Different Methodologies for Indian License Plate Detection

Atharvraj Patil[1], Jayesh Ingale[2]

[1]Department of Electrical Engineering, College of Engineering, Pune, India
[2]Department of Electrical Engineering, College of Engineering, Pune, India

---

Abstract - *With the aid of a picture produced by security cameras, automatic license plate detection systems (ALPDS) have the potential to automatically track and identify the vehicle by capturing and recognizing the number plate of any vehicle. It can be used for a variety of practical purposes, like recording vehicle numbers at toll gates, tracking down stolen vehicles using CCTVs, and more. Detecting the license plate from the input image is the primary stage in automatic license plate recognition systems, which are frequently employed for purposes like effective parking management, updating law enforcement, systematic toll way management on roads, etc. There are numerous methods for detecting license plates. This study focuses on various techniques for reading license plates from input images of vehicles. Here, three key methods—edge detection, object detection with YOLOv4 (You Only Look Once) and WPOD-NET (Warped Planar Object Detection Network)—are covered. This paper also provides a description of each method's benefits and downsides.*

***Key Words*: Edge detection, License plate detection, WPOD-NET, YOLOv4.**

1. INTRODUCTION

Massive growth of the vehicular sector has raised the issues of parking management, traffic control, tracking stolen vehicles, etc. Also, tracking individual vehicles has become a tedious task for all organizations. To avoid issues and for effective parking management, the Automatic Number Plate Recognition Systems (ANPRS) are designed. License plate recognition is the base of ANPRS. It has a wide range of applications which uses extracted license plate details to create automated solutions for various problems like access control, tolling, border control, traffic control, finding stolen cars, efficient parking management and so on. The primary step in ANPRS is license plate detection and extraction.

This paper discusses different methods of license plate extraction where the vehicle's static image is fed to the system which further processes the image to extract the license plate. Each of the techniques discussed in this paper has various advantages and drawbacks. Like, a method may require low processing power, but it fails for different unconstrained scenarios. Another method may be efficient for real time applications but then it requires high processing power leading to limitations of hardware. Thus, further sections describe in detail about various methods of license plate detection.

2. LITERATURE SURVEY

In [1], the Automatic Number Plate Recognition System (ANPRS) is divided into 3 main steps namely,

- Vehicle Detection
- License Plate Detection
- Optical Character Recognition (OCR).

There were various algorithms available for the process of object detection like Fast Region Based Convolutional Neural Network (RCNN), Single shot multibox detector (SSD), You Only Look Once (YOLO), etc. After comparing various algorithms for the process of vehicle detection they found the YOLO algorithm as the most suited for this work due to its advantage of quick real-time object detection and used it for vehicle detection, Warped Planar Object Detection Network (WPOD-NET), and Optical Character Recognition (OCR) for License Plate (LP) detection. They have employed various datasets for training this network. These datasets consider various factors, such as the LP angle (frontal and tilted), the distance from vehicles to the camera (near, intermediate, and far), and the area where the photographs were taken. Finally, after training the model on all four datasets average accuracy obtained was 89.33%. The earlier YOLOv2 vehicle detection and OCR-NET were developed and put into use using the DarkNet framework, but the proposed WPOD-NET is implemented using the TensorFlow framework. In [4], this project report presents development on new approaches for the extraction of license plates. Their proposed algorithm was based on video acquisition, license plate region extraction, plate character segmentation, and character recognition. This project presents a simple straightforward license plate extraction technique. The method's four main stages—converting RGB images to grayscale, applying Gaussian blur, performing morphological operations, and determining the license plate's precise location—are

based on the Edge Detection algorithm. It uses a character segmentation algorithm to detect the characters on the number plate. Templates are created for each of the alphanumeric characters (from A-Z and 0-9). The mean squared error method is used for image similarity. Each segmented character of the license plate is compared with all the standard template characters and the error is recorded and the final output is displayed on the monitor.
The proposed system has the following limitations like the camera should be of good quality to extract correct characters from license plates. It should have proper lighting. It doesn't work in different illumination conditions and although accuracy is high, mean squared error leads to low computational results.

The implementation of the proposed system can be extended for the recognition of number plates of multiple vehicles in a single image frame. User-friendly android applications can be developed for traffic surveillance management systems. Also, character recognition can be done using various deep learning algorithms as they yield more accuracy. (Graphical Processing Units) GPUs can be used to achieve more performance in terms of computational time.

3. DIFFERENT METHODOLOGIES

3.1 Edge Detection

Digital images have edges, which are notable local changes in intensity. An edge is a collection of linked pixels that identifies the boundary between two distinct sections. Edge detection is the method of image pre-processing used to recognize points from the input image and in simple words, it identifies sharp changes in the brightness of the image. It is a fundamental stage in image processing, computer vision, and picture pattern recognition. Edge Detection Operators are of two types:

- Gradient–based operator: First-order derivations are computed using gradient-based operators in digital images, such as the Sobel, Prewitt, and Robert operators.

- Gaussian–based operator: Second-order derivations are computed using Gaussian–based operator in a digital image like Canny edge detector and Laplacian of Gaussian.

The input image is firstly converted from RGB to grayscale to simply reduce complexity: from a 3D pixel value (R, G, B) to a 1D value. Many tasks do not fare better with 3D pixels (e.g., edge detection). To reduce the noise, we need to blur the input image with Gaussian Blur then convert it to grayscale. Image noise is a distortion in the image that results from a camera malfunction or from reduced visibility brought on by shifting weather conditions. Noises are another term for the random change in pixel intensity levels. There are many varieties of noise, including salt & pepper noise and Gaussian noise. For noise removal, we implemented an iterative bilateral filter. A bilateral filter is used for smoothening images and reducing noise, while preserving edges. More effectively than the median filter, it offers the mechanism for noise reduction while preserving edges. The process of binarizing a picture results in an output image that only has two-pixel values, white and black pixels. The work of detecting license plates will be simpler if the binarization step is carried out first since edges will be visible more clearly in the binary image.

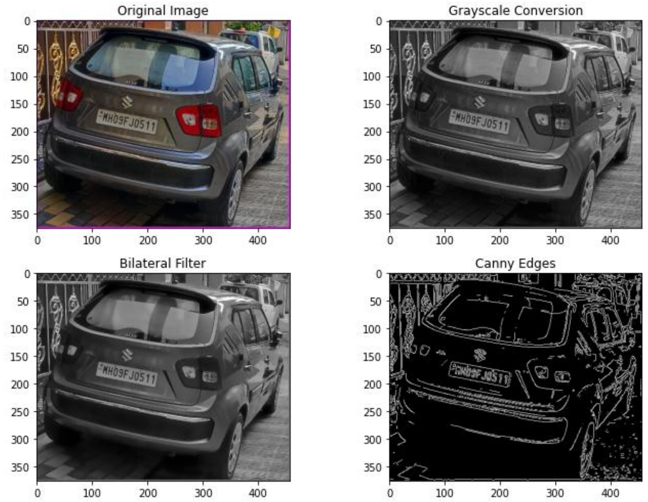

Fig -1: Image Preprocessing on input image

Our image is ready to discover contours after preprocessing shown in figure 1. The remaining contours are sent for additional processing once any contours having an area of less than 30 are eliminated. Every contour is roughly approximated to form a polygon, and if a contour is quadrilateral in shape (has four sides), it is expected to be the license plate. The contours are then created, and the resulting image is as follows:

Fig -2: Extracted license plate

Results for leveled image and tilted image:

As the image shown in figure 3 is leveled, therefore the detection and extraction are done properly as depicted in 4.

Fig -3: Detected license plate for levelled image

Fig -4: Extracted license plate for levelled image

For Tilted Image:

As shown in figure 5 and 6, the region of interest detected is not accurate.

Fig -5: Incorrect license plate detection

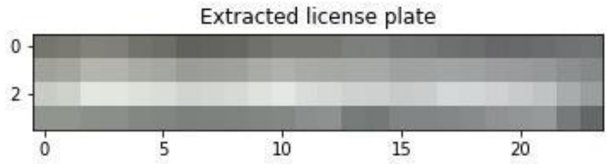

Fig -6: Incorrect license plate extraction

This algorithm fails when the image is more tilted. Also, it works only if the input image is of good quality. As this algorithm is not efficient enough in real time, we have tried other algorithms for license plate localization.

3.2 YOLOv4

You only look once (YOLO) is a family of one-stage object detectors that are fast and accurate. All the YOLO models are object detection models. Object detection models are trained to look at an image and search for a subset of object classes. When found, these object classes are enclosed in a bounding box and their class is identified. Object detection models are typically trained and evaluated on the COCO (Common Objects in Context) dataset, which contains a broad range of 80 object classes. Real-time is particularly important for object detection models that operate on video feeds, such as self driving cars, etc. The other advantage of real time object detection models is that they are small and easy to wield by all developers.

YOLOv4 makes real time detection a priority and conducts training on a single GPU.

The original YOLO (You Only Look Once) was written in a custom framework called Darknet. Darknet is a very flexible research framework written in low-level languages and has produced a series of the best real-time object detectors in computer vision.

The general architecture of YOLOv4 includes the following:

Backbone is the deep learning architecture that basically acts as a feature extractor. The neck has additional layers between the head and the backbone. The head, often referred to as the object detector, essentially locates the area where the object may be present. Methods such as "bag of freebies" (BoF) can improve object detecting accuracy without raising inference costs. Bag of Specials (BoS) are those plugin modules and post-processing techniques that only slightly raise the inference cost but greatly increase object detection accuracy. The general architecture of YOLOv4 is depicted in figure 7.

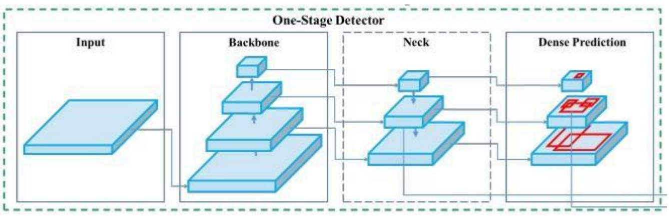

Fig -7: General architecture of YOLOv4(One stage detector)

Selection of architecture for the entire network is a very tedious task as there can be a lot of possibilities. Thus, depending upon some factors like number of convolutional layers, number of parameters and additional characteristics which include bag of specials, freebies, etc, the final architecture of YOLOv4 can be presented as:

CSPDarkNet53 (Head) - SSP + PANet (Neck) - YOLOv3(head)

The main idea behind the YOLOv4 object detector is that it divides the image into evenly sized grids, and then predicts the boundary boxes and corresponding probability of each grid. Owing to this, YOLOv4 uses CIoU as loss function as it considers distance between center points and overlapping area, and it uses Mish activation function.

As discussed, the YOLOv4 network has the ability of detecting 80 classes, but for our real time application detection of unnecessary classes leads to reduction of efficiency and increases time of operation. Thus, in order to deal with this problem, we decided to train a custom YOLOv4 detector for a single class i.e., license plate. Further in order to create a custom detector the primary requirement is a good dataset of images and labels so that the detector can be efficiently trained to detect objects. We used the images from Google Open Images and created a dataset of 5000 images and are split into training and validation dataset in 80:20 ratio. All the images in the training dataset were labeled with the bounding box coordinates and further these labels are converted in YOLOv4 format. The parameters for the training process were set to optimal values found by fine tuning it: batch=64, subdivisions=16, learning_rate=0.001, max_batches = 6000, policy = steps, steps=4800, 5400. Finally, we received an accuracy of 91%. The results of YOLOv4 custom detector are shown in figure 8.

Fig -8: Result of YOLOv4 custom detector

Fig -9: Extracted license plate

Even though the YOLOv4 custom object detector has an advantage of high-speed processing making it apt for real time applications, it sometimes fails in unrestricted situations when the LP may be significantly altered as a result of oblique perspectives.

3.3 WPOD-NET

Warped Planar Object Detection Network (WPOD-NET) is a novel Convolutional Neural Network (CNN) capable of detecting and rectifying multiple distorted license plates in a single image. The methodology of this network is complex and is described further. The network is initially fed by the vehicle detection's output that has been downsized. An 8-channel feature map with object/non-object probability and affine transformation parameters is the result of the feed forwarding. Let's first consider an imaginary square with a constant size surrounding a cell's center in order to extract the warped LP (m, n). A portion of the regressed parameters are then utilized to create an affine matrix that converts the hypothetical square into an LP region if the object probability for this cell is greater than a specific detection threshold. There are a total of 21 convolutional layers in the proposed design. All convolutional filters have a constant size of 3x3. Except for the detection block, the entire network employs ReLU activations. There are four max pooling layers with a size of 2x2 and a stride of 2, which lowers the input dimensionality by a factor of 16. Two parallel convolutional layers make up the detection block's last layer. The architecture of WPOD-NET can be seen in figure 10.

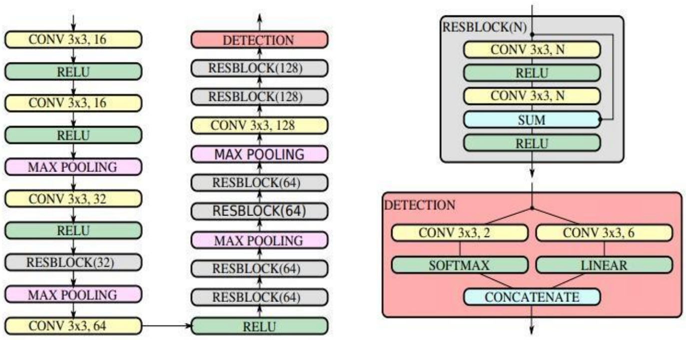

Fig -10: Architecture of WPOD-NET

A dataset of 196 photos is utilized to train the WPOD-NET, comprising 105 images from the Cars Dataset, 40 images from the SSIG Dataset (training subset), and 51 images from the AOLP (Application-oriented License Plate) dataset (LE subset). Four of the corners of the LP in each image are manually annotated. Most of the LPs in the Cars Dataset's chosen photos are European, but there are also a lot of American and other types of LPs. Brazilian and Taiwanese LPs are depicted in images from SSIG and AOLP, respectively. Data augmentation is done on the dataset because it is minimal in size.

Various augmentations are done on images such as rectification, centering, scaling, rotation, mirroring and translation. Due to hardware limitations, we studied WPOD-NET architecture and decided to use pre-trained weights trained on the dataset mentioned above. The average accuracy obtained on all the dataset is 91%. According to the WPOD_NET results, the suggested strategy performs significantly better than previous methods in complex datasets with LPs collected at severely oblique viewpoints while maintaining good results in more controlled datasets.

4. CONCLUSIONS

The above discussed methods work well in different scenarios. As per discussed, edge detection algorithm requires low computation power, but may not work well in unconstrained scenarios. While YOLOv4 is known for real time applications but struggles to detect small sized objects on the other hand WPOD-NET works well with small, oblique and distorted license plate images and in unconstrained scenarios but requires high computation power. Hence considering the requirement of application, one can opt for any of these algorithms.

REFERENCES

[1] S´ergio Montazzolli Silva, Cl´audio Rosito Jung, "License Plate Detection and Recognition in Unconstrained Scenarios", Institute of Informatics - Federal University of Rio Grande do Sul Porto Alegre, Brazil, ECCV 2018.

[2] Cheng-Hun Lin, Yong-Sin Lin, Wei-Chen Lui, "An Efficient License Plate Recognition System Using Convolutional Neural Networks", IEEE ICASI 2018

[3] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline", School of Computer Science and Technology, University of Science and Technology of China, Hefei, China (2018).

[4] Arpitha D, Archana T, Rajashree Y G, Priya G, "Automatic License Plate Recognition for Real-Time Videos", Visvesvaraya Technological University Belagavi, Karnataka.

[5] V Gnanaprakash, N Kanthimathi, N Saranya, "Automatic number plate recognition using deep learning", Bannari Amman Institute of Technology, Department of Electronics and Communication Engineering, Sathyamangalam, Erode, Tamilnadu, India, ICCSSS 2020.

[6] Muhammad Tahir Qadri, Muhammad Asif, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition", 2009 International Conference on Education Technology and Computer.

[7] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017.

[8] Prof.Amit Kukreja, Swati Bhandari, Sayali Bhatkar, Jyoti Chavda, Smita Lad, "Indian Vehicle Number Plate Detection Using Image Processing", Apr -2017.

[9] Ibtissam Slimani, Abdelmoghit Zaarane, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun, An automated license plate detection and recognition system based on wavelet decomposition and CNN,Array, Volume 8,2020.

[10] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana and S. Apoorva, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning,"

2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), 2018.

[11] Lubna, N. Mufti, and S. A. A. Shah, "Automatic Number Plate Recognition:A Detailed Survey of Relevant Algorithms," Sensors, vol. 21, no. 9, p. 3028, Apr. 2021.

[12] J.-S. Chou and C.-H. Liu, "Automated Sensing System for Real-Time Recognition of Trucks in River Dredging Areas Using Computer Vision and Convolutional Deep Learning," Sensors, vol. 21, no. 2, p. 555, Jan. 2021.