2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)
Manipal University Jaipur, Sep 28-29, 2019

Anonymous Vehicle Detection for Secure Campuses: A Framework for License Plate Recognition using Deep Learning

Crystal Dias
Department of Computer Science and Engineering
Manipal University Jaipur
Rajasthan, India
crystaldias0504@gmail.com

Astha Jagetiya
Department of Computer Science and Engineering
Manipal University Jaipur
Rajasthan, India
astha.jagetiya@gmail.com

Sandeep Chaurasia
Department of Computer Science and Engineering
Manipal University Jaipur
Rajasthan, India
sandeep.chaurasia@jaipur.manipal.edu

***Abstract*— Automatic license plate recognition is being widely used for numerous applications since its inception. The ability to procure license plate numbers accurately has been beneficial in maintaining traffic rules, parking enforcement, and security. In this paper, we have discussed the results of using ALPR for recognition of anonymous vehicles entering our university campus. We used deep learning for license plate localization and Tesseract OCR for license plate recognition. By doing so we could read the license plates of vehicles entering a particular campus and verify if the vehicle is authorized by comparing it with a predefined list of authorized vehicles. To efficiently extract these number plates we have trained our model using Faster RCNN and tuned it to get the best output. The results of which have been discussed in this paper. Further, the image processing techniques used for preprocessing the identified number plate have been mentioned here. For character segmentation and character recognition, we have used tesseract. While training our model for number plate extraction the minimum loss obtained was 0.011 with RMSprop optimizer at initial learning rate 0.002.**

Keywords*— *License plate recognition, OCR, Faster RCNN, Tesseract, image processing, character segmentation, character recognition.

I. INTRODUCTION

The ability of computers to process images and other visual data to something meaningful has aided many domains including medical, security, monitoring and engineering [1]. With an increasing number of automobiles, maintaining security and traffic rules with the help of manpower is tedious. A camera captures the image of the vehicle's number plate. This image is then sent for further processing. The output of this image is the character on a number plate in textual form [1]. But now with the help of deep learning and OCR for number plate recognition, this task has become much easier. A standard ANPR project consists of three steps: Number plate localization, character segmentation, and character recognition. The trained RCNN model is applied to a webcam where it localizes number plates of vehicles. Once the plate has been detected the image of this number plate is then processed using suitable image processing filters to prepare it for character segmentation. Each character in this processed image is then separately recognized as an alphanumeric character.

II. RELATED WORK

The main process followed for Automatic Number Plate Recognition includes majorly of 3 process – Plate localization, Character segmentation, and Character recognition. Over the years different approaches have been used to follow up this process with varying results. Some of these researches are as follows:

For plate localization, the authors of [2], have used an algorithm based on vertical edge recognition and colour alteration in grey scale image. They further used fuzzy logic and a region growth algorithm for character segmentation. Ultimately, 85% of correctness was obtained.

In [3], creators have utilized strategies for dark degree fortifying and sifting treating to help differentiate on a tag and to diminish commotion. The OTSU calculation was then used to build up the worldwide thresholding esteem and concentrate edges and make a second situating on the tag with projection technique bringing about a paired tag picture making it simple for division and acknowledgment

Authors of paper [4], have proposed a method to identify the spatial and temporal features of vehicle trajectory and have used these detected plates via ANPR to analyse vehicle activity patterns. Later to evaluate the accuracy performance of classification a sample of 20,000 plates and a mean percentage error of 4.76% was obtained.

III. EXPERIMENTAL SETUP

A. Framework

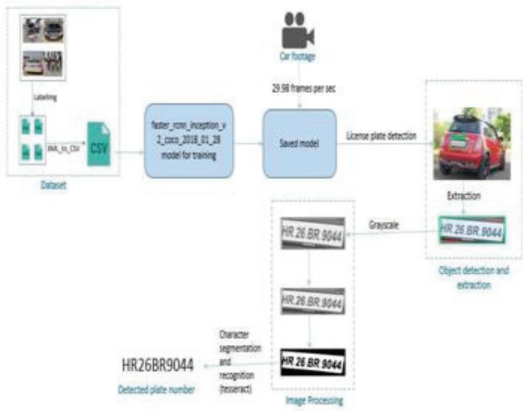

Fig. 1. The framework of proposed work.

For making the training dataset we labelled the number plates of cars in raw images and saved it in XML format, consequently merging the data of these XML files into a CSV file. After that with the help of transfer learning, we used a pre-trained model to train on our dataset. This model is saved and is used to detect license plate from vehicles in video footage. The detected license plate is then extracted and undergoes various image processing operations. Finally, Tesseract OCR is used to convert the text on the license plate into a string format. This framework is depicted in Fig. 1.

B. Dataset

Keeping in mind the unique characteristics of Indian number plates, we have created the dataset with images of Indian vehicles. It consists of total of 323 images (including both four-wheeler and two-wheeler vehicles), which is further divided into training (261 images) and testing (62 images) datasets. Table I. Shows the distribution of the images in the two datasets:

TABLE I. DISTRIBUTION OF IMAGES

	Training dataset	Test dataset
Four-wheeler	153	34
Two-wheeler	108	28
Total	261	62

The dimension of the images varies, with a minimum being 518×864 pixels and maximum being 570×960 pixels. The larger the image the longer it takes to train the classifier.

For each image, we labelled the area of number plate as 'Number_plate' using the application named LabelImg, which stores the information of the labelled image in XML file, in Pascal VOC format. Then the data of these XML files are stored in .csv files ('train_label.csv' and 'test_label.csv'), so that they can be further used for training and testing the models. Fig.2 shows the images collected for the dataset

Fig. 2. Dataset images.

IV. APPROACH

After creating the dataset, we annotated these images using the software LabelImg. Using the application we were able to label the number plates on the vehicles and stored the co-ordinates of the annotations in XML format for training. For training, we used faster R-CNN.
We developed the automatic license plate recognition system in following main steps:

a) Training using Faster R-CNN
b) Image processing
c) Tesseract OCR [7]

A. Faster R-CNN [6]

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. [6] By using R-CNN we were able to work on a nearly cost-free region proposal that uses full-image Convolution features, as proposed by authors in [6]. For this project we have used Tensorflow Object detection API to train Faster R-CNN pretrained model (Faster R-CNN inception V2 model trained on COCO dataset).

Faster R-CNN comprises of two parts. In the first part convolution networks propose the regions and in the second part a detector uses the proposed regions. The entire system is a complete network for object detection. A feature map is created with the images provided as input to the convolution network. Then, a separate network is used on this feature map to predict region proposals, instead of using selective search algorithm. A RoI layer is used to reshape these region proposals. These images are then classified and the offset value of the bounding boxes is predicted. [6]

The input to a Region Proposal Network (RPN) may be a raw image (of any size), and therefore the output may be a set of rectangular object proposals. This method is modelled with a totally convolutional network. to get region proposals, a tiny low network is fell over the convolutional feature map output by the last shared convolutional layer between the 2 modules of the system. This little network takes as input within the variety of AN n × n spatial window of the convolutional feature map, later these slippery windows are mapped to lower dimensional options. The features generated are fed into two sibling fully connected layers (box-regression layer and box-classification layer). Fig.3 shows how a Faster R-CNN works for object detection (detecting number plates in our case).[6] Fig. 4 shows object detection carried out on a raw image.

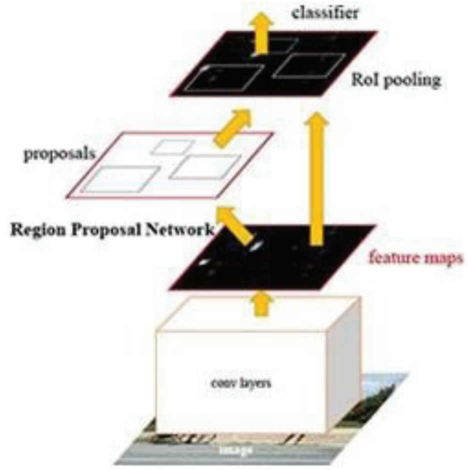

Fig. 3. Working of Faster R-CNN [6]

B. Image Processing

The number plate detected in the new image undergoes image processing so that the characters are identified properly. First of all, the combination of top-hat and black- hat morphological transformations is used to get the image of maximum contrast. The top-hat filter is used to enhance bright objects of interest in a dark background, and the black-hat operation is used to do the opposite, i.e., enhance dark objects of interest in a bright background. The top-hat image is added to the grayscale image and then the black-hat image is subtracted from it. As a result, the image with high contrast is obtained. Second, Gaussian smoothing filter and medianBlur filter is used to blur the image and remove noise. This finally gives us the image that can be used for character segmentation and recognition. Fig.5 shows the results after each filter is used.

C. *Tesseract OCR*

After doing image processing on the extracted number plate, we have used tesseract, an OCR engine, to extract text from the number plate. The processing of image follows a systematic pipeline. Firstly a connected component analysis is carried out in which outlines of the components are stored. The outlines are then gathered together into Blobs. Following which the blobs are organized into text lines, which are analyzed for fixed pitch or proportional test. According to the kind of character spacing, text lines are broken into words. Text with fixed pitch is chopped immediately by character cells. Proportional text is broken into words using definite spaces and fuzzy spaces. Tesseract was the first to handle black on white text efficiently. [7] Fig. 6 shows the output after using Tesseract OCR in string format

Fig. 4. Object detection on a raw image

Fig. 5. Results of the filters used for image processing. 1.Gray-Scale conversion 2.Top-hat and black-hat morphology 3. Gaussian smoothing and medianBlurr filter.

Fig. 6. After applying Tesseract OCR

V. RESULT

RMSprop (Root mean square prop): divides the learning rate by an exponentially decaying average of squared gradients. The momentum term is set to 0.9.

Momentum optimizer: prevents oscillations so that the stochastic gradient decent tends to travel in the same direction to arrive at the local minima.

Adam: running averages of both the gradient and the second moments of the gradient are used.

While training our model we observed the losses for different optimizers and different learning rates for each optimizer. Table II. Shows the losses obtained when different learning rates were applied to the model on using the momentum optimizer (which is a flexible version of stochastic gradient descent (SGD) [5]), RMSprop and Adam optimizer. Fig. 7 Shows the change in loss over 12,000 steps for different learning rates.

TABLE II. LOSSES OBTAINED WHILE VARYING THE LEARNING RATE FOR VARIOUS OPTIMIZERS

Learning rate	Momentum	RMSprop	Adams
0.0002	0.041	0.016	0.043
0.0001	0.050	0.036	0.020
0.00002	0.028	0.120	0.019
0.0003	0.015	0.012	0.066
0.0004	0.028	0.020	0.099
0.002	0.027	0.011	0.551

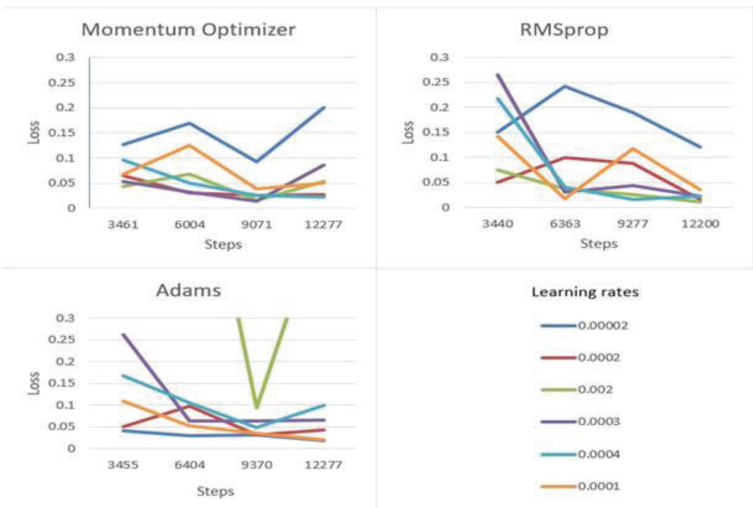

Fig. 7. Loss through steps over different learning rates

VI. CONCLUSION

RMSprop gave the best result when the learning rate was set to 0.002. Applying this model, we could localize the license plate entering a given campus. Following which, with the help of above-mentioned filters we obtained a high contrast and noise-free image of the identified number plate and lastly with the help of Tesseract OCR we were able to extract the alpha-numeric characters from the plate in string format.

In this proposed work change in light intensity of surroundings on number plates would make character recognition difficult. In future, changes could be made to the OCR to help accommodate light intensity on the number plate.

This framework can further be used for smart traffic: identifying the number plates of vehicles breaking traffic rules or for optimized parking: keeping an account of vehicles in parking.

REFERENCES

[1] Nuzulha Khilwani Ibrahim, Emaliana Kasmuri, Norazira A Jalil, Mohd Adili Norasikin, Sazilah Salam, Mohamad Riduwan Md Nawawi, License plate recognition (LPR) : A review with experiments for Malaysia case study. 2014.

[2] Carlos A. B. Mello, Diogo C. Costa, A complete system for vehicle license plate recognition, 2009.

[3] Chaofeng Lan, Fengchen Li, Yingjian Jin, Xuemei Sui, Shouqiang Kang, Liping Zhang, Research on the license plate recognition based on image processing, 2015.

[4] Yuyan Sun, Xinyun Zhou, Limin Sun, Shuixian Chen, Vehicle activity analysis based on ANPR system, 2014.

[5] James Sutton, Deep Dive into Object Detection with Open Images, using Tensorflow, Algorithmia Blog, October 2017.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016.

[7] Ray Smith, An Overview of the Tesseract OCR Engine, Google Inc. 2007.

[8] Lisheng Jin, Huacai Xian, Jing Bie, Yuqin Sun, Haijing Hou, and Qingning Niu. License Plate Recognition Algorithm for Passenger Cars in Chinese Residential Areas. 2012.

[9] Satadal Saha, Subhadip Basu, Mita Nasipuri. Automatic Localization and Recognition of License Plate Characters for Indian Vehicles. 2011.

[10] Sebastian Ruder, An overview of gradient descent optimization algorithms, http://ruder.io/optimizing-gradient-descent/, 2016.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, Rethinking the Inception Architecture for Computer Vision, 2015.

[12] Ross Girshick, Fast R-CNN, 27 September 2015.

[13] Hogne Jorgensen, Automatic License Plate Recognition using Deep Learning Techniques.

[14] Zied Selmi, Mohamed Ben Halima, Adel M. Alimi, Deep Learning System for Automatic License Plate Detection and Recognition, 2017.