix is a specific table layout that allows for the evaluation of the performance of a supervised learning algorithm.

The precision of a model tells us how often the predicted true value is actually true. It is given by the formula:

$$Precision = (True\ Positive) / (True\ Positive + False\ Positive) \qquad (2)$$

The precision values for each of the evaluated models has been described in figure 2.

Figure 3 is a sample video frame from our model that is showing the confidence rating of the license plate prediction made.

5. Conclusion

The modern techniques of conventional neural networks have been used to build a model that is both efficient and effective to address traffic management. Every functionality of the existing systems has been used and in many instances a new approach has been discovered. Unlike most other solutions using object detection, we provide an end to end system which includes the detection as well as extraction of the relevant information from the license plates. From the experiment, we observe that our model has the highest precision of 92.2%, and offers better performance as compared to similar existing models. Although it must

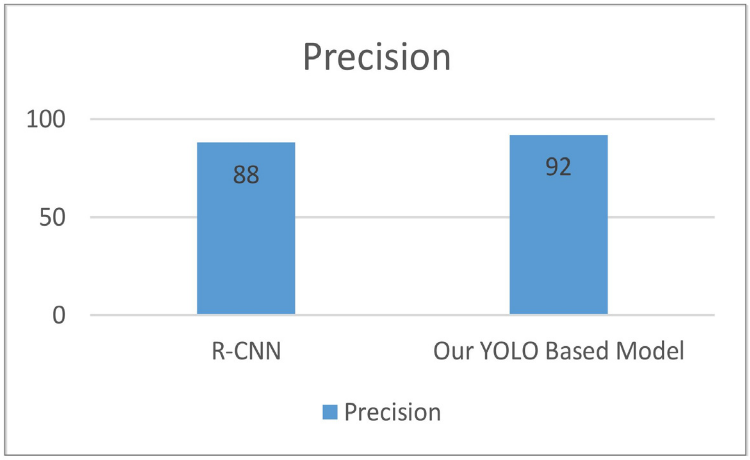

Figure 2. Accuracy

Figure 3. Detection Display

be noted that due to the inherit design of the YOLO algorithm our model fails to perform well in scenarios where the size of the license plates may be too small and clustered together. In such situations we are also unable to extract the features of the license plate.

References

[1] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition, *In*: Proceedings of the IEEE, 86 (11) 2278-2324.

[2] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, (2016), p. 779-788.

[3] Chang, Shyang-Lih., Chen, Li-Shien., Chung, Yun-Chung., Chen, Sei-Wan. (2004). Automatic license plate recognition, *In*: IEEE Transactions on Intelligent Transportation Systems, 5 (1) 42-53.

[4] Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S. (2018). A New CNN-Based Method for Multi-Directional Car License Plate Detection, *In*: IEEE Transactions on Intelligent Transportation Systems, 19 (2) 507-517.

[5] Laroca, R. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector, 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, p. 1-10.

[6] Hsu, G., Chen, J., Chung, Y. (2013). Application-Oriented License Plate Recognition, *In*: IEEE Transactions on Vehicular Technology, 62 (2) 552-561.

[7] Smith, R. (2007). An Overview of the Tesseract OCR Engine, *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Parana, (2007), p. 629-633

[8] Palekar, R. R., Parab, S. U., Parikh, D. P., Kamble, V. N. (2017). Real time license plate detection using openCV and tesseract, 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, p. 2111-2115