Automatic Number Plate Recognition Using Deep Learning

K. Prema[a], P. Arivubrakan[b], and V. Suganya[c]

[a,b,c]
Assistant Professor, Department of CSE, Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science and Technology

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Automatic Number Plate Recognition has been a topic due to many practical applications. However, lots of the solutions which can be current still not robust in real-world situations, commonly based on many constraints. ANPR systems typically have three stages: Automatic Number Plate detection, character segmentation and character recognition. It need higher precision or almost perfection, since its failed to identify the NP may possibly lead to a failure in the stages that are next. Many approaches search first for the vehicle and then its NP in order to reduce processing time and eliminate positives that are false. Although ANPR has been frequently addressed in the literature, many studies and solu-tions are still not robust enough on real-world scenarios. These systems where dealing with images, the accuracy depends on many parameters like camera, lighting conditions etc. Many computer vision tasks accuracy always depends on feeding huge number of training data or called as dataset. As of these limitations with computer vision, Deep Learning arise.The accuracy and performance of any applications like ANPR using DL gives descent output but still there is a demand of proper dataset annotation.

Keywords: Data Collection,Labelling Images,Convolutional Neural Network, Tensorflow,Python.

1. Introduction

Automatic Number Plate Recognition (ANPR) is commonly used in many countries for many applications like Ticketless parking fee management, car theft prevention etc. ANPR systems consists of three stages: Number Plate detection, character segmentation and character recognition. The primary stages needs greater accuracy as it depends on whole system accuracy and efficiency. A lot of approaches search first for the vehicle and then its NumberPlate in order to reduce processing time and eliminate false positives. Although ANPR has been frequently addressed in the literature, many studies and solu-tions are still not robust enough on real-world scenarios. These systems where dealing with images, the accuracy depends on many parameters like camera, lighting conditions etc. Many computer vision tasks accuracy always depends on feeding huge number of training data or called as dataset. As of these limitations with computer vision, Deep Learning arise.The accuracy and performance of any applications like ANPR using DL gives descent output but still there is a demand of proper dataset annotation. The DL performance depends on training data. Larger the training data, larger the machine learn patterns. So, here collected large number of images from different types if dataset to detect the License plate in a given image. Depends on the angle the images took, there is no sure of visibility of license plate properly in dataset which affects directly to the accuracy. Secondly, the aspect ratio for all the LP is not same. Together gives numerous amount of false positives out of the model. YOLOv3 is a deep learning model that uses convolutional layers and 5maxpooling layers. Dealing with video processing, each frame should be evaluated independently leads to temporal redundancy.

2 Scope of number Plate Recognition

Automatic Number Plate Recognition (ANPR) systems is becoming greater importance in road traffic analysis and in many applications including police forces. Many ANPR systems are deployed in different fields like traffic control system, highway tolling systems, parking areas controlling, control of the access to cities centers, etc. All these existing systems are deployed using basic infrared cameras in order to see the License plate and a software to detect the plate and find the license number using OCR techniques and image processing techniques. This proposes YOLO based end to end solution provides in a single prototype.

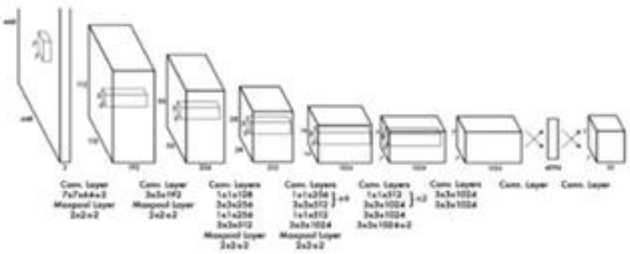

Fig. 1. YOLO Model Using Convolutional Neural Network

YOLO is DL model is used popularly for object detection. YOLO is a single shot detection which uses single neural network for completing the task. YOLO divides images into regions and predict bounding box which has more accuracy of object to be identified. Each region has its own probability. Compared with other DL models which uses more recursive techniques leads to more processing time, YOLO only look once at the whole image. Hence YOLO is fast. Lot of study required to understand the configuration of YOLO model and its weights. Everything is implemented using tensorflow framework. Input images is converted into grey scale. To match the model, the image is resized into 448 into 448. In python opencv function loads image in BGR format, So while displaying it again needs to convert back to RGB using following equation:

RGB_img= cv2.cvtcolor (BGR_img,CV2.COLOR, BGR2RGB)

This implementation can able to detect LP and givens decent accuracy. YOLO uses other layers named by keras layers. In the configuration file, the model is not sequential.

3 Modelling and Design Phase

Firstly, we need to prepare a data set of the vehicles with number plate and later annotate them with LabelIMG application. So, after finishing the annotation part, we need to train the model according to our requirements. After finishing the training, we need to check the model whether our input features are mapping with the new input images or not. Once after completion of this training, testing will come into the picture and check the prediction score and accuracy of the input images we parsed. So, when this was perfectly done, the model will start detecting number plate and using Optical Character Recognition (OCR) technique, it will perform char-acter segmentation in order to recognize the number plate. So, after completion of number plate recognition, it will check with the database whether it is a authorized vehicle or not.

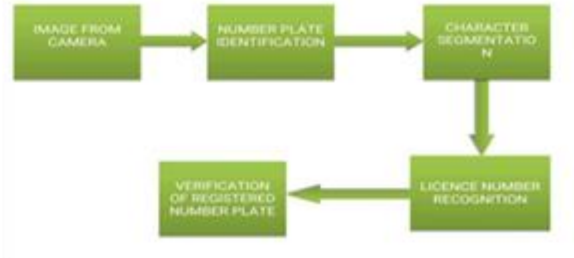

Fig. 2. General Architecure

The dataset contains images taken from different types of datasets available in the internet. Some of the datasets are videos where images extracted by taking frames, most of the images have proper LP visibility. Stabilization method is used here. The images were taken with two cameras. The cameras used were: I-Phone 11 pro max and OnePlus 7 pro. So as the cameras are different all the images resolution are different. This is because of many parameters like autofocus, bit rate, focal length and optical image stabilization. All license plates are not same even in same country. It differs from place to place, vehicle type, brand type etc. Figure 5.1 shows LPs found in the training data.For training, images need to be annotated. The best tool for doing this job is labelImg. Labelled data contains data of images with their corresponding bounding box coordinates and class number. That is, the bottom left and top right (x,y) coordinates and the class. LabelImg saves a .txt file containing the information above mentioned for each image.

To train custom data, YOLO custom configuration needs to be implemented. In the Darknet program, many configuration files are needed in order to setup the whole neural network structure. In the Number Plate detection phase, there was only one class which is NumberPlate itself. The path indicated the number of images processed at each training step, which was further rectified by further divisions according to the computer performance.The reason for testing is to find mistakes. Testing is the way toward attempting to find each possible shortcoming or shortcoming in a work item. It gives an approach to check the usefulness of segments, sub-gatherings, congregations as well as a completed item. It is the way toward practicing programming with the goal of guaranteeing that the Software framework lives up to its necessities and client desires and doesn't flop in an inadmissible way.

The reason for testing is to find mistakes. Testing is the way toward attempting to find each possible shortcoming or shortcoming in a work item. It gives an approach to check the usefulness of segments, sub-gatherings, congregations as well as a completed item. It is the way toward practicing programming with the goal of guaranteeing that the Software framework lives up to its necessities and client desires and doesn't flop in an inadmissible way. Training is starting using darknet command with all the configuration files mentioned above with pertained weights. Pretrained weights are initial weights given as starting point in the training. After training the object detection test is needed whether LP is detecting properly or not. The Bounding box for prediction of object is given.

A methodology for framework testing incorporates frame-work experiments and structure systems into an all-around arranged arrangement of steps that outcomes in the effective development of graphical portrayal. The testing procedure must co work test arranging, experiment configuration, test execution, and the resultant in-formation assortment and assessment. Testing speaks to an intriguing peculiarity for the examination framework.

Consequently, a progression of testing is performed for the proposed framework before the framework is prepared for client acknowledgment testing.

Fig. 3. Character Recognition using YOLO

Number plate character segmentation have been done us-ing YOLO is presented at Figure 3. The comparison have been performed to predict the accuracy between YOLO and OpenCV. Here, we conclude that accuracy of YOLO is 99% and accuracy of OpenCV is 93% .

Fig. 4. Accuracy achieved using YOLO

Figure 4 describes that our model is massively successful with both vehicle detection and character recognition. For NP detection, the accuracy is above 90% confidence, and our training showed a final loss of 0.287. For character recognition, our model had a final test loss of 0.225 and claimed an accuracy of 93%. This model worked well for most fonts that we tested, but it is apparent that further work is required to improve it. Envelope-transient simulations show the modeled output current, and the model enhanced capability to reproduce both the current discontinuity and the slow transient due to traps emission. The emission time constants are however not very accurate, and this can be due by the strong non-linearity of traps time constants versus bias and temperature, as explained previously.

To increase the accuracy for the character segmentation, the proposed system given decent accuracy compare to existing system. For the fastest output processing speed, we need high end GPU system.

4 Conclusion

In summary, we were able to understand the architecture of the convolutional neural network (CNN) and its training procedure and understand the whole procedure of YOLO custom model. Furthermore, we were able to increase the prediction precision by training a small dataset. The aspiration is detecting more foreign and Indian license plates with the trained international dataset.

5 Future Enhancements

It is very difficult to setup this advance technology of ANPR systems because of many reasons. All these limitations can be solved by developing a software which will run on low versions of PC, angels, speed and size in which the plate would be displaying on the camera field of view should be considered. The real time ANPR where it can be installed over vehicles can be useful for the police forces.

References

1. Masood, S. Z., Shu, G., Dehghan, A., Ortiz, E. G. (2017, March 28). License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks. Retrieved November 29, 2017, from https://arxiv.org/pdf/1703.07330.pdf.
2. Zhou, Y., Nejati, H., Do, T., Cheung, N., Cheah, L. (2016, Au-gust 7). Image- based Vehicle Analysis using Deep Neural Net-work: A Systematic Study. Re- trieved October 15, 2017, from https://arxiv.org/pdf/1601.01145.pdf.
3. Krause, J. Deng, J. Stark, M. and Li, F. F. (n.d.). Collecting a Large-Scale Dataset of Fine-Grained Cars, from http://ai.stanford.edu/ jkrause/papers/fgvc13.pdf.
4. F. Oladeji, “Developing a License Plate Recognition System with Ma- chine Learning in Python,” Devcenter, 03-Aug-2017. [Online]. Available:https://blog.devcenter.co/developing-a-license-platerecognition-system-with-machine-learning-in-python-787833569ccd. [Accessed: 01-Nov-2017].
5. YOLOv3: An Incremental Improvement Joseph Redmon Ali Farhadi, University of Washington.