Detection and Recognition of License Plates by Convolutional Neural Networks

Zahra Taleb Soghadi

A Thesis
In the Department
Of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master of Science (Computer Science) at
Concordia University
Montréal, Québec, Canada

October 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Zahra Taleb Soghadi

Entitled: Detection and Recognition of License Plates by Convolutional Neural Networks

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

________________________ Chair

Dr. A. Hanna

________________________ Examiner

Dr. T. D. Bui

________________________ Examiner

Dr. C. Poullis

________________________ Supervisor

Dr. Ching Y. Suen

Approved by ________________________

Dr. Leila Kosseim, Graduate Program Director

December 16, 2019 ________________________

Dr. Amir Asif, Dean

Gina Cody School of Engineering & Computer Science

Abstract

Detection and Recognition of License Plates by Convolutional Neural Networks

Zahra Taleb Soghadi

Concordia University, 2019

The current advancements in machine intelligence have expedited the process of recognizing vehicles and other objects on the roads. The License Plate Recognition system (LPR) is an open challenge for many researchers to develop a reliable and accurate system for automatic license plate recognition. Several methods including Deep Learning techniques have been proposed recently for LPR, yet those methods are limited to specific regions or privately collected datasets.

In this thesis, we propose an end-to-end Deep Convolutional Neural Network system for license plate recognition that is not limited to a specific region or country. We apply a modified version of YOLO v2 to first recognize the vehicle and then localize the license plate. Moreover, through the convolutional procedures, we improve an Optical Character Recognition network (OCR-Net) to recognize the license plate numbers and letters.

Our method performs well for different vehicle types such as sedans, SUVs, buses, motorbikes, and trucks. The system works reliably on images of the front and rear views of the vehicle, and it also overcomes tilted or distorted license plate images and performs adequately under various illumination conditions, and noisy backgrounds. Several experiments have been carried out on various types of images from privately collected and publicly available datasets including OPEN-ALPR (BR, EU, US) which consists of 115 Brazilian, 108 European, and 222 North American images, CENPARMI includes 440 from Chinese, US, and different provinces of Canada and UFPR-ALPR includes 4500 Brazilian license plate images; images of those datasets have several challenges: i.e. single to multiple vehicles in an image, license plates of different countries, vehicles at different distances, and images taken by several types of cameras including cellphone cameras. Our experimental results show that the proposed system achieves 98.04% accuracy on average for OPEN-ALPR dataset, 88.5% for the more challenging CENPARMI dataset and 97.42% for UFPR-ALPR dataset respectively, outperforming the state-of-the-art commercial and academics.

Acknowledgments

First, I would like to show my appreciation to my supervisor Dr. Ching Y. Suen for his excellent support, brilliant ideas, insights, and comments. It was my honor to work under his supervision. I am grateful for every single moment that he spent to review my research thesis draft.

I would like to thank my examiners Dr. T. Bui and Dr. C. Poullis, and my colleague Ali K. Hmood for their constructive comments. Precious comments and feedbacks from them are deeply appreciated.

My special thanks to Dr. Marleah Blom, the Executive Assistant of CENPARMI for all her administrative assistance and her positive energy and, Mr. Nicola Nobile, the research manager of CENPARMI, for all his technical support and help.

Last but not least, I want to express my heartfelt thankfulness and gratitude to my parents, who believed in me and supported me in my life which, their guidance and their words of wisdom enlightened my way, also my beautiful sisters with their positive words and their support.

Finally, I would like to dedicate this thesis to my husband because of his understanding, patience, and support and to my amazing daughter that I breathe for her.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter provides an introduction to this thesis. Section 1.1 introduces the automatic license plate recognition (ALPR) application and describes the motivation of developing a robust ALPR system. Section 1.3 defines our contribution. Lastly, Section 1.4 gives an overview of the structure of this work.

1. 1 Background and Research Motivation

Building an Automatic License plate recognition system (ALPR) with high accuracy and high speed from low resolution or tilted pictures is still a challenging task among researchers. Forensics, traffic control, and public parking are the main applications for the ALPR systems. Governments and private companies normally use cameras specifically designed for character recognition and vehicle detection [1]. But as a matter of the fact, on the roads, specifically, when there is no governmental camera, police car's cameras are highly affected due to the car's speed, angles, or weather condition, and they do not provide acceptable resolution under certain weather conditions. The license plates convey much information from the vehicles, so, having such a system is vital. To mention some of the challenges we can point out to the low quality, blur, and uneven illumination as well as various types of font, number of characters, their size, color, direction, and complex background in different countries or even different provinces within a country.

Figure 1-1. The examples of different license plate within a single country, Canada [2]

As illustrated in Figure1-1, notwithstanding too many efforts that have been done in the earlier years, the license plate recognition system is still a challenging problem. Not all the typefaces on current license plates are clear to read for machines. For instance, there are some letters and numbers that are similar in terms of design; to mention some, we have several problematical classes of letters and numbers which create complexity to distinguish them from one another, specifically, from far distances and under various conditions:{'B' '8' '3' 'E' '6' 'G'},{'W' and 'M'}, {'Q' 'D' 'O' '0'},{'S' '5'}, {'Z' '2'}, {'T' '7' 'I' '1' 'L'}, {'P' 'R'}, {'A' '4'}, {'F' 'E'}, etc.

Figure 1-2. Complex combination of letters and numbers leads to unreadability [3]

In Figure 1-2, there is a complex combination of letter 'B' and number '8', which made the situation even worse for a machine to distinguish the characters.

Without a doubt, the improvement of the vehicle detection, license plate recognition and optical character recognition have taken more advantage of the recent development in deep learning techniques. The Deep Convolutional Neural Network (CNN) is one of the best machine learning techniques used for vehicle and license plate recognition systems [4], [18], [20], [35].

Numerous commercial ALPR systems utilizing deep learning techniques, for instance, Open-ALPR (http://www.openalpr.com/), Sighthound (https://www.sighthound.com), GENETEC Automatic license plate recognition (https://www.genetec.com/), and Amazon Rekognition (https://aws.amazon.com/rekognition/). Most state-of-the-art systems are limited to frontal view of the vehicles or they are not able to recognize different license plates from different regions, for example, some of them can solely distinguish US license plates and some of them are capable of recognizing Chinese or Brazilian license plates which are suitable only for parking validation and toll monitoring.

In this thesis, we propose new LPR system that performs well using a modified version of YOLO v2 [6], [7] in order to detect and recognize vehicles and license plates, also a customized existing Optical Character Recognition Network (OCR-NET) [4] for character recognition.

In Figure 1-3, there is some examples of dissimilar license plates within a single country, United State of America, where each state uses its own design, logo, character's font, size, color, and quote.

Figure 1-3. United State License Plates for 50 states [5]

Numerous ANPR methods are reliable when trained to match plates from a single province or territory, but they are not successful when attempting to identify plates from other regions due to variations in format, font, color, layout, and other plate features [8]. Some regions allow customized plates, which can yield multiple variations in a particular region [5], [8], [10].

In addition, due to the influence of different weather conditions and various illumination and to make the LPR systems powerless of reading the license plates correctly, some vehicle proprietors to avoid ALPR systems and road-rule enforcement cameras [9], have used complex methods to reduce the legibility of their license plates. For instance, using reflecting Vinyl Vehicle with lettering to enhance the qualities and reduce the level of contrast, the system could not read the letters and numbers accurately, leading to misidentification.

Moreover, the resolution of the images taken by different cameras vary in distance, and the angle of the vehicle is also different from the cameras. Besides, some existing ALPR systems are able to recognize a certain type of vehicles, for instance, only sedans and buses, and they are not capable of recognizing all types of vehicles i.e. aforementioned vehicles, SUV, trucks and motorbikes.

1. 2 Problem Definition

There are separate tasks in the ALPR system, detecting the vehicles, localization of the license plate and recognition of the characters. The study introduced in this thesis is concentrated on the all three aforementioned tasks.

Having a large and realistic dataset is necessary to solve the problem of the ALPR system, specifically, when the goal is to resolve the ALPR task for more than one region and scenario.

The problem of realistic data are the high level of noise and the situation will be even worse when intentionally, you use some data with high distortions and images with low quality from far distances. However, very complex rules and methods arise to cope with ALPR system problem [4],[20],[35], most of them still have difficulty to cope with tilted and far license plates.

On the other hand, some approaches have problematic issue to detect all types of vehicles [4],[35], most of them are not able to recognize characters on motorbikes, because the size of license plates on motorbikes are smaller than other vehicles and most of pre-trained models cannot classify them easily. Moreover, the traditional machine Learning are not able to make a comprehensive ALPR system, but through deep learning model approach, it is possible to cope with aforementioned problems.

1. 3 Thesis Contributions

Based on the Deep Learning Convolutional Neural Network, we introduce a system which is capable of detecting several license plates in a real world image, and is able to only detect all types of vehicles (sedan, SUV, bus, motorbike, and trucks) as well as different sizes of license plates with dissimilar foreground and background colors, from various angles and distances under different weather conditions. Moreover, we improved an existing OCR-NET [4], in order to recognize different font faces and dynamic lengths of characters on a license plate.

Additionally, we have done several experiments and evaluations on public and private datasets to evaluate our system in detect, locate, and recognize vehicles, license plates, and characters. The experimental results show that our proposed method outperforms the state-of-the-art commercial and academic ALPR systems in most challenging scenarios.

1. 4 Thesis Overview

Related works and earlier techniques in the Literature Review chapter are summarized. In this section, we have a comparison of other academic performances, processing rules, and their classifiers. Furthermore, the pros and cons of their methods are noted. In the Theoretical Background and System Methodology chapter, we review some relevant theoretical backgrounds about Deep Learning and Convolutional Neural Networks, then explain our approach and methods in detail such as the features, neural network model structures, datasets, platforms, hardware and deep learning techniques. It also introduces a novel system of license plate recognition and its component functions. The experimental work conducted in this study is illustrated in the Experimental Results and Discussion chapter. In this section, we have different results on various datasets and show the impact of several activation functions and threshold variables.

Finally, the conclusions and future work chapter reviews the thesis and suggests possible study pathways in the future.

CHAPTER 2

Literature Review and Theoretical Background

1. Literature Review

This chapter presents a brief review of the previous studies on license plate detection and recognition using machine learning and deep learning techniques. Also, this chapter presents the background theory on neural networks and convolutional neural networks. The current license plate recognizers can be classified to two approaches, employing classical machine learning methods and deep learning procedures.

Generally, License plate recognition algorithms in images or real-time videos consist of three different processing levels:

- Detecting the vehicle
- Locating the license plate region, and
- Recognizing characters.

Due to the diversity of license plates, different image angles, complex alphanumeric ordering, unpredictable font faces, blurry and noisy images, and inadequate illumination circumstances, these different steps are considerably challenging. Consequently, most approaches work only under limited situations like fixed illumination, restricted vehicle velocity, monochrome backgrounds, and standard font faces. Various methods have been proposed for license plate detection and recognition, and we provide a brief review of them in this chapter.

1. 1 Machine Learning and Computer Vision Based Algorithms

For license plate detection, the following elements should be considered at least:

- Size of the license plate: the license plates from different regions vary in size.
- Location of the license plate: a license plate can be positioned in the rear or front of a vehicle.
- Background of the license plate: a license plate can have various background colors or images based on vehicle regions.

In reference [11], the K-means algorithm was used to recognize the license plate characters from an image. The K-means algorithm in this study was developed by proposing an automatic cluster number determined by filtering scale-invariant feature transform (SIFT) key points. They performed a 6-layer cascaded classifier for license plate localization by applying global edge and local Harr-like features. The authors achieved a 94.03% accuracy on 578 images from Chinese license plates with 3502 characters, by applying several methods consisting of image binarization, vertical edge detection, horizontal and vertical image projections, and modified K-means segmentation algorithm. They also applied the Tesseract OCR software for character recognition. These methodologies are time-consuming and require several preprocessing techniques.

In reference [12], a combination of the K-Nearest Neighbors algorithm and the Multi-Class Support Vector Machines (KNN-SVM) model for the Iranian License plate recognition system was developed. To reduce the noise of images, they applied K-NN algorithms which is sufficient for big datasets. Multiple SVMs classification models with RBF kernel have been employed to resolve the license plate character recognition problem. The SVMs model has enhanced the performance of the K-NN in character recognition specifically for similar characters. The authors claimed that their system obtained a 97.03% accuracy ratio for all their experiments. Although the system has an acceptable accuracy ratio, the most significant problem is in what the system is unable to recognize the screw from the character in some images. For instance, it is complicated to distinguish number '1' plus a screw from number '9' in Persian license plate's fonts.

In reference [13], a K-Nearest Neighbors algorithm was proposed with pre-training steps to recognize numbers and letters on multi-style license plates. For instance, single-line and double-line, and complex backgrounds and character's colors on Korean and the United State license plates. The authors evaluated 50 minutes video of 138 different vehicles with various styles and reached above 99% accuracy under 50 millisecond processing time on characters recognition, but the system has limitation of recognizing Korean and US license plates.

In reference [14], the authors used the morphological procedures such as the Fuzzy transformation and Fuzzy logic edge detection algorithm to extract the location of license plates. Furthermore, they adopted character segmentation and template matching by utilizing correlation to recognize the license plate characters and achieved a 90.18% accuracy ratio in the extraction of number plates and a 79.30% accuracy ratio in character recognition which is not acceptable in terms of character recognition.

The work discussed in [15] is based on single-level wavelet transform, and their algorithm performed acceptably in different situations on Indian license plates. They conducted their experiments under various lighting conditions, and on different vehicle models with various shapes, sizes, and colors of license plates. The input colored images were 400 × 300 pixels, and they used 250 license plate images. In order to segment the characters, they plotted the vertical frequency's energy curve for license plates with two lines. The authors also utilized a statistical correlation-based method of template matching for character recognition. To overcome failure in determining the difference between some complicated letters and numbers such as 'O' and '0', they considered some special properties of each character, such as the aspect ratio of the character's horizontal to vertical length. Their method performed a 97.33% accuracy for plate localization, a 95.93% accuracy for character segmentation, and a 95.6% accuracy for character recognition. However, the system has adequate accuracy but incapable of recognizing complicated license plate's backgrounds and needs many preprocessing steps.

In reference [16], they discussed a new method for text localization and recognition in natural scene images with complex backgrounds. Their approach had several steps;" superimposed text regions in an image was extracted based on character descriptor features like Bounding box, Perimeter, Euler number, Horizontal crossings" [16]. Then they used SVM classifiers to test if the text region included letters or not. They did line segmentation by applying horizontal profiles. Subsequently, they segmented each character by utilizing vertical profiles. Toward character recognition, they used Optical Character Recognition (OCR) tools. Their various accuracy outcomes for different methods such as the Ostu algorithm, AdaBoost and SVM were 64.40%, 75.04%, %78.80% respectively. Although this methodology works properly with complex backgrounds, we need a more accurate system for license plate character recognition.

In reference [17], a system for Indian vehicle license plates detection with different font faces (Arial, Courier and Times New Roman) was tested on a small private database. In the suggested approach, the morphological operations, and horizontal/vertical edge histogram were applied to localize and character segmentation. Character recognition was performed by employing a two-layer feed-forward backpropagation Artificial Neural Network (ANN). In this methodology, license plate localization, character segmentation, and character recognition had accuracies of 92%, 92%, and 87% respectively which are not acceptable for such a time-consuming algorithm.

1. 2 Deep Learning Neural Network Algorithms

Though in recent years, the deep learning techniques for license plate detection and recognition produced a better result, due to the various weather conditions, the skew of images, complicated backgrounds, and various types of license plate font faces, designing a practical, and reliable application still faces a significant problem.

In reference [18], a two-step method was proposed for license plate localization and skew correction and region refinement based on multi-level thresholding binarization for Chinese license plates. The algorithms detected the license plate region using four borderlines and corners and then, utilized the single-shot multi-box detector (SSD) [50], classified the license plate type in the image. The study evaluated the accuracy of the introduced system on 600 vehicle images with a size of 800 * 800, and in complicated environments. They applied Optical Character Recognition (OCR) methods to segment and recognize the license plates numbers and letters. The outcomes indicated that the license plates in 579 images was localized precisely with a 96.5% accuracy and processing speed was up to 27 frame per second (FPS). The SSD method with the Mobile-Net performed a real-time license plate detection and classification in a single network with a 97.33% accuracy.

A CNN (convolution neural network)-GRU (gated recurrent unit) model was produced in [19]. They employed CNN for feature extraction and GRU for sequencing character recognition purposes without segmentation. Suvarnam et.al trained a two-dimensional CNN pattern by providing resized input images. Unlike the sliding window approach, their suggested algorithms enabled the hidden layer to access the features of the entire image. Their study showed that an ALPR system using the combination of CNN-GRU performed a better result than utilizing a layout coordinating method and CNN. The accuracy of the stated program is 90% for 5000 training images, which is not sufficient for an accurate ALPR system.

Based on deep learning, in reference [20], the authors designed a Vehicle License Plate Recognition System (VLPRS) by utilizing the Xilinx PYNQ board for Chinese license plates. They also compared the performance and accuracy of YOLOv2 [6] and YOLOv3 [7] to recognize license plates from a single image and employed Convolutional Neural Network (CNN) toward character recognition. Hou et.al performed license plate recognition on PYNQ-Z1 (the first Zynq board to support PYNQ) [22]. The images which they used in this experiment were captured by cameras in a natural environment. First of all, they detected the license plates based on YOLOv2 and then, by adopting OpenCV [21], they cropped the license plate location from images and used CNN on Caffe in order to recognize the characters. Finally, the detection part reached the accuracy of 99.35% with 12. 19 milliseconds for one image and the accuracy of the character recognition engine is 97.89%.

In [23], a hierarchical system for license plate recognition detected vehicles by applying YOLOv2. Formerly, they used SVM OAR (one against rest) architecture with the HOG values to identify the vehicle's license plates. In the last step, the authors presented a vertical projection on the license plate images to identify the position of characters and later divided them into single characters. Their proposed LPRCNN model is constituted of "two convolutional layers, two max-pooling layers, two fully connected layers, and one output layer" [23]. The outcomes indicated that their algorithms reached a 96.12% accuracy ratio for car detection and a 94.23% accuracy for plate detection. They claimed that by employing LPRCNN, they obtained a 99.2% accuracy for character recognition on blurry and skewed images.

In this paper [25], a convolutional neural network structure was introduced for Chinese license plate character recognition. In order to reproduce the condition of blur images affected by severe weather, various illumination, different velocity, varied skews and other factors in real-time videos, the authors utilized salt and pepper noise and affine transformation. To overcome the low recognition rate caused by the complicated formation of Chinese characters, they applied various continuous convolution layers to obtain more features of images. The input of the convolution layer is a majority of two-dimensional matrices of the prior layers, which convolves the input matrix with the convolution kernel and generates a multiplicity of two-dimensional feature graphs by utilizing RELU function. The original dataset randomly assigned 20000 images to the training set and the test set, and 65 classes of Chinese characters were labeled. The results produced by the stated network had a 99.3% accuracy. They realized that the performance of the network structure can be improved by maximizing the number of iterations and enhancing the learning rate of attenuation procedure.

In this work [4], Silva et.al proposed a fully Automatic LPR system, using CNN for unconstrained scenarios. To detect the vehicle and recognize the license plates and characters they used a modified YOLO v2 on VOC dataset and their system was able to only detect cars and buses. They also, introduce a novel system (WPOD-NET) in order to detect and unwrap of distorted license plates by creating an affine transformation matrix per direction cell. Furthermore, their result on different datasets indicates that their system is able to cope with different situations and it had a competitive result to commercial and academic datasets. Their proposed pipeline achieved 89.33% accuracy in average on OPEN ALPR [46] (https://github.com/openalpr/benchmarks), SSIG [48], Car dataset [42], and ALOP (http://aolpr.ntust.edu.tw/lab/) datasets.

In reference [35], the authors utilize two object detection CNN and both networks used the YOLO v2 architecture due to robustness and high accuracy of this network. Their system had two different parts, the first network was retrained to detect only the license plates, and the second network was trained to segment and recognize character of detected license plates. Their system achieves 97.6% accuracy on an annotated dataset of Norwegian license plates collected by the authors. However, their proposed method was only able to cope with different illumination (day/night) and it was not able to detect oblique license plates.

2. Theoretical Background

2. 1 Convolutional Neural Networks

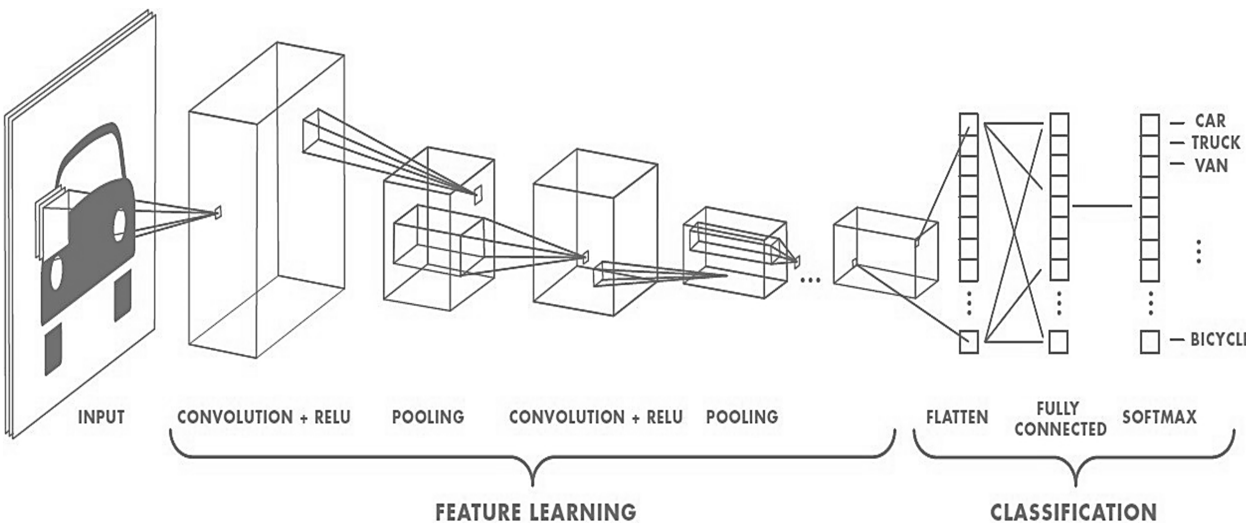


Figure 2- 1. Convolutional Neural Network Schema [27]

The concept of Convolutional Neural Networks (CNNs) first proposed by LeCun in 1989. CNN is a deep learning algorithm that is able to classify the objects from a given image by assigning learned weights and biases. Compared with other classification algorithms, the CNN needs less pre-processing and it is able to learn complex patterns mapped to high dimensional features. Inspired by CNN, many problems in the field of computer vision have been solved. Object detection [28], [29], [32], [33], object recognition, optical character recognition [25], [30], [34] face detection and license plate recognition [31], [35], [36], [37], [38], [40] are all tasks where CNNs have achieved state-of-the-art results.

The downside of the CNNs is that they usually are extremely computationally intensive for large datasets. Without preprocessing, the CNN requires more perceptron to see the patterns and it increases the time per training sample and the number of epochs to reach convergence. Hence, in some circumstances, preprocessing can make a positive effect to reduce the computation time. However, it may decrease the accuracy of the system. The output is a classification of the image, and the overall results can be improved by applying a loss function to minimize it over training.

The convolution is a mathematical operation on two different functions. Usually, the convolution between functions f and g is $f * g$. In image processing, due to the pixel-wise and non-continuous digital images, the convolution formula for two dimensions is:

$$(f * g)[m,n] = \sum_{j=-J}^{J} \sum_{i=-i}^{I} f[m-i, n-j] * g[i,j] \tag{2.1}$$

Convolution has been done over Four sets $-I, -I+1, \dots, I-1, I \; and \; -J, -J+1, \dots, J-1$.

After the convolutional step, a pooling function is utilized on the resulting convolutional response map. The pooling function is a function to gradually reduce the spatial size of sub-region of the convolutional response map by sampling it to a smaller response map [24], [35].

Max pooling is the most used pooling function, and it returns the maximum value in each sub-region in the convolutional response map. The advantages of using pooling are that it reduces the quantity of parameters and regularizes the CNN slightly.

2. 2 Learning rate

The learning rate is a parameter which determines the rate of convergence of a neural network. Actually, the gradient descent algorithm with backpropagation for each iteration, often changes the weights too much. The weights will over correct themselves during learning phase and they miss the local minimum. This missing leads to an actual increase of the loss function. In order to prevent this problem, the small learning rate parameter μ, is multiplied with the gradient of the loss function $\Delta E(W)$. The formula for this term is:

$$\Delta W^{t+1} = \mu \nabla E(W^t) \tag{2.2}$$

where _Wt+1 is the change in the weight space in the iteration step t + 1.

Intuitively, the learning rate determines the step size used in each iteration. With a high learning rate, the process of reaching a local minimum is speed up, but the step size may be too large "overshooting" the actual minimum. Thus, a low as possible learning rate while the training process still uses

2. 3 Momentum

The momentum is a small variable α, where the weight adjustment ΔW^t is also dependent on previous weight adjustments done in the last iteration.

The formula for usage this term is like this:

$$\Delta W^{t+1} = (1-\alpha)\mu \nabla E(W^t) + \alpha \Delta W^{t-1} \tag{2.3}$$

The range of momentum is something between [0, 1]. It means, when the value of α is 0, the weight changes for next iteration are just dependent on the gradient of the loss function.

For $\alpha = 1$, the weight change is solely dependent on the last weight adjustment. A most used value for momentum is 0.9 [4], [35], the small value of α, while the constancy of the convergence is kept sensibly intact, considerably can speed up the learning process. Instinctively, momentum benefits the gradient descent algorithm to avoid get stuck on plateaus and small local minima.

The reason is that the gradient of the loss function becomes very small and slows the gradient descent. This deceleration process such that plateaus, and small local minima take advantage of the momentum to delay, and they are faster to overcome without increasing the overall learning rate.

2. 4 Weight decay

The simple way to regularize the loss function is using the weight decay. A zero mean Gaussian effectively changes the loss function to:

$$\widetilde{\mathrm{E}} = E(W) + \frac{\lambda}{2} W^2 \tag{2.4}$$

where, $E(W)$ is the previous loss function and λ is the weight decay parameter. When the new loss function applies to the gradient decent algorithm with momentum, the formula is:

$$\Delta W^{t+1} = (1-\alpha)\mu \nabla E(W^t) + \alpha \Delta W^{t-1} - \mu\lambda W^t \tag{2.5}$$

The term $-\mu\lambda W^t$ displays in what way the weights are charged proportional to its size dependent on the value of λ. Since the penalty is the proportional to the size of the weights, the most penalty applies to large weights, which leads to force the network to find a resolution only using smallmagnitude parameters. With the reduction in freedom in the model, the weight decay prevents overfitting and forces the network to generalize more.

The normal values for weight decay are between [0.0005-0.002] in deep neural networks [4], [35].

2. 5 Batch normalization

Batch normalization is a regularization step implemented to resolve internal covariate shift. Internal covariate shift is the concept when parameters of a network change, the distribution of network activations changes as well. It means, once weights in a certain layer is updated, the allocation of output vectors from that layer is also changed. The aforementioned output vectors are input vectors for the next layer and force the layer to adjust to the drift in input distribution. This problem causes slowing down in the learning phase.

Batch normalization resolves this problem by making it look like all layer inputs are normalized.

2. 6 Transfer Learning

Train a CNN from scratch is hard and time-consuming. Usually, an untrained CNNs have random weight matrices and biases. Also, for learning something, they need massive datasets for training. On the other hand, the early convolutional layers train general filters and those filters learn to detect simple and low-level patterns such as color blob detection and edge detection [51]. Because deep CNNs trained on general tasks and they have many adjustable layers, they are good candidates for transfer learning. There are two main strategies for transfer learning:

a) CNN as a fixed feature extractor

   This strategy, only the last classifier layer is replaced, and the rest of CNN is treated as a fixed feature extractor for the new dataset. The second last layer is normally fully connected, and it computes N-dimensional vectors called CNN

   codes for each image. The CNN codes have the information of activated neurons in the fully connected layer. They can also be considered as the features extracted from an image. The only task to use the pre-trained CNN is to train a new linear classifier designed for the new dataset [35].

b) Fine-tune the CNN

   another strategy for transfer learning is to retrain the whole or parts of the pre-trained CNN to replace the last classifier layer. CNN can be retrained with the new dataset without changing the architecture or reinitialize the weights which are called fin tuned. Due to overfitting concerns, it is possible to keep the early layers unchanged, as they detect the generic features.

comparison with CNN as a fixed feature extractor strategy, in this method, the feature extractors are not fixed, and they changed to represent the new dataset. This method also takes a long time and gives a higher representational power of the data [35], [51].

Factors that affect transfer learning

In order to use transfer learning in a task [51], two basic features should be considered. The size of the new dataset, and the similarity between the dataset for the new task and the original dataset used by the pre-trained CNN. The combination of these two features give four different scenarios:

1. Small dataset and low similarity: This is the hardest scenario, and it is very difficult to get a good result from it. Fine-tuning may give overfitting problems because the small dataset is inadequate to retrain the whole network. If in its place, a new classifier is trained on top of the CNN, the features for the CNN codes will not represent the new dataset well because of the low similarity between the new and original dataset. The only alternative which may work, is to train a classifier on top of an earlier and more generic layer.
2. Small dataset and high similarity: It is not also a good option to use for fine-tuning manners with small datasets and high similarity. However, since the similarity is higher than the first scenario, a linear classifier can be trained directly on top of the last fully connected layer and it gives better results.
3. Large dataset and low similarity: Due to the larger dataset, fine-tuning cannot cause any overfitting. However, the low similarity is still a drawback, but if the pre-trained CNNs architecture is similar to the whole subject, fine-tuning is recommended. The alternative choice is to solely use some parts of the architecture; however, this is not a direct task. Another decision is to train a CNN from the beginning. This option is reasonable since the dataset is satisfactorily enormous and it gives the most freedom to choose an adequate architecture.
4. Large dataset and high similarity: Large dataset and high similarity scenario is the most appropriate one. Due to the large dataset and high similarity fine-tuning is efficient and one can be sure that the learned features for the original dataset are also suitable for the new dataset.

CHAPTER 3

System Methodology

This chapter presents the methodology of our proposed system. This Section explains the pipeline and the methodology of License Plate Recognition System (LPR) and three different convolutional neural networks used for vehicle detection, license plate detection and rectification, and character recognition.

3. 1 Overview

This work is composed of three principal parts, (1) vehicle detection, (2) license plate detection, and (3) optical character recognition. As illustrated in Figure 3-7, the first step is to detect vehicles in an input image then in each detection region, we utilize Warped Planar Object Detection Network (WPOD-NET) [4] which is explained in 3.2.3, as a semi black box for license plate localization that transforms tilted license plates and rectifies them to a rectangular shape like frontal or rear views. These improved detected license plates are fed into an OCR Network for character recognition task.

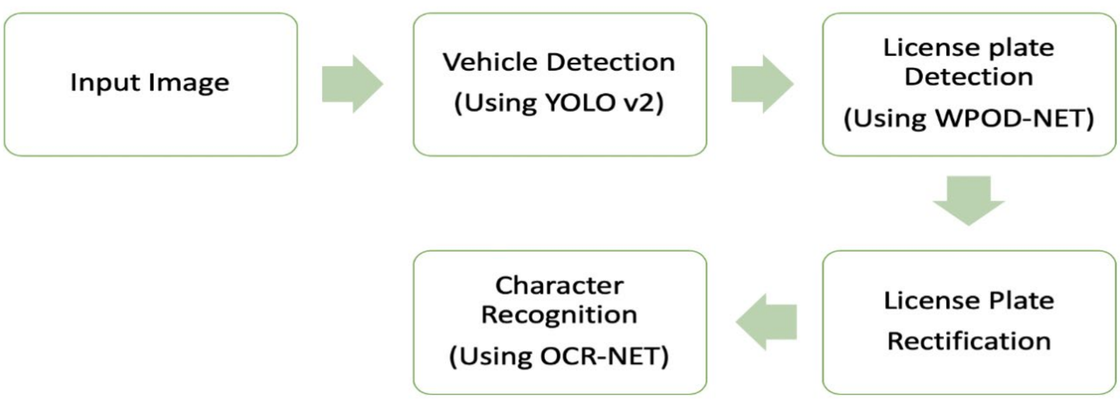


Figure 3- 1. The proposed system flowchart

3. 2 The proposed License Plate Recognition System (LPR)

3.2.1 Vehicle Detection

Due to the importance of vehicle detection, many datasets provide car detection such as COCO [43], ImageNet [39], and Pascal Visual Object Classes (VOC) [45]. Therefore, we decided to work on one of these studied models in order to make an accurate vehicle detector. YOLO version 2 [6] (2016), also called YOLO9000 is an improved version of the original YOLO network. As it utilizes a multi-scale training technique, it is able to run in different sizes and offer a trade-off between speed and accuracy. YOLO v2 considerably outperforms Faster R-CNN with ResNet [44] when the prediction accuracy is considered [46]. YOLO v2 using Darknet-19 network over PASCAL-VOC 2012 dataset which can detect the 20 Pascal object classes, are the best matches to have a maximum precision in vehicle detection and keep running time low.

As [41] mentioned, there is a rule which indicates that the large input images provide a better object detection, specifically for smaller objects, but in the negative side, they raise the cost of computation. Furthermore, when the angle of the camera is almost "0" with the frontal or rear views of vehicle, the proportion of the license plate and the vehicle bounding box is definitely high. Conversely, this ratio is considerably low for oblique and lateral images. Therefore, in order to detect the license plate region adequately, oblique views should be resized to a larger dimension in comparison with frontal views. A simple and fast technique based on aspect ratio of the vehicle bounding box is proposed in [4] and the mathematical formula of resizing factor f_{sc} is:

$$fsc = \frac{1}{\min\{W_v, H_v\}} min\left\{D_{min}\frac{max\{W_v, H_v\}}{min\{W_v, H_v\}}, D_{max}\right\} \tag{3.1}$$

where W_v and H_v are the width and height of the vehicle bounding box. Besides, $D_{min} \leq fsc$ $\min\{W_v, H_v\} \leq D_{max}$, hence, those D_{min} and D_{max} restrict the range for the smallest dimension of the resized bounding box. Based on our experiments and trying to keep a good compromise between accuracy and running times, D_{min}= 287 and D_{max}= 588.

Due to the high prediction accuracy, YOLOv2 is preferred to the other networks. For the pre-processing of the images, the size of input images resized to 416 × 416 pixels due to the higher speed and accuracy in comparison with other dimensions, and the input image has 3 channels (RGB). The proposed architecture has a total of 30 convolutional layers, and the size of all convolutional filters varied from 32 to 1024. Leaky and ReLU activation functions are used throughout the network, except in the detection block where a linear activation function is utilized.

Five max pooling layers with 2×2 size and stride 2 are employed in order to reduce the input dimensionality by a factor of 16. The route and reorganized layers represent the pass-through layer in the YOLOv2 architecture. The fine-tuned features are routed from the 16th layer, turned into 13 × 13 resolution and concatenated with the features from the 27th and 24th layers. Both the vehicle detection network and character recognition network are the improved YOLOv2 architecture with some adjustments and altered parameters which are different from the original implementation to fit the CENPARMI dataset and increase the accuracy of whole system for some public datasets and to detect more types of vehicles which are discussed in chapter 4. The architecture of the proposed network with additional layers and modifications are shown in Table 3-1 see above.

Table 3- 1. The architecture of proposed YOLO v2

NO.	Layer	Filters	Size	Activation Function	Input	Output
0	conv	32	3×3 / 1	Leaky	416 × 416 × 3	416 × 416 × 32
1	max		2×2 / 2		416 × 416 × 32	208 × 208 × 32
2	conv	64	3×3 / 1	ReLU	208 × 208 × 32	208 × 208 × 64
3	max		2×2 / 2		208 × 208 × 64	104 × 104 × 64
4	conv	128	3×3 / 1	Leaky	104 × 104 × 64	104 × 104 × 128
5	conv	64	1×1 / 1	ReLU	104 × 104 × 128	104 × 104 × 64
6	conv	128	3×3 / 1	Leaky	104 × 104 × 64	104 × 104 × 128
7	max		2×2 / 2		104 × 104 × 128	52 × 52 × 128
8	conv	256	3×3 / 1	Leaky	52 × 52 × 128	52 × 52 × 256
9	conv	128	1×1 / 1	ReLU	52 × 52 × 256	52 × 52 × 128
10	conv	256	3×3 / 1	Leaky	52 × 52 × 128	52 × 52 × 256
11	max		2×2 / 2		52 × 52 × 128	26 × 26 × 256
12	conv	512	3×3 / 1	Leaky	26 ×26 × 512	26 ×26 × 256
13	conv	256	1×1 / 1	Leaky	26 ×26 × 256	26 ×26 × 512
14	conv	512	3×3 / 1	Leaky	26 ×26 ×512	26 ×26 ×256
15	conv	256	1×1 / 1	Leaky	26 ×26 × 256	26 ×26 × 512
16	conv	512	3×3 / 1	Leaky	26 ×26 × 512	26 ×26 × 1024
17	max		2×2 / 2		26 ×26 × 512	13 ×13 × 1024
18	conv	1024	3×3 / 1	Leaky	13 × 13× 1024	13 × 13× 512

19	conv	512	1×1 / 1	Leaky	13 × 13 × 512	13 × 13× 1024
20	conv	1024	3×3 / 1	Leaky	13 × 13× 1024	13 × 13× 512
21	conv	512	1×1 / 1	Leaky	13 × 13× 256	13 × 13× 512
22	conv	1024	3×3 / 1	Leaky	13 × 13×1024	13 × 13× 1024
23	conv	1024	3×3 / 1	Leaky	13 × 13× 1024	13 × 13× 1024
24	conv	1024	3×3 / 1	ReLU	13 × 13× 1024	13 × 13×64
25	route	17				
26	conv	64	1×1 / 1	Leaky	26 × 26×512	26 × 26× 64
27	reorganize		/2		26 × 26× 64	13 × 13 × 256
28	route	27, 24				
29	conv	1024	3×3 / 1	Leaky	13 × 13× 1280	13 × 13× 1024
30	conv	125	1×1 / 1	Linear	13 × 13× 1024	13 × 13× 125
31	detection					

For transfer learning, the pre-trained convolutional network Darknet19 [6] on PASCAL-VOC is used. The original Darknet19 includes 24 layers and 19 layers are convolutional, and 5 layers are max pooling. In order to retrain the network, we utilized 69 images from OPEN ALPR dataset (BR) [47], 264 manually annotated images from the CENPARMI dataset, and 133 images from the OPEN ALPR dataset (US). In addition, the number of images used for retraining the YOLO v2 network is 60 percent of each aforementioned datasets.

The vehicle detection network was trained for 80200 batches with a batch size of 64 which is over 100k epochs and the weights are refined by additional samples of annotated license plates. The number of epochs is notably high and only after 200 epochs the networks started to work acceptably. It means that the network has the potential of cutting down the training time in different conditions. To increase the accuracy of the network, the high number of epochs is selected, and the end-user never faces such time-consuming training time. Besides, to avoid over-fitting the training data, data augmentation and batch normalization are applied to efficiently adjust the model, which means there is no limitation to enlarge the training phase. During training, the learning rate starts from 0.001 and after 200 batches, it raised to 0.01. Due to the fragile gradients, we started with a lower learning rate to avoid the divergence of the model. The momentum is set to 0.9 and the weight decay is set to 0.0005.

We performed several changes and refinements to YOLO v2, in order to classify different types of vehicle with a small extra re-training the whole system. For instance, we examined different activation functions and pooling factors to have a more accurate system.

3.2.2 License Plate Detection Using WPOD-NET

After detecting the vehicles, the output image from positive detections is resized before being fed to Warped-Net license plate detection [4]. We used this network as a semi black box, and we did perform a small change and refinement to the threshold value and the bounding box size of license plate. In order to understand this network, we should notice that the license plates have mostly rectangular shapes and they are planar, which are attached to vehicles for identification reasons. S.M Silva and C.R. Jung proposed a novel CNN called Warped Planar Object Detection Network. They said that: "This network has learned to detect license plates in different distortions and regresses coefficients of an affine transformation that “unwarps” the distorted license plate into a rectangular shape resembling a frontal view" [4]. The WPOD-NET was produced using visions from YOLO, Single Shot Multi-Box Detector (SSD) [50], and Spatial Transformer Networks (STN) [49]. Fast multiple object detection and recognition can be done by utilizing YOLO and SSD. However, they are unable to perform spatial transformations and they only generate the rectangular bounding boxes for each detection. On the other hand, STN can detect non-rectangular areas, but it is not able to handle multiple transformations at the same time, and it can perform simply a single spatial transformation over the entire input [4]. The detection process using WPOD-NET is illustrated in Fig. 3. 8.

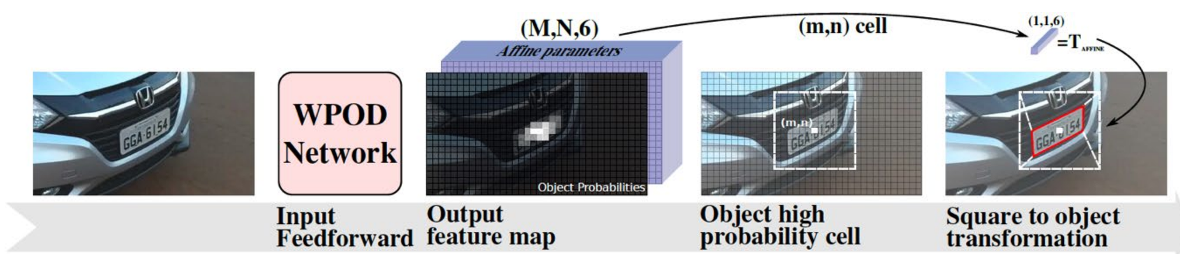


Figure 3- 2. Fully convolutional detection of planar objects [4]

In the beginning, the network is fed by the resized output of the vehicle detection section. The feed-forwarding outcomes in an 8-channel feature map that encodes object or non-object probabilities then affine transformation parameters. To extract the warped license plate, imagine a square of a fixed size around the center of a cell. If the object possibility for this cell is higher than the given detection threshold, the part of the regressed parameters is applied to make an affine matrix that transforms the imaginary square in a license plate area. Therefore, the license plate can simply unwrap toward a horizontally and vertically aligned object.

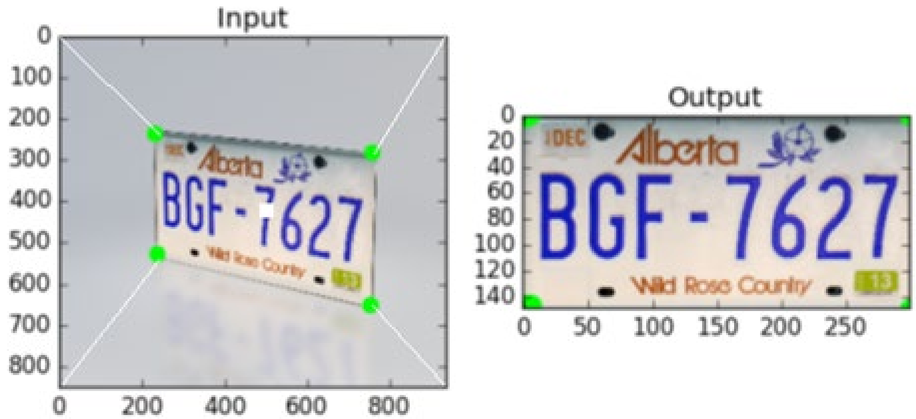


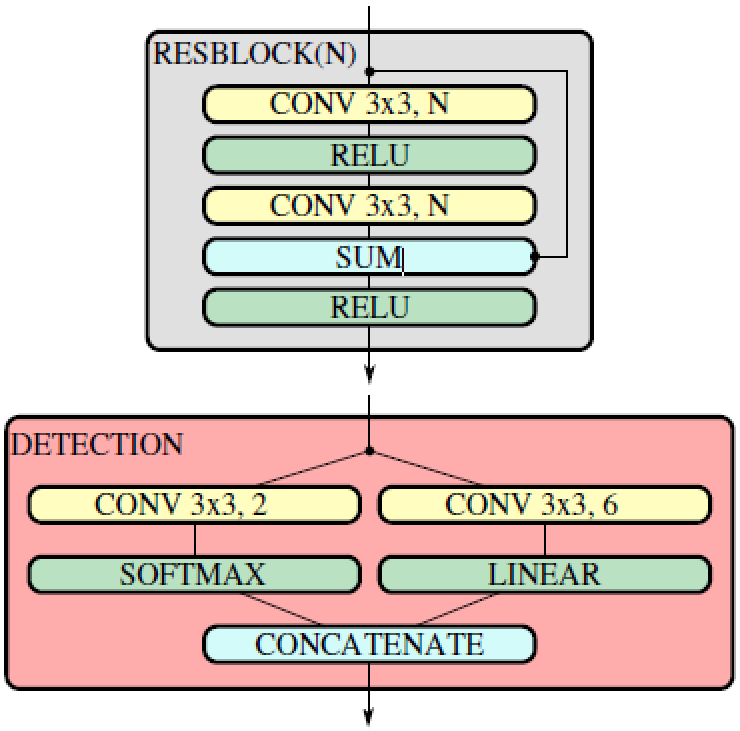
Figure 3- 3. Transformation of the oblique license plate through WPOD-NET

The proposed architecture of WPOD-NET has a total of 21 convolutional layers, where 14 are inside residual blocks [4], [44], [46]. The size of all convolutional filters is fixed in 3 × 3. ReLU activations are used throughout the entire network, and for the detection block a linear activation function is utilized. To reduce the input dimensionality, there are 4 max pooling layers of size 2×2 and stride 2 by a factor of 16. At the end, the detection block includes two parallel convolutional layers. One layer is used to infer the probability, with a SoftMax activation function, and the second one for regressing the affine parameters, with linear activation (or, equivalently, using the identity $F(x) = x$ as the activation function) [4].

Table 3- 2. Detailed WPOD-NET architecture [4]

NO.	Layer	Filters	Size	Activation Function	Input	Output
0	conv	16	3×3 / 1	ReLU	240× 80 × 16	240× 80 × 16
1	conv	16	3×3 / 1	ReLU	240× 80 × 16	240× 80 × 16
2	max		2×2 / 2		240× 80 × 16	120 × 40 × 16
3	conv	32	3×3 / 1	ReLU	120 × 40 × 16	120 × 40 × 32
4	Resblock	32	3×3 / 1		120 × 40 × 32	120 × 40 × 32
5	max		2×2 / 2		120 × 40 × 32	60 × 20 × 32
6	conv	64	3×3 / 1	ReLU	60 × 20 × 32	60 × 20 × 64
7	Resblock	64	3×3 / 1		60 × 20 × 64	60 × 20 × 64
8	Resblock	64	3×3 / 1		60 × 20 × 64	60 × 20 × 64

9	max		2×2 / 2		60 × 20 × 64	30 × 10 × 64
10	Resblock	64	3×3 / 1		30 × 10 × 64	30 × 10 × 64
11	Resblock	64	1×1 / 1		30 × 10 × 64	30 × 10 × 64
12	max		2×2 / 1		30 × 10 × 64	15 × 5 × 64
13	conv	128	3×3 / 1	ReLU	15 × 5 × 64	15 × 5 × 64
14	Resblock	128	3×3 / 1		15 × 5 × 64	15 × 5 × 64
15	Resblock	128	3×3 / 1		15 × 5 × 64	15 × 5 × 64
16	detection					



For $i = 1,2,3,4$ indicate the four corners of a license plate, consider $p_i = [x_i, y_i]^T$, which starts clockwise from the top-left corner. Besides, consider $q_1 = [-0.5, -0.5]^T, q_2 = [0.5, -0.5]^T, q_3 = [0.5, 0.5]^T, q_4 = [-0.5, 0.5]^T$ as the corresponding vertices of a standard square centered at the origin. Assume an input image with the height $H$and width W, and network stride of $N_8 = 2^4$ for four max-pooling layers, the feature map of the network output includes an $M \times N \times 8$, where $M = H/N_8$,$N = W/N_8$, and for each pixel (m, n) in the feature map, there are 8 values which should be assessed. Let (v_1, v_2), two values of the license plate/non-license plate possibilities, and the other 6 values $(v_3, \dots, v_8)$ utilize to make the local affine transformation $T_{m,n}$:

$$T_{m,n}(q) = \begin{bmatrix} \max(v_3, 0) & v_4 \\ v_5 & \max(v_6, 0) \end{bmatrix} q + \begin{bmatrix} v_7 \\ v_8 \end{bmatrix}, \tag{3.2}$$

where to ensure that the diagonal is positive and to avoid unwanted mirroring, the max function used for v_3 *and* v_6 [4], [26]. WPOD-NET is trained by a dataset with 196 images, 105 images from Cars dataset [42], 40 from SSIG dataset [48], and 51 from ALOP dataset. The 4 corners of license plates manually annotated, and the locations of the four license plate corners are adjusted by applying the spatial transformations. Different augmented test images also obtained from manually labeled samples by Silva and et al. For such systems, using augmented data is vital. In order to cover different scenarios, various transformations are used on the small dataset, for instance, rectification, centering, scaling, rotation, mirroring, and etc. [4].

3.2.3 Character Recognition Using OCR-NET

In order to segment and recognize of the characters over the output license plates from WPOD-NET, a modified YOLO network is employed, and the architecture of this network is shown in Table 3. In this network, to cope with different scenarios and variety of regions, we use a mixed of dissimilar datasets and augmentation of them to train the system such as European, US, and Brazilian license plates. The synthetic data consist of a 7 characters string on a textured background and then performing random operations, such as rotation, transformation, noise, and blur. As we mentioned in chapter 1, there are undeniable similarities between some characters and numbers in the most used font faces of license plates, for instance, O/Q/D/0, S/5, E/B/8, 2/Z/7, A/4, I/1/T, 6/G and etc. For text recognition in the OCR applications, most of these misclassifications can be handled through applying adjacency and semantic analysis which is impossible in terms of license plate recognition, because the characters and numbers of license plate do not carry any meaningful terms and the order of characters are almost meaningless. Another problem in license plate character recognition is the order of numbers and characters varying in different regions. For example, the format of Brazilian license plates is 3 letters followed by 4 numbers (ABC-1234) which is completely dissimilar to Quebec license plates, which have (A12 BCD) format and the personalized plates vary in the order of letters and numbers.

In order to overcome those problems, a pre-trained OCR network [40] and all layers from 1 to 11 were transferred from YOLO-VOC network. To reduce the chance of losing crucial information, the size of input image is 240 × 80 in 3 channel (RGB) which is almost the double size of the license plates, and the output image is 30 × 10. By experiments, this output size has enough horizontal space to distinguish the 7 characters of a license plate. To avoid nonlinearity the additional 3 layers were added and optimized using Adam optimizer.

In our proposed method, we have done many experiments on activation functions using in each layer and for each refinement. The new result shows the important role of activations in convolutional neural networks. Modification on those functions could make a huge difference in terms of accuracy of the whole system, especially in complex backgrounds and far distances.
The majority of activation function in this model is Leaky that was discussed prior. In the end, ReLU activation function for 3 layers of our network was chosen. The final architecture of the proposed network is shown in Table 3-3.

Table 3- 3. The modified architecture of OCR-NET

NO.	Layer	Filters	Size	Activation Function	Input	Output
0	conv	32	3×3 / 1	Leaky	240× 80 × 32	240× 80 × 32
1	max				240× 80 × 32	120 × 40 × 32
2	conv	64	3×3 / 2	ReLU	120 × 40 × 32	120 × 40 ×64
3	max		2×2 / 1		120 × 40 × 64	120 × 40 × 64
4	conv	128	3×3 / 1	ReLU	120 × 40 × 64	120 × 40 × 128
5	conv	64	1×1 / 2	ReLU	120 × 40 × 128	60 × 20 × 64
6	conv	128	3×3 / 1	Leaky	60 × 20 × 64	60 × 20 × 128
7	max		2×2 / 1		60 × 20 × 128	30 × 10 × 128
8	conv	256	3×3 / 1	Leaky	30 × 10 × 128	30 × 10 × 256
9	conv	128	1×1 / 1	Leaky	30 × 10 × 256	30 × 10 × 128
10	conv	256	3×3 / 1	Leaky	30 × 10 × 128	30 × 10 × 256
11	conv	512	3×3 / 1	Leaky	30 × 10 × 256	30 × 10 × 512
12	conv	256	3×3 / 1	Leaky	30 × 10 × 512	30 × 10 × 256
13	conv	512	3×3 / 1	Leaky	30 × 10 × 256	30 × 10 × 512
14	conv	80	1×1 / 1	Linear	30 × 10 × 512	30 × 10 × 80
15	detection					

3. 3 Datasets

The main reason of developing this LPR system is to create an accurate method which works properly in a variety of complicated scenarios, such as close or far distances, various illuminations, tilted and oblique license plates, blurry and noisy images, and different background and font faces of license plates as well as different regions around the world. Hence, three datasets were chosen for evaluation and test of which two of them are available online, in detail Open-ALPR (BR, EU, US) consist of 115 Brazilian, 108 European, and 222 North American images [47], which cover many different situations, as summarized in the first part of Table 1, UFPR-ALPR (https://web.inf.ufpr.br/vri/databases/ufpr-alpr),which includes 4500 images from 150 moving vehicles, 1200 of images are from sedan, and buses with different license plate background color (gray and red) and 300 of them are from motorbikes. We also evaluate and test one private dataset, namely CENPARMI (Center for Pattern Recognition and Machine Intelligence) dataset which includes 440 images generally rear view and the mixture of oblique, noisy, different distances from Chinese, US, and different provinces of Canada and mostly Quebec. The three important different variables which we considered in this work are license plate angle (frontal, rear and oblique), the distance from vehicles to the camera (close, intermediate and far), and the region of the license plate respectively. The more challenging dataset for our proposed methodology in terms of license plate distortion is the CENPARMI dataset, which contains multiple vehicle in a single image and mostly tilted license plates from different distances with high-resolution taken by iPhone 6 and X cellphone cameras, and they have the most license plate distortion but they are still legible for humans. For the evaluation part, the most challenging images from aforementioned datasets were chosen. 20 percent of all dataset images used for evaluation part and 20 percent of images tested by the system and 60 percent of them used for training the vehicle detection and character recognition part.

3. 4 Positive, Negative and False Positive

In our work, a positive is a plate detected and read correctly, a negative is a plate not detected and read and a false positive is non-plate detection or a plate read incorrectly. We considered negative and false positive as failure for the system. Furthermore, if a plate is not detected by the license plate detection section, a negative will happen and there is no backup methodology to find the plate which is not detected. The system will not show any warning to the end-user and the only way to find out the error is to check the images manually. Conversely, if a false positive happens, it means a non-plate object is detected as a license plate, it does not have a negative result for the system. The detected non-plate object will be sent to the character recognition section and the system will not find all the required characters to classify it as a license plate. This false positive also should be manually checked to assure that a real license plate is not discarded by the system. In the character recognition module, if one or more characters are misclassified, a negative will happen and only that particular license plate should be manually checked. Then again, in character network system, a false positive will happen when incorrect prediction accrued, for example, the system predicts ‘I’ instead of ‘1’, so there is no other choice to distinguish the error except manually checking the predictions.

3. 5 System Requirements and Project Setup

The proposed WPOD-NET was implemented using TensorFlow framework, while the initial YOLOv2 vehicle detection and OCR-NET were created and executed using the DarkNet framework, Keras and OpenCV. A Python wrapper was used to integrate the two frameworks. The hardware used for our experiments was a MAC OS with an Intel Corei7 2.8 GHz processor 4 Core, with 16 GB of RAM and with a Radeon pro 555 GPU. With that configuration, we were able to run the full LPR system with an average of 6 images Per Second for all datasets. This time is highly dependent of the number of vehicles detected in the input image. Hence, incrementing the vehicle detection threshold will result in higher FPS, but lower recall rates.

CHAPTER 4

Experimental Results and Discussion

The experimental results will be presented in this chapter. In addition, we report the experimental analysis of our LPR system with a comparison with state-of-the-art methods and commercial systems. Section 4.1 illustrates the experimentation with the selection of the confidence threshold values for vehicle detection and character recognition. Section 4.2 discusses experimentation with different activation functions. Section 4.3 compares and summarizes all experimental results describes the selection of the best parameters settings from the previous sections. Section 4.4 shows the performance of the system on the CPU and GPU processing units.

4. 1 Confidence Threshold Value

The YOLO network returns a confidence score for each predicted bounding box and if this confidence score is above a certain threshold value, the bounding box in the final prediction will be shown. Therefore, there is a potential impact directly the false positive and false negative ratio. In other words, if we choose a high value for the threshold, the network might make a prediction with less confidence and it leads to the false prediction or negative result. On the other hand, when the lower threshold is chosen, the network makes predictions without precision which increases the number of false positive. Therefore, with the respect to the impact of negative and false positive on the accuracy of the system, the threshold for the character recognition should be lower than the vehicle and license plate networks. In our proposed pipeline, we start with the following thresholds: {0.3, 0.7} for vehicle detection network (YOLO v2), {0.4, 0.8} for the license plate detection network (WPOD-NET) and 0.5 for character detection and recognition network (OCR-NET). To analyze the impact of different confidence threshold, several experiments were conducted. Finally, the decision was made to choose the threshold value of 0.45 for vehicle detection network (YOLO v2), 0.45 for the license plate detection network (WPOD-NET) and 0.3 for character detection and recognition network (OCR-NET).

For the LPR system, the number of undetected vehicles and the number of false positive for each threshold value level were analyzed. The highest number of false positive happened when the threshold value was low (~ 0.3). On the other hand, the number of undetected vehicles increased when the threshold value became high (0.7). The number of undetected vehicles for OPEN-ALPR (BR) dataset were about 7 out of 100 images when we only changed the threshold value from 0.7 to 0.65 and the number of false positives were 1 in this case. For the threshold value of 0.4, the number of undetected vehicles is 0 and the false positive were 7.

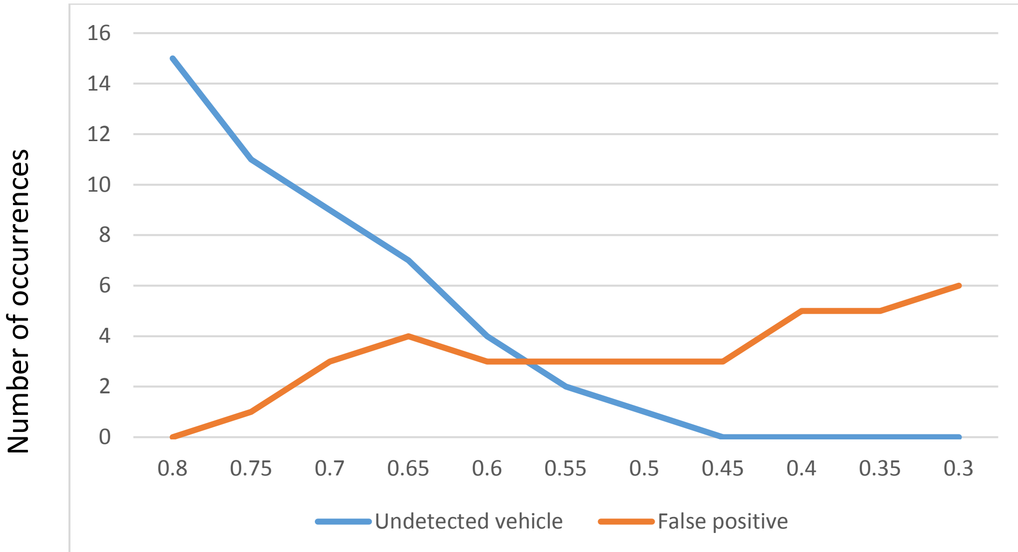


Figure 4- 1. Number of undetected vehicles versus false positives for the YOLO v2 network

The higher threshold for vehicle detection leads to lower prediction of vehicles which is not the results that we expect from a fully automated LPR system. In order to create a balance between the number of undetected vehicles and the number of false positives, the value of 0.45 was chosen for vehicle detection threshold with 1 undetected vehicle and 3 false positives.

The selected threshold was chosen for WPOD-NET is strongly related to the bounding box around the license plate. In the original version of WPOD-NET, the size of the bounding box was D_{min}= 288 and D_{max}= 608, but by doing many experiments the final threshold value was set to 0.45 and the bounding box changed to D_{min}= 287 and D_{max}= 588. In this case, the system faced the minimum false positive and these changes even have a positive impact on character recognition. Besides, by reducing the bounding box size around the license plate, the false positive remarkably decreased.

For OCR-NET network, to analyze the impact of confidence threshold we need three different categories. In the first category, we considered that one or more characters are misclassified or missing. For the second category, we assumed that all characters or a few of them are correctly classified. In the third category, analysis of the false positives, meaning the characters which are wrongly classified, for example, '1' instead of 'I'.

The threshold value for this network started from 0.8 and decreased to 0.3. The results of all experiments on OPEN-ALPR (BR) dataset are shown in Figure 4-2.

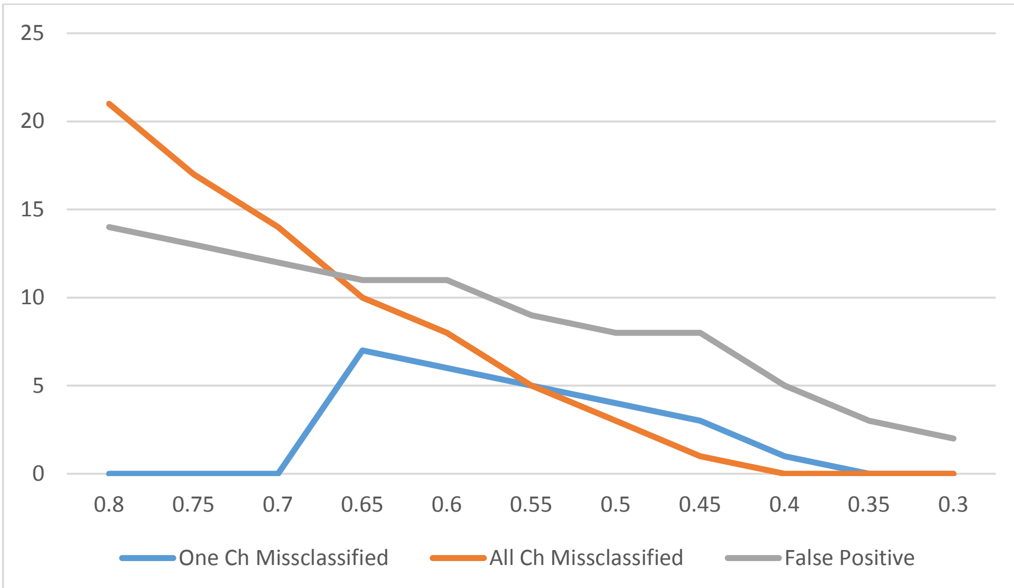


Figure 4- 2. Number of misclassified characters versus false positives for the OCR-NET

4. 2 Impacts of Different Activation Functions

In order to achieve a higher accuracy in our work, we evaluated and refined different parameters of the network. One of the most important factors that can affect the whole system's accuracy is the activation function. As mentioned, the activation function defines the output of each layer and it acts as an on and off switch for a network.

To evaluate our network, the ReLU activation function applied for all layers of the YOLO v2 vehicle detection network, except the last layer which is a linear activation function to detect the different types of vehicles.

The results indicate that the system cannot work properly by using ReLU only. Due to some fragile gradients which are perished in the training phase. In fact, if only this activation function is used, some of the data points lead to inactivity. Then, the activation function for the bottom half of the network were replaced with leaky activation functions and the system leads to failure. In the next step of our experiment on vehicle detection, all layers were replaced with leaky and the accuracy was less than 90% for all mentioned datasets which were not acceptable for an LPR system.

Then the last setting was carried on by changing the activation functions one by one and see each activation function impacts on certain challenging images. The results show that the proposed architectures for the YOLO v2 network and the OCR-NET are the most reliable setting. Furthermore, we evaluated ELU activation functions for those specific layers which we replaced by ReLU. Table 4.1 indicates the impacts of different activation functions on our methodology for only YOLO v2 Network.

Table 4- 1. The dissimilarity among three different activation functions on YOLO v2 network

Activation function	Formula	Open ALPR			CENPARMI
		EU	BR	US	TOTAL
ELU	$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & for\ x \leq 0 \\ x & for\ x > 0 \end{cases}$	77.73%	77.45%	75.11%	64.53%
Leaky	$f(x) = \begin{cases} 0.01x & for\ x < 0 \\ x & for\ x \geq 0 \end{cases}$	93.52%	91.23%	96.23	84.49%
Our Final Architecture	Leaky + ReLU	**98.54%**	**98.20%**	**96.39%**	**88.50%**

However, the ReLU and Leaky activation functions have a similar accuracy on Open ALPR (US) dataset, the system was not able to recognize the trucks by using Leaky function for all the layers. Also, the system faced more false positives in terms of vehicle detection.

4. 3 Example of Results and Discussions

In the following pages, there are some examples of the vehicles that our method detected accurately from different datasets.


```
Scanning /tmp/output/IMG_7665_0car_lp.png
        LP: FLS2702
```

Figure 4- 3. A cubic truck parked on the street, from CENPARMI

As seen in Figure 4-3, the system accurately detects and recognizes all the characters. However, the license plate is attached in a place that is not common for license plates. The system did not face any false positive, although there is a rectangular street scene text.


```
Scanning /tmp/output/TABRWET_0car_lp.png
        LP: TABRWET
```

Figure 4- 4. Complex background from CENPARMI dataset

In Figure 4-4 the vehicle is far from the camera and the license plate has a complicated background, but the system coped well with the situation and was able to recognize all the characters correctly.

Scanning /tmp/output/IMG_20170227_151008_0car_lp.png
LP: RG4293V

Figure 4- 5. One of the most challenging images in CENPARMI dataset

As illustrated in Figure 4-5, the image has a dark shadow and an oblique license plate attached to a cubic truck, and the system could perfectly detect and recognize the vehicle and license plate.


```
Scanning /tmp/output/KALSI_0car_lp.png
        LP: KALSI
Scanning /tmp/output/KALSI_1car_lp.png
        LP: FKM3590
```

Figure 4- 6. Multiple license plates in a single image from CENPARMI dataset

In Figure 4-6, two vehicles at different distances from the camera are shown and the proposed system successfully detected and recognized their license plates.

The following images are the samples of the pictures that the proposed system had difficulty to detect and recognize the vehicles, license plates, and characters partially or completely.


```
Scanning /tmp/output/JSG9648_0car_lp.png
        LP: JSG9648
Scanning /tmp/output/JSG9648_1car_lp.png
        LP: 0K06067
Scanning /tmp/output/JSG9648_2car_lp.png
        LP: 02224
```

Figure 4- 7. Multiple license plates in a single image from OPEN ALPR (BR) dataset

As seen in Figure 4-7, the system precisely detected 3 license plates but with the far vehicle, the system had problem in recognizing the characters correctly. Only 4 right digits of 7 characters from the far vehicle were detected by the system.

```
Scanning /tmp/output/wts-lg-000189_0car_lp.png
         LP: K43L
```

Figure 4- 8. The dark shadow and noisy image from OPEN ALPR (US) dataset

In Figure 4-8, the real license plate characters are “FK4W3L”, but the system only predicted few characters correctly. Due to the dark shadow and the low-resolution image, the system was unable to detect ‘F’ and ‘W’ from the license plate.


```
Scanning /tmp/output/test_055_0car_lp.png
        LP: N0450AM
Scanning /tmp/output/test_055_1car_lp.png
        LP: E5103Y
```

Figure 4- 9. One of the challenging images in OPEN ALPR (EU) dataset

In Figure 4-9, the closest vehicle to the camera is detected and the characters were recognized correctly, but the furthest vehicle from the camera that the characters on the license plate are “PD5108Y” could not be recognized correctly due to the blur and its far distance. The system only finds 4 characters correctly which are: “510Y”

The images which are shown below caused the false positive for the system, where there is no license plate but recognized some characters in different datasets.


```
Scanning /tmp/output/eu3_0car_lp.png
        LP: FWE50
Scanning /tmp/output/eu3_1car_lp.png
        LP: WE50
```

Figure 4- 10. The image from OPEN ALPR (EU) dataset

There is no other plate in Figure 4-10 as it illustrated, but the system copied mistakenly the parts of characters of the detected license plate and developed a false positive. The reason that such problem happened in this picture is because the threshold value for vehicle detection has been selected intentionally small in order to detect all the vehicles in different images.


```
Scanning /tmp/output/JRD2238_0car_lp.png
        LP: JRD2238
Scanning /tmp/output/JRD2238_1car_lp.png
        LP: JPCN
```

Figure 4- 11. The image from OPEN ALPR (BR) dataset

The system in the first network detected 3 vehicles in Figure 4-11, and in the WPOD-NET recognized 2 license plates. In the OCR-NET, while one of the predicted license plates did not have more than 3 visible characters in the image, by mistake, the system finds an extra ‘N’.

For the negative part, a negative will occurred if the system was unable to detect a vehicle. Also, a negative will occurred if the character network could not recognize a detected license plate's characters.

A few negative samples are shown in the following:

Figure 4- 12. The image from OPEN ALPR (BR) dataset

The license plate is too close to the camera in Figure 4-12, and the system has difficulty to recognize the characters of up-close license plates due to the selected bounding box size of the license plate, and the system is unable to detect the plate correctly.

Figure 4- 13. A challenging image from OPEN ALPR (US) dataset

The resolution of the image in Figure 4-13 is considerably low and it is even harder for a human to read the characters correctly at the first glance. The dark shadow made the situation worst and the system was unable to recognize the characters.


```
Scanning /tmp/output/53858J-79562F-2920V-80961G_0car_lp.png
        LP: 53858J
Scanning /tmp/output/53858J-79562F-2920V-80961G_1car_lp.png
        LP: 31BPQ
Scanning /tmp/output/53858J-79562F-2920V-80961G_2car_lp.png
        LP: 79562F
```

Figure 4- 14. The mixture of motorbikes and a sedan in an image from CENPARMI dataset

In Figure 4-14, there are four motorbikes and a sedan. The system only detected 3 vehicles in the image and recognized 100% correctly characters on the license plates.

As previously mentioned, the reason that the proposed system is unable to detect all the vehicles in this image is the selected threshold value. If we decrease the threshold of YOLO v2 network, the system is capable to find all 5 vehicles. For the other test samples, the false positive will increase significantly. In order to have a more accurate system, we ignored such misclassifications to avoid false positive.

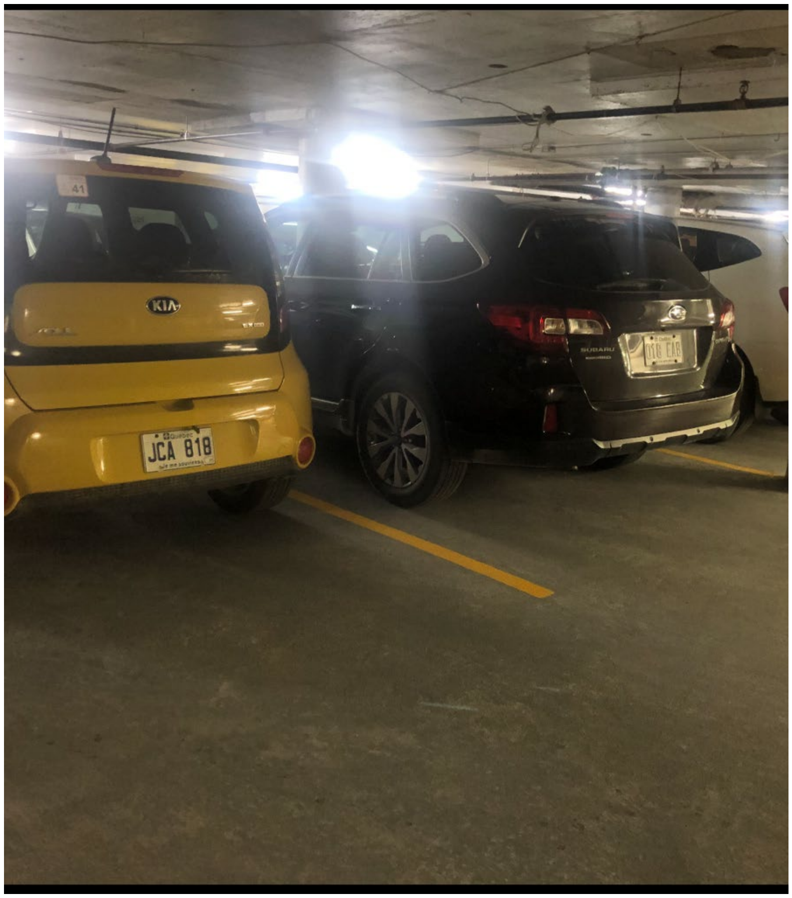
The following images show some of the challenges for the system that it coped with perfectly. The images are all from CENPARMI dataset, one of the most challenging datasets with the variety of images from Chinese, Taiwanese, US and Canadian license plates and they are purposely taken by high resolution cameras and from different complicated scenarios.

```
Scanning /tmp/output/IMG_20171011_165932_0car_lp.png
        LP: AHF3006
Scanning /tmp/output/IMG_20171011_165932_1car_lp.png
        LP: 7323MN
Scanning /tmp/output/IMG_20171011_165932_2car_lp.png
        LP: 1868DQ
Scanning /tmp/output/IMG_20171011_165932_3car_lp.png
        LP: 6452DS
```

Figure 4- 15. Chinese license plates with a resolution of 4032×3024


```
Scanning /tmp/output/IMG_20171015_124505_0car_lp.png
        LP: 2625WD
```

Figure 4- 16. An up-close Taiwanese license plate with the dimension of 4032×3024




```
Scanning /tmp/output/IMG_7411_0car_lp.png
        LP: 01GEAB
Scanning /tmp/output/IMG_7411_1car_lp.png
        LP: JCA818
```

Figure 4- 17. Light reflection on the far license plate, image is from CENPARMI dataset

In Figure 4-17, the illumination condition and the reflection of the light on the license plate attached to the right SUV made the situation complicated for recognizing the characters. But the proposed OCR-NET predicted 5 out of 6 characters correctly. While, the OPENALPR application was not able to detect the whole plate.

4. 4 Comparison with Commercial Automatic LPR Systems

According to the literature, a detected license plate by WPOD-NET network is a plate where the bounding box around the license plate is correctly adjusted and the OCR-NET is capable to recognize all the characters. The correctly recognized plate using OCR-NET is a license plate where all characters in a plate has a confidence over the threshold value. If the confidence is lower than the threshold, the network avoids making a prediction to reduce the risk of a false positive.

The overall prediction accuracy is calculated by the given formula:

$$Accuracy = \frac{True\ Positive}{True\ Positive + False\ Positive + False\ Negative} \quad (4.1)$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4.2)$$

The accuracy results of different datasets and commercial Automatic LPR systems are shown in Table 4-2.

Table 4- 2. Results of different commercial ALPR and academic dataset

	Open-ALPR			CENPARMI		UFPR-ALPR
	EU	BR	US	QC	US	
OPEN ALPR	96.30%	85.96%	-	90.54%	77.63%	96.47%
SIGHTHOUND	83.33%	94.73%	-	62.12%	52.76%	91.87%
Amazon Rekognition	69.44%	83.33%	-	-	-	-
Silva et al. [4]	93.52%	91.23%	89.24%	88.43%	71.34%	90.11%
Ours	**98.54%**	**98.28%**	**97.39%**	**96.77%**	**80.23%**	**97.42%**

The challenging part of the UFPR-ALPR dataset is the two-line license plate used for the motorbikes. Both OPENALPR application and our proposed methods are able to recognize the second line accurately, the OPENALPR completely ignores the first line, but our method predicts only one character of the first line.

The reason that the SIGHTHOUND application was not successful to deal with CENPARMI dataset is that this application has a major problem with tilted license plates, and most of images in CENPARMI dataset are from oblique plates.

Some of results on Open ALPR (BR) dataset is shown in the following table:

Table 4- 3. 50 Brazilian License plates from Open ALPR dataset

no.	license plate	System prediction	Accuracy
1	OCX4764	0CX4764	100%
2	OKS0078	0KS0078	100%
3	NZ06276	NZ06276	100%
4	PJG0783	PJG0783	100%
5	0UH9191	0UH9I9I	77.77%
6	JSP7678	JSP7678	100%
7	0KV8004	0KV8004	100%
8	0DJ1599	0DJI599	87.5%
9	0YJ9557	0YJ9557	100%
10	PJB2414	PJB24I4	87.5%
11	0LA1208	0LAI208	87.5%
12	0UP9563	0UP9563	100%
13	AY09034	AY09034	100%
14	AZJ6991	AZJ699I	87.5%
15	0ZU5764	0ZU5764	100%
16	PJD2685	PJD2685	100%
17	0ZG3580	0ZG3580	100%
18	0LC4728	0LC4728	100%
19	JQV5526	JQV5526	100%
20	NZW2197	NZW2I97	87.5%
21	0UG6157	0UG6I57	87.5%
22	JIY4434	JIY4434	100%
23	PJY5472	PJY5472	100%
24	0KL0817	0KL08I7	87.5
25	PJB7392	PJB7392	100%
26	NYY1710	NYYI7I0	77.77%
27	NTK5785	NTK5785	100%
28	PJV9741	PJV974I	87.5%
29	FZB9581	FBZ958I	87.5%
30	PJU2853	PJU2853	100%
31	PJP8208	PJP8208	100%

32	PJH0957	PJH0957	100%
33	PJI5396	PJI5396	100%
34	NZJ6581	HJ658I	87.5%
35	0ZV6697	0ZV6697	100%
36	0EL1145	0ELII45	77.77%
37	0UM7311	0UM73II	77.77%
38	0UN4297	0UN4297	100%
39	PJT4884	PJT4884	100%
40	NZM5430	NZM5430	100%
41	MYX3152	YX352	77.77%
42	PJY8509	PJY8509	100%
43	PJF4224	PJF4224	100%
44	PJJ4955	PJJ4955	100%
45	PJP2783	PJP2783	100%
46	0LB4809	0LB4809	100%
47	PWC0633	PWC0633	100%
48	NYL3614	NYL36I4	87.5%
49	NTV0498	NTV0498	100%
50	JRV1942	JRVI942	87.5%

The reason that the system has difficulty to distinguish ‘1’ from ‘I’ in Brazilian dataset is because of the font face of the number ‘1’. This problem can be solved by re-training the system with more annotated Brazilian datasets.

Figure 4- 18 A sample of Brazilian license plate with the number ‘1’

4. 5 Speed testing of the system

The speed of the proposed system certainly depends on the number of vehicles in a single input image. For instance, an image with the resolution of 1024×768 with three vehicles needs

0.12 second for detection and character recognition with an Intel Corei7 2.8 GHz processor 4 Core, 16 GB of RAM and a Radeon pro 555 GPU. The most time-consuming part of our system is the WPOD-NET, which is responsible for license plate localization and correction in terms of obliqueness, and the fastest section is character recognition network.

CHAPTER 5

Conclusions and Future Work

5. 1 Conclusions

This thesis has proposed a pipeline to tackle the ALPR task based on deep learning techniques. We utilized a modified version of YOLO v2 on Pascal-VOC to detect the vehicles. A WPOD-NET system was employed in order to localize and rectify the distortion of license plates. Furthermore, through the convolutional procedures, we modified a YOLO based Optical Character recognition network (OCR-NET) to recognize the license plate numbers and letters within a cropped license plate from different regions.

Our method performs accurately for different vehicle types such as sedans, SUVs, buses, motorbikes, and trucks. The system works equally well on images of the front and rear views of the vehicle. The system overcomes and performs adequately in different conditions, noisy backgrounds, and tilted/distorted license plate images. We evaluate several experiments and refine the network parameters in order to achieve a better result under a variety of conditions, and our experimental results show that the proposed system outperforms the state-of-the-art commercial and academic methods for challenging images. Our proposed system achieves 98.04% accuracy on average for OPEN-ALPR dataset, 88.5% for challenging CENPARMI dataset and 97.42% for UFPR-ALPR dataset respectively.

5. 2 Future work

In order to have a universal LPR system, we are looking to train our model to recognize Arabic, Persian characters as well as Chinese characters and enable the system to classify more than 35 Latin characters. There are many different annotated datasets for non-Latin characters that we can take advantage of, although it needs a more powerful GPU and may take several days to train such a system.

Our proposed method has the ability to be imported to a mobile device with appropriate processing speed. Some front-line mobile phones (Apple iPhone 8, X, XR, Google Pixel 2, 3, etc.) have GPUs and by modifying and downsizing the system, it is possible to have an application for mobile phones to detect and recognize license plates with the high accuracy at a satisfactory speed.

References

[1] Automatic License Plate, https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/a/ Automatic_number_plate_recognitio.htm, last access: May 7, 2007.

[2] The License Plate Game – Florida Style, https://justlikefamilyhomecare.wordpress.com /2015/ 08/, last access: Nov 25, 2015.

[3] Never get a ticket again by changing your license plate to something unreadable! https://www.reddit.com/r/ShittyLifeProTips/comments/bltm64/lpt_never_get_a_ticket_again_by_changing_your/last access: May 10, 2019.

[4] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," In Proc. of European Conference on Computer Vision, Munich, Germany, Sep 2018, pp. 593–609.

[5] Complete set 50 USA license plates, https://www.ebay.com/itm/complete-set-50-usa-license-plates-states-plus-bonus-plates-mancave-usa-lot-/264087061175 last access: Aug 08, 2019.

[6] Redmon, J., Divvala, S., Girshick, R., Farhadi. A.: "You Only Look Once: Unified, Real-Time Object Detection." In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA, pp. 779–788. IEEE Jun 2016.

[7] Redmon, J., Farhadi, A.: "YOLO9000: Better, Faster, Stronger." In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6517–6525. IEEE, Jul 2017.

[8] Personalized license plate, Government du SAAQ Québec, https://saaq.gouv.qc.ca/en/ vehicle-registration/ personalized-license-plates/ last access: Jan 01,2019.

[9] License-plate spray foils traffic cameras, https://www.washingtontimes.com/news/2003 /Jul./3/20030703-120901-3612r/ last access: Jun 03, 2003.

[10] Standard License Plates, https://www.dmv.ca.gov/portal/dmv/detail/pubs/plates/ standardplates last access: Jan 01, 2019.

[11] Zheng L., He X. "Character Segmentation for License Plate Recognition by K-Means Algorithm." In proc. of Image Analysis and Processing, ICIAP 2011. Lecture Notes in Computer Science, vol 6979, pp. 444-453, Springer. Berlin, Heidelberg 2011.

[12] Sahar S. Tabrizi; Nadire Cavus, "A Hybrid KNN-SVM Model for Iranian License Plate Recognition," In Proc. of 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria, Procedia Computer Science, Vol. 102, no 5, pp. 588-594, 2016.

[13] Soungsill Park, Hyoseok Yoon and Seho Park, “Multi-Style License Plate Recognition System using K-Nearest Neighbors,” KSII Transactions on Internet and Information Systems, vol. 13, no. 5, pp. 2509-2528, 2019.

[14] R. Mukherjee, A. Pundir, D. Mahato, G. Bhandari, and G. J. Saxena, “A robust algorithm for morphological, spatial image-filtering and character feature extraction and mapping employed for vehicle number plate recognition,” In Proc. of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2017, pp. 864-869.

[15] H. Rajput, T. Som and S. Kar, "An Automated Vehicle License Plate Recognition System," in Computer, vol. 48, no. 8, pp. 56-61, Aug 2015.

[16] K. S. Satwashil and V. R. Pawar, "Integrated natural scene text localization and recognition," In Proc. of International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2017, pp. 371-374.

[17] J. Jagtap and S. Holambe, "Multi-Style License Plate Recognition using Artificial Neural Network for Indian Vehicles," In Proc. of International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, India, 2018, pp. 1-4.

[18] Y. Luo, Y. Li, S. Huang and F. Han, "Multiple Chinese Vehicle License Plate Localization in Complex Scenes," In Proc. of IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, Jun 2018, pp. 745-749.

[19] B. Suvarnam and V. S. Ch, "Combination of CNN-GRU Model to Recognize Characters of a License Plate number without Segmentation," In Proc. of 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 317-322.

[20] X. Hou, M. Fu, X. Wu, Z. Huang and S. Sun, "Vehicle License Plate Recognition System Based on Deep Learning Deployed to PYNQ," In Proc. of 18th International Symposium on Communications and Information Technologies (ISCIT), Bangkok, Thiland, Sep 2018, pp. 79-84.

[21] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," In Proc. of the 35th International Convention MIPRO, Opatija, Croatia, May 2012, pp. 1725-1730.

[22] B. Hutchings and M. Wirthlin, "Rapid implementation of a partially reconfigurable video system with PYNQ," In Proc. of 27th International Conference on Field Programmable Logic and Applications (FPL), Ghent, Belgium, Sep 2017, pp. 1-8.

[23] C. Lin, Y. Lin and W. Liu, "An efficient license plate recognition system using convolution neural networks," In Proc. of IEEE International Conference on Applied System Invention (ICASI), Chiba, 2018, pp. 224-227.

[24] Aggarwal, C. C. (2018), neural networks and deep learning, Springer, pp. 9-54, Berlin. Germany ISBN: 9783319944623. 9783319944630 (online).

[25] P. Wu, Z. Huang and D. Li, "Research on the character recognition for Chinese license plate based on CNN," In Proc. of 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, Jun 2017, pp. 1652-1656.

[26] G. E. Nasr, E. Badr, and C. Joun, "Cross entropy error function in neural networks: Forecasting gasoline demand," In Proc. of FLAIRS Conference, pp. 381–384, 2002.

[27] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. https://towards datascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 last access: Dec 15, 2018.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," In Proc. of IEEE, vol 5, Nov 1998, pp.1:541–551.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, June 2014, pp. 3431–3440.

[30] D. Hubel and T. Wiesel, "Receptive fields and functional architecture of money striate cortex," Journal of Physiology, vol. 6, pp. 215–243, 1968.

[31] R. Yang, H. Yin and X. Chen, "License Plate Detection Based on Sparse Auto-Encoder," In Proc. of 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, May 2015, pp. 465-469.

[32] L. Zuo, H. Sun, Q. Mao, R. Qi and R. Jia, "Natural Scene Text Recognition Based on Encoder-Decoder Framework," In Proc. of IEEE Access, vol. 7, pp. 62616-62623, 2019.

[33] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," In Proc. of IEEE

Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017.

[34] K. Wang, B. Babenko, and S. Belongie. “End-to-end scene text recognition.”, In Proc. of IEEE International Conference on Computer Vision, pages 1457–1464, Washington, DC, USA, Nov 2011.

[35] Jørgensen, Hogne. Master thesis with the title of: “Automatic License Plate Recognition using Deep Learning Techniques.” Norwegian University of Science and Technology, Trondheim, Norway, Jul 2017.

[36] Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S.: “A New CNN-Based Method for Multi-Directional Car License Plate Detection.” In Proc. of IEEE Transactions on Intelligent Transportation Systems 19, vol 2, pp. 507–517. Feb 2018.

[37] Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gon¸calves, G.R., Schwartz, W.R., Menotti, D. “A robust real-time automatic license plate recognition based on the YOLO detector.” CoRR abs/1802.09567, 2018.

[38] Hsu, G.S., Ambikapathi, A., Chung, S.L., Su, C.P. “Robust license plate detection in the wild.” In Proc. of 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 Lecce, Italy, Aug 2017.

[39] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z.Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L. “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision (IJCV) vol 115, pp. 211–252, Sep 2015.

[40] Silva, S.M., Jung, C.R. “Real-time Brazilian license plate detection and recognition using deep convolutional neural networks.: In Proc. of 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 55–62, Niteroi, Brazil, Oct 2017.

[41] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K. “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors.” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3296–3297, Hawaii, USA, Jul 2017.

[42] Cars dataset https://ai.stanford.edu/~jkrause/cars/car_dataset.html last access : Dec 21,2017.

[43] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P.Zitnick, C.L. “Microsoft coco: Common objects in context.” In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – In Proc. of ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014).

[44] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, Nov 2016, pp. 770-778.

[45] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” International Journal of Computer Vision, vol. 111, pp. 98–136, Jan. 2015.

[46] A. Veit, M. J. Wilber, and S. J. Belongie, “Residual networks are exponential ensembles of relatively shallow networks,” CoRR, vol. abs/1605.06431, 2016.

[47] OPENALPR dataset benchmark, https://github.com/openalpr/openalpr last access: October 2018.

[48] Gonalves, G.R., da Silva, S.P.G., Menotti, D., Schwartz, W.R. “Benchmark for license plate character segmentation.” Journal of Electronic Imaging, vol 5, 1–5. 2016.

[49] Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k. “Spatial transformer networks.” In Proc. of Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R.(eds.) Advances in Neural Information Processing Systems (NIPS). vol.28, pp. 2017–2025 Montreal, Canada, Dec 2015.

[50] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu and Alexander C. Berg. “SSD: Single Shot Multi-Box Detector.” In Proc. of ECCV, 2016, vol. 9905, pp. 21-37, Amsterdam, Netherlands, Oct 2016.

[51] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features indeep neural networks?” In Proc. of Advances in Neural Information Processing Systems (NISP) 27 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3320–3328, Curran Associates, Inc., 2014.