Recognising License Plate from Image Data using Deep Learning

by

Rupam Golder
18301040
Sadman Rahman
18301037
MD. Isteak Shams
18301228
Jawad Hasan
18301242
Fahim Muhammad Bakhtiar
18301095

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rupam Golder

Rupam Golder
18301040

Sadman

Sadman Rahman
18301037

shopnil

MD. Isteak Shams
18301228

Jawad

Jawad Hasan
18301242

Fahim Muhammad Bakhtiar

Fahim Muhammad Bakhtiar
18301095

Approval

The thesis titled "Recognising License Plate from Image Data using Deep Learning" submitted by

1. Rupam Golder (18301040)
2. Sadman Rahman (18301037)
3. MD. Isteak Shams (18301228)
4. Jawad Hasan (18301242)
5. Fahim Muhammad Bakhtiar (18301095)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Septmeber 20, 2022.

Examining Committee:

Supervisor:
(Member)

Tawhid

MD. Tawhid Anwar
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Maintaining surveillance and security on the roads has become a significant challenge, which is why it has become necessary to conduct proper methods to control this problem. To ensure security and safety on the roads, methods such as Automatic Number Plate Recognition is implemented so that all crimes and other security-related issues may scale down. In this paper, two versions of YOLO are used. The first one is YOLOv5, and afterward, the most recent model of YOLO, which is YOLOv7. These models are used so that we may get better results compared to all other previously used models. Eventually, EasyOCR is used to extract the characters from the number plate. The proposed models are tested on the LP dataset, our custom dataset consisting of 10,700 images. 93.8% and 95.6% accuracy are acquired from YOLOv5 and YOLOv7, respectively. However, the main goal of this research paper is to prove that YOLO is superior to other models in terms of object detection. In addition, YOLOv7 provides us with improved results compared to YOLOv5.
Keywords: License Plate, EasyOCR, YOLOv5, YOLOv7.

Dedication

We dedicate the report to Almighty God, our parents and our advisors. Without their participation, attention, and support, we would not have gotten this far. We owe a debt of gratitude to them. Many thanks to them.

Acknowledgement

We want to start by giving thanks to the Almighty GOD, without whom we could not have finished our thesis. Also, we'd like to extend our gratitude to MD. Tawhid Anwar sir, our primary supervisor, and to Rafeed Rahman sir, our secondary supervisor, for all of the help and guidance they've given us during this project. We could always count on his assistance.
Finally, we owe a great deal to our father and mother; without their unwavering support, none of this would have been possible. Our success to this far would not have been possible without their prayers and encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANPR Automatic Number Plate Recognition

CNN Convolutional Neural Networks

DNN Deep Neural Networks

LP Licence Plate

NS Number of Samples

TN True Negative

TP True Positive

YOLO You Only Look Once

Chapter 1

Introduction

Safety on the road has become a big issue for the people not only in this country but around the world. As with each passing day, the endangerment is just getting much worse than before. The safety of people on the roads plays a major part in people's minds when going out. In this case surveillance and security on the roads become a major issue for the people of every country. To make sure the security in every sectors, many countries have increased the uses of video as public surveillance for the monitoring of people in order to prevent domestic crimes on the streets. The importance of surveillance cameras or CCTV cameras in our daily life is increasing day by day. With each passing day seeing how digital data is collected, stored, analyzed, and shared which has changed how video surveillance systems are being done during the last ten years. In developed cities and countries like China and USA, for smart security movement security cameras plays a big part in city movement. For the sake of gathering data and in order to make prediction for integrated analytic AI and deep learning are being used the most. So, surveillance cameras have become one of the most ubiquitous and common technologies to track vehicles on the street. Some of these cameras are capable of 360-degree video or infrared vision. Some of these cameras which are used in day-to-day life for public surveillance are; Bullet cameras, Dome cameras, PTZ cameras, Poles of Mobile Surveillance, Thermal cameras, Automated license plate readers or the ALPR. Even though having a huge number of options for the video surveillance systems, still there are some flaws and difficulties in the process. However, in rainy or muddy weather or during night time number plate detections are not accurate enough and for surveillance purposes, LP detection plays a vital part. In most of the country with each passing year, the population is also growing at an alarming rate. So is the number of vehicles. So, it is quite tough to keep track of all these vehicles' license plates. Therefore, for this reason, crime on the street has also increased. To find out which vehicle has either committed a crime or did not pay the toll or maybe even kidnapped someone recognizing the license plate is a must. And that's where License Plate detection plays one of the most vital roles. This is how number plate detection helps to boost the security of a sector by increasing its security level by detecting number plates from cars.This research aims to properly detect all kinds of license plates from different regions and extract the characters from that detected license plate properly for different purposes. We utilize the dataset LP-Detection to further study in license plate detection using deep learning. This dataset has around 10,700 images and a single class which is the license plate class. This research illustrates ways to detect license plate from images. We implemented YOLOv5 [26] and YOLOv7 [17] in which we achieve high levels of accuracy and compared other models such as CNN [2], Tensorflow [11], OpenCV [16] with the most recent model which is YOLOv7. This study compares the performance of other models with the implemented models.

1.1 Research Problem

Making sure of the safety and proper surveillance on the roads has become one of the most alarming issues around the world. Recently around the world vehicle theft, kidnapping, not maintaining proper rules, and not giving tax or tolls have increased by a large margin. So, to make sure of security and surveillance of roads together, detection and recognition of license plates together is the top priority. However, there are several approaches have already been used to ensure road safety and security. But among them, detecting License plates, extracting the number from the Plate, and recognizing the number has become one of the most vital and useful methods. Besides, there are so many models have been used to detect and recognize license plates with the help of either an updated proposed model like RP-net [15] or using many existing models like CNN , OCR [24], YOLOv3 [14], TensorFlow and many other models. But it sees an overlap or creates some problems which results in less accuracy and not being able to detect properly. The main problem is detecting the license plate. Also, for oblique angles, it is still difficult to fully recognize the license plate. As the algorithms used in this case still are not able to fully centralize the license plate without some sort of distortion for the method to recognize it properly. In bad weather conditions and because of dirty license plates it is still hard for the module to recognize the character properly. Moreover, after detecting the plate sometimes suggested modules become confused when there are many texts on the plate. It fails to recognize the exact texts which refer to the registration number of the car. As a result, vehicle and road safety cannot be ensured properly.

1.2 Research Objectives

Our main goal is not only to detect and recognize the License Plate together but also to make sure that this approach is faster and gives us more accurate results than other proposed models. So, to reach the goal perfectly, we are proposing an improved and refined Deep Learning model with the help of some additional already existing models like TensorFlow, OCR, and some algorithms to properly detect and recognize license plates at the same time from an oblique angle and in any sort of environment.

- Detecting and recognizing LP from any oblique image of CCTV or any footage used in the dataset.
- Detecting and recognizing LP in any kind of weather condition.
- Extracting the exact text (Registration number of LP) from the LP.
- Using different modules, and libraries and modifying or updating them to make sure the detection and recognition are faster and more accurate.

There may be other choices, but additional research is needed.

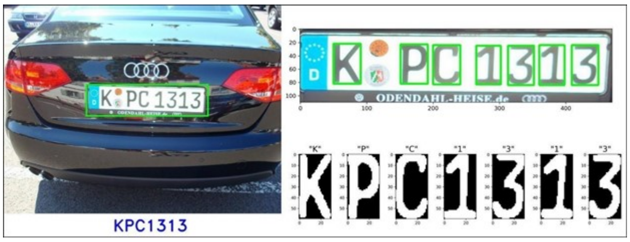

Figure 1.1: LP Detection.1

Figure 1.2: LP Detection.2

1.3 Thesis Structure

In chapter 1, we discussed the Research problem and research objectives. In chapter 2, we discussed Literature review and deep learning. In chapter 3, we discussed Methodology and proposed models. In chapter 4, we discussed Dataset and data augmentation. In chapter 5, we discussed Implementation and results. In chapter 6, we discussed the Conclusion and future work.

Chapter 2

Background Study

2.1 Literature Review

For ensuring security, surveillance systems use Automatic Number Plate Recognition to identify vehicles. It is being used in several sectors, like parking lot management, automated toll collection, traffic monitoring, and traffic law enforcement which are all applications of Automated Number Plate Recognition. Steps for how the Number Plate Recognition works are a major discussion topic in recent times.
In the following paper “Number Plate recognition using Improved Segmentation” [12] advances with the goal of detecting the number plate area, breaking down the characters in the number plate, and also optical character recognition. The base of ANPR is built upon multiple methodologies that contain Artificial Neural Network (ANN), Probabilistic Neural Network, OCR, MATLAB, Configurable Method, and Sliding Concentrating Window but [12] used the template matching technique and implemented it on the ANPR system to recognize number plates on vehicles. The proposed method in [12] to detect license plates includes Binary Image Processing where the method extracts characters from binary images. This process has an accuracy rate of 98 percent among 9745 images assuming the number plate frame’s edges are plain and perfect. Image color is given a high preference here in order to detect the license plate and also to extract the characters in the number plate. As greyscale and also color processing is done within the suggested method in this paper along with adaptive thresholding it is easier to figure out if the image needs to be black and white or not. Also, Contrast expansion is also done in case of improving the sharpness of the image and to adjust the noise Median Filtering is done. For character segmentation, MATLAB is used and also feature extraction is used to perform Zonal Density. Databases of alphanumeric characters templates are used where characters from A-Z, a-z, and 0-9 are all present for comparison. In the end, an accuracy of 82.6 percent was obtained after taking images from static vehicles for Indian number plates.
Whereas the paper “Automatic License Plate Recognition using Extracted Features” [3] focuses mainly on plate identification and image quality. Focusing on the image quality while designing this ALPR model and also these three License Plate Localization, Segmentation of character and Recognizing Optical character are the main focus of this paper [3]. License plate localization identifies the license plate region from the image, then character segmentation identifies all of the characters on the number plate, and after that OCR detects all of the segmented characters

using template matching. It is observed that a license plate recognition algorithm was used to detect inclined number plates from binary images, this algorithm gives 97 percent accuracy but can identify only one kind of license plate. The Vertical Edge Detection Algorithm which is used in [3]. Comparing to Sobel approach, it's 7-9 times faster, which requires a noticeable edge, the accuracy rate of this method is 92.5 percent. After dilating the gathered picture with a flat disk as a structural element, a grayscale image is formed using the [3] approach, which then increases light sections surrounded by dark regions. Following the Sobel process, the image is normalized to remove unwanted lines and then converted to a binary image using the image normalization method. A histogram diagram generally shows the region with different intensities, this sudden change of intensities might give wrong results which is why a low pass average filter is used to smoothen the abrupt changes. After discovering the number plate's region, the character is recognized and extracted using the character segmentation process. This procedure converts the image into a binary image and removes tiny blobs of color from the image. The characters are removed from the number plate in row main orders. The algorithm for character recognition (ALPR) performs OCR using template matching at the end of the procedure. When comparing two photos, the cross-correlation establishes the degree of resemblance. A total of 500 real-time images were used to evaluate the ALPR approach, with an accuracy rate of 84.8 percent in under 0.5 seconds.

The following paper "Improved OCR based automatic vehicle number plate recognition using features trained neural network" [4] primarily concerned with traffic management and the enforcement of traffic laws. It has become very important to make a recognition system that is automated and fast. ANPR could be the solution for a lot of problems as it can detect the license plate even after the vehicle is used in different conditions, like different weather conditions and light conditions. To get a good output of the image, the color image is converted to HSV and it is converted to a binary image for future steps. In edge detection, mainly two types of processes are used. One is horizontal and vertical edge detection and the other is Sobel edge detection. It is important that the localization algorithm accuracy is high otherwise Character recognition technique would not work properly. There are five steps to follow in localization which are localization using technique of projection, clipping band vertically, selecting proper band, clipping band horizontally , correction of skew, and segmentation. After that comes character recognition from extracting features and this approach utilizes the fact that there are some features which every character has such as like corners, ending, and bifurcation and this makes it faster and easier. After that, the feature sets are trained to utilize a neural network, and then it prepares to train multiple images to get better accuracy. The whole system gives an accuracy of 94.45 percent and that too in less than 1 second. The system has been applied to 300 national and international motor vehicles and got the result after that.

The authors in the paper "Performance Analysis of Vehicle Number Plate Recognition System Using Template Matching Techniques" [7], shares their view of how to determine a vehicle's number plate by capturing an image with a digital camera. In this process, the number plate is scanned through computer vision from a real image of the vehicle. Later, a number is being extracted for a unique recognition code to identify a specific vehicle. This process uses various algorithms, mainly including Normalized Cross Correlation and Phase Correlation Algorithms to recognize vehi-

cles using template matching. In this paper we mainly see Gray Scale Conversion, Sobel Edge Detector, Dilation, Erosion, Holes Filling, Opening and Closing and Normalized Cross Correlation for the theories of VNPR. The general NPR methodologies include input image where the first stage is to take images with a digital camera of 5 megapixel where the inputs are 1403×677 or 1932×2576 or 960×960 pixels. Then it is converted into an image which is in grayscale and with help of a Sobel edge the edges are detected to recognize the rectangular plates. After that, dilation helps make the objects in the image bigger and erosion is unfavorable of dilation. The structure of the rectangle is also identified through dilation. Filling holes algorithm is used to fill the holes which are created due to the dilation process. After that, the image is smoothed out using the erosion technique. After smoothing the image, a 2-D median filter with mask 3x3 is used, and some items that are not rectangular and have no probability of being plates are removed. Next the segmentation is done. Here, the license plate is divided into several parts according to the number of characters on the plate individually. Then character recognition is done for the validation of characters. The letters and digits are cut into 70x70 blocks. Then the phase correlation and normalized cross-correlation are used to match with the characters in the database. During these 687 samples were obtained to evaluate the process, with 467 being recognized by normalization and 436 being recognized by phase correlation. The average accuracy was 67.98 percent on normalized cross-correlation and that of 63.46 percent was for phase correlation.
After that, in the paper "Number Plate Recognition Using an Improved Segmentation" [1], the automated identification method used for security purposes It will not permit suspicious cars to enter any security zone, as their information is already stored in the dataset. This system is mainly divided into two models. The software model is the most important one. Here the NPR algorithm is used and divided into some parts which are implemented in MATLAB. At first, a picture needs to be taken and stored in JPEG format for later use. Then with the help of image processing, the noises are removed from the JPEG file. After those noises are removed from the pictures, the number plate is extracted, which can be done in two ways. Then the plate character segmentation and character recognition are done. In the hardware model, first, the data is sent and then with the help of the microcontroller, the data is compared with the standard data to confirm the authorized numbers. The system helps to detect the number plate and the algorithm has been used on a lot of images. This project is able to replace the system of manual entry. The accuracy rate or extraction of plate region is 96 percent and the accuracy of character recognition is 93 percent in this project. The average execution time is 45 seconds where the image quality is 480*640.
Later in the paper "License Plate Detection and Recognition in Unconstrained Scenarios" [8], it is mentioned that ALPR is currently the most popular method being used with the help of OCR and DEEP LEARNING. The problem that mostly hinders this process are tight angle pictures and rough environment. This particular paper uses a complete ALPR system which helps in recognizing the license plate in any unconstrained scenarios with multiple camera setups also adding the mass use of synthetically warped version of real images so that the network is able detect all the license plate from scratch with the help of augmentation and allowing a reformation before the OCR. This particular ALPR system was developed with the help of OCR, to maintain the success rate of an already developed network with the

name of YOLO-v2. Also, to make the images into the right size a resizing factor was used along with Warped Planner Object Detection Network which was developed to detect the LP in any tough condition and also to make sure that the distortion is less in the provided image. To recognize the LP faster and to separate objects from non-object insight from an existing network called WOOD-net was used. Also, this network was used to make sure that the unwrapped LP is in the right shape. Various augmentation was used in this paper such as rectification, aspect ratio, centering, scaling, rotating, mirroring, translation, cropping, color-space, annotation; the network was trained to be able to give very distinct visual characteristics from any manual sample. For testing the network of this paper two dataset one from BR and one from EU was used. So that this particular ALPR system can be used in various regions' cars. This paper saw an increase in 5 percent accuracy for both the BR and EU dataset with the accuracy climbing to 89.33 percent for the AOLP RP dataset. Though there are some problems here with the detection oblique images of LP still it saw a 7percent increase in accuracy compared to the other papers for a single region dataset.
We can see in paper, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline" [10], detecting and recognizing LP has been one of the top most priorities in almost all the countries for proper surveillance and security in the roads. From controlling traffic to preventing car theft and also to collect tolls in the roads LP has become the face of all. This paper focuses on getting better accuracy using a large dataset only using a single method whereas, most of the papers focuses on a small stat set and uses two methods which makes the process slower and less accurate. An architecture called Roadside Parking net was designed which was able to do the LP detection and recognition in a single process. The model known as RP-net has two models. The first one known as the detection model, is a deep convolutional neural network which has 10 layers to process the image. It basically supplies the feature map with three siblings, that are fully connected layers for box prediction. The later part known as the recognition module is used for the ROI layer and to predict the regions classifiers are used and to use the number of the plate. The dataset used in this paper is one of the largest LP detection datasets which is (over 250000 cars image) CCPD with many sub datasets. It basically collects images from a pivotal car parking management of China. Because of the large number of datasets, a loss factor is taken into consideration to make sure that the accuracy is still kept high. The dataset is mainly trained into two different parts. Among them, localization loss and classification loss . This model produces a better accuracy rate compared to other papers and also provides all in better FPS.
Finally, the paper "Deep Learning System for Automatic License Plate Detection and Recognition" [5] focused on the method of properly detecting the LP in low light and from various angles. It [5] mentions that Most authors in their papers suggested the method to LP detecting in three ways which consists of LP detection, character segmentation and recognition. Using context shapes, morphological operations in case of gray image, contour detector, colors-based approach is often used to identify and recognize LP of different regions. Also, methods such as sliding window, fuzzy logic, CNN are used to properly segment the characters. Among a lot of segmentation algorithms, the main ones are projection algorithm, mathematical morphology, contours, local and adaptive threshold. For character recognition multi-stage classification schemes with three layered CNN model and CNN model are used. Here [5]

a Morphology filter which uses hat transformation is used to get the proper color saturated image and also to get every small detail of the image. A Gaussian Blur filter is used to make sure that the picture does not have any unwanted noise and also for removing any details that are not needed. Also, an Adaptive Threshold algorithm was used. It [5] used a Geometric Filtering to make sure that the processed image is of the right size. Later a CNN model is used to make sure that the image is maintaining the proper criteria and the method DEEP LEARNING architecture is used to find out the difference between LP and non-LP. Finally, a classifier made by the author is used to check the boundary box, keeping in mind with all the complexities the authors firstly derived from a RGB to grayscale image then using various methods to make change in threshold and finally using the same contour technique to make sure of the accuracy of the segmentation. This paper used the following data-sets which are LP (AOLP, CALTECH) and non-LP (Microsoft Research Cambridge Object Recognition) characters. The proposed model in terms of precision, recall and f-scoring rate sees the success rate respectively 94.8,96.2, 95.4 and 95.1 (Caltech, ac, le, rp) which indicates the increase in accuracy in all aspects.

2.2 Deep Learning

The characteristics retrieved from the image pertain to the “graphics” of the model, and the object’s choice is quite problematic. It was essential in classifying performance in the past, but it was also labor demanding and subjective expert work that manually derived characteristics. In addition, we cannot manually extract many of the features accurately. Consequently, a technique that would automatically identify appropriate functionality for a problem with a defined logic endeavored after.
Deep learning is an AI characteristic that replicates data-processing function of the human brain’s for object recognition. Artificial neural networks automatically extract data from collected samples by learning the proper representation and applying a solid model. This automatic removal proof is accurate to computer vision, cutting-edge image technology, recognition of objects, and image recall models. (Bengio, Courville, & Vincent, 2013) [9]. Deep learning is based on widely used ANNs that apply mathematical models in the biological, neural, centralized animal nervous system and brain-inspired learning algorithms. Neuronal networks consist of one or more deep neuronal learning layers, which combine the bulk of the Artificial Brain, composed of numerous hidden layers. ANN expanded more extensively via the flow of information and dispersed biological communication nodes but still varies in several ways from the human brain. The term “deep” is used here to signify that this network contains more than one layer. Initiators and tech businesses utilize profound learning and IoT technology to enhance agricultural yield [13]. Deep learning makes use of Neural Networks to understand how the human brain operates.

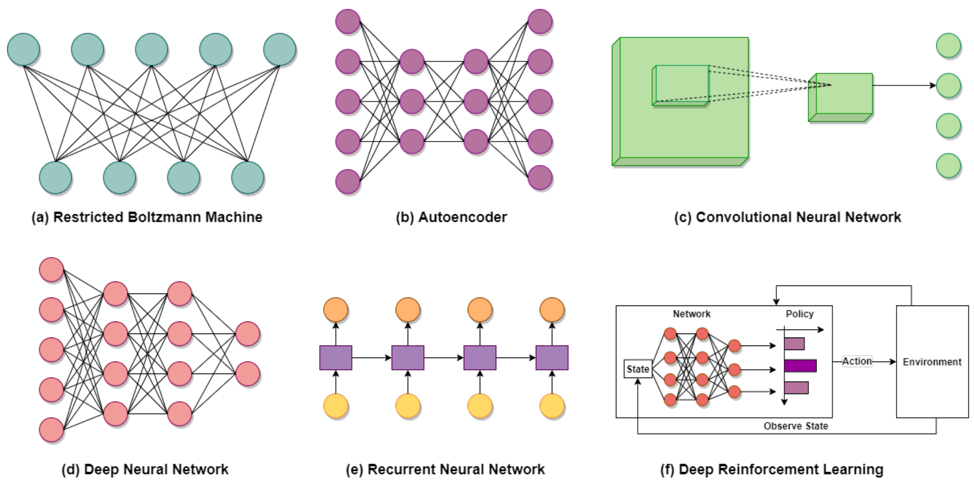

Figure 2.1: The structures of different deep learning models.

These networks are like distinct nodes (or places) linked in a single layer in the human brain. The more the number of layers, the stronger the network gets. Information travels between nodes using signals. The relevant weights are applied to the nodes when these signals are received. Heavier-weight knots will increase the impact on the other neighboring nodes. The weighted inputs are transformed to outputs afterward. The entire system required costly hardware since vast data, including multiple complex computations, had to be processed.
The complete procedure may be conducted using the image and processing description, object sensing network, and model optimization for insect and pesticide identification.

2.3 YOLOv3 [29]

YOLO or You Only Look Once is an architecture introduced by Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi which is used in deep learning. YOLO uses a completely different methods than the other algorithms used for Object detection. The YOLO machine learning algorithm uses deep convolutional neural network trained features to identify objects. Also, YOLO is really popular in the recent research works as it has a high accuracy and also it can be used in a real-time or real-time applications. The name You Only Look Once came from the concept of the algorithm as it only looks once at the image that has been inputted and only needs one forward propagation pass to do the predictions.
YOLOv3 is based on the idea: “A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.”[29]
YOLO works as a real-time object identification similar to CNN. CNN are systems that use classifiers as their foundation. These use input images as organized arrays of data to find patterns between the images. A grid is created from an image using the Yolov3 method.

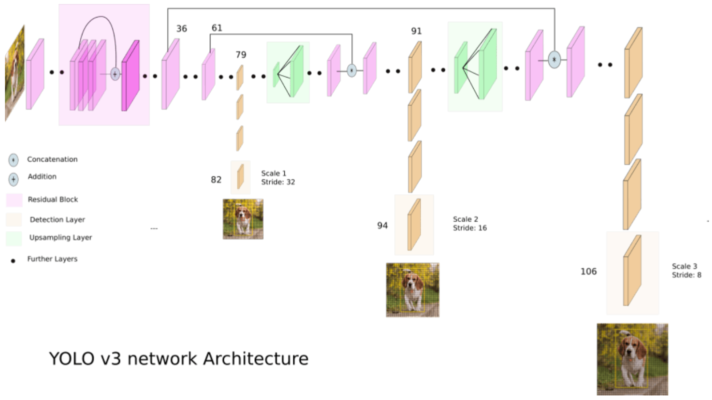

Figure 2.2: Network Architecture Diagram of YOLOv3 [29]

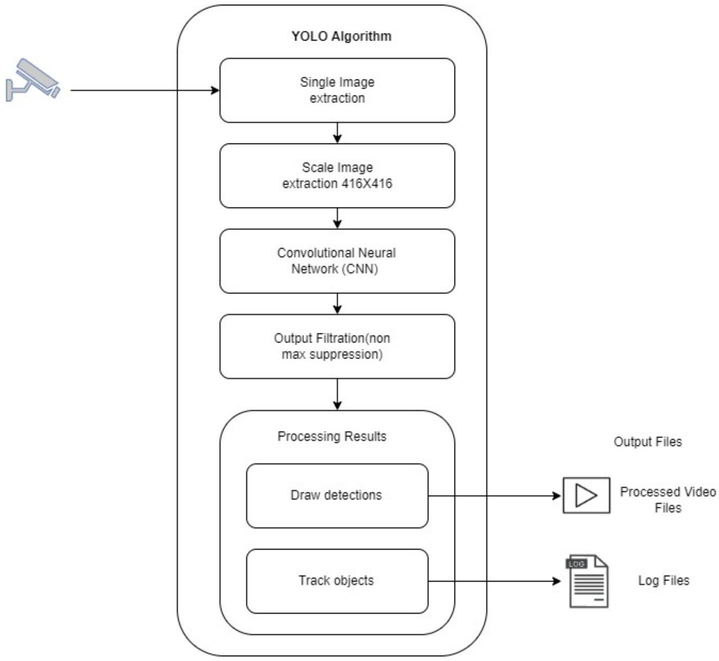

Figure 2.3: Workflow of YOLOv3

Each step is to picture a certain number of anchor or boundary boxes surrounding highly scored objects that belong to established classifications in each grid cell. The dimensions of the ground truth boxes from the original dataset, which identify the most prevalent sizes and shapes, are clustered to construct these boxes. YOLO is different from other algorithm like Fast R-CNN and R-CNN as both are trained to carry out classification and bounding box regression simultaneously.

2.4 Convolutional Neural Networks (CNN) [2]

One of the most common deep learning models for detecting license plates is CNN. For recognizing the patterns, and detecting the plates from the cars CNN is one of the most used models. This is a model which has many contemporary designs also in other people's opinion CNN is basically comprised of neurons which is better for object detection and accepting feedback. This model consists of three-dimensional neurons, the spatial dimension of the input, and depth.

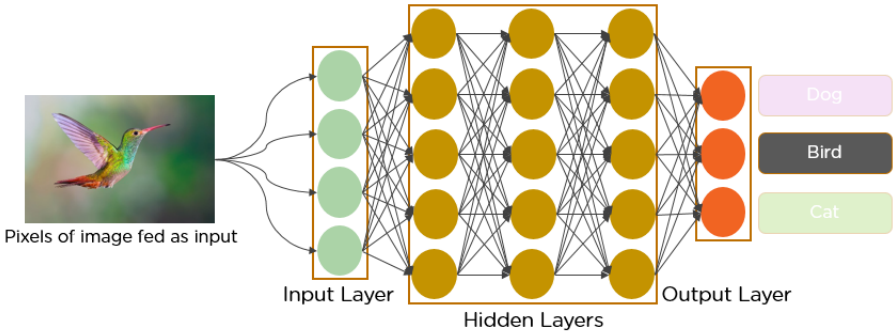

Figure 2.4: CNN Layers.

The involute layer, pooling layer, and entirely linked layers are the three layers of CNN architecture[Figure 2.4]. The neurons output connected to local areas input to compute the scalar produced product between its weights and the associated regions with volume which was entered as an input are defined by the convolutional layers. An image that was given as the input is transformed into a vector and it goes through a two-dimensional weight set, filter, or kernel. This input data and kernel were used for the dot product as the kernel is systemically applied over the image that was taken as an input. This two-dimensional array known as the featured map is produced from this.

Also a function name ReLU is specifically used to excite the kernel for specific

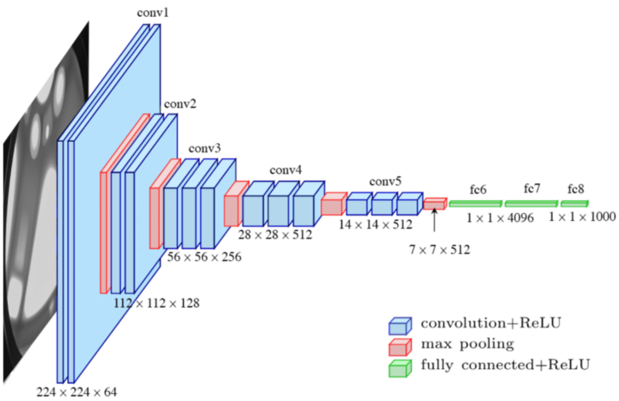

Figure 2.5: CNN.

characterization at a particular input spatial point. The activation map goes via the convolutional layer after which is transmitted over to the pooling layer. The main purpose of the pooling is to reduce the dimensional and complexity of the architecture. So finally the linked layers include neurons connected directly in the two neighboring layers, without any layers connected beyond them.

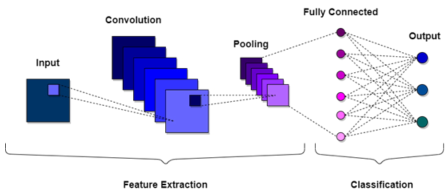

Figure 2.6: Convolution layers.

2.5 Tensorflow [11]

TensorFlow is a platform that is used for machine learning application by using symbolic math libraries which uses dataflow and differentiable programming to perform and complete the task. Also, for the purpose of training datasets and inference of deep neural networks Nowadays, tensorflow is among the most used approaches. In order to work with preprocessed data, build a suitable model for the required work, and also train that particular model for the expected model TensorFlow is the perfect model. It basically works by taking the inputs in a multi-dimensional array known as tensors and by taking it in one end and going through the multiple layers of the system it produces an outcome in the end. The tensor itself is called a vector or many n dimensions of a matrix by which almost all kinds of data can be represented which is a big plus [Figure:2.7].

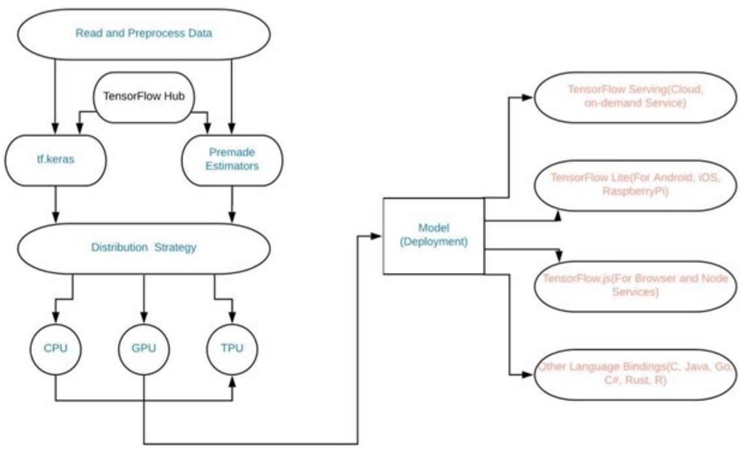

Figure 2.7: Model Of TensorFlow-1.

The main results are conducted inside a graph which is a computation that is done during the training of the model and also it is really helpful as it can be run on multiple CPU and GPUs even on mobile platforms[Figure:2.7]. Basically, a node and edge are used for the process of this model where the node contains the mathematical equation which gives all the endpoints and outputs. The edges are there to explain the relationship between the outputs and inputs. Here we have used the TensorFlow model so that the license plate can be detected from the preprocessed image dataset. As such the model was proposed so that the license plate can be

easily detected from all sorts of angles and resize it so that it becomes easy to extract the numbers from it.
The images were trained and tested first by generating the TFrecords and then by training the model with the TensorFlow the object detection was done. [Figure:2.8]

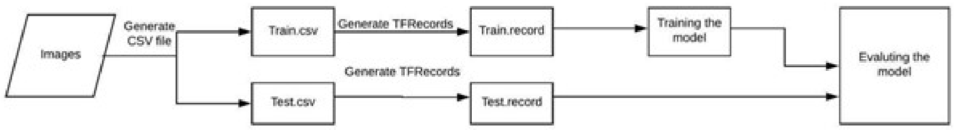

Figure 2.8: Model Of TensorFlow-2.

Chapter 3

Methodology

We suggest an approach that includes the acquisition of the dataset, training the YOLOv5 and YOLOv7 model extracting the numbers using EasyOcr, and finding the accuracy. To evaluate the model properly we divided the dataset into test and train sets. To remain true to the original study we decided to stick to the pre-stated training, testing in the way that was described in the research to ensure comparability. For the training, testing, and extraction the acquired dataset was divided about 8:2 for the train and test set to run the models on them and get the desired results [Figure 3.1].

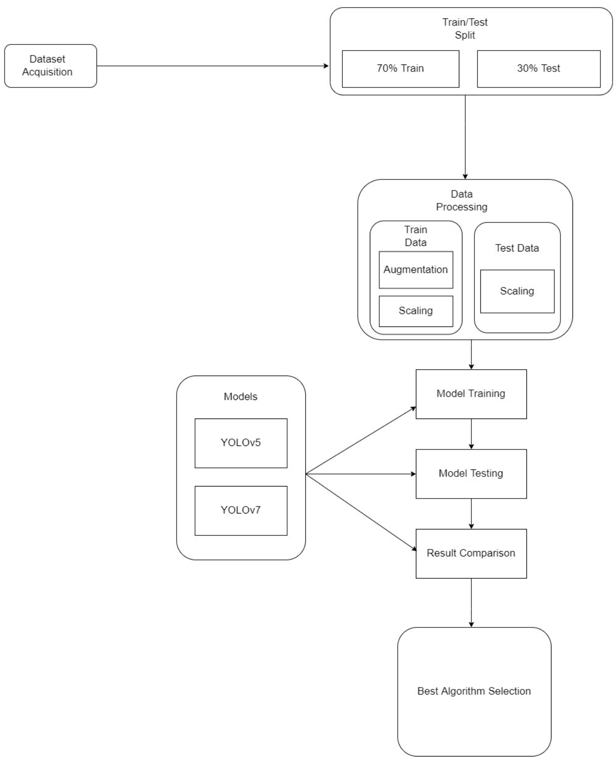

Figure 3.1: Proposed method of the research.

3.1 Implemented Models

Deep learning and Machine learning algorithms are one of the most popular and used algorithms nowadays. These algorithms are used to analyze, extract, and properly categorize the training set from the test system and from unknown data. In our paper we have used some of the deep learning and machine learning algorithms which are basically used to detect, recognize and extract car's license plate's data. So, in our dataset, we have used three of the most common deep learning and machine learning algorithms and tested each of these algorithm's performance which are: CNN, TensorFlow, YOLOv5, YOLOv7 and EasyOCR algorithms. After that, we have compared first two algorithm's results. Further down, we'll go through the models we used.

3.1.1 YOLOv5 [26]

The object detection method known as YOLO(initially proposed by Joseph Redmon) which stands for 'You only look once' separates images into a grid system. In the grid, each cell is in charge of finding objects within of it. As a result of its efficiency and precision, this technique is among the most well-known for object detection. Glenn Jocher introduced YOLOv5 in 18th May of 2020 using the framework named Pytorch. The dataset that is mainly used for training the YOLOv5 is coco dataset and it is a family of compound-scaled object identification models, which also contains capabilities for TTA, ensembling the model, evolution of hyperparameter, and export to ONNX, CoreML, and TFLite.
Since, YOLOv5 model is used for object detection, license plate in this case, we will employ transfer learning techniques, asses its effectiveness, use it for inference and even convert it to different file formats like ONNX and TensorRT. For training, we will be taking a custom dataset of license plate which has 432 images in total. As the dataset was in VOC format we had to convert it to YOLO format to train it. The dataset images are divided into 80 percent for training, 10 percent for test and 10 percent for validation. We will be training the YOLOv5s model with few parameters cause of limitations of our machine even though it provides us with better results when the models are bigger. As our license plate dataset is small, transfer learning is expected to produce better results cause a model gives better results only when trained from scratch. This way of utilizing and applying transfer learning is known as fine-tuning. Which means when a model is already trained for one task can perform another similar task with some adjustment or tweaks to the model. Fine tune can achieve significant improvements after the model gradually adjusts the pretrained features to the custom data. After fine tuning we can evaluate the performance of the model on custom dataset over training, validation or test dataset split. After receiving positive training results, our model is prepared for inference. After inference, test-time augmentations (TTA) can be used to further improve the predictions' accuracy. Each image is enhanced (using a horizontal flip and three different resolutions), and the final prediction is an ensemble of all these augmentations. The TTA must be abandoned if the Frames-Per-Second (FPS) rate is constrained because the inference takes 2-3 times longer. When our model is completed after training and inference it is saved as '.pt' file extension which is a common PyTorch convention. This model then can be exported to other file formats

as ONNX and TensorRT from PyTorch models with 'export.py' script.

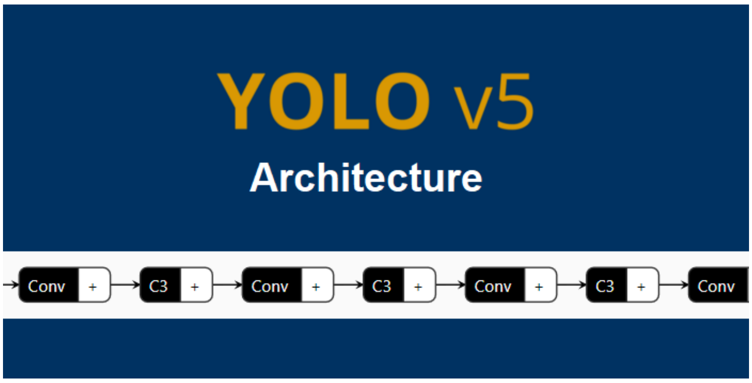

Figure 3.2: YOLOv5 Architecture. [28]

Figure 3.3: YOLOv5 Detection.

3.1.2 YOLOv7 [17]

At the time of object detection there are some challenges like limitation of data, detecting multiple numbers of objects, tracking from different spatial scales and aspect ratio of same image, speeding in real-time etc. may occur. Until now, there are so many models released in the field of Computer Vision for detecting objects at the same time solving these challenges. Although, among all of the challenges increasing the speed of detection in real-time is a major issue in which most of the popular models poorly perform except one method. That is YOLO algorithm. YOLO (You Only Look Once) algorithm is a regression-based Convolutional Neural Network (CNN) that, in a single run, predicts classes and bounding boxes for the entire image rather than just the intriguing portion. Not only, speed but also it generates accurate results with a few backgrounds error. Besides, it has an outstanding learning capability which makes it to learn the description of objects pretty well. However, aiming to update the model's accuracy and speed, there are so many versions of YOLO algorithm has been introduced by the researchers. Those are, YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6 and YOLOv7. Among these, YOLOv7 is the latest version of YOLO algorithm which outperforms all previous versions of YOLO and other object detectors based on the speed and accuracy ranging from 5 to 160FPS as much. Moreover, it reaches the maximum accuracy among all other real-time object detection approaches while achieving 30 frames per second or more. The backbones in YOLOv7 are not pre-trained on ImageNet. Instead, the complete COCO dataset is used to train the models. From YOLOv4, Scaled YOLOv4, and YOLO-R, the architecture is developed. [23]

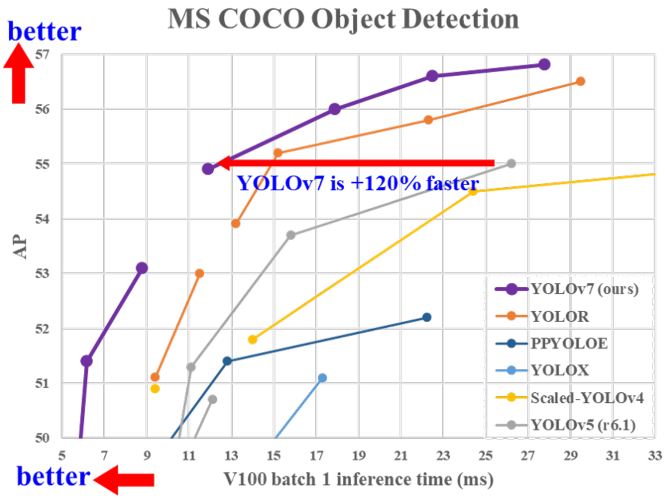

Figure 3.4: Comparing YOLOv7 models performance with other object detectors[22]

The computational building unit of the YOLOv7 backbone is called the E-ELAN. It draws influence from earlier studies on network effectiveness. It was created by looking at the following elements that affect speed and accuracy. Elements are memory access cost, I/O channel ratio, element wise operation and activations gradient path. Simple words, the E-ELAN design improves learning for the framework. The ELAN computational block serves as its foundation. Besides, different models are needed for various uses. While some require extremely realistic models, others put speed first. Model scaling is done to meet these needs and make the model compatible with different computer devices. At the time of scaling the model, there are some parameters which should be followed. Parameters are, resolution (input image's size), width (number of channels), depth (number of layers), stage (number of feature pyramids). The researchers of the YOLOv7 study demonstrate that a compound model scaling strategy can further optimize it. For concatenation-based models, width and depth are scaled in this case coherently. However, another method called Re-parameterization is used for enhancing the model after training. It lengthens the training process but yields better inference outcomes. Levels of models and Module level that ensemble re-parametrization are the types of re-parametrizations used to finish models.
In this research paper, YOLOv7 algorithm has been used to detect license plates from car. At the beginning, a custom coco dataset is trained with the help of yolov7.pt weights which is a pre-trained model of yolo and configuring yaml files of the model which holds the number of classes that requires in training the dataset.

Weights in CNN algorithms are used so that some robust images can be learn perfectly. To function for particular use case, it only needs to learn the final (or perhaps final few) layers. Moreover, it helps to use images which are comfortable enough to read and fits suitably with the framework. After training the dataset, detection from images gives a high accuracy which is better than other versions of yolo algorithm.

3.1.3 EasyOCR [24]

EasyOCR is a kind of python library which makes Optical Character Recognition (OCR) simpler for computer vision developers to implement for extracting text from image. Basically, OCR is a method/ technique which helps text in typed, handwritten or print format to convert into machine readable format. This technique can be applied in a document which is scanned such as PDF, in an image of a document, in a snapshot of a scenario or can be used to extract subtitle text from an image. To do this machine conversion work more easier developers uses EasyOCR python library. This library can be used over 70+ languages which is actually created by Jaided AI company. Moreover, EasyOCR is also implemented in PyTorch library, which is an open-source framework of machine learning applied in computer vision and processing of natural language developed mostly by Fakebook's AI Research Lab. Besides, the user who have a GPU capable of using CUDA software of NVIDIA this PyTorch library can be boosted the speed of text detection and OCR speed drastically.

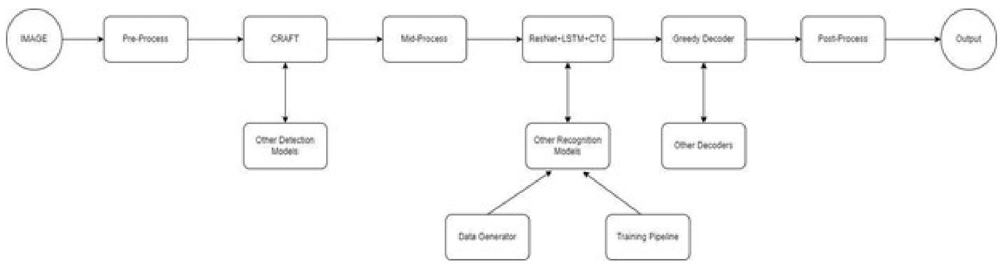

Figure 3.5: EasyOCR Model.

EasyOCR package has less dependency which actually makes it easier to configure our OCR development surroundings. The package can be installed with just writing a single pip command.

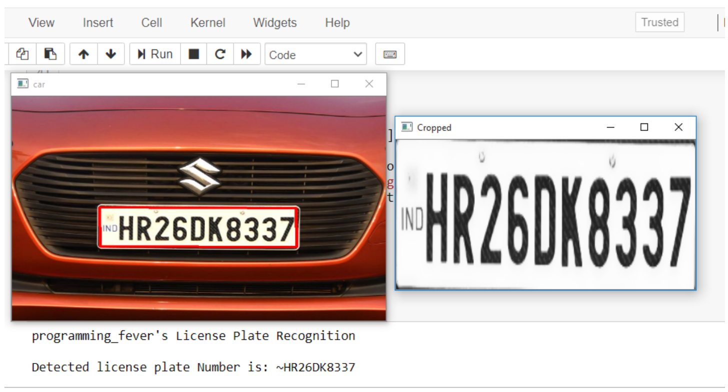

Figure 3.6: Working Of EasyOCR

Once it's done installing a import statement needs to be written to import the package in the project. Now, to run or execute OCR we need to write two lines of code to initialize OCR. First one for reader class and other one to recognize the function from image via read text function [Fig:10]. However, its highly recommended to install this package with OpenCV another popular library of computer vision in real time. Another important feature of EasyOCR is it can work with the default-BGR color channel ordering in OpenCV. So, it does not need to switch colors of the channels when the image is done with loading. After doing all of the inner works of the project the model correctly detecting and extracting the text from the image according to many people.

Chapter 4

Dataset Analysis

For our license plate detection, a custom dataset named LP-Detection was created by mixing three datasets together also with pictures that were self-taken for this dataset. This dataset has only class which is the license plate class. The three datasets that were used in making this dataset were yolo_plate_dataset [18].South America License Plates- in which there are images from a parking lot cctv camera, license plate images acquired from a video, also many images which were augmented into different scenarios [19], Car License Plate Detection [20].This three were also mixed with pictures of cars taken in Bangladesh.So, in total 10700 images were used in this dataset.

Figure 4.1: Dataset Image-1

Figure 4.2: Dataset Image-2

Figure 4.3: Dataset Image-3

In which different regions number plates are included which creates a variety in the dataset.The images contain rear, frontal, sideways images of a car where their license plate can be seen. Also, these images are augmented to create tough condition in which the model would test if it could detect the license plate from the images. Mainly the images were rotated sideways, upside down, noise were added during the augmentation.The challenge in this dataset is to detect the lp properly in from many different images whether they have noise, they are sideways or rotated or they are from different region. The LP-Detection dataset is split into two sets which are Train and Val(validation). The ration of which it is split is 70/30%. The dataset LP-Detection contains a total of 10700 images which were split into (train/val) which has 1 class which is the license plate class and it is used for this research paper.

4.1 Data Augmentation

For the Deep Neural Network (DNN) models, a lot of data is needed because more image sample in the dataset will provide more information, decrease of uncertainty in data, gives a higher precision, gives accurate mean values, spot outliers which will skew the data in a smaller way or form and provide proportion that is of lower error. As a result, the model will produce more accurate value from dataset. Especially, for classes of cars license plate, as much data will be in the dataset the model will generate more accurate detection value of license plates. Data augmentation is the technique of creating additional data units from current data in order to artificially raise the amount of data. This includes enhancing the dataset by making extra changes to the data or by applying machine learning models to create points which are of new information in the latent space of the original data. There are some popular augmentation techniques such as, flipping, rotation, scaling, cropping, translating and adding noise in the images are used to make the dataset more stable and accurate to detect by DNN models [21].

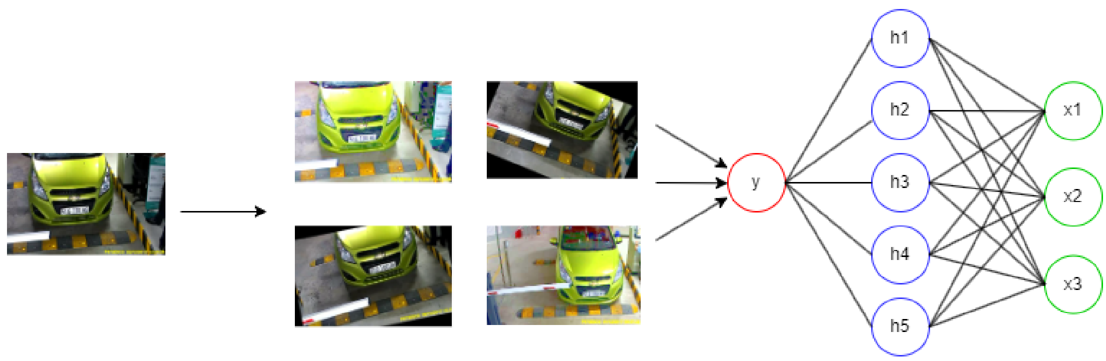

Figure 4.4: Workflow Of Data Augmentation

A high-level summary of the Data Augmentation Parameters which has been used in the dataset of this research paper is given in Table 4.2. There are five distinct Augmentation Types shown in the table. Each form of augmentation displays the Value Range and Direction specific to that type.

Table I

Data Augmentation

Type of Augmentation	Range of Value	Direction
Rotation (With degree)	-45 to 45	Clockwise/ Anti-clockwise
Width shifting (In proportion to overall width)	-0.075 to 0.075	Left/Right
Height shifting (In proportion to overall height)	-0.075 to 0.075	Top/Bottom
Zooming X-axis (With a percentage value)	0 to 60	X Axis
Zooming Y-axis (With a percentage value)	0 to 60	Y Axis
Adding Gaussian Noise (With a percentage value)	0.6	–

Table 4.1: Parameters of Augmentation Technique

First of all, the Rotation (With degree) augmentation technique has been applied both Clockwise and Anti-clockwise having a value which has range from -45 to 45. After that, Width shifting (In proportion to overall width) has been applied in both right and left direction with a range of value from -0.075 to 0.075. Also, Height shifting (In proportion to overall height) having a similar range of value of Width shifting technique from -0.075 to 0.075 has been applied. But this time, it is used in top and bottom direction. Moreover, in Zooming X-axis (With a percentage value) and Zooming Y-axis (With a percentage value) techniques, images have been zoomed with a value range of 0 to 60 in both X-axis and Y-axis direction. Lastly, in noise adding augmentation technique, noise density has been added with a range value of 0.6 which means noise has been applied in 60% pixels of each image. We can therefore quickly understand data augmentation parameters from this table. Furthermore, images were randomly inverted on the horizontal axis. Also, every pixel value was scaled from 0 to 1 and labeled as x. The following is the min-max scaling process:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

At last, augmented images and its data has been merged with previous images (images which used before augmentation) and its data and prepared a large dataset to train the model.

Chapter 5

Implementation and Result Analysis

This section explains how the suggested models for detecting license plates are implemented. In order to determine accuracy and loss, we performed 100 epochs on the training and validation data sets. Before training the dataset. We have resized all of the images into 640×640 pixels. A workstation equipped with an AMD Ryzen 5 5600 4.2 GHz CPU, 16 gigabytes of RAM, and an RTX 3070 GPU was used to do the computation.

5.1 Results

We have implemented yolov5 and yolov7 models on our custom dataset named LP-Dataset and achieved the below results.

Figure 5.1: License Plate Detection-1

Figure 5.2: License Plate Detection-2

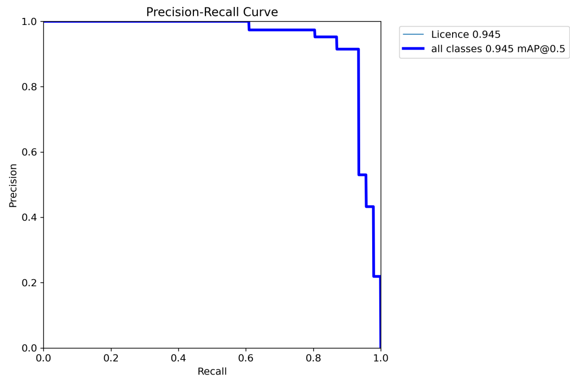

Figure 5.3: PR Curve of YOLOv7 model

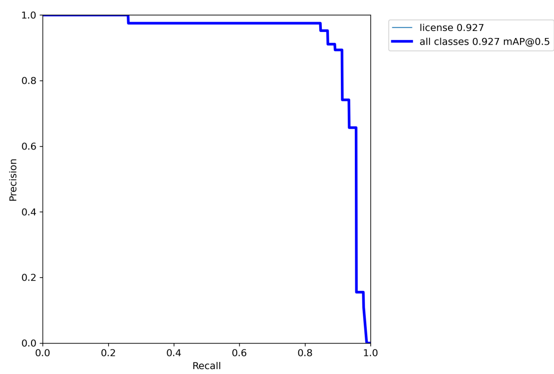

Figure 5.4: PR Curve of YOLOv5 model

The above graphs Figure 5.3 and Figure 5.4 are Precision-Recall Curve of the models yolov7 and yolov5 respectively.
We determine the values of Precision and Recall from the following equations:

$$\text{Precision} = \frac{\text{TP(Truth Positive)}}{\text{TP(Truth Positive)} + \text{FP(False Positive)}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{TP(Truth Positive)}}{\text{TP(Truth Positive)} + \text{FN(False Negative)}} \tag{5.2}$$

We plot the values of precision in the y-axis and the values of recall in the x-axis. The area that falls under the Precision-Recall curve is said to be the Average Precision (AP). The value of AP is required to find the value of mAP. The mAP@0.5 that we get from these figures are 0.945 for yolov7 and 0.927 for yolov5.
A common statistic for evaluating the accuracy of an object detection model is the Mean Average Precision(mAP). The equation that we use to compute the value of mAP is:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{5.3}$$

As we have only one class in our dataset, i.e., license. The value of n becomes 1 and the sum of n interpolated Precision values equals the AP of the models. So, the mAP for both yolov7 and yolov5 are actually the AP itself for both models.

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{5.4}$$

Here, in the Figure 5.5, the value of mAP is already high after 20 epochs from when the training starts, after that the value of the mAP is oscillating. Since, the threshold of Interaction over Union (IoU) is 0.5, the model recognizes the bounding box as a license plate, even if half of the predicted bounding box doesn't intersect with the actual bounding box. Hence, the model doesn't give us the true value. So, in the Figure 5.6, we start the threshold from 0.5 and increment it by 0.05 and find the mAP during every increment. In this stage, we add all the mAP from each increment and divide it with the number of total increments to find the value.
Before the training, the accuracy of the model was quite low in Figure 5.6, but the

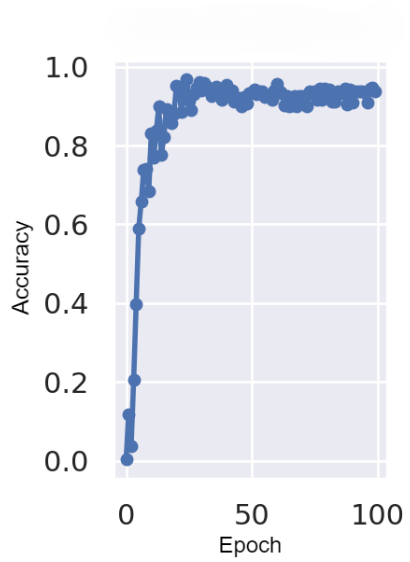

Figure 5.5: map@0.5 graph of YOLOv5 model

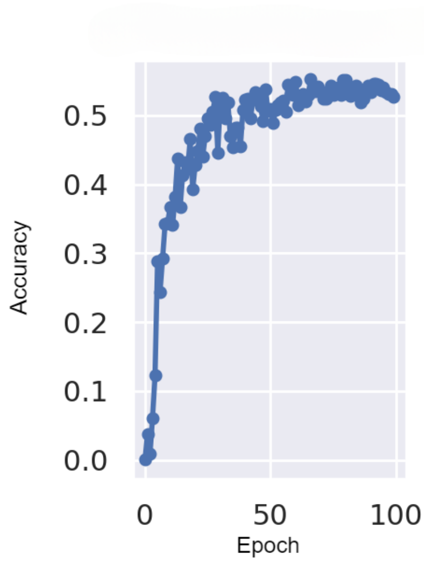

Figure 5.6: map@0.5:0.05:0.95 graph of YOLOv5 model

accuracy increased steadily as we were training the model. The model was being trained for 85 epochs until it converged and it started oscillating the value around 0.55. Though the mAP is quite low in mAP@0.5:0.05:0.95, we can confirm from this evaluation that our model is learning since the accuracy is increasing gradually. This method of mAP is the best evaluation metric for object detection.

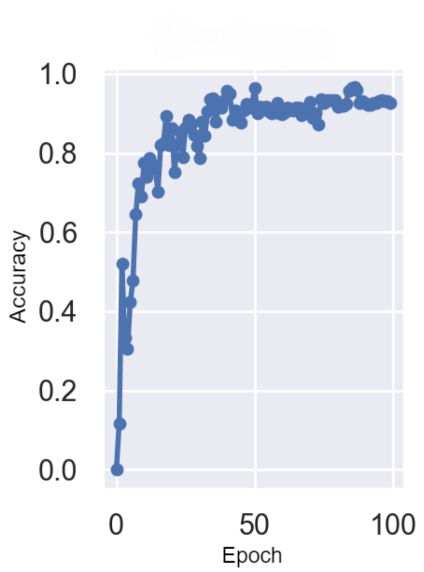

Figure 5.7: map@0.5 graph of YOLOv7 model

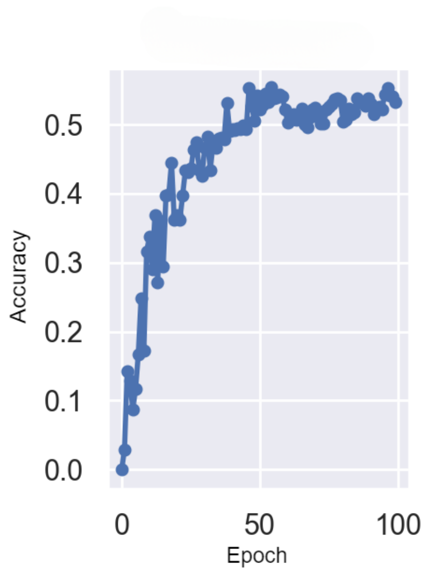

Figure 5.8: map@0.5:0.05:0.95 graph of YOLOv7 model

As seen in the mAP graph of yolov5, the same case is being noticed in yolov7. The value of mAP when the threshold is 0.5 is suddenly increased as the threshold value of IoU is too low, the model training is already done when it is 30 epochs, after that the models value goes back and forth after 30-40 epochs. Whereas, in Figure

5.8 the mAP value increases gradually after each training of the dataset is done and finally the machine stops learning after almost at the end of training when the value is around 0.6.

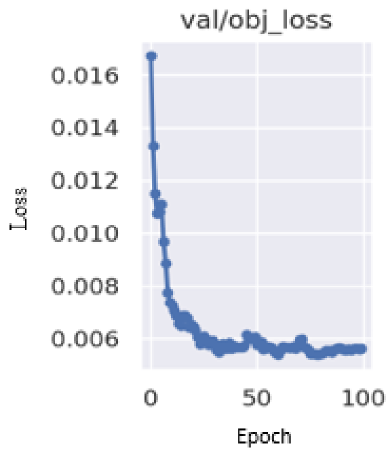

Figure 5.9: Validation loss history of YOLOv5 model

Figure 5.9 represents the validation loss of yolov5. This graph has an initial loss of 0.016 while the training started but the loss started to decrease significantly after 25 epochs and then maintained a stable state while the loss ended with a value of 0.0057 at the final epoch. The overall value of the graph demonstrates this yolov5 model has a pretty decent loss in validation data..

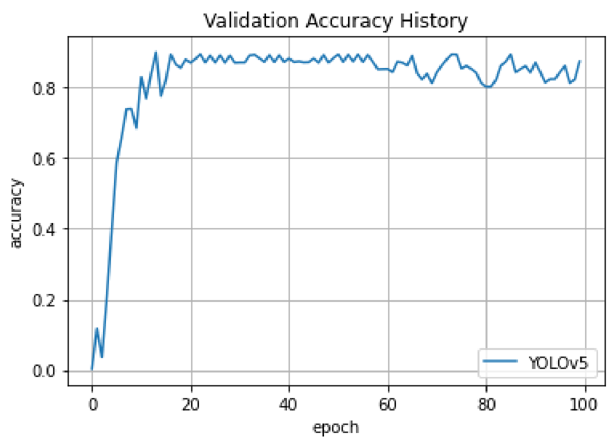

Figure 5.10: Validation accuracy history of YOLOv5 model

Moreover, Figure 5.10 represents the overall validation accuracy of the model. The validation images are used to test the performance of the model after training. From the graph we can see that, the model yolov5 was learning the dataset and was improving its accuracy until 15 epochs, after that the accuracy of the model became stable. The model finished with an average confidence score of around 0.938.

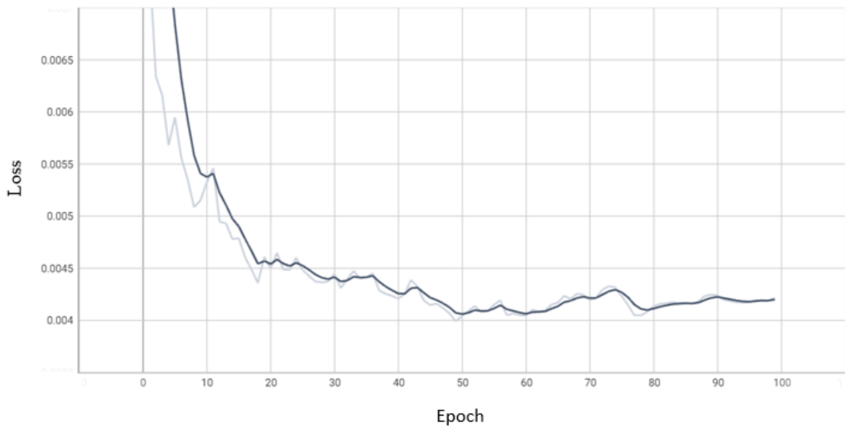

Figure 5.11: Validation loss history of YOLOv7 model

In this validation loss Figure 5.11, the y-axis denotes the loss of the model whereas the x-axis denotes the number of epochs that are being used for the yolov7 model to train. The loss starts with a very high value but gradually decreases when the model learns the dataset. Here, we can observe from the yolov7 model, that the validation loss for the object starts from around 0.007 but after 100 epochs the loss comes down to 0.00425.

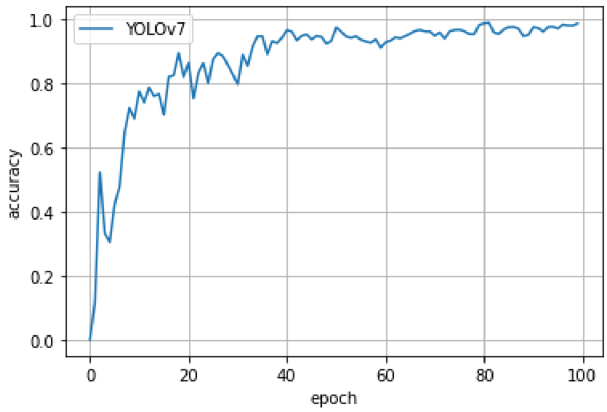

Figure 5.12: Validation accuracy history of YOLOv7 model

While Figure 5.12 is the validation accuracy graph, like the precision and recall graph, the y-axis of this graph also increases along with the number of epochs. The y-axis in this graph represents the accuracy of the yolov7 model and the horizontal x-axis of this graph denotes epochs. The confidence score of validation accuracy increases from the minimum which gradually increases and the value of accuracy stables after 30 epochs finally giving us an average accuracy of 0.956.

5.2 Applying EasyOCR

After detecting the license plate the detected bounding box of license plate is cropped and then EasyOCR is applied on that bounding box to extract the characters from the license plate. The easyocr contains over 80+ library of languages which means it supports over 80 languages character to be extracted. The ocr extracts all the characters from the license plate successfully and saves it.

Figure 5.13: Character Extraction Using EasyOCR-1

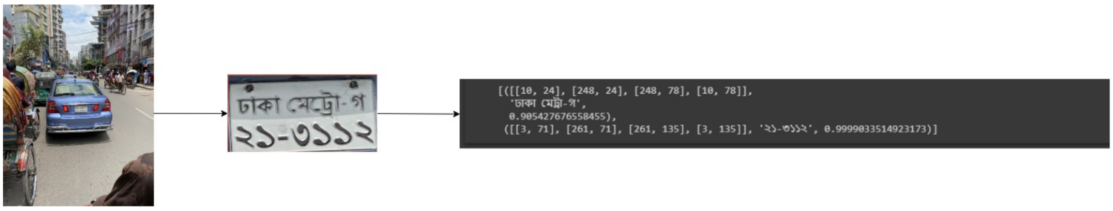

Figure 5.14: Character Extraction Using EasyOCR-2

5.3 Real-Time Implementation

The model YOLOv7 was used in real-time detection on videos that were self-taken. The trained model was used in the videos and produced very accurate results in detecting the license plate that was seen in the video. The detection rate depends on the quality of the video and camera. Though the real-time detection rate and accuracy of YOLOv7 exceeds the numbers of YOLOv5 by quite a margin. It achieves 114 FPS in inference speed and also is 127FPS faster and 10.7% more accurate than YOLOv5. So, the version 7 gains 15 FPS and is 21 FPS inference speed faster comparing to version 5 of YOLO. In the videos (self-taken) that were tested for real-time detection in YOLOv7 produced very accurate results in detecting the license plates.

Figure 5.15: License Plate detection in Real-time-1

Figure 5.16: License Plate detection in Real-time-2

5.4 Comparing with other research works

We compared our results with Paper-A [26], IR-LPR: Large Scale of Iranian License Plate Recognition Dataset, Paper-B [25], Real-time license plate detection and recognition using deep convolutional neural networks, Paper-C [6], Bangladeshi License Plate Detection and Recognition with Morphological Operation and Convolution Neural Network & Paper-D [27], License plate detection for multi-national vehicles – a generalized approach. The findings of the detailed comparison will be discussed in the next section.

5.4.1 Comparison with Paper-A: IR-LPR: Large Scale of Iranian License Plate Recognition Dataset [26]

Researchers in this paper have used the YOLOv5 model to detecting the object. As this paper uses a pytorch implementation rather than a fork one and also they opted for YOLOv5 because of it's working mechanism which is different from other models as it divides the images into grid system and also for it's better accuracy and speed. The authors used a total of 20,967 car images for this research. First, they used the normal YOLOv5 models and then they implemented an extra model for detecting the faster detection. According to our paper the comparison between the paper for each model is shown in table-1. All the models of our research paper generated more accuracy than the models of paper. The accuracy for yolov5-s shows an accuracy for 86%, for yolov5-x is 86.7%, Detectron2 is 91.3%, yolov5-s(+dummy dataset) is 86.2%, yolov5-x(+dummy dataset) is 86.2% and for the Detectron2(+dummy dataset) is 93.3% whereas our dataset (LP-Detection) which we trained with our custom yolov5 and yolov7 model produced an accuracy of 93.8% and 95.6%. From this table we can see that an already increased accuracy is seen. In all cases of yolov5 model and also for the Detectron2 for the case of 1005 accurate LP reading our models produced a better accuracy then all of those models mentioned.

Table II

Comparison with Paper-A

Papers	**Model Name**	**Accuracy**
Paper-A	YOLOv5-s	86%
Paper-A	YOLOv5-x	86.7%
Paper-A	Detectron-2	91.3%
Paper-A	YOLOv5-s(+dummy dataset)	85.2%
Paper-A	YOLOv5-x(+dummy dataset)	86.2%
Paper-A	Detectron-2(+dummy dataset)	93.3%
Our Implemented Model	YOLOv5	93.8%
Our Implemented Model	YOLOv7	95.6%

Table 5.1: Comparing with the Paper-A

5.4.2 Comparison with Paper-B: Real-time license plate detection and recognition using deep convolutional neural networks [25]

Authors of this paper proposed a model where they extracted the parts of the car whether it might be the frontal or the rear part of the car based on the LP annotation. A LP will always be in the dataset was always assumed. Also, by adjusting the region and based on all of this they proposed a model where they used end to end ALPR method which was based to identify the license plate from the vehicle using a hierarchical Convolutional Neural Network (CNN). Basically, the image of LP went through the CNN two times to get recognized. They used a dataset consisting of 196 images. The Proposed method of this research paper resulted in an accuracy of 89.15% for SSIG, 65.62% for UFPR-ALPR, 85.19% for OpenALPR without redundancy and for with redundancy in case of weighted voting for SSIG the accuracy was 92.41%, for UFPR-ALPR it is 85.42% and in case of majority voting for SSIG the accuracy was 87.81%, for UFPR-ALPR it was 81.25%. On the contrary our research provided a much better accuracy than the proposed model of the author's paper. From the 10700 images that we trained our yolov5 and yolov7 model we gained a accuracy of 93.8% and 95.6%. Which clearly indicates a better trained model and the accuracy gained are also higher which also suggests to a better detection model.

Table III

Comparison with Paper-B

Papers	Model Name	Accuracy
Paper-B (Without Redundancy)	SSIG	89.15%
Paper-B (Without Redundancy)	UFPR-ALPR	65.62%
Paper-B (Without Redundancy)	OpenALPR	85.19%
Paper-B (Weighted Voting)	SSIG(With Redundancy)	92.41%
Paper-B (Weighted Voting)	UFPR-ALPR(With Redundancy)	85.42%
Paper-B (Majority Voting)	SSIG(With Redundancy)	87.81%
Paper-B (Majority Voting)	UFPR-ALPR(With Redundancy)	81.25%
Our Implemented Model	YOLOv5	93.8%
Our Implemented Model	YOLOv7	95.6%

Table 5.2: Comparison with Paper-B

5.4.3 Comparison with Paper-C: Bangladeshi License Plate Detection and Recognition with Morphological Operation and Convolution Neural Network [6]

This research paper suggests a method of four modules which are detection of license plate then extraction of the license plate, character extraction and finally recognizing the characters. For the detection of license plate the authors of this paper first binaries the grayscale images then convert them and connect them to their components. Then the detected area is checked if it is within the aspect ratio and then it run through the deep learning tool which is CNN. The proposed model of this paper displayed an accuracy of 93.78% while detecting the license plates. On the contrary our research provided a much better accuracy than the proposed model of the author's paper. From the 10700 images that we trained our yolov5 and yolov7 model we gained a accuracy of 93.8% and 95.6%. Which clearly indicates a better trained model and the accuracy gained are also higher which also suggests to a better detection model.

Table IV

Comparison with Paper-C

Papers	**Model Name**	**Accuracy**
Paper-C	CNN	93.78%
Our Implemented Model	YOLOv5	93.8%
Our Implemented Model	YOLOv7	95.6%

Table 5.3: Comparison with Paper-C

5.4.4 Comparison with Paper-D: License plate detection for multi-national vehicles – a generalized approach [27]

In this paper, the proposed methods by the authors were, Fuzzy inference system based region of interest (ROI) identification, License plate detection (LPD) using local recursive analysis and Local image features based license plate verification. These methods were tested on 2200 images during several hazardous conditions to calculate and accurately detect 5379 license plates among 5945 total vehicles. This paper came up with a success rate of 90.5% accuracy. The suggested solution beats both traditional and deep learning approaches in terms of performance and has the potential to be used with international automobiles. Although it has such good percentage of detecting a number plate accurately, still the success rate of these methods were much lesser than yolov5 and yolov7 that was being computed in our paper with a dataset of image 10700 which had an accuracy of 93.8% and 95.6%. The improvement in detecting license plate successfully is quite noticeable in our paper comparing to this one.

Table V

Comparison with Paper-D

Papers	Model Name	Accuracy
Paper-D	ROI + LPD	90.5%
Our Implemented Model	YOLOv5	93.8%
Our Implemented Model	YOLOv7	95.6%

Table 5.4: Comparison with Paper-D

Chapter 6

Conclusion and Future Work

Failing to identify and detect the license plate makes it really difficult to maintain security in the parking lots, in the streets, and also to keep track of cars in the streets. So, the work aims to create an automated license plate detection method for detecting license plates from cars in an effective and more accurate way. In most places the detection of the license plate is done in a moderate way or in many cases they don't have any detection models available. So, a model that can be run anywhere and in any specs that would detect the license plate from an image and extract the number in a more accurate way would be perfect for surveillance and safety. Though it is technically difficult to detect the license plate from the images of a car where the image might be blurry, from a tough angle, where the camera might not be good enough, light conditions- keeping in mind to all these problems we are aiming to design a model which will produce better results in all these difficult situations and produce better accurate results. As with each passing day, we are depending more and more on machine learning algorithms and models to process everything so these technologies will allow us a more secure and safe street for us to roam around and will create more interest in this particular field.We made our dataset full of different region car license plate and augmented it in a way that it is possible to detect the license plate in many conditions. The suggested research sought to make sure that license plates are being able to be detected with a high accuracy rate with the help of yolov7. Though the dataset contains images of cars from different regions still there are many things that can be worked on. More data could be added into this dataset in order to cover all the regions Also for augmentation and real-life images of cars can be added to make sure that the license plates are being detected even in very tough and unique condition. Also, this work was done on the basic yolov7 while being customized to the need of detecting the license plate as necessary but it can be supervised more in a way to get more accurate detection and better accuracy. The possibility of doing extensive data augmentation always adds a dimension that can produce even greater results. Finally this version of research dose not satisfy the issue of detecting license plate in bad lighting condition or from very bad angled images which can be solved in the future by adding more extensive augmentation and also adding more layers to the data. We anticipate that this work of ours contributes to the advancement of the future research regrading license plate detection using yolov7 and popular recognition tasks.

Bibliography

[1] A. Dama and R. Todmal, "Number plate recognition using improved segmentation," *Int. J. Innov. Res. Stud.(IJIRS)*, vol. 2, pp. 162–168, 2013.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv e-prints*, Nov. 2015.

[3] N. Saleem, H. Muazzam, H. Tahir, and U. Farooq, "Automatic license plate recognition using extracted features," in *2016 4th international symposium on computational and business intelligence (ISCBI)*, IEEE, 2016, pp. 221–225.

[4] B. V. Kakani, D. Gandhi, and S. Jani, "Improved ocr based automatic vehicle number plate recognition using features trained neural network," in *2017 8th international conference on computing, communication and networking technologies (ICCCNT)*, IEEE, 2017, pp. 1–6.

[5] Z. Selmi, M. B. Halima, and A. M. Alimi, "Deep learning system for automatic license plate detection and recognition," in *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, IEEE, vol. 1, 2017, pp. 1132–1138.

[6] G. Rabbani, M. Aminul Islam, M. Anwarul Azim, M. Khairul Islam, and M. M. Rahman, "Bangladeshi license plate detection and recognition with morphological operation and convolution neural network," in *2018 21st International Conference of Computer and Information Technology (ICCIT)*, 2018, pp. 1–5. DOI: 10.1109/ICCITECHN.2018.8631937.

[7] G. Sharma, "Performance analysis of vehicle number plate recognition system using template matching techniques," *Journal of Information Technology & Software Engineering*, vol. 8, no. 2, pp. 1–9, 2018.

[8] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 580–596.

[9] J. Wäldchen and P. Mäder, "Machine learning for image based species identification," *Methods in Ecology and Evolution*, vol. 9, no. 11, pp. 2216–2225, 2018.

[10] Z. Xu, W. Yang, A. Meng, *et al.*, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 255–271.

[11] P. Prakash, S. Premanathan, S. Surwase, M. A. Vigil, and S. Bohare, "License plate detection and facial analysis using tensorflow deep learning algorithm," *Journal of Computational and Theoretical Nanoscience*, vol. 16, pp. 3246–3251, Aug. 2019. DOI: 10.1166/jctn.2019.8171.

[12] G. T. Sutar, A. M. Lohar, and P. M. Jadhav, *Number plate recognition using an improved segmentation*. Citeseer, 2019.

[13] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8787–8796.

[14] S. Shen, L. Wang, S. Duan, and X. He, "Car plate detection based on yolov3," *Journal of Physics: Conference Series*, vol. 1544, p. 012 039, May 2020. DOI: 10.1088/1742-6596/1544/1/012039.

[15] Y. Cao, S. Liu, P. Zhao, and H. Zhu, "Rp-net: A pointnet++ 3d face recognition algorithm integrating rops local descriptor," *IEEE Access*, 2022.

[16] V. Sharma, M. Jain, T. Jain, and R. Mishra, "License plate detection and recognition using opencv–python," in Jan. 2022, pp. 251–261, ISBN: 978-981-16-8247-6. DOI: 10.1007/978-981-16-8248-3_20.

[17] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[18] https://drive.google.com/file/d/1KLK-DWgT3VoQH4fcTxAt2eB3sm7DGWAf/view?fbclid=IwAR0yAeEKmRhLbJQTQYbLaXRSwuYA04a-n8bHt1fZycAXYu-HARM0R_bEakM.

[19] *Artificial mercosur license plates - mendeley data*, https://data.mendeley.com/datasets/nx9xbs4rgx/2.

[20] *Car license plate detection | kaggle*, https://www.kaggle.com/datasets/andrewmvd/car-plate-detection.

[21] *Data augmentation | how to use deep learning when you have limited data*, https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/, (Accessed on 09/20/2022).

[22] *Github - wongkinyiu/yolov7: Implementation of paper - yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, https://github.com/WongKinYiu/yolov7.

[23] *Introduction to yolo algorithm for object detection | engineering education (enged) program | section*, https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/.

[24] *Jaided ai: Easyocr*, https://www.jaided.ai/easyocr/.

[25] S. M. S. C. R. Jung, *Real-time license plate detection and recognition using deep convolutional neural networks - sciencedirect*, https://www.sciencedirect.com/science/article/abs/pii/S1047320320300237#section-cited-by.

[26] *Large scale of iranian license plate recognition dataset*, https://arxiv.org/abs/2209.04680.

[27] *License plate detection for multi-national vehicles – a generalized approach | springerlink*, https://link.springer.com/article/10.1007/s11042-019-08199-4.

[28] *Object detection algorithm—yolo v5 architecture | by surya gutta | analytics vidhya*, https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef.

[29] *Yolov3 (you only look once) deep learning algorithm. all you need to know… | by afroz chakure | apr, 2021 | medium | datadriveninvestor*, https://medium.datadriveninvestor.com/yolov3-you-only-look-once-12de76ad74d5.