Real Time Bangladeshi License Plate Detection and Recognition

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Md. Saimom Islam	**160204011**
Tauhidul Islam	**160204031**
Md. Yeasir Arafat	**160204093**
Homaira Alam	**160204102**

Supervised by

Qamrun Nahar Eity

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

July 2021

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Qamrun Nahar Eity, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Md. Saimom Islam
160204011

Tauhidul Islam
160204031

Md. Yeasir Arafat
160204093

Homaira Alam
160204102

CERTIFICATION

This thesis titled, **"Real Time Bangladeshi License Plate Detection and Recognition"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in

July 2021.

Group Members:

Md. Saimom Islam	**160204011**
Tauhidul Islam	**160204031**
Md. Yeasir Arafat	**160204093**
Homaira Alam	**160204102**

Qamrun Nahar Eity
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Professor Dr. Mohammad Shafiul Alam
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

We have received a great deal of guidance, assistance and encouragement throughout the writing of this dissertation. At first, we would like to thank our supervisor, Qamrun Nahar Eity, whose proficiency was inestimable in the formation of the research topic and methodology notably. We also would like to thank our head of the department, Professor Dr. Mohammad Shafiul Alam for his uncountable patience to hear every problem and thus to solve these with a friendly mind. All the faculty members of Computer Science and Engineering department have always been there for us, no matter what time it is and how much time it takes to figure out our problems. Next, we would like to applaud our parents for their prudent admonition and compassionate appreciation. Finally, our friends deserve credit for not only supporting us in pursuance of our problems and findings, but also for offering delightful relaxation to rest our mind from our research materials.

Dhaka
July 2021

Md. Saimom Islam

Tauhidul Islam

Md. Yeasir Arafat

Homaira Alam

ABSTRACT

All countries require license plates for road vehicles. Bangladesh Road Transport Authority (BRTA) made a generalized guideline for automobile license plate. The dimension and features of all license plates are identical.

Automatic number plate detection and recognition system has several uses. Numerous number of works have already been done to make the detection and recognition process efficient and effective. An enormous number of work on Bangla vehicle plate detection and recognition has been done before but very little work has been done on real time using videos. It is difficult in segmenting Bangla characters. The successes of previous works, for the most part have been restricted to rectify localization and acknowledgment of the plates whose pictures are captured from the frontal or the rear of vehicles with insignificant angular varieties. In consequence, most Bangla ANPR frameworks encounter problems when the license plates are slanted and not viable in real-time detection. In this paper, we attempted to address the matters. We note that our anticipated technique can be connected to non-Bangla license plates as well by making it more universal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Automatic Number Plate Recognition

A number plate is a unique identification of a vehicle and is connected to a vehicle for official acknowledgment purposes. Automatic Number Plate Recognition (ANPR) is an programmed framework which is able of perusing vehicle permit plates without human interference through the apply of high speed picture, discovery of characters inside the given pictures, confirmation of the character patterns as being those from a vehicle license plate, character acknowledgment to convert an image to text.
There are a few types of ANPR system which are: Fixed, Mobile, Portable, ANPR connected with CCTV etc. A discussion on Fixed ANPR and Mobile ANPR system is given below.

Fixed ANPR System

Fixed ANPR uses infrared (IR) cameras. They can be mounted at high fixed points, such as parkades, street signs, road lights, interstate overpasses or constructions etc. Camera program is made competent of recognizing the designs that make the license plates and interpreting the letters and numbers into a computerized setup. Then, the samples are compared in actual time to form a list of plate numbers that have a place to a set of vehicles of concern. If the framework recognizes a match, an alarm rings or an caution message is sent to the dispatcher or other relegated staffs. It has various applications like:

1. High Occupancy Toll (HOT) lanes.
2. Traffic data collection for analysis of road usage.
3. Parking management.

4. Speed enforcement.

5. Infrastructure monitoring.

6. Travel time management.

7. Nonstop observation of high speed or high-crime areas etc.

Figure 1.1: Fixed ANPR system

Mobile ANPR System

Mobile ANPR mainly uses vehicle-mounted IR cameras.Numerous cameras are arranged on the motor vehicle. As the vehicle proceeds, it captures the images of the license plates from different vehicles on every side of it. Then, it checks captured license plates against one or more databases straight away and immediately alerts the officers of hits. Mobile ANPR is designed to cope up with camera shake, highly changeable picture quality, and oblique angles. It has numerous applications like:

1. Allows the police to search for stolen vehicles, vehicles with invalid or any other violation.

2. This technology allows police officers to recognise plates of all the parked vehicles around them.

3. Improves circulation avoiding traffic jams caused by improper parking.

4. Increases spatial cognizance for advanced officer safety.

5. Enables the police to check for parking violations etc.

Figure 1.2: Mobile ANPR system

A graph has been given below which shows the usage of different types of ANPR cameras from 2015 to 2021.

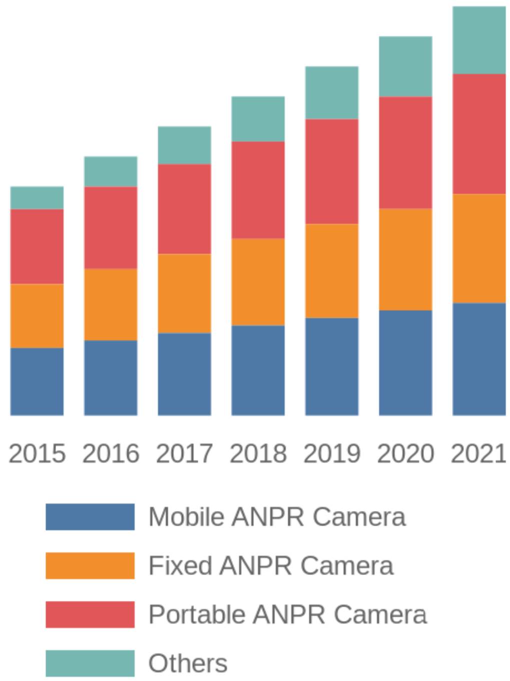

Figure 1.3: Use of Different ANPR Camera Over Years

Table 1.1: Advantages and Disadvantages of Fixed and Mobile ANPR System

ANPR System	Advantages	Disadvantages
Fixed ANPR Systems	1. Can be used 24/7. 2. Reading is more error free. 3. Coverage is extended. 4. If modern technology is used then it can be cost efficient. 5. Very useful on motorway routes and expressway.	1. Limited to a certain place. 2. Cannot follow crime trends. 3. Costly to move to another location. 4. Costly to maintain and repair. 5. Needs a clear plan on deployment, response etc.
Mobile ANPR Systems	1. Can be used when required. 2. Can be deployed at particular places. 3. Can be moved without cost to take after a hot spot or an issue in a range. 4. Response real time 5. Gives officers the capability to distinguish a vehicle which has matched against a hot-list which they may not otherwise have halted. 6. Can get hits from the fixed ANPR cameras and react in genuine time.	1. Vulnerable to destruction. 2. Less dependable at scanning license plates. 3. Connected to cars which can break down, need servicing etc. 4. Relies on officers on the ground to set up and keep an eye on the framework.

Portable ANPR Systems	1. Can be used when needed. 2. Set up is easy. 3. Can be used in any location. 4. Does not depend on vehicles. 5. Similar advantages as mobile systems mentioned above.	1. Difficult to handle in vehicles. 2. Similar disadvantages as mobile systems mentioned above.
ANPR connected to CCTV (Local Dominance, private use)	1. Can be used 24/7. 2. Camera range is amplified with constrained additional costs. 3. Can track vehicles. 4. Can be utilized in car parking, shopping malls, petrol stations etc. 5. Enhanced possibility of getting picture of the driver.	1. City environment, less specific to traveling culpability. 2. In case no suitable agreements in place, disagreement of interest between local officials and policing targets. 3. Ineffectual positioning of cameras (too tall, centered on pedestrian range etc). 4. Need of suitable regulations.

1.2 Technological Difficulties

In spite of the fact that, the thoughts of ANPR are basic, applying them into practise is more tough. It will not be correct to say that all ANPR frameworks can be utilized within the same way or can make the same outputs. Many frameworks work with lower detail cameras making lower quality pictures and less correct reads. More up to date ANPR have progressed qualifications, with infrared abilities to empower interpreting number plates and capturing photographs in destitute brightness circumstances or at night. It is acknowledged that the common interpreting rates for ANPR is 90% to 94%, in idealize circumstances and it is backed by advanced frameworks. The older ANPR frameworks are strikingly untrustworthy, being intensely condemned for misinterpreting number plates and making 'hits' on guiltless drivers. In spite of the reality that ANPR innovation has created astonishingly over the last few years, concerns around the exactness and trustworthiness of ANPR frameworks stay.

There are numerous troubles that the framework must be able to manage. Some of which are:

1. The picture or video of poor resolution since the plate is too distant away or from the use of a low-grade camera.
2. Hazy pictures due to movement obscure.
3. Bad lighting, low contrast, reflection or obscurity.
4. Any object obscuring the plate or a portion of the plate like soil, tow bar etc.
5. Utilizing diverse textual styles, well known for vanity plates. A few nations do not permit plates which kills the issue.
6. Need of association between nations. Cars from distinctive nations can have the similar number but diverse design of the plate etc.

Few of the above said issues can be illuminated within the computer program, but it is essentially left to the hardware side of the framework to work out arrangements to these issues. Expanding the peak of the camera may dodge issues with objects such as: other vehicles darkening the plate but it instigates other issues like altering the expanded skew of the plate. Few small-scale frameworks permit few blunders in the license plate.

1.3 Does ANPR bring us close to a total surveillance society

A few individuals stress that for CCTV surveillance to be 'total', it has to have the control to spot (anyone, anyplace, anytime) and classify people. Individuals ought to be cautious of being observed and there needs to be a surety of reaction from specialists to acts of non-conformity.

Being computerized and automated, ANPR frameworks have an expanded capacity to save and look at data, an expanded capacity to pinpoint suspects without having to observe for behavioral designs.

In most cases, the populace subject to CCTV perception (open street frameworks) is hidden to the eyewitnesses, so they cannot productively recognize and categorize people in open location. Most CCTV systems cannot however routinely connect a person's picture to a database. ANPR cameras are associated to databases comprising information on the total enlisted driving populace. In any case, stand-alone ANPR frameworks have their downsides over CCTV.

First and foremost, they are bounded to the driving people. Not everything and everyone be watched by it. The details on the familiar driver of the vehicle of interest can be wrong, as the driver may not be the selected attendant of the vehicle or the databases used in affiliation with ANPR may not be modern at the time when the framework double-checks the information with respect to the vehicle registration number. ANPR systems can match, look at and spread person data at high speeds. These are many of the reasons why ANPR might bring us closer to a maximum surveillance society. One of the foremost downsides of both CCTV and ANPR systems is the degree to which these observation systems create an authoritative reaction to non-conformity. Most of the CCTV systems are not observed on a regular basis and CCTV staff have other works or do not check all the cameras continuously.

The differences between ANPR and CCTV is given below:

Table 1.2: Differences Between ANPR and CCTV

ANPR	**CCTV**
Digital	Mostly analogue
Fast	Slow
Increased capacity of storing data and analysis	Limited data storage capacity
Recognition and tracking is automatic	Manual tracking
Low coverage, limited to roads	Extensive coverage, including pedestrian areas
Driving population	Everyone
Diminished number of administrators; decreased 'operator' bias (programmed detection of suspects)	High number of administrators; profoundly skilled; danger of 'operator' bias (administrator chooses on suspicious behaviour)

1.4 Real Time Systems

Real time system implies that the system is subjected to real time, i.e., reaction ought to be ensured within a specific time limitation or system ought to meet the required deadline, else risk serious results, including disappointment. In real-time advanced picture processing, the ordinary handling time per test must be less than the analyzing phase, which is the testing rate. A framework with an accurate time restriction can not be treated productive in case it produces the proper process or the proper reply after a definite due date. In genuine time frameworks time or rightness is as vital as coherent rightness of a program. The rightness of a real-time framework is depended on the rightness of the outcome and convenience. Real time entities are of two types which are: Hard real time systems and soft real time systems.

1.4.1 Hard Real Time System

In hard real time system due date should be met with error-free reaction or the framework will proceed to a complete disappointment. An affirmation of continuously meeting the hard time limit is required. Examples incorporate air traffic control, vehicle subsystems control, atomic power plant control etc.

1.4.2 Soft Real Time System

The soft real time systems perform assignment nearly within the defined due date. They do not assure a hard time limit. Assignment can be carried out even after the time has passed. Examples incorporate mixed media transmission and reception, networking, cellular systems, web sites, administrations and computer games etc.

1.5 Use of ANPR in Other Countries

ANPR systems have been in practise in many countries in the world. Strict implementation of number plate standards has helped the early improvement of ANPR systems.

1.5.1 For Law Enforcement

Australia

A few State Police Forces and the Department of Justice utilize both fixed and portable ANPR frameworks. In 2005, New South Wales Police Force Highway Patrol were the first to utilize fixed ANPR framework. Later, they started to utilize a portable ANPR framework with three infrared cameras fitted to Highway Patrol fleet in 2009. The framework distinguishes unregistered, stolen vehicles, disqualified or suspended drivers.

United States

About 71% of all US police divisions utilize a few form of ANPR framework according to a report of 2012. Portable ANPR is becoming a surprising component of metropolitan policing methodologies and insights gathering. It is additionally utilized for recovery of stolen vehicles, distinguishing proof of wanted criminals and revenue collection from people who are troublesome on city, or observing for ”Amber Alerts”.

Canada

Government, common, and metropolitan police administrations over Canada utilize ANPR computer programs. It is additionally utilized on toll routes and parking authorization organizations.

Belgium

The city of Mechelen employs an ANPR framework since September 2011. It is used to filter all cars crossing the city. Cars recorded on ’black lists’ create an caution within the dispatching room, so they can be capturing by a patrol.

United Kingdom

Vehicle movements are recorded by about 8000 cameras capturing between 25 and 30 million ANPR ‘read’ records regularly. These records are stored for up to two years. It can be accessed, examined and utilized as proof by UK law authorization offices.

1.5.2 For Electronic toll collection

Ontario

Ontario’s 407 ETR highway uses a blend of ANPR and radio transponders to toll vehicles coming in and leaving the road.

Portugal

Old streets of Portugal have toll stations where drivers can pay with cards and also lanes where there are electronic collection systems. But most new streets only have the choice of an electronic toll collection system.

South Africa

In South Africa, ANPR is employed for the etoll fee collection. Car owners driving into or out of the city has to pay a fee. The amount of tolls passed relies upon the distance travelled on the particular freeway.

Sweden

In Sweden, ANPR is employed for the Stockholm congestion tax. Car owners driving into or out of the city must pay a fee which relies on the time of the day. From 2013, for the Gothenburg congestion tax, which also includes vehicles passing the city on the main streets.

1.5.3 For Vehicle Speed Control

ANPR is utilized for limiting motor vehicle speed in Australia, Austria, Belgium, Dubai, France, Italy, Netherlands, Spain, South Africa, UK, and Kuwait. It monitors vehicles' travel time between two fixed points and examines the average speed.

1.6 Study of Bangladeshi Vehicle License Plate

Bangladesh Road Transport Authority which is called as BRTA, gives vehicle license plates for motor vehicles in Bangladesh. The license plates in Bangladesh use the Bengali alphabet and numbers. The basic format of license plates in Bangladesh is "city - vehicle class letter and number - vehicle number". An example is given below.

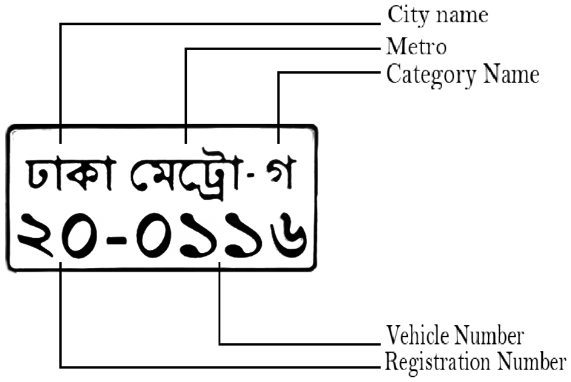

Figure 1.4: A sample of BRTA Standard License Plate

The license plates are connected on both the front and back of the vehicle. The letters and numerals qualified to be used in the license plate are:

অ ই উ এ ক খ গ ঘ ঙ
চ ছ জ ঝ ত থ ঢ ড ট
ঠ দ ধ ন প ফ ব ভ ম
য র ল শ স হ
০ ১ ২ ৩ ৪ ৫ ৬ ৭ ৮ ৯

Each of the letters carries the identities of different vehicles which are given below.

Table 1.3: Letters and the identities of different vehicles

Letters	**Identities of different vehicles**
অ	Medium personal goods (3.5 to 7.5 tons)
ই	Agricultural vehicle – power tiller, tractor)
উ	Heavy private goods – bottle carrier
এ	Motorcycle (small, up to 50 cc)
ক	Motor car (small, up to 1000 cc)
খ	Motor car (medium, 1001 to 1300 cc)
গ	Motor car (medium, 1301 to 2000 cc)
ঘ	Private passenger (Crossover, SUV)
ঙ	Private three wheeler tempo
চ	Micro bus for private service
ছ	Health service vehicle
জ	Minibus for public service
ঝ	Minibus for private service
ট	Heavy public goods (7.5 to 22 tons)
ঠ	Vehicles for dual-purpose
ড	Medium public goods
ঢ	Private articulated vehicle
থ	4 stroke CNG Auto rickshaws
দ	Private CNG Auto rickshaws
ন	Light public goods (up to 3.5 tons)
প	Taxi
ফ	Public Auto tempo
ব	Minibus public Service

Continued on next page

Table 1.3 – continued from previous page

Letters	Identities of different vehicles
ভ	Motor car (extra-large, 2001 cc and above)
ম	Light personal goods (up to 3.5 tons), Delivery vehicle (up to 2.5 tons)
য	Any vehicle of Prime Minister's office
র	Any vehicle of President's office
ল	Motorcycle (large, over 125 cc)
শ	Vehicles for special purpose
স	Minibus for private service
হ	Motorcycle (medium, 51 to 125 cc)

From the pictures given below, it can be spotted that the public and private license plates have the same format and they have two rows. License plates of Government and military vehicles have only one row. Yellow is the background color of government owned vehicles and the foreground is black. It has one Bangla character which indicates the registration type which is accompanied by a five digit registration number. Military vehicles' license plates have an arrow and the acronym of the name of the military force at the starting. It is then followed by a four digit registration number.

(a) Public

(b) Private

(c) Government

(d) Military

Figure 1.5: Different Bangladeshi License Plate

There are twenty nine work places of BRTA that are working around the country. These work places are documented in Table 1.4.

Table 1.4: Vehicle Registration Area

ঢাকা মেট্রো (উত্তর)	ঢাকা মেট্রো(দক্ষিণ)
ফরিদপুর	ময়মনসিংহ
গাজীপুর	নারায়ণগঞ্জ
টাংগাইল	মানিকগঞ্জ
চট্টমেট্রো	ফেনী
রাঙামাটি	নোয়াখালী
কুমিল্লা	কক্সবাজার
বি-বাড়িয়া	রাজশাহী
নওগাঁ	বগুড়া
রংপুর	দিনাজপুর
পাবনা	সিরাজগঞ্জ
খুলনা	ঝিনাইদহ
কুষ্টিয়া	যশোর
মৌলভীবাজার	বরিশাল
সিলেট	

License plates are exceptionally distinctive from one nation to another nation. Because of this, it is not compelling to use an ANPR framework created in one nation to another nation without particular adjustments.
There are some traits of Bangladeshi license plates which can be composed inside the Bangla language. Maximum of the languages within the global have separate characters in a phrase but, the characters in a Bangla word are frequently joined on the top via a horizontal line. That line is referred to as Matra. Some Bangla characters are comprised of two or greater isolated regions. There can be a few disjoint parts in a character either at the top or at the bottom.
Those extraordinary traits of Bangla characters make it fairly tough to fragment. In contrast to various nations, a variety of differences can be seen among many of the license plate designs in Bangladesh.
A large range of Bangladeshi license plates have two rows in which the primary row carries the registration place and its kind and the second row includes the registration number. There can be an additional row at the top or at the bottom of the registration plate containing a few extra records. This extra information also makes it extra challenging to extract the registration information from the license plates.

1.7 Motivation

In recent years, ANPR has become more and more important due to the growing number of vehicles on the roads. In the context of Bangladesh, ANPR can play a vital role. Many problems can be solved with ANPR system. A pie chart is given below which shows the global market of ANPR by its' different applications in 2016.

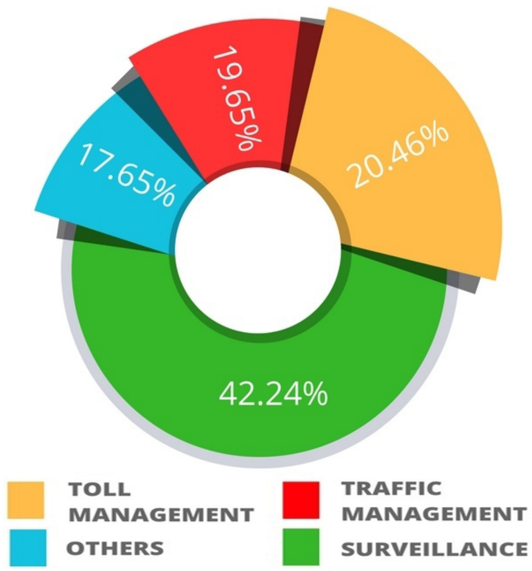

Figure 1.6: Global ANPR Market by Applications in 2016

1.7.1 Importance in Reduction of Vehicle Theft and Identifying Stolen Vehicles

As stated in the crime data of Dhaka Metropolitan Police (DMP), about 102 cases have been filed at police stations over car theft in last three months. Twenty-three cars were stolen in July while 30 cars were stolen in August. In September, the gangs stole 49 cars. Sources at the Detective Branch (DB) of police said around 20 car theft gangs are active in the capital and they steal 20-25 cars every month. Recently, instances of car theft have increased alarmingly. To reduce this occurrence, if the vehicle is lost or stolen, the owner can notify the police and give the license plate number to the them. Then the stolen vehicle can be detected using license plate detection technique with their database images.

Figure 1.7: Detecting stolen vehicle

1.7.2 Importance in Toll Collection

The most challenging work faced by the travelers, is waiting in the toll plaza since manual work can take more time and payment methods are not easier. It leads to wastage of fuel by waiting as well as valuable time to a great scale which leads to traffic congestion on the express way paving a way for air pollution. If the waiting time is long, it often results in drivers getting irritated which can engage them in a petty quarrel over people and toll attendants.
Collecting the tolls and maintaining the records of different vehicles and transaction of money is a laborious process. By using ANPR, the delay on taking tolls can be eliminated by cashless tolling and it is rapidly becoming the most inventive technology for the trav-

elers who pass through the toll plaza. The cash payment is more difficult for collecting, transferring, recording and managing purpose since fraudulent and burglary are serious scenarios of manual payment methods.

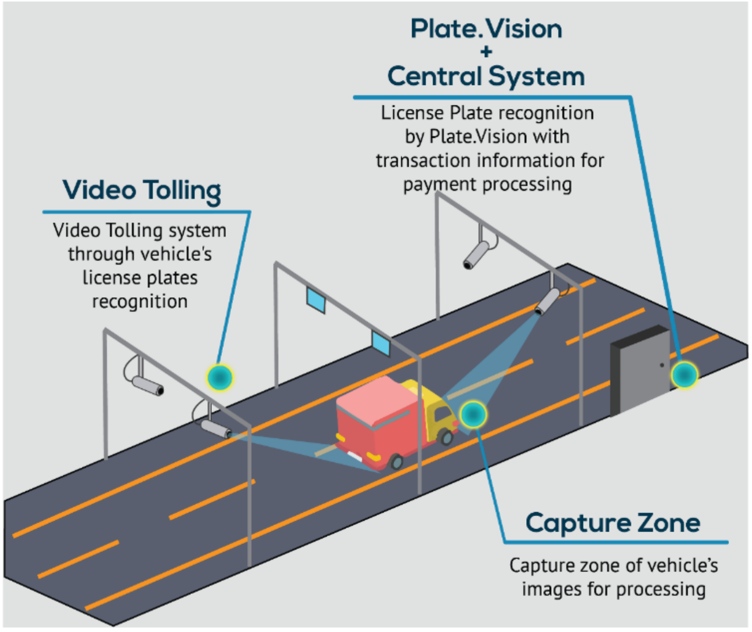

Figure 1.8: ANPR for automatic toll collection

1.7.3 Importance in Traffic Management

In Bangladesh, the traffic circumstance is disorganized. As traffic jam causes an immense loss financially, it is necessary to improve the traffic management system. ANPR can help to ensure portability. It can measure vehicle speed. Travelling above the speed limit dramatically reduces a driver's chance of stopping if something sudden happens. It can accurately determine the average speed of vehicles moving on a section of road by identifying their license plate at both ends of a travel. When vehicles pass through different road signs, the license plates are showed with a safety message.

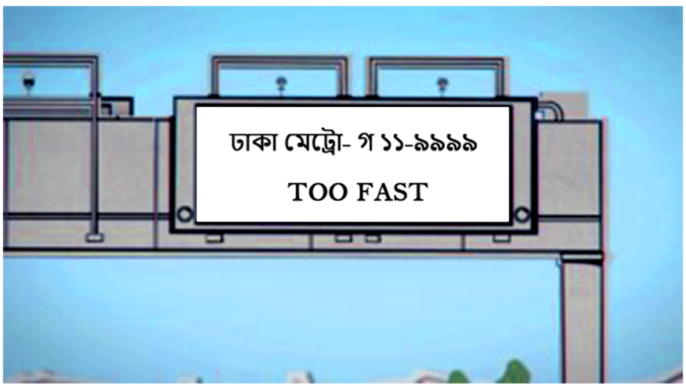

Figure 1.9: Safety message with license plate if a vehicle is over speed

If ANPR is used, it will simplify the manual labor and reduce the error opportunity. It can recognize the license plate from vehicle in real time and this process is automatic and full time. So, the need of manpower for this purpose will reduce. If a vehicle is unregistered, it is not possible for the traffic police to know that. By using ANPR, it can be easily checked in database if the vehicle is registered or not.

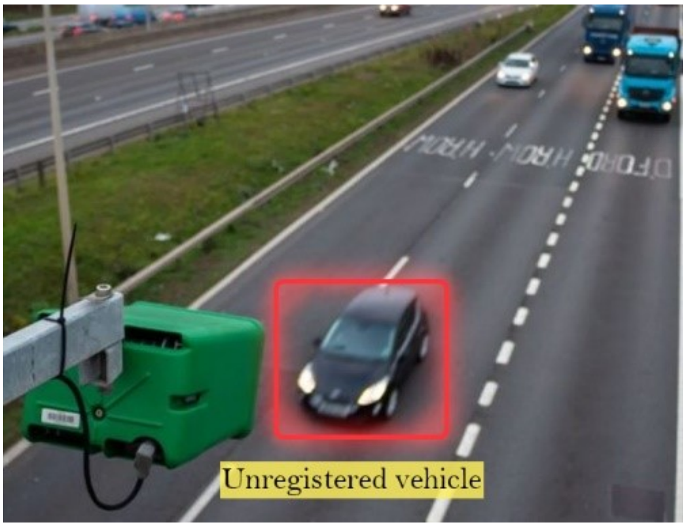

Figure 1.10: Unregistered vehicle detection

Sometimes it becomes very difficult to identify for the traffic police to recognize vehicle owner who has violated the traffic regulations or drive in too much speed. Therefore, it becomes quite impossible to arrest and penalize them because the traffic police might not be able to extract license number from the moving vehicle because of the speed. Therefore, ANPR frameworks can be utilized as one of the solutions to these problems.

1.7.4 Importance in Parking Management

ANPR systems are an efficient and inexpensive way to monitor parking management. Using advanced ANPR system we can ensure angled, mirrored plates do not interrupt the ability to detect and interpret each number plate, no matter what the time of day or weather it is.

Using ANPR, plates can be captured at entry and exit, images can be securely saved and time stamped. Then, the vehicles which violated the parking rules and conditions can be processed and ownership details can be obtained. It is beneficial when delivering fines to drivers. ANPR cameras ensure that the vehicles do not outstay their allotted parking time or park without permission. The security for both parking operators and users can be improved by using ANPR. In car parking management system, ANPR cancels out the need for parking enforcement officer. For the high-accuracy of ANPR readings and

24/7 operation, they are economical than most individual and so it offers an additional dependable service. Parking management personals usually notice that traffic personnel and ANPR systems work well along. Staffs can trust ANPR to produce the mandatory info, minimize the time they spend on the roads. The traffic flow during peak hours can also be improved.

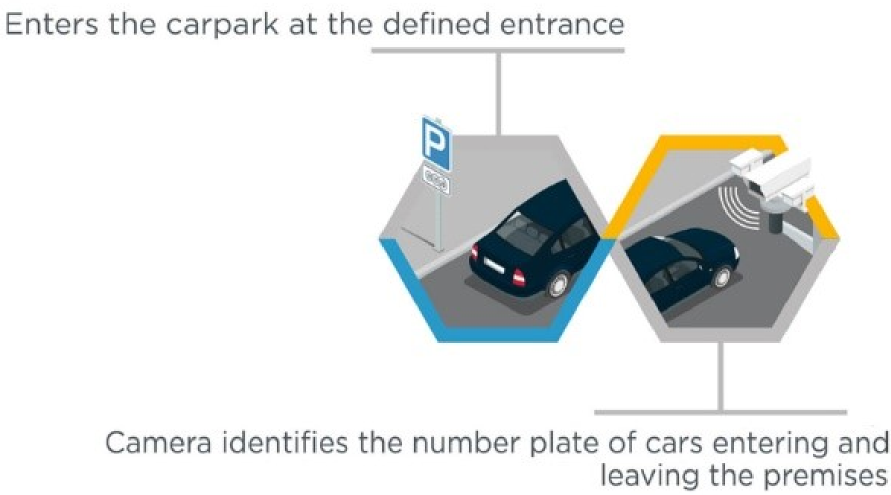

Figure 1.11: ANPR in parking management

1.7.5 Added Security

ANPR for the most part acts as a deterrent. The information that the number plate is being recorded and checked is sometimes enough to prevent criminal behaviour from before. ANPR offers an additional measure of security for both public and personal use.

1.7.6 Easy and Efficient

Installing a heavy-duty security gate or having a manual check system is an effective way against intruders, but they can both be incredibly time-consuming. It is always necessary to consider how easy it is for people you want to grant access to, such as employees ad delivery vehicles, to get in and out. An ANPR system is incredibly easy and efficient. People can come and go as required without needing to do anything, but it will still be known who has entered the premises.

1.7.7 Cost Effective

ANPR technology is one of the most cost-effective solutions for managing car park. It is possible to cut costs and reduce the need for security personnel when this smart solution is chosen. Companies can also issue fines to anyone picked up by the ANPR system that should not be on their private property or anyone that has exceeded the maximum time limit of staying.

1.7.8 Stand Alone

ANPR cameras can operate in such a way where all the information is entirely processed so that no additional computers, or software licenses are needed. These cameras also have optical character recognition software installed which enables all images to be analysed directly. As they are stand-alone solutions, ANPR cameras are quick, safe and light to install.

1.7.9 Can be used as Evidence

ANPR systems can provide the details regarding when a person was at the premises, whenever they are required. The images taken by this camera can be used as evidence and can provide beneficial information that can be used in inspections. It can easily be proved when the vehicle in question was on the premises.

1.7.10 Congestion Control

ANPR systems can be used to manage blockage by charging for vehicle entrance to the central business areas at peak hours. Traffic congestion causes financial losses, also it has additional a destructive effect on road safety. ANPR can assist to enhance mobility and safety.

Chapter 2

Literature Review

2.1 Introduction

ANPR has three basic steps which are: license plate localizing and detection, segmentation and recognition. Due to the immense use of license plate recognition, different methods and techniques have been developed for these stages. In this chapter, we present a survey of the methods and techniques that are used in ANPR in different prior works.

2.2 Methods Used in Related Works

- **Mask R-CNN:**
  Mask R-CNN is a deep neural sample(DNN) concentrated to solve sample segmentation issues in machine learning or computer vision. In other phrases, it can split up disparate items in a picture or a video. It offers the bounding boxes, classes and masks, if an image is given.

  In [1], they used it for Image segmentation. Their offered strategy of pre-processing license plate images has two steps: conducting segmentation and transforming the segmented license plates into consistent rectangular perspectives. The model was trained for 100 epochs and the batch size is set approximately to 100. For other hyper parameters, used the same values.

- **Harris Method and Shi-Tomasi Method:**
  The idea of the Harris method is to discover points on the basis of the intensity fluctuation in a local neighborhood: a little portion around the properties should indicate a enormous intensity alteration when compared with windows shifted in any direction.

J. Shi and C. Tomasi made a tiny adjustment to it which shows improved outcomes compared to Harris Corner Detector.

In [1], for perspective transformation they generated Shi-Tomasi corners reconstructed from Harris corners.

- **YOLOv3:**
  The full form of YOLOv3 is You Only Look Once version 3. It is a accurate time object spotting method that recognizes precise objects in videos, live feeds, or images. YOLO is a Convolutional Neural Network (CNN) which does object detection. CNN's are primarily classifier-based schemes which is used to process input images as organized arrays of data and recognizes prototypes between them. YOLO is much quicker than different networks and still sustains preciseness. One more benefit of using it is that it can generalize extremely well. So, it performs strikingly well on samples that it has never been trained. It also can strain background noises from the authentic samples very effectively. Another advantage of using YOLO is that it can observe and pinpoint stuffs in an image at the same time using only one convolutional network which remarkably decrease segmentation and recognition time furthermore.

  In [2] and [3], they used method YOLOv3 based CNN to detect the license plate. In [2], they divided their work in three stages which are: license plate detection and localization, digit recognition and character identification and character recognition.Using bounding box coordinate given by YOLOv3 network they sliced number plates.

  In [4], a YOLO based network is used as these are said to be the fastest networks.

- **Bounding Box:**
  Bounding Box Concept is that it makes an invisible rectangle that provides some points of reference for detecting the object and creates a collision box for the object. By selecting X and Y coordinates of each picture data, interpreters draw these rectangles over images and bordering the object of interest. Machine learning algorithm can easily find what they are searching for. It is one of the most well-known image annotation schemes in deep learning. This strategy can cut costs and intensify annotation productiveness. This helps to calculate and recognize what a vehicle looks like.

  Bonding box on binary image is used in [5] for segmentation. A discrete database of binary images of Bangla numbers (0-9) and words like "Dhaka", "Chotto", "Metro" etc. and letters like ”Ka”, ”Kha”, ”Ga” etc. was made for recognizing segmented characters.

In [6], at first contours of the characters were detected and then characters were extracted using bounding box algorithm.

For each connected components, a green bounding box was formed in [7].

- **Median Filtering:**
  Median Filtering is basically used for expelling the noise from an image. By expelling the noise from an image, this process is very effective for further processing. It is widely used in modern image processing because, while removing the noises, it also stores the edges. The main thing is to swap the gray value with the median value of the gray levels of the neighbourhood of that pixel of the image. It is effective using median instead of average operation.

  In [5], noise removal is one of their steps to segment the characters. For that step, they used Median filtering for removing the noise from their images.

  [7] and [8], both used 2D Median filtering for removing the noise from their images.

- **Morphological Operations:**
  Morphological Operation is a non-linear operation which is dependent on the structure of an image. Instead of numerical values morphological operation depends on pixel values of matching ordering. It is basically suitable for binary images but it can also be applied on gray-scale images.

  Morphological techniques create an image with a smaller size template which is called a structuring element. Structuring elements located at all pixels of the image and matched with the neighbourhood pixels. The element “fits” with the neighbourhood pixels or “hits” with the neighbourhood. If the experiment is successful for a specific location it creates a new binary image with non-zero pixel value.

  Morphological operation is used in [5] and [7].

  In [9], there may still exist noises such as: little gaps, bulges etc. in the picture after the segmentation. These noises are evacuated utilizing mathematical morphology.

- **Morphological Erosion:**
  Erosion is a process in the field of morphology. It is basically applied on binary

images. Later, it was extended to be used in gray-scale images as well. Erosion operation erodes the regions with boundary. Basically it erodes foreground pixels, that is s why foreground pixels shrink and gap along with those regions become bigger.

Morphological erosion is done in [5], [10], [7], [11], [9].

- **Morphological Dilation:**
  Dilation is another fundamental operation in morphology. It increases the size of the objects, fills the holes and broken areas, increases the brightness of the objects and is used later in Opening operation.

  Morphological dilation is done in [5], [10], [7], [11], [9].

- **2D convolution Operation:**
  2D convolution operation is performed to enhance the contrast of the image. 2D convolution involves both horizontal and vertical directions in 2 dimensional spatial domain. Convolution is often used for image processing such as: smoothing, sharpening, and edge detection of images. Then the image is then converted to binary image.

  For the detection of edge, 2D convolution operation is performed in [7], which enhances the contrast of the image.

- **Faster R-CNN:**
  Faster R-CNN is an object detection design presented by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015, and is one of the famous object detection structures that uses convolution neural networks. Faster R-CNN have 3 parts which are: Convolution layers, Region Proposal Network (RPN), Classes and Bounding Boxes prediction.

  In [12] and [**?**], it is used to detect number plate.

- **CNN Model:**
  CNN is a sort of deep learning model for processing data that contains a network design, such as pictures, and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is a mathematical construct that's ordinarily composed of three types of layers or building blocks which are: convolution, pooling, and completely connected layers. The primary two, convolution and pooling layers, perform feature extraction, while the third, a completely connected layer, maps the extracted features into the final output, such

as classification. A convolution layer plays a key part in CNN, which is composed of a stack of numerical operations, such as convolution, a specialized sort of linear operation. In advanced images, pixel values are stored in a two-dimensional (2D) framework and a small grid of parameters called the kernel, an optimizable feature extractor, is connected at each picture position, which makes CNN's exceedingly productive for picture processing, since a feature may occur anyplace within the picture.

In [13], CNN model is used to detect number plate. They did not apply any data augmentation or processing steps. License plate was cropped from the main image first. Then, digits and characters were sliced in 32X32 pixels. At last, they cropped individual character from the license plate regions.

In [14], the Convolution Neural Network classifier is trained using 25*25 Width and Height sized images. CNN has several layers that are used for feature extraction and compute hidden layers weights of the network.

- **Top-hat Transform:**
  In mathematical morphology and digital image processing, top-hat transform is an operation that extricates little components and subtle elements from given pictures. There exist two sorts of top-hat transform: the white top-hat transform is characterized as the contrast between the input picture and its opening by a few organizing components, whereas the dark top-hat transform is characterized dually as the contrast between the closing and the input picture. Top-hat transforms are utilized for different picture preparing tasks, such as feature extraction, background equalization, picture improvement, and others.

  It is used in [6] to gray version of input color image.

- **Feature Extraction:**
  Feature extraction is a process of dimensionality decrease by which an initial set of raw information is decreased to more reasonable groups for processing. Feature extraction is the name for strategies that select and /or combine factors into features, successfully decreasing the amount of information that must be processed, while still precisely and completely describing the initial information set. The method of feature extraction is valuable when you ought to diminish the number of assets required for processing without losing vital or important data. Feature extraction can also decrease the amount of excess information for a given analysis. Also, the reduction of the information and the machine's endeavors in building variable combinations (features) encourage the speed of learning and generalization steps within

the machine learning handle.

It is used in [8].

- **Gaussian Blurring:**
  In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring a picture by a Gaussian function named after mathematician and researcher Carl Friedrich Gauss. It is a broadly utilized effect in graphics software, typically to decrease picture noise and diminish detail. The visual effect of this blurring method is a smooth blur taking after that of viewing the picture through a translucent screen, particularly distinctive from the bokeh effect created by an out-of-focus lens or the shadow of an object beneath regular light. Gaussian smoothing is additionally utilized as a pre-processing stage in computer vision algorithms in order to upgrade picture structures at diverse scales. The pixels closest to the center of the kernel are given more weight than those which are far away from the center. This averaging is done on a channel-by-channel basis, and the normal channel values become the new value for the filtered pixel.

  Gaussian blurring is used in [6] and [10].

- **Adaptive Thresholding:**
  Adaptive thresholding is the technique where the threshold value is calculated for smaller regions so, there will be different threshold values for different regions. It is used to convert an image containing gray scale pixels to just black and white pixels.

  Image is converted to binary image by using adaptive thresholding in [6].

  Local adaptive thresholding algorithm is used for binarizing in [15].

- **Connected Component Analysis Method:**
  Connected Component Analysis method is used to recognize Bengali license plate. In 2D image, connected components are collection of pixels with the same value, which are connected to each other through either 4-pixel, or 8-pixel connectivity. 4-pixel connectivity groups all pixels that contact each other on either of their four faces, while 8-pixel groups pixels that are connected along any face or corner. In 3D, connectivity options, at least for rectangular pixels, are 6, 18, and 26 (faces, faces+edges, faces+edges+corners). In order to find the objects in an image, an operation is used that is called Connected Component Analysis (CCA). This operation takes a binary image as an input. Generally, the false value in this image is associated with background pixels, and the true value indicates foreground or object pixels. Such an image can be created with thresholding. After a thresholded image

is given, CCA produces a new labeled image with integer pixel values. Save value pixels belong to the same object.

In [9],a recursive algorithm is used for connected component labeling operation.

In [8], Connected Component Analysis (CCA) is done to label the component.

- **Edge Detection:**
  Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image. It is one of the basic steps in image processing, pattern recognition in images and computer vision. When we process very high-resolution digital images, convolution techniques come to our rescue. There are various methods in edge detection, and the following are some of the most commonly used methods- Prewitt edge detection, Sobel edge detection, Laplacian edge detection, Canny edge detection.

  Sobel edge detection method is used in [10]. Sobel method basically finds the changes in pixels values inside the picture and computes the gradient to discover the edge directions inside the picture.

- **Gray-scale conversion Algorithm:**
  Advanced descriptor-based picture recognition frameworks frequently work on gray-scale pictures, with little being said of the component utilized to convert from color to gray-scale. Typically since most analysts accept that the color-to-gray-scale strategy is of little consequence when utilizing robust descriptors. The output of each gray-scale algorithm is between 0 and 1. There are few gray-scale algorithms which are: Averaging (aka "quick and dirty"), Luma or Luminance, Desaturation, Decomposition, Single color channel, Custom of gray shades, Custom of gray shades with dithering (for example, horizontal error-diffusion dithering).

  In [5], [10], [14] gray-scale is used. In [7], RGB to Gray-scale transformation is done to convert the three-dimensional (3D) pixel value (R, G, B) to a two-dimensional (2D) value subsequently decreasing the computational complexity.

- **Otsu's Binarization Algorithm:**
  Binarization is the method of changing information features of any substance into vectors of binary numbers to form classifier algorithms more productive. In a straightforward case, changing an image's gray-scale from the 0-255 range to a 0-1 range is binarization. Otsu's algorithm is one of the classical algorithms presented

by Nobuyuki Otsu in 1979. The algorithm works by comprehensively looking for the edge that minimizes the weighted within-class variance, or put another way maximizes the between-class variance.

Otsu's thresholding method is used to produce binary image in [8].

- **HSI Color Model:**
  The HSI (Hue, Saturation, Intensity) color model decouples the intensity components from the color carrying data (Hue and Saturation) in a color picture. It is a perfect apparatus for creating algorithms based on a color description that is normal and intuitive to people. The Hue component portrays the color itself within the shape of a point between [0,360] degrees. 0 degree mean red, 120 implies green, 240 implies blue. 60 degrees is yellow, 300 degrees is magenta. The Saturation component signals how much the color is contaminated with white color. The extend of the S component is [0,1]. The Intensity range is between [0,1] and 0 implies dark, 1 implies white.

  In [9], RGB to HSI Conversion is used. To detect the black license plate pixels, intensity parameter of HSI color is used.

- **Haar Classifier:**
  Haar Cascade classifier is a successful object detection approach that was proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. Haar-like features are advanced picture features utilized in object recognition.

  It is used in [14].

- **Sobel's mask operator:**
  The Sobel operator, sometimes called the Sobel–Feldman operator or Sobel filter, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges. When we apply this mask on the image it prominent vertical edges. It simply works like as first order derivative and calculates the difference of pixel intensities in a edge region. This mask will prominent the horizontal edges in an image.

  After making the image gray-scale, Sobel filters is used to extract the edging image in [15].

- **Template Matching:**
  Template matching is a procedure in digital picture processing for finding little parts

of a picture that match a template picture. It can be utilized in manufacturing as a portion of quality control, a way to explore a versatile robot, or as a way to identify edges in pictures. The most challenges in the template matching assignment are: occlusion, the discovery of non-rigid changes, illumination, and background changes, background clutter, and scale changes.

For template matching in [11], template pictures of each character in each text style that become the references for the comparison are made and put away in a database. Then, the template matching strategy made a continuous search to discover whether a similar format exists inside the region. For the most part, to create the template matching usable in genuine practice, the size of the candidate pictures is normalized to a predefined measurement, which is precisely the same as that of the template pictures.

- **Histogram Based Approach:**
  A histogram is utilized to summarize discrete or nonstop information. In other words, it gives a visual interpretation. This requires focusing on the most focuses, facts of numerical information by appearing the number of information focuses that fall within a indicated extend of values (called "bins"). It is comparable to a vertical bar chart.

  Histogram equalization is used in [15]. It is also used in [16]

- **Optical Character Recognition:**
  Optical character recognition or optical character reader (OCR) is the electronic or mechanical change of pictures of typed, written by hand, or printed content into machine-encoded content, whether from a filtered report, a photo of a report, a scene-photo (for example the content on signs and billboards in a landscape photo) or from subtitle content superimposed on a picture.

  It is used in [17]. The best result from the OCR would be taken as the converted text value. The distinctive stages were passed through Tesseract OCR and their particular results passed through the ASCII filter.

  In [15] and [16], the character images are passed to the OCR module for recognizing purpose.

- **Square Tracing Algorithm:**
  The thought behind the square tracing algorithm is very straightforward; this can be attributed to the fact that the algorithm was one of the primary attempts to extricate the contour of a binary pattern. This algorithm works as follows; in order

to extricate the contour of the pattern, after finding a dark pixel, move left and each time standing on a white pixel, turn right, until experience the start pixel once more. The dark pixels strolled over will be the contour.

In [11], for identifying the regions of the plate-candidate objects, a contouring algorithm is utilized to discover out the closed boundary objects. A number of candidate judgment algorithms are prepared on the binary picture. One of the contour algorithms named the square tracing algorithm is utilized to isolated plate objects.

- **Hough Transform Algorithm:**
  The Hough Transform is an algorithm licensed by Paul V. C. Hough and was initially designed to recognize complex lines in photos (Hough, 1962). Since its beginning, the algorithm has been adjusted and upgraded to be able to recognize other shapes such as circles and quadrilaterals of particular types.

  In [15], for license plate detection the combination of the Hough Transform and Contour algorithm is used which produces higher accuracy and faster speed.

- **Hidden Markov Model (HMM):**
  Hidden Markov Model (HMM) is a statistical Markov model in which the framework being modeled is expected to be a Markov process – call it X – with unobservable states. HMM expect that there's another process Y whose behavior depends on X. The objective is to memorize about X by watching Y.

  The HMM (Hidden Markov Model) model is used for character recognition in [15].

Table 2.1: Summarization of Literatures

Paper	Data Set	Methods	Accuracy
[1]	Used own dataset	R-CNN model for segmentation	Recognition accuracy for challenging normal is 45% Recognition accuracy for ip-skewed is 86% Recognition accuracy for vp-skewed is 25%

[2]	Used own dataset	YOLOv3 for Number Plate Detection ResNet-20-based CNN with the samples to identify the individual character Bounding Box Coordinate provided by YOLOv3 for segmentation	85% accuracy for number plate detection 92.7% accuracy for character recognition
[3]	Used own dataset	YOLOv3 model based CNN for Number Plate Detection, Segmentation, Character Recognition	The accuracy is around 97% for number plate detection.
[5]	Did not need dataset	Bounding Box on filtered Binary Image for Segmentation of characters and words	Did not learn anything.They Segmented and recognition by hard coding.Maximum Number plate's Number and character can be segmented properly
[12]	Used own dataset	Faster R-CNN to detect the Number Plate	91.6% accuracy for number plate detection
[13]	Used own dataset	CNN Model to detect the Number plate and character recognition	Highest accuracy is 88.7%
[6]	Used own dataset	Morphological Transformation for gray scale image Gaussian blur is used to remove noise for segmentation Bounding box for character extraction Deep learning architecture is used for recognition	93% accuracy for number plate detection 98% accuracy for character segmentation 98% accuracy for recognition
[10]	Used own dataset	MATLAB functions are used	80% accuracy.
[7]	Did not need dataset	MATLAB functions are used to detect number plate and for character recognition	90% accuracy for close range

[14]	Did not need dataset	HAAR characteristic based classifier to discover license plate Class letter separator with a offered procedure CNN for acknowledging class letters	96.92% precision for detection 94.61% accuracy for class letter segmentation 90.90% accuracy for recognition rate with real-time performance
[11]	Used own dataset	Square tracing algorithm for finding number plate bounding contour Rotation matrix for skewness Horizontal and vertical estimates with threshold to divide Bangla characters and digits effortlessly	Detection precision is 93% Segmentation precision is 98.1% Recognition precision is 88.8%
[9]	Did not need dataset	HSI color model Geometrical properties Candidate Regions Intensity Histogram	The precision of detection is 85%.
[17]	Used own dataset	Tesseract OCR machine on processed image for License plate detection	The final result was very close to the genuine value on the plates.
[8]	Used own dataset	Sobel's mask operator Otsu's thresholding method Connected Component Analysis Feature extraction	License plate detection accuracy 95% Character segmentation accuracy 99% License plate recognition accuracy 91%
[15]	Used own dataset	Sobel filters Local adaptive thresholding Hough Transform Contour algorithm OCR module HMM (Hidden Markov Model)	License plate detection accuracy 98.76 for both image set% Character segmentation accuracy 97.61 for both image set% OCR module accuracy 97.52 for both image set% Whole system accuracy 92.85%
[18]	Used own dataset	Faster R-CNN	Not mentioned.

[4]	Used own dataset	YOLO based network	Segmentation 99%, recognition 93%.
[16]	Used own dataset	Histogram based approach, Optical character recognition (OCR)	Not mentioned.

2.3 Study on ANPR Systems of Other Countries License Plate

In this section, we are going to discuss about works that are done on Saudi Arabian, Vietnamese and Chinese license plate recognition.

2.3.1 ANPR system for Saudi Arabian license plate

In [8], they completed their research using main 4 steps. They used image pre-processing then they detected the LP, segmented the characters and then finally recognized the characters.
For license plate detection they used Radial Basis Function network (RBF). At first, it converts the gray-scale image into binary and performs edge detection. For edge detection, they used Sobel's mask operator. Then, they performed dilation. Flood Fill algorithm is used to fill the gaps. Then they did noise removal and finally contour of the selected area is used.

Then, by using threshold value, they fixed a range to obtain the license plate only. Objects outside that range were removed. Then they used image processing operation and prepared the license plate for character recognition.After that using feature extraction method they recognized the Arabic characters and the numbers. Two RBF NNs are used to train the width and length of the image.

Then, license plate is extracted from the image. The borders of the plates are removed. The x and y coordinates of top left and bottom right corners of the LP are used to crop the license plate.

After removing the plate borders, the segmentation step starts. The process is based on thresholding and Connected Component Analysis (CCA). On the cropped LP from the preceding step, the steps that has been followed are: Otsu's method to convert an image in binary format, Median filtering for noise removal and CCA to label the objects.

The final step is character recognition which is done by feature extraction. The features help to determine non-number images and Arabic words. This study has total 10 number of features, which are 8 pixel's feature and 2 ratio features.
The accuracy of license plate detection is 95%, character segmentation is 99% and license plate recognition is 91%.

Figure 2.1: Saudi Arabian License Plate

2.3.2 ANPR system for Vietnamese license plate

In [15], they completed their research using major 4 steps which are: image pre-processing, then LP detection, segmentation of the characters and then finally Optical Character Recognition (OCR).

At first some image processing operations were applied to prepare images for detecting the license plate. After converting the image in gray-scale, Sobel filters is used for detecting edge from the image. Local adaptive thresholding was used for image binarization.

Then for detecting the license plate, they used Hough Transformation and Contour Algorithm combined from the edging image. It has higher accuracy and faster speed so, it can be used in actual time systems.

Then they used plate-candidate algorithm to segment the characters. Then HMM(Hidden Markov Model) was used for character recognition.

They took images in two camera positions which are:

1. Airport office where angles are: right, left - 30°. The images are taken during 10-12 A.M. It is denoted by A.

2. Random places where angles are: right, left - 30° or straight. The images are taken in morning or night with flash light. It is denoted by B.

For license plate detection step, the accuracy for image set A is 99.27%, 98.2% for image set B and 98.76% for A and B combined.
For character segmentation, the accuracy for image set A is 98.05%, 97.13% for image set B and 97.61% for A and B combined.
For OCR module, the accuracy for image set A is 97.82%, 97.19% for image set B and 97.52% for A and B combined.
For the whole system, the accuracy is 92.85%.

Figure 2.2: Vietnamese License Plate

2.3.3 ANPR system for Chinese license plate

[18] is a work on Chinese license plates. Compared with the license plates of other countries, the kinds of Chinese license plates are more diversified. The present plates are 1992 standard. It consists of the one-character territorial short form, a character of the alphabet and five numerals or characters of the alphabet. Previously, all number plates used the five-number classification.

Chinese license plates has four traits. They are listed below:

1. Color characteristics.
2. Text features.
3. Geometric features.
4. Texture features.

They applied a database consisting one hundred and ninety photos of license plates. These pictures have disparate surroundings, angles and illumination circumstances. One to two observation points are involved with every image. There are four stages to train the framework. The first two stages train the networks which are used in Faster R-CNN. The networks are: region proposal and detection networks. The last two stages unifies the two networks from the previous two steps in a manner that a single network is produced for detection. The learning rate of initial two stages are fixed larger than the last two stages as those are adjustment stages.

The initial step is to train a RPN employed in Faster R-CNN. Then, they trained a fast R-CNN adopting the RPN from the first stage. To unite the two networks, they employed interchangeable training and re-training RPN utilizing weight contributing Fast R-CNN.

To completely figure out the system, they used balanced accuracy to calculate the capability to precisely do classifications and the capability to detect every applicable targets.

Figure 2.3: Chinese License Plate

2.3.4 ANPR system for Brazilian license plate

In [4], They are employing DL(Deep Learning) methods studied to be analytically costly, averting small targets, turns into even more essential when real time system is needed. They used a YOLO based network as these are said to be the fastest networks even swifter than Faster R-CNN which does detection and recognition applying region proposals not having the need of picture pyramid and sliding windows. In experiments, the FAST-YOLO performed twenty class detection and categorization in 800×600 pixels images in less than 5.5ms or about 180 frames per second(FPS).

A new network was built which was motivated by the YOLO structure for detecting and recognizing characters, with basic practical dissimilarities to fit in with thirty five classes (0-9, A-Z excluding O because it is spotted in conjunction with the number 0). It outputs a feature map with similar screen proportion of license plate where height is one-third times smaller than width.
The vehicle front view and its license plate are spotted by utilizing an unique classifier organized in a cascaded style. The initial layer detects the front view from the inserted image, and second layer extracts the number plate from the spotted front view picture.

To attain a better settlement between precision and execution times, the classifier is centered on the FAST-YOLO structure. This network was constructed and trained to control twenty diverse classes of things and executes at two hundred FPS. We hypothesized that the FAST-YOLO network altered for two classes could own the ability to help both of the works in sole network when implemented in a cascaded manner.

In license plate detection step, first process implies the entire picture and searches only for front views. Any license plate discovered is rejected. Then, the recognized front views are cropped and delivered to the similar network and solely the result associated to license plate is utilized. If numerous license plates are noticed, solely the one with the maximum chance is stored.

For character detection and recognition, first the number of max pooling layers needed to bring down from five to three so that it can keep the fine outcome coherency by escaping numerous amplitude cutbacks. Next, to preserve the network extent equivalent to FAST-YOLO and still granting to utilize as much transfer learning as achievable. Then the initial 11 layers of YOLO is utilized, discontinuing on the 12th layer, as it consists of the 14th max pooling in the system. At last, for the betterment of non-linearity, 4 additional layers were attached and trained from the very beginning.

If a letter is identified in the plates associated to digits, it is shifted by the digit that has the extensive occurrence in the confusion matrix acquired with the training samples. The similar method is employed when a digit is spotted in the plates associated to letters.

The system successfully detected and recognized all 7 characters of a registration plate in 63.18% of the test set and 97.39% when acknowledging at least five accurate characters. The accuracy of segmentation of the characters was 99%, and the system could accurately recognize 93% of the characters.

Figure 2.4: Brazilian License Plate

Chapter 3

Methodology

3.1 Proposed Work Flow

The working structure of the proposed system is shown below. It consists of video acquisition, vehicle and license plate detection, cropping the license plate, some image preprocessing like- gray scale, binarization, character segmentation, template matching and finally license plate recognition.

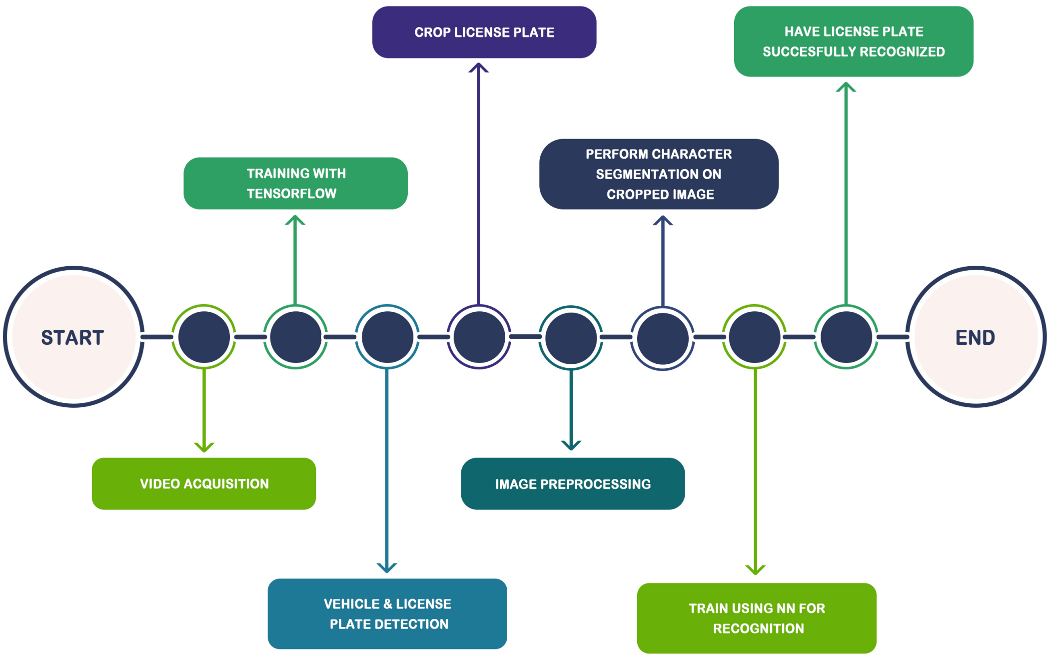

Figure 3.1: Work Flow

3.1.1 Video Acquisition

In our work, we used a phone to capture few videos and pictures which were captured inconstantly from distinct angels on moving vehicles on the streets. The videos were captured in color mode. The standard of the videos were diverse based on the neighbouring circumstances.

Figure 3.2: Video Acquisition

3.1.2 Training with Tensorflow

Data labeling is the way of detecting and marking data samples. This can be done manually but is also performed or assisted by software. It is a necessary element for supervised learning. Both input and output information are marked for classification to provide a learning base for future processing.
We trained our system to identify vehicles and license plates in images. We provided the system with numerous images of diverse types of vehicles from which it learned the common features. So, it was able to precisely identify the vehicles and license plates in

unmarked pictures.
To label and remark the training and test pictures, LabelImg was utilized which is an publicly-available image annotation tool. We did the labeling of the objects from images as car, bike, license plate.
The annotations were stored as XML files. The XML files were then transformed into CSV files. The CSV files were modified into tfrecord format. At last, the tfrecord files were employed as input information to train the models.

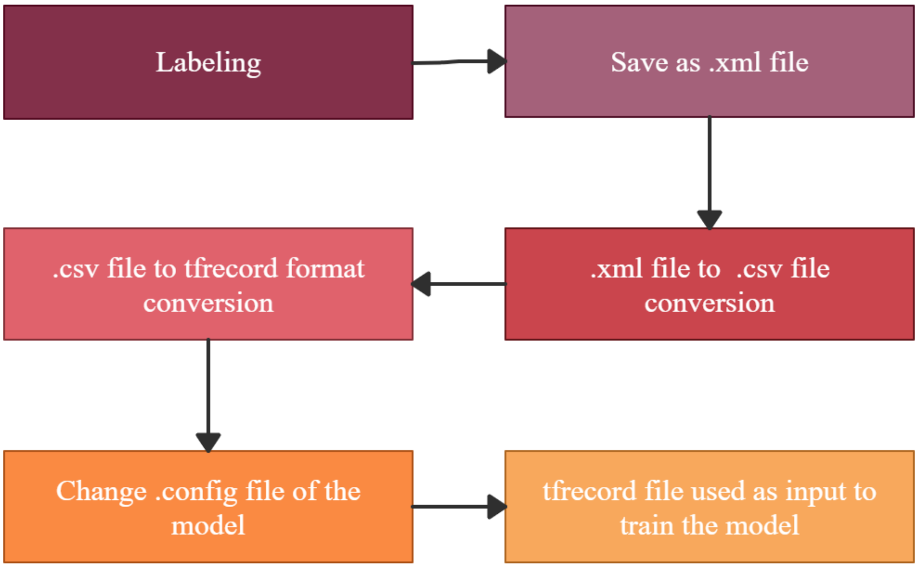

Figure 3.3: Data Processing for Training the System to Detect LP

The model could be examined with additional test specimens, but due to its excellent outcome in every test in the detection and recognition of characters and numerals further tests were not done.

3.1.3 Vehicle and License Plate Detection

We used four different models to train the system to detect vehicle and license plate and compared each of their accuracy and speed. The models are: Faster R-CNN, SSD Mobilenet V1 COCO, SSD Mobilenet V1 FPN COCO and SSDlite Mobilenet V2 COCO. The basic architecture and working procedure is discussed in the next section.

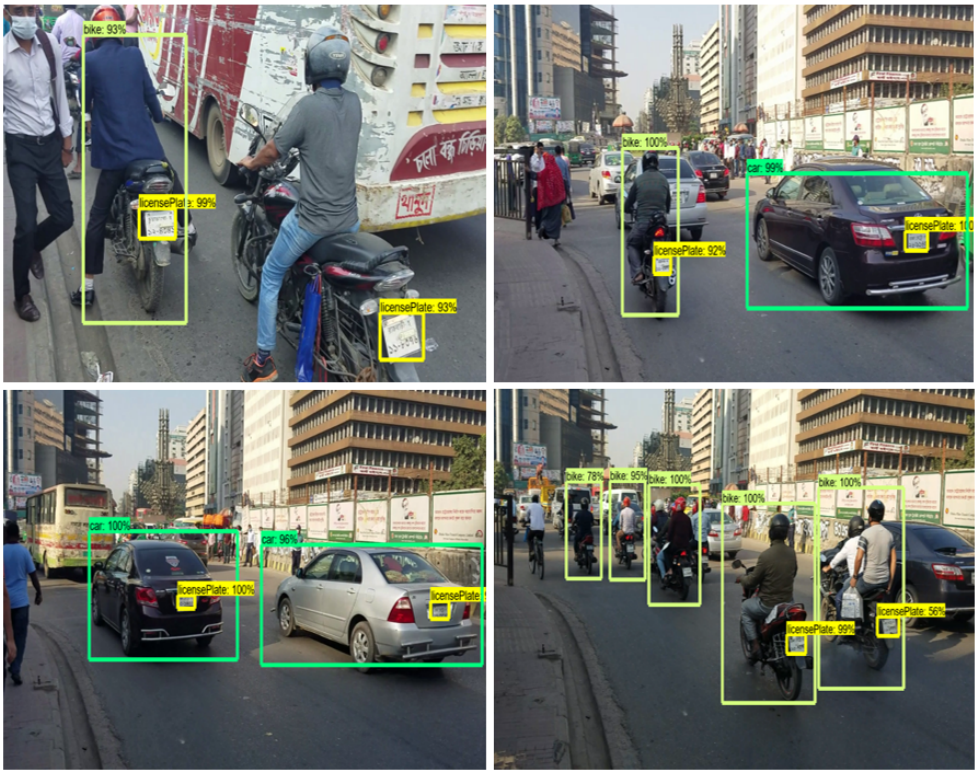

Figure 3.4: Vehicle and License Plate Detection

3.1.4 Crop License Plate

From the detection step, we get multiple four coordinates value of bounding boxes. It also gives the class for all the bounding boxes. According to that class, license plate is cropped using bounding box. And then, the cropped license plate image is saved.

Figure 3.5: Some of the License Plates after Cropping

3.1.5 Image Pre-processing

Image pre-processing is the title of the activity on pictures at the lowermost level of abstraction whose aim is an advancement of the picture information that conceals undesirable deformities or upgrades a few picture features vital for advanced handling. It does not increment picture data content. These activities do not increment picture data but they diminish it in case of measuring entropy data.

Some of the pre-processing methods that are used in our system are listed below:

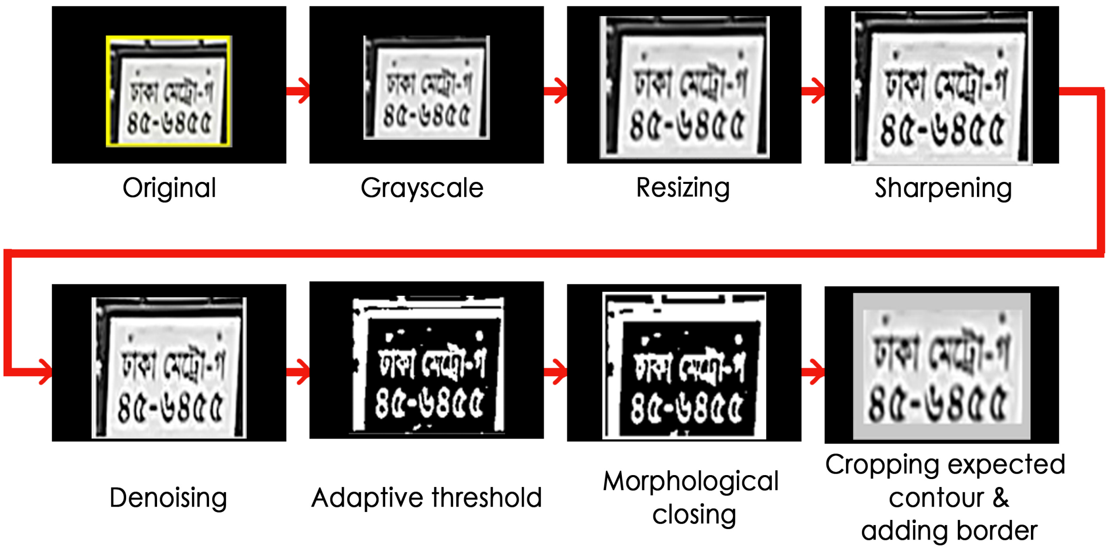

Figure 3.6: Flow Chart of Image Pre-processing steps

Gray-scale Conversion

Using cvtColor function an image is converted from one color mode to another different one. In case of a change to-from RGB mode, the arrangement of the channels ought to indicate specifically (RGB or BGR). The primary color format in OpenCV is frequently published as RGB but it is really BGR. So in a standard color image the fundamental byte will be an 8-bit Blue element, the second byte will be Green, and the third byte will be Red. The fourth, fifth, and sixth bytes will be the second pixel (Blue, Green, Red).

A natural approach to alter a color picture 3D array to a gray-scale 2D array is, to take the mean of the red, green, and blue pixel values to get the gray-scale estimate. This combines the light or illuminated light contributed by each color band into an acceptable gray estimation. The algorithm uses the formula:

$$\frac{R+G+B}{3}$$

for all pixels in an image so it can convert to gray-scale images.

To our eyes, green appears about ten times intense than blue. Through numerous recurrences of thoroughly planned attempts, psychologists have estimated how distinctive we see the red, green, and blue light. They have given us a diverse group of weights for the average of our channel to get total luminosity.

Sharpening

Upgrading the high-occurrence elements in a picture, improves visual quality. The image is quoted in any upgraded method that sharpens the image and highlights the edges and details in an image. . Image sharpening includes a signal in the first image that is proportional to the high-pass filtered form of the first image.

The first picture is first sifted by a high-pass filter that pulls out the high-frequency components, and after that, an ascended variant of the high-pass channel outcome is included in the first picture, in this way creating a sharpened picture of the initial. Homogeneous regions of the signal, i.e., where the signal is consistent, are not unobstructed. The sharpening process can be represented by:

$$S_{i,j} = x_{i,j} + \gamma F(x_{i,j})$$

Here,

$x_{i,j}$ is initial pixel value at the coordinate i, j.

F is high pass filter.

γ is the adjustment variable which is greater that or equal zero.

$S_{i,j}$ is the sharpened element at the coordinate i, j.

The key to an useful sharpening operation lies in the preference of high-pass filtering operation. Conventionally, linear channels have been utilized to execute the high-pass channel, nevertheless, linear procedures can head to unsatisfactory consequences if the initial picture is adulterated with disturbance. A trade-off between disturbance attenuation and edge highlighting can be obtained if a weighted median channel with relevant weights are employed. The weight we used is:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Denoising

We edited the image using a Non-local Means Denoising method with a few computational enhancements. Noise anticipated being a Gaussian white noise. We applied it on our gray-scale images.

Gaussian blur is also used. It is a kind of blurring technique that employs a Gaussian function which uses the normal distribution for examining the alteration to apply to every pixel in the picture. Formula of Gaussian function for one dimension:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Here,

σ is the normal deviation of the Gaussian distribution.

Adaptive threshold algorithm is used also.

Binarization

Binarization is done using Otsu's thresholding method. The algorithm broadly identifies marginal class change as a weighted sum of two class changes and reduces marginal class change.

$$\delta_w^2 = \omega_0(t)\delta_0^2(t) + \omega_1(t)\delta_1^2(t)$$

Here,

ω_0 and ω_1 are weights which are the chances of the two classes separated by threshold t.

δ_0^2 and δ_1^2 are differences of these two classes.

Dilation

We applied dilation to our binary images. It continuously extends the boundaries of the pixel regions of the foreground and develops the pixel fields. The gaps in the regions become even smaller.

Erosion

Erosion is used in binary images to take off the borders of sections of the foreground pixels. Thus, the areas of the pixels on the top decrease in size and holes between the regions become greater.

3.1.6 Character Segmentation

Bangla characters are exceptionally troublesome to fragment since of their diverse complications in Bangladeshi number plates . To cope with those, an algorithm is offered that employs horizontal and vertical projections with thresholds to fragment Bangla characters and digits correctly.

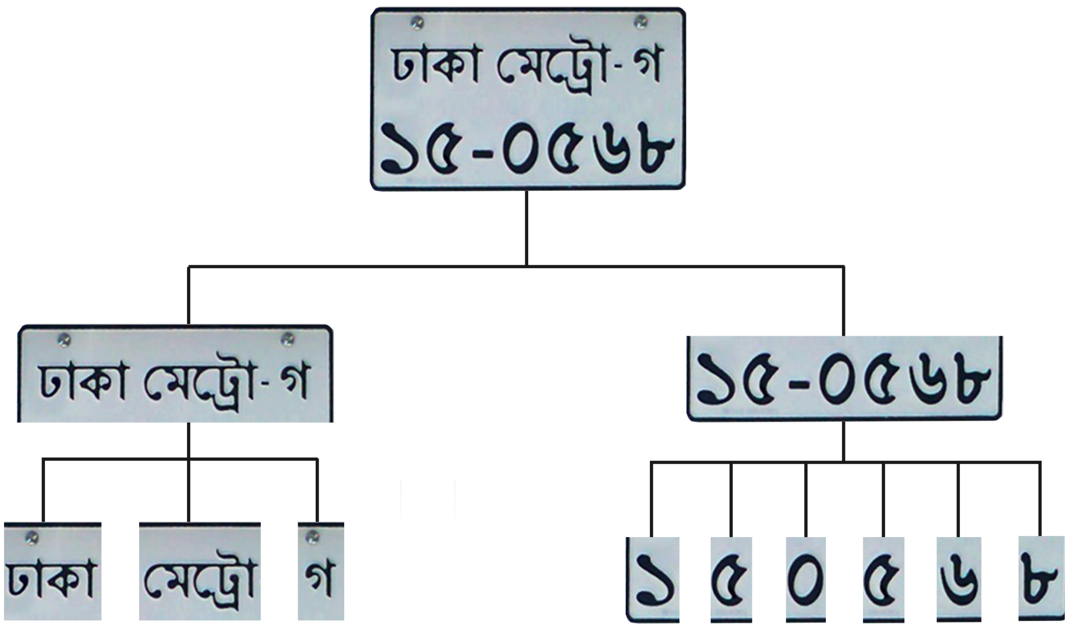

Figure 3.7: Segmentation Process

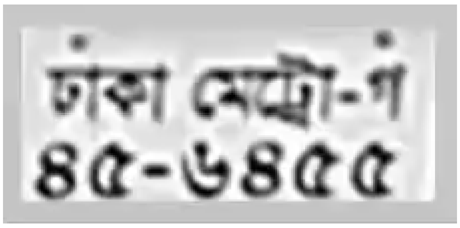

Figure 3.8: Before Segmentation

The segmentation process has two steps which are:

1. **Separating the rows:**
   Bangladeshi number plates contain two parts of text.Horizontal projection is performed by using inspecting general quantity of row pixels which sections the rows in two distinctive lines. The row with the minimal values of horizontal pixels are the beginning or the ending of a row in the plate.

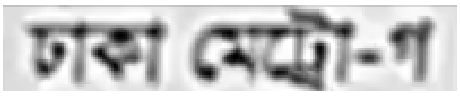

Figure 3.9: After Horizontal Segmentation 1st Row

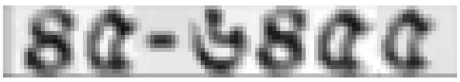

Figure 3.10: After Horizontal Segmentation 2nd Row

2. **Separating the characters and digits:**
   Vertical projection is used to isolate each Bangla character and words.By expelling the repetitive region from each outcome of the partitioning process the desired characters and digits can be established.

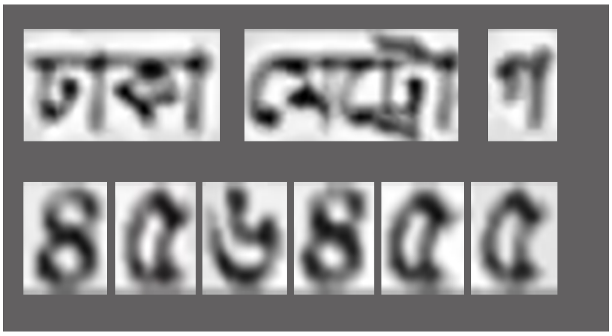

Figure 3.11: Segmented Bangla Words and Digits

3.1.7 Train System using Neural Network for Recognition

We trained our system using neural network. A neural network is a sequence of procedures that endeavors to acknowledge fundamental connections in a set of information through a system that imitates the idea of how the human brain works. In this sense, neural networks cite to systems of neurons, either natural or fabricated in nature. It can adjust to switching input, so the network creates the finest conceivable outcome without having to update the output basis.

Neural Network

A neural network operates comparably to the neural network of human brain. A "neuron" is a scientific function that assembles and categorizes data agreeing to a precise design. It carries a solid likeness to factual strategies such as curve fitting and regression analysis.

It contains layers of linked junctions. Each junction is a perceptron and is comparative to numerous linear regression. The perceptron sustains the signal delivered by a several linear regression into a stimulation work that is nonlinear. In a multi-layered perceptron (MLP), perceptrons are organized in lined layers. There are three types of layers which are:

1. **Input Layer:** The input layer collects input styles.
2. **Output Layer:** The output layer has classifications or output alerts to which input designs may also map. As an instance, the styles may additionally encompass a listing of portions for specialised signs about safety.
3. **Hidden Layer:** Hidden layers great-tune the enter weightings until the neural network's fringe of mistake is negligible. It is hypothesized that hidden layers extrapolate exquisite features in the input data which have predictive manage with

recognize to the outputs. This depicts characteristic extraction, which finishes a application comparative to factual approaches which includes important component evaluation.

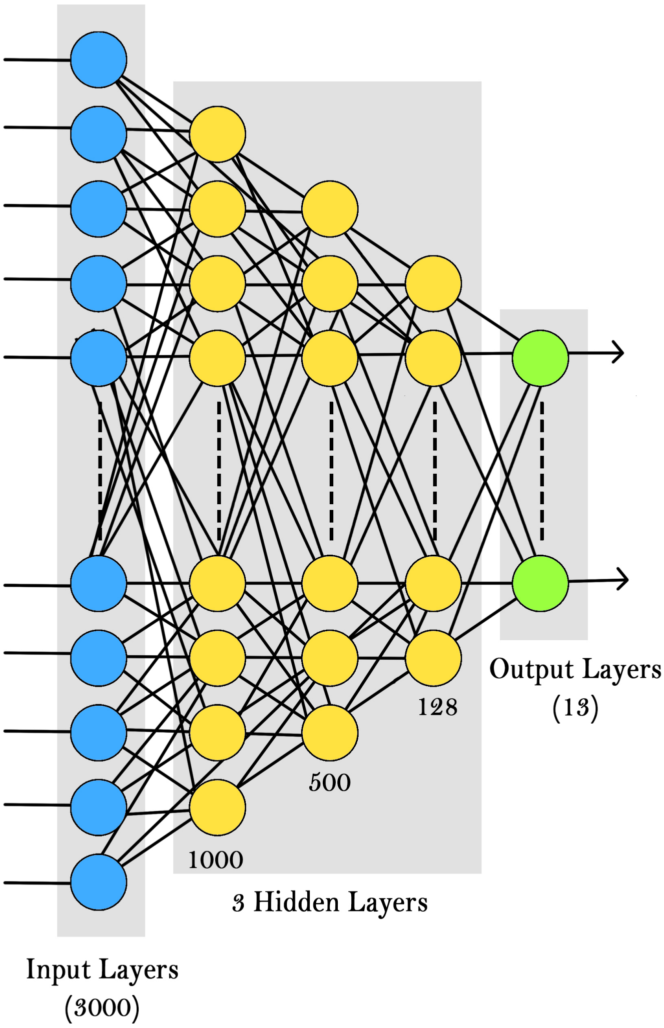

Figure 3.12: Neural Network of Our System

The hyper-parameters we used are:
Batch Size = 50
Number of Iterations = 10000
Input Dimension = 100*30 = 3000
Number Features = 784
Output Dimension = 13
Learning Rate = 0.01
Hidden layer = 3

For the first and second hidden layers:
We used Linear function and for activation function we used Leaky ReLU (Rectified Linear Unit). Leaky Rectified Linear Unit, or Leaky ReLU, is a kind of activation function

established on a ReLU, but it blends a little slope for negative values rather than a flat slope. The slope coefficient is decided previously before training, i.e. it is not determined during training. This sort of activation work is prevalent in tasks where we may suffer from inadequate gradients.

For the third hidden layer:

We used Linear function and for activation function we used Sigmoid. A sigmoid function is a numerical function having a distinctive ”S”-shaped curve or sigmoid curve. A usual case of a sigmoid function is the logistic function. It helps in lessening the time needed for creating models, there is a major disadvantage of information loss because of the derivative of a short-range.
It is characterized by the equation:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x).$$

Table 3.1: Sigmoid and Leaky ReLU Functions Comparison

Name	Equation	Range
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	(0, 1)
Leaky ReLU	$f(x) = \begin{cases} 0.01x; & x < 0 \\ x; & x \geq 0 \end{cases}$	$(-\infty, \infty)$

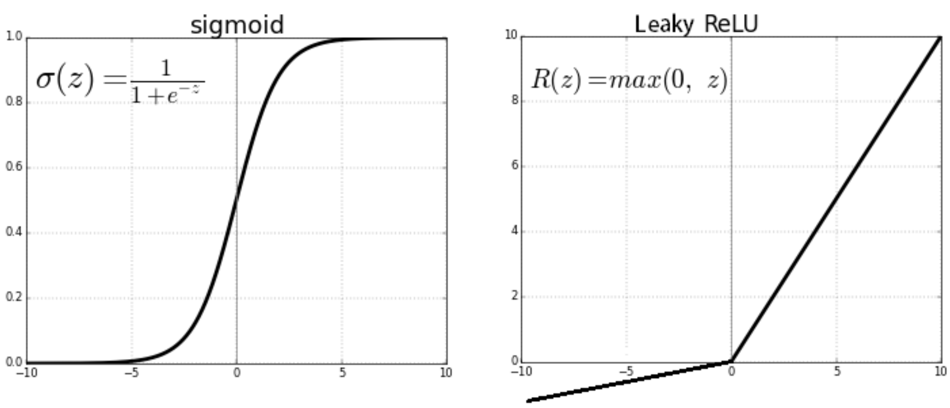

Figure 3.13: Sigmoid and Leaky ReLU Functions Graph Plot

Pictures of each character in each text style that becomes the references for the comparisons are made and put away in a database. At that point, system persistently search to discover whether a comparable reference exists inside the range. For the most part, to make it usable in real practice, the estimate of the candidate pictures are normalized to a predefined measurement, which is precisely the same as the reference.

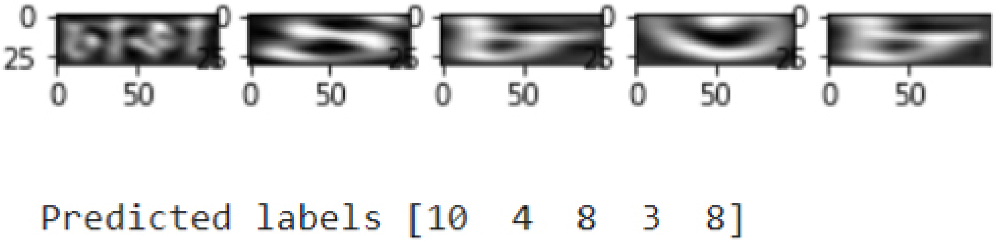

Figure 3.14: Prediction Using NN

3.1.8 License Plate Recognition

We trained our system using neural network. The outputs were satisfactory giving approximately 88.8% match between the trained samples and the test samples. After recognizing the license plates will be saved in a CSV file in text format. We have more work to do in terms of showing the output in a formal way which we intend to finish in the near future.

34	DhakaMetroGa-350374	04-07-21 15:41
35	DhakaMetroGa-456455	04-07-21 15:41
36	DhakaMetroGa-	04-07-21 15:41
37	DhakaMetroGa-456455	04-07-21 15:41
38	3MetroGa-	04-07-21 15:41
39	Ga-Ga-Ga-DhakaDhaka	04-07-21 15:42
40	Ga-DhakaGa-DhakaDhaka	04-07-21 15:42
41	DhakaGa-Ga-Dhaka	04-07-21 15:42
42	DhakaMetroGa-160757	04-07-21 15:42
43	DhakaMetroGa-160754	04-07-21 15:42
44	DhakaMetroGa-160754	04-07-21 15:42
45	MetroDhakaMetroGa-160754	04-07-21 15:42
46	MetroDhakaMetroGa-160754	04-07-21 15:42
47	DhakaMetroGa-3754Ga-	04-07-21 15:42
48	DhakaMetroGa-3754Ga-	04-07-21 15:43
49	DhakaMetroGa-147838	04-07-21 15:43
50	DhakaMetroGa-147838	04-07-21 15:43
51	DhakaMetroGa-147838	04-07-21 15:43
52	DhakaMetroGa-788	04-07-21 15:43

Figure 3.15: Screenshot from the CSV file after Recognition

3.2 Models used to train for vehicle and license plate detection

3.2.1 Faster R-CNN

Here in Faster R-CNN, the term R-CNN means Region based Convolutional Neural Network. It is a Modified model of R-CNN which Overcomes some issues of R-CNN. It is faster than R-CNN which is one of the main advantages of it.

1. At first, an input image is taken and passed through Convolutional Network which returns feature maps of the image.
2. .Then, it uses Region Proposal Network also called as RPN. RPN is applied on feature maps which returns object proposals.
3. A new layer was proposed called ROI (Region Of Interest) pooling that extracts feature vectors which are equal length from all proposals in the given image which makes it faster.
4. Finally the proposals are passed through classifier layer to classify and output the bounding box for objects. It used RPN to create bounding boxes.
5. Like R-CNN, Faster R-CNN has not multiple stage such as: region proposal generation, feature extraction, and classification using SVM.it creates a network which consists of only a stage.

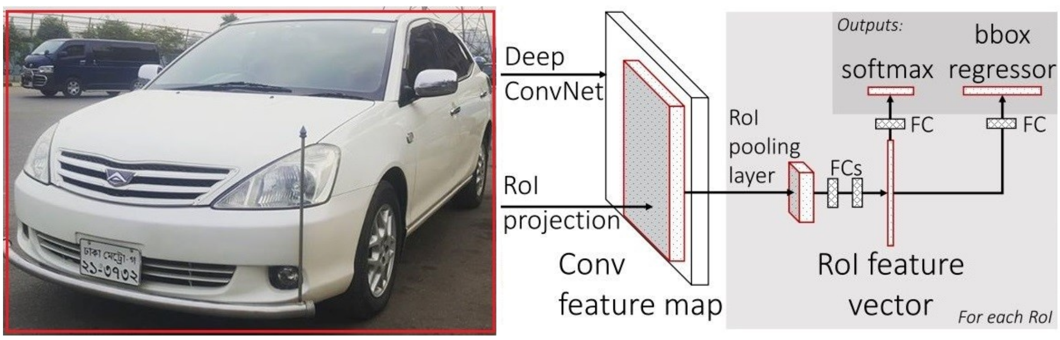

Figure 3.16: Faster R-CNN

The image shown above using one stage instead of three stages (region proposal generation, feature extraction, and classification using SVM). The Model simply takes an image as input and returns class probabilities (Softmax Layer) and bounding boxes (FC Layer) of the detected object.

The accuracy of faster R-CNN is very high but the whole processing time is far below what a real time processing needs. so, in this real time detection and recognition work, we did not use this model because it will not give us desired output within desired time.

3.2.2 SSD Mobilenet V1 COCO

Next model which has been used in this work is SSD Mobilenet V1 COCO. Here SSD stands for Single Shot multi-box Detector and COCO stands for Common Objects in Context. It is a SSD Network which performs object detection in real time. Mobilenet SSD is used only for face detection but SSD Mobilenet V1 COCO can detect objects as well.

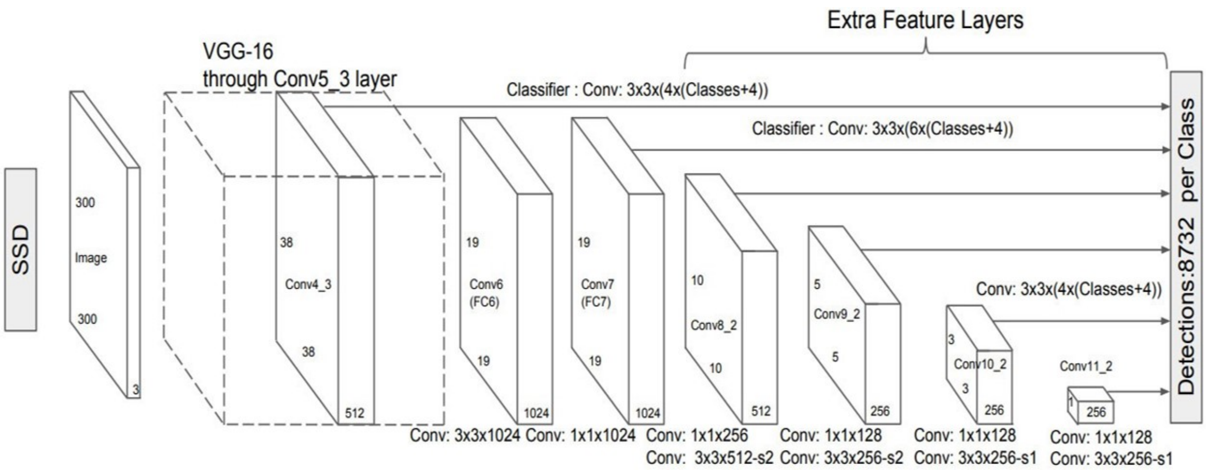

Figure 3.17: SSD Mobilenet V1 COCO

It predicts bounding box and class simultaneously in a single shot. For base network a truncated version of VGG16 is chosen. An input image is passed through all convolution layers to generate multiple feature maps of different sizes.

In Our Real time Number Plate Detection , The accuracy and speed of SSD Mobilenet V1 COCO is high. The accuracy may not be as high as faster R-CNN but, it can be used for real time detection because of its faster performance.

3.2.3 SSD Mobilenet V1 FPN COCO

Another model that is used is SSD Mobilenet V1 FPN COCO. Here, FPN means Feature Pyramid Network.

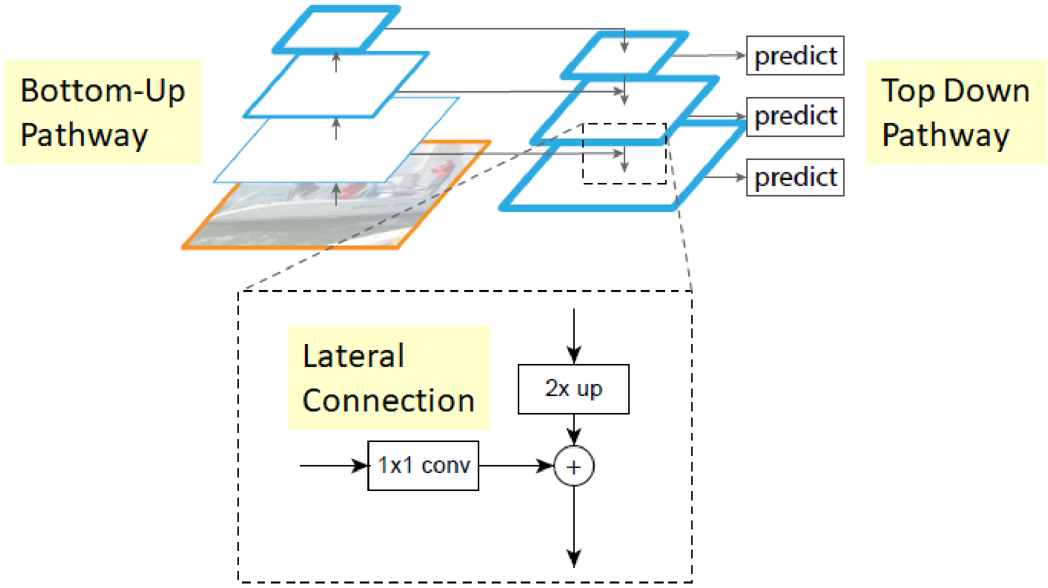

Figure 3.18: SSD Mobilenet V1 FPN COCO

The working procedure of it is given below:

1. Here, bottom up pathway is the feed forward for the backbone CNN.
2. One pyramid level for each stage is defined.
3. The output of last layer of each stage is used as reference of feature maps for enriching top down pathway.
4. Each lateral connection merges feature maps of the same size from bottom up and top down pathway.
5. Finally, 3X3 convolution is applied on each merges map.

As like Faster R-CNN, the accuracy of SSD Mobilenet V1 FPN COCO is very high but it takes a lot of time to detect an object which means the speed of it is very slow. In real time detection and recognition, it will not give desired output. Because of this problem, it cannot be used for real time detection.

3.2.4 SSDlite Mobilenet V2 COCO

The next one is SSDlite Mobilenet V2 COCO.It is also called SSDlite M2. It is a of SSD and Mobilenet V2. SDLite is a mobile-friendly alternate of SSD.

Based on Bottleneck Residual Block (BRB), the SSDlite MobileNet V2 can achieve approximately 8.3 times compression rate compared with the SSD and can maintain the same level of correctness. The model size of it is lower than other high-accuracy models but it can only acquire 5 frames per second (fps) on a high-end embedded CPU. It still do not accommodate the requirement of real-time processing.

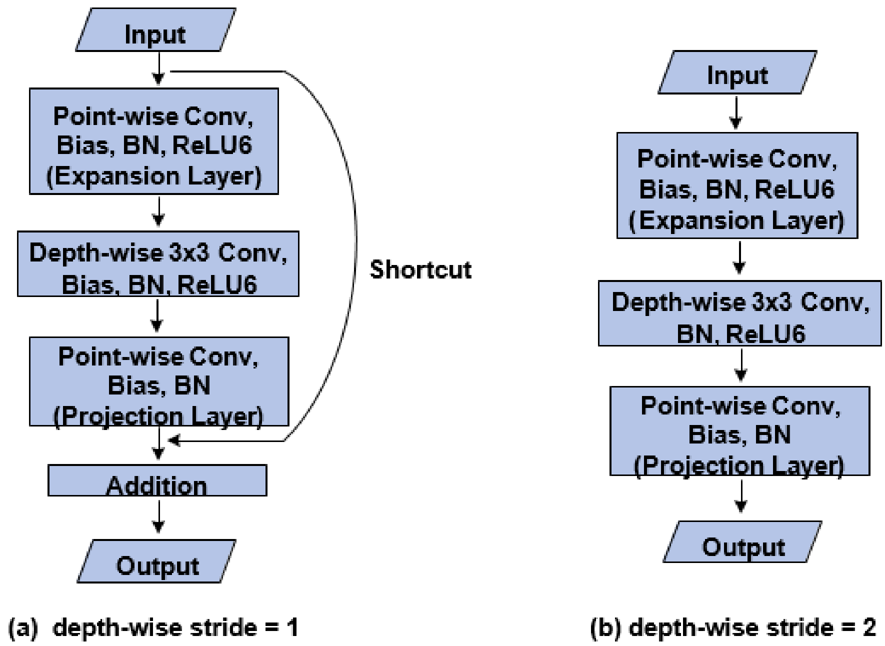

Figure 3.19: The structure of bottleneck residual block

The Bottleneck Residual Block is:

1. There are two types of blocks. One is residual block with stride of 1. Another one is block with stride of 2 for downsizing.
2. There are 3 layers for both types of blocks.
3. This time, the first layer is 1×1 convolution with ReLU6.
4. The second layer is the depth-wise convolution.
5. The third layer is another 1×1 convolution without any non-linearity. It is believed that if ReLU is employed again, the deep networks have the power of a linear classifier on the non-zero volume part of the result domain.

The SSD is a popular framework for object detection. It has two components which are: feature extractor and bounding box predictor. The feature extractor is also called

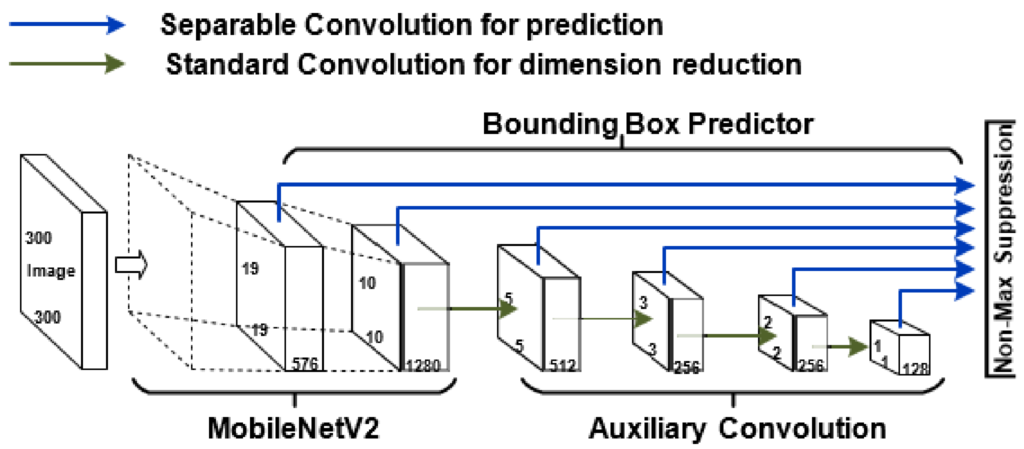

Figure 3.20: The network architecture of SSDlite Mobilenet V2

base network. It is commonly a truncated classification network such as VGG-16. It is followed by a set of auxiliary convolutional layers which permits features extraction at several scales and decrease the input size of each following layer. The bounding box predictor is a small cluster of convolutional filters applied to anticipate category scores and box offsets for a fixed set of default bounding boxes. It generates multiple box for the same object after 10000 steps. This problem was solved after 20000 steps. The speed of it is very high but the accuracy is a bit low than the SSD Mobilenet V1 COCO. Because of this problem, it cannot be used for real time detection.

Table 3.2: Training parameters for the models which has been used in vehicle and license plate detection

Model	Batch Size	Learning Rate	Steps
Faster R-CNN	1	0.0002	100000
SSD Mobilenet V1 COCO	24	0.004	20000
SSD Mobilenet V1 FPN COCO	24	0.4	10000
SSDlite Mobilenet V2 COCO	24	0.004	20000

Chapter 4

Experimental Result

In this chapter, we have added classification and localization loss graphs for each models that are used to train the system for detection license plates. The localization graph shows loss occurred when the object was not detected. The classification graph shows the loss occurred while not being able to detect each class.
For Faster RCNN, we trained our model for 100k steps, for SSD Mobilenet V1 COCO 20k steps, for SSD Mobilenet V1 FPN COCO 10k steps and for SSDlite Mobilenet V2 COCO 20k steps.
If we continued to train our system further, it will become over fitted.

4.1 Localization Graphs

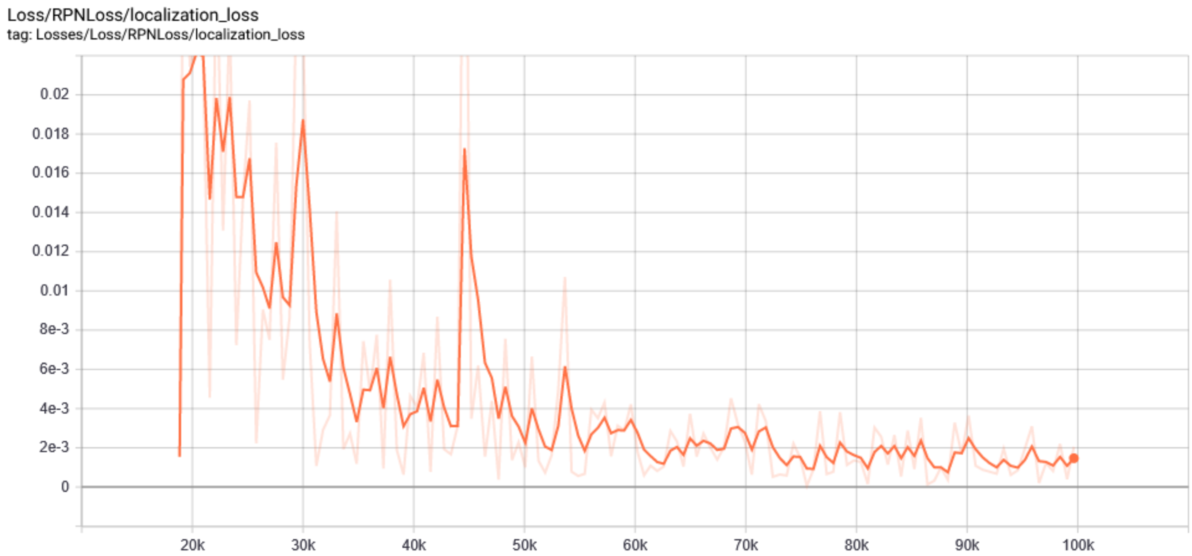

Figure 4.1: Localization Graph of Faster RCNN

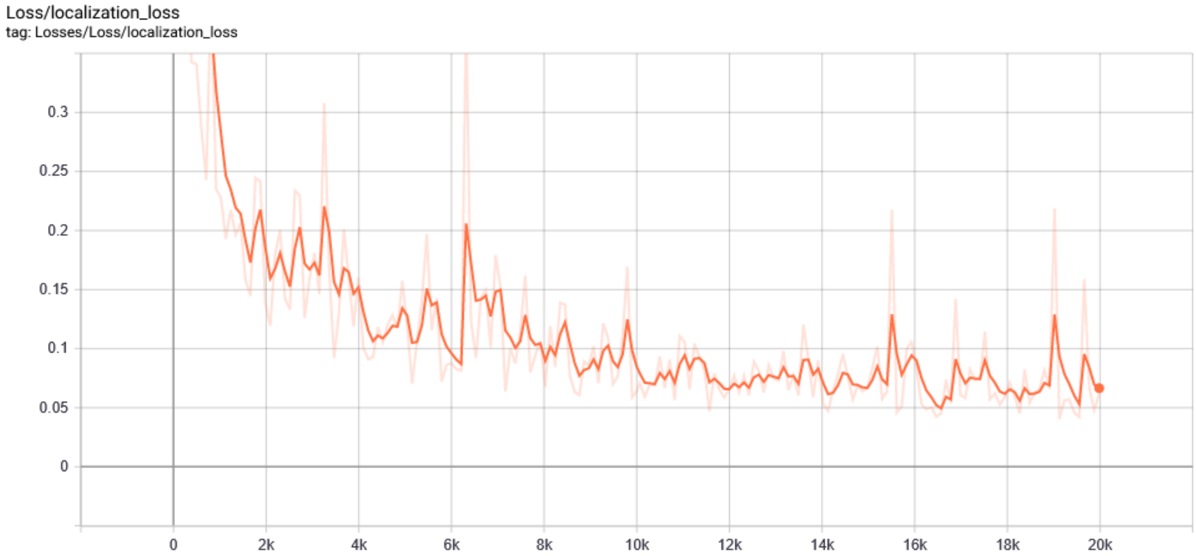

Figure 4.2: Localization Graph of SSD Mobilenet V1 COCO

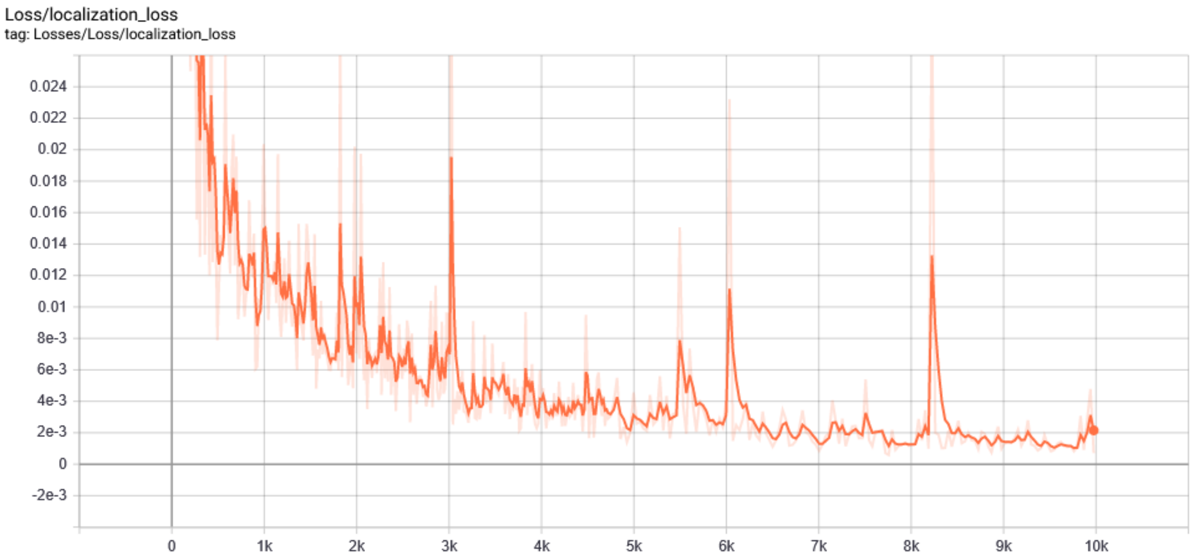

Figure 4.3: Localization Graph of SSD Mobilenet V1 FPN COCO

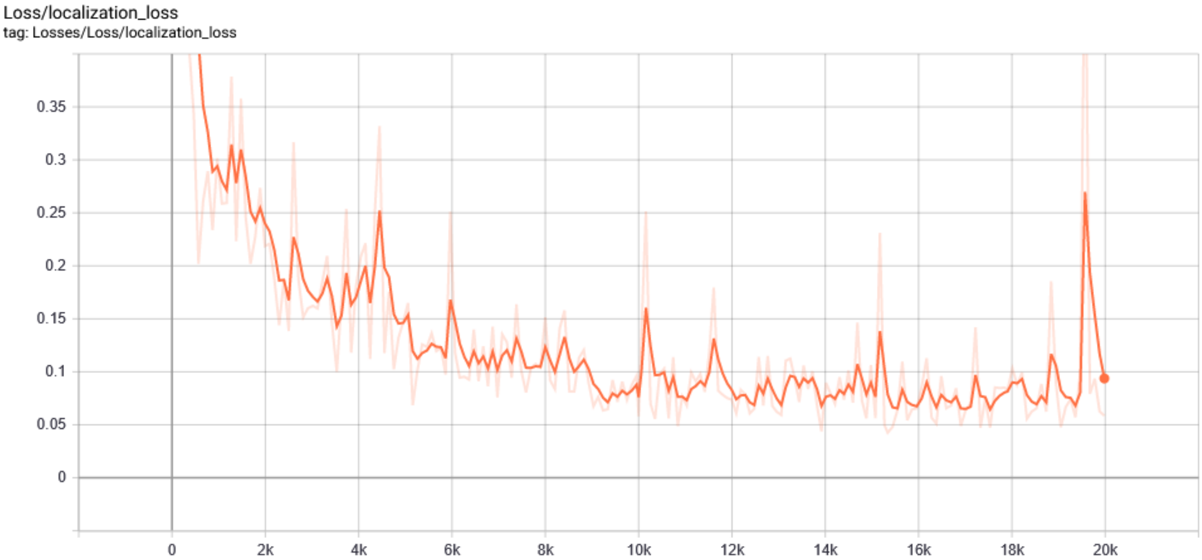

Figure 4.4: Localization Graph of SSDlite Mobilenet V2 COCO

4.2 Classification Graphs

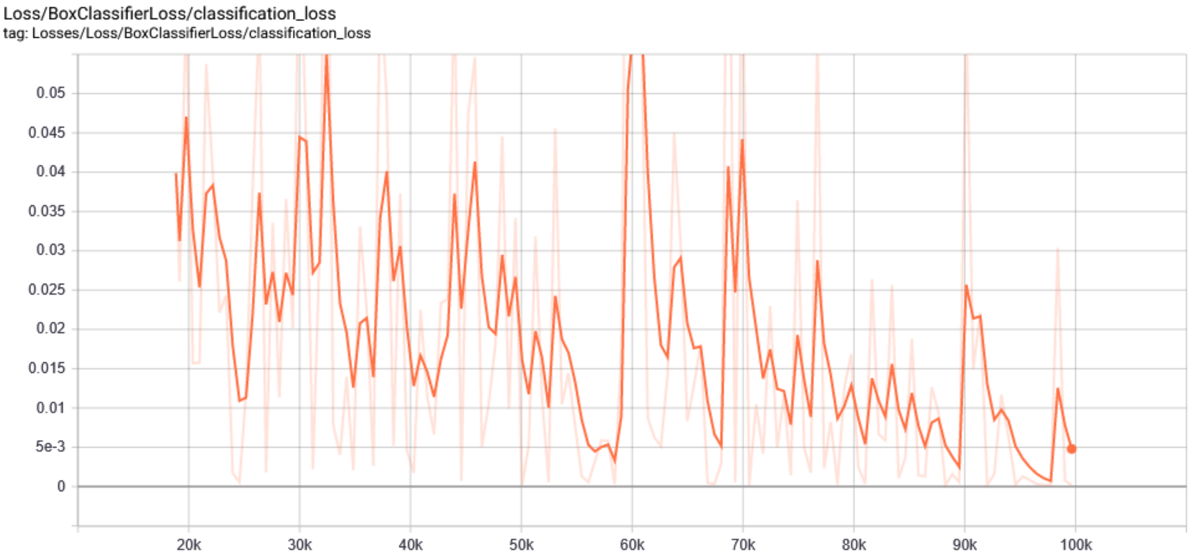

Figure 4.5: Classification Graph of Faster RCNN

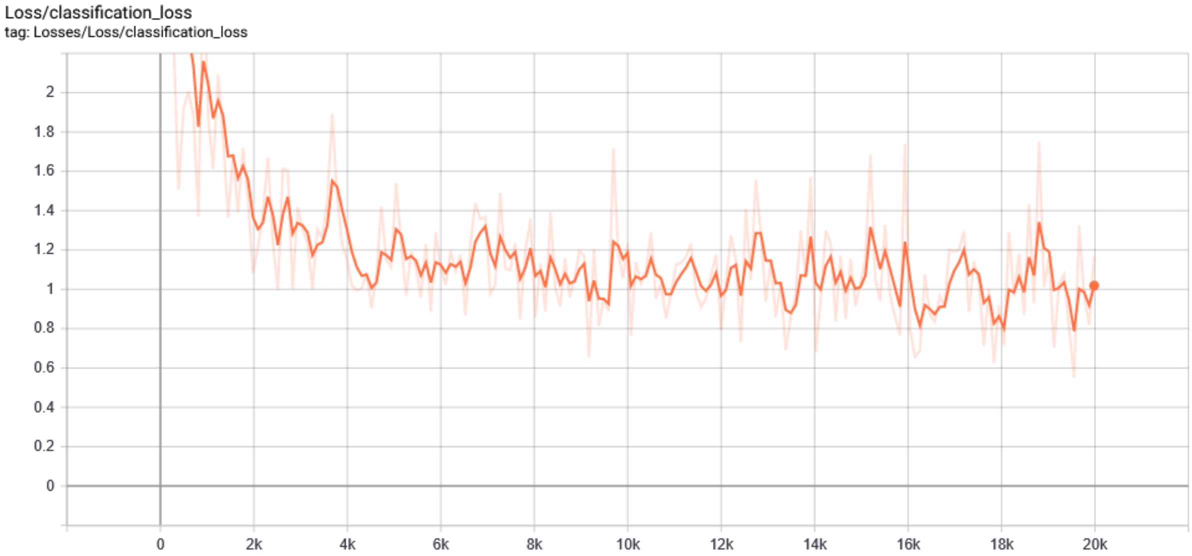

Figure 4.6: Classification Graph of SSD Mobilenet V1 COCO

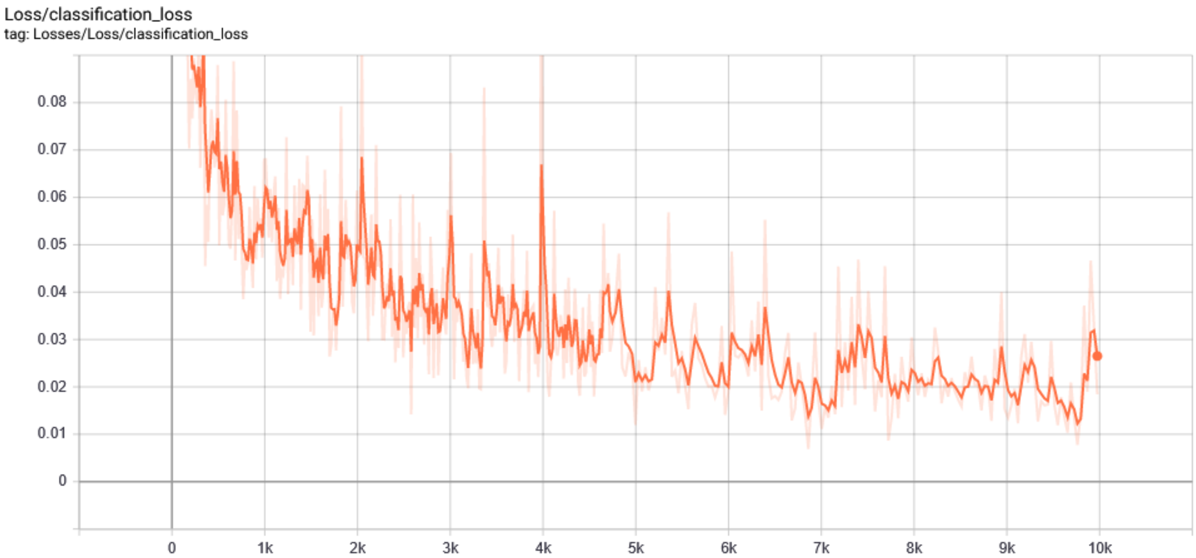

Figure 4.7: Classification Graph of SSD Mobilenet V1 FPN COCO

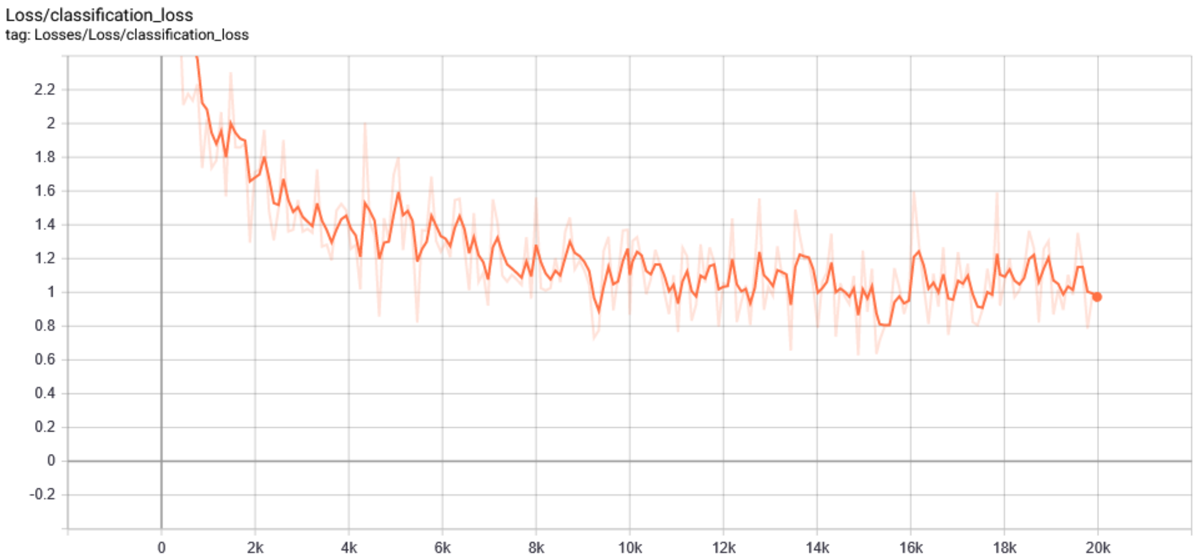

Figure 4.8: Classification Graph of SSDlite Mobilenet V2 COCO

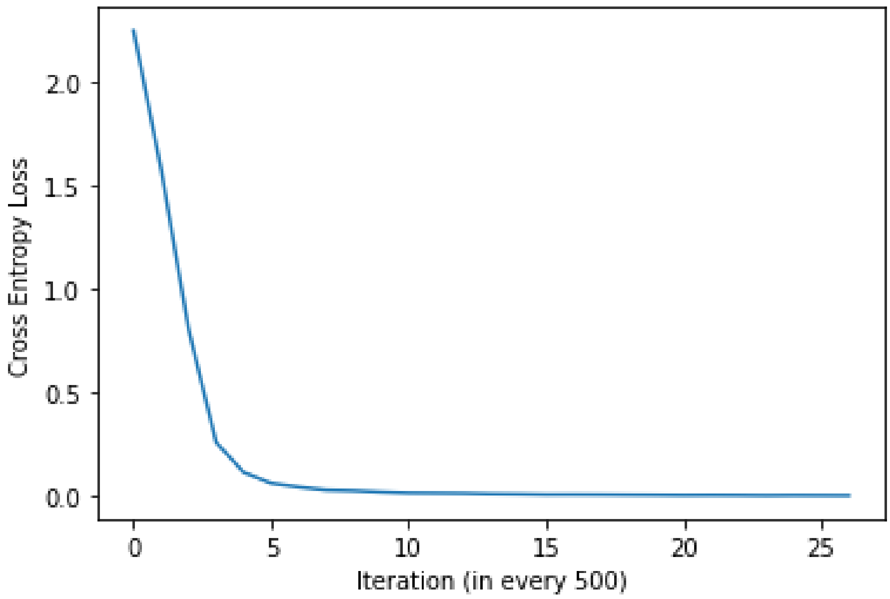

Figure 4.9: Loss Graph Of Recognition

In this work, mobile camera is used to take few videos at 5 to 10 meters separations from camera to vehicle with the distinctive background. These algorithms are applied on videos of Bangladeshi vehicles snapped in distinctive light conditions, such as night, sunny, and cloudy. The parameters that are utilized in the sifting operation to qualify ROI from the candidate list emerges as the experimental outcome of the system.

As we used video as our input, there were approximately 25 vehicles in that video.

Table 4.1: Experimental Result

Step	**No. of Instances**	**No. of success**	**Success Rate (%)**
Detection	25	23	92.6%
Segmentation	25	20	80.3%
Recognition	25	22	88.8%

In the existence of complex background and exceedingly fluctuating license plate designs in the input, the normal detection rate of our system is 92.6%, segmentation is 80.3% and recognition is 88.8%. It demonstrates that the outcome of our system is better or quite similar than that of the other algorithms in recognition Bangla license plate.

Chapter 5

Difficulties and Limitations

5.1 Difficulties We Have Faced

Some of the difficulties we faced while working are listed below:

- We captured the images and videos using one camera. At first we tried to capture image and videos in free road. But, due to the high speed, we were unable to capture license plate clearly. So we had to go to traffic signals, speed breakers where vehicles slow down.
- Some of the vehicles were missed because of other vehicles passing in front of them because we had only one camera.
- We could not find data set or collect enough images of buses, trucks etc.
- We tried to do segmentation without vertical and horizontal partition. The segmentation code we did at first segmented the license plate pretty well. But, the segmented fragments were unsorted. So, we could not use that code and had to do it again by adding vertical and horizontal segmentation and a sorting function.
- We used contour to draw bounding box. The system detects many contours which we do not need. So, we manually had to find a range which gives us the contours that is needed. It took time to find the correct range.

5.2 Limitations of the System

No system is infinitely precise. There must be some limitations to any systems. 100% precision of systems is not possible.

- The video frame rate should be 25fps or less.
- The resolution of the video should be 720 pixels. If it is more than that, the image may get clearer, but the processing takes a lot of time which is not feasible to use in real time.
- As we could not find data sets for bus, trucks, we could not train our system with those. So for now, our system can detect cars and bikes and their license plates.
- We took the videos in Dhaka. So no vehicle with other registration area was found. So we could not train our system with that.
- In Bangla language, there are two overlapping vowels. They are rarely used in the Bangla license plates. Our proposed segmentation algorithm may not able to segment these overlapping vowels as we did not train our system with them.
- Decision tree can be used in character recognition step to make prior decision about the registration area and type. This will decrease the execution time and improve the precision of the system.
- In order to apply our recommended system in real-time applications more effective and efficient way, algorithms can be instrumented in hard wire and parallel appliances, which need a lot of exploration in these fields.
- Some vehicle license plates are too old and blurry. So, it is difficult to recognize the license plates of those vehicles.
- The videos taken are in day light. So, the system cannot recognize in night mode.

In future work, we hope to improve our system to solve the above mentioned problem.

Chapter 6

Conclusion

6.1 Summary

In this work, an attempt has been made to detect and recognize the Bangladeshi license plate and how it can be made useful to people. As there is no public data set to our knowledge about the Bangla license plates, we created our own data set consisting of license plates for vehicles. As deep neural network models mostly depend on data, we believe that, a more diverse data set for training our model will produce a better result for our test data set. Also, by using a diverse data set, we will be able to classify more types of vehicles in the future without much alteration of any part in our model.

6.2 Suggestions for future work

This ANPR framework works very well however, there is still room for advancement. In spite of the fact that we had many things in mind that we wanted to implement, which we could not do due to a few challenges we have confronted. Below is some suggestions for future works:

- An automatic stolen vehicle detection system can be implemented. For stolen vehicle detection, the owner of the vehicle will need to inform the police about the occurrence and give the license plate number. If the vehicle with that license plate is detected anywhere, an alarm will ring and the person who monitors all the cameras will get to know it.

- An automatic toll collection feature can be implemented. The roads, bridges or flyovers where toll is needed, with the help of cameras the license plate will be detected and recognized. From the database, the owner of the vehicle will be searched and will get notified about the toll amount and the last date to pay the toll.

- Today progressive innovation took Automatic Number Plate Recognition (ANPR) systems from difficult to set up, limited costly, fixed based applications to simple portable ones in which "point to shoot" strategy can be used. This is possible because of the creation of software which ran on cheaper PC based and also a non specialist hardware in which there no need to deliver pre-defined direction, angels, speed and measure in which the plate would be passing the camera field of see.

- Small cameras which can read license plates at high speed, together with little, more durable processors can fit in police vehicles, allowed law enforcement personals to watch every day with the advantage of license plate recognition in real time.

- This ANPR framework speed can be increment with a high-resolution camera which can be able to capture clear pictures of the vehicle.

- The OCR strategy is sensitive to misalignment and to diverse sizes, so we have to make distinctive kinds of formats for different specifications.

- The factual investigation can also be utilized to characterize the likelihood of detection and recognition of the vehicle number plate.

- At present there are few limits on parameters such as the vehicle, text on the vehicle number plate, skew in the picture that can be expelled by improving the algorithms further.

- ANPR solutions that are accessible within the market do not offer a standardized set for all the nations. Each company has got to be given a well-optimized system for different parts/regions of the world since the same framework as created is not adequate and has to be planned according to the region where deployed, keeping all the influencing factors in considerations. OCR engines frequently are optimized for particular nations. It has to be made sure if the desired nations are supported within the library or engine that is installed on the camera. Each ANPR solutions system given by vendors has possessed qualities and shortcomings. The finest among these is the one that caters to the requirements of the region in a recognized framework affecting the conditions of that zone.

References

[1] K. Roy, A. M. S. Khan, M. Z. A. Ali, S. R. Simanto, N. Mohammed, M. A. Atick, S. Islam, and K. M. Islam, "An analytical approach for enhancing the automatic detection and recognition of skewed bangla license plates," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–4, IEEE, 2019.

[2] S. Abdullah, M. M. Hasan, and S. M. S. Islam, "Yolo-based three-stage network for bangla license plate recognition in dhaka metropolitan city," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–6, IEEE, 2018.

[3] N. Saif, N. Ahmmed, S. Pasha, M. S. K. Shahrin, M. M. Hasan, S. Islam, and A. S. M. M. Jameel, "Automatic license plate recognition system for bangla license plates using convolutional neural network," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 925–930, IEEE, 2019.

[4] S. Montazzolli and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pp. 55–62, IEEE, 2017.

[5] R. A. Baten, Z. Omair, and U. Sikder, "Bangla license plate reader for metropolitan cities of bangladesh using template matching," in *8th International Conference on Electrical and Computer Engineering*, pp. 776–779, IEEE, 2014.

[6] M. Z. Abedin, A. C. Nath, P. Dhar, K. Deb, and M. S. Hossain, "License plate recognition system based on contour properties and deep learning model," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 590–593, IEEE, 2017.

[7] M. T. Shahed, M. R. I. Udoy, B. Saha, A. I. Khan, and S. Subrina, "Automatic bengali number plate reader," in *TENCON 2017-2017 IEEE Region 10 Conference*, pp. 1364–1368, IEEE, 2017.

[8] K. W. Maglad, "A vehicle license plate detection and recognition system," *Journal of Computer Science*, vol. 8, no. 3, p. 310, 2012.

[9] K. Deb, M. K. Hossen, M. I. Khan, and M. R. Alam, "Bangladeshi vehicle license plate detection method based on hsi color model and geometrical properties," in *2012 7th International Forum on Strategic Technology (IFOST)*, pp. 1–5, IEEE, 2012.

[10] M. S. Mashuk, M. A. Majid, N. Basher, and T. R. Rahman, "Automatic detection of bangla characters in bangladeshi car registration plates," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, pp. 166–171, IEEE, 2010.

[11] A. C. Roy, M. K. Hossen, and D. Nag, "License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–6, IEEE, 2016.

[12] T. Islam and R. I. Rasel, "Real-time bangla license plate recognition system using faster r-cnn and ssd: A deep learning application," in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, pp. 108–111, IEEE.

[13] M. S. Rahman, M. Mostakim, M. S. Nasrin, and M. Z. Alom, "Bangla license plate recognition using convolutional neural networks (cnn)," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, 2019.

[14] M. Atikuzzaman, M. Asaduzzaman, and M. Z. Islam, "Vehicle number plate detection and categorization using cnns," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–5, IEEE, 2019.

[15] T. D. Duan, T. H. Du, T. V. Phuoc, and N. V. Hoang, "Building an automatic vehicle license plate recognition system," in *Proc. Int. Conf. Comput. Sci. RIVF*, vol. 1, pp. 59–63, Citeseer, 2005.

[16] C. Patel, D. Shah, and A. Patel, "Automatic number plate recognition system (anpr): A survey," *International Journal of Computer Applications*, vol. 69, no. 9, 2013.

[17] R. R. Palekar, S. U. Parab, D. P. Parikh, and V. N. Kamble, "Real time license plate detection using opencv and tesseract," in *2017 international conference on communication and signal processing (ICCSP)*, pp. 2111–2115, IEEE, 2017.

[18] Z.-K. Huang and L.-Y. Hou, "Chinese license plate detection based on deep neural network," in *2018 International Conference on Control and Robots (ICCR)*, pp. 84–88, IEEE, 2018.

Appendix A

Implemented Codes for Appendix

```
import numpy as np
import os
import tensorflow as tf
from utils import label_map_util
from utils import visualization_utils as vis_util
import natsort
import torch
import torch.nn as nn
import cv2
import glob
from datetime import datetime
from csv import writer

cap = cv2.VideoCapture('VID_20210408_132024.mp4')

# What model to download.
MODEL_NAME = 'new_graph'

# Path to frozen detection graph. This is the actual model that is used for the object detection.
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'

# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join('numberPlate_training', 'labelmap.pbtxt')

NUM_CLASSES = 3


# Deep Neural Network Model for segmented image recognition
class DeepNeuralNetworkModel(nn.Module):
    def __init__(self, input_size, num_classes):
        super().__init__()
```

```
        # 1st hidden layer
        self.linear_1 = nn.Linear(input_size, 1000)
        # Non-linearity in 1st hidden layer
        self.relu_1 = nn.LeakyReLU()

        # 2nd hidden layer
        self.linear_2 = nn.Linear(1000, 500)
        # Non-linearity in 2nd hidden layer
        self.relu_2 = nn.LeakyReLU()

        # 3rd hidden layer
        self.linear_3 = nn.Linear(500, 128)
        # Non-linearity in 2nd hidden layer
        self.sigmoid_3 = nn.Sigmoid()

        self.linear_out = nn.Linear(128, num_classes)

    def forward(self, x):
        # 1st hidden layer
        out = self.linear_1(x)
        # Non-linearity in 1st hidden layer
        out = self.relu_1(out)

        # 2nd hidden layer
        out = self.linear_2(out)
        # Non-linearity in 2nd hidden layer
        out = self.relu_2(out)

        # 3rd hidden layer
        out = self.linear_3(out)
        # Non-linearity in 3rd hidden layer
        out = self.sigmoid_3(out)

        probas = self.linear_out(out)

        return probas


# Defining Neural Network Model
segmentationModel = DeepNeuralNetworkModel(input_size=3000, num_classes=13)
# Loading trained segmentationModel
segmentationModel.load_state_dict(torch.load('new_graph/templateMatching.pkl', map_location="cpu"))


def append_list_as_row(file_name, list_of_elem):
    with open(file_name, 'a', newline='') as write_obj:
```

```
        csv_writer = writer(write_obj)
        csv_writer.writerow(list_of_elem)


# Read the model from the file
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.compat.v1.GraphDef()
    with tf.io.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
   max_num_classes=NUM_CLASSES,

   use_display_name=True)
category_index = label_map_util.create_category_index(categories)


def load_image_into_numpy_array(image):
    (im_width, im_height) = image.size
    return np.array(image.getdata()).reshape(
        (im_height, im_width, 3)).astype(np.uint8)


def pre_process(image):
    # pre-processing
    global crop
    img = cv2.resize(image, None, fx=2.5, fy=2.5, interpolation=cv2.INTER_CUBIC)
    temp_img = cv2.resize(image, dsize=(222, 118), interpolation=cv2.INTER_CUBIC)

    # Removing older preprocessed images
    files = glob.glob('temp\\preprocess\\*.png')
    for file in files:
        os.remove(file)

    # Sharpening
    kernel = np.array([[0, -1, 0],
                       [-1, 5, -1],
                       [0, -1, 0]])
    img = cv2.filter2D(img, -1, kernel)

    # Denoise
```

```
    img = cv2.fastNlMeansDenoising(img, None, 20, 15, 3)

    ad = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 55, 1)  # (55, 1)

    im = cv2.resize(ad, dsize=(222, 118), interpolation=cv2.INTER_CUBIC)
    # adding border
    row, col = im.shape
    im = cv2.rectangle(im, (0, 0), (col, row), (255, 255, 255), 6)

    # Morphological Closing
    kernel = np.ones((3, 3), np.uint8)
    im = cv2.morphologyEx(im, cv2.MORPH_CLOSE, kernel)

    # threshold image
    ret1, threshed_img = cv2.threshold(im, 127, 255, cv2.THRESH_BINARY)

    # find contours and get the external one
    contours, heir = cv2.findContours(threshed_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # with each contour, draw boundingRect
    for c in contours:
        if 10000 < cv2.contourArea(c) < 20000:
            # get the bounding rect
            x, y, w, h = cv2.boundingRect(c)
            # draw a green rectangle to visualize the bounding rect
            temp_img = cv2.rectangle(temp_img, (x, y), (x + w, y + h), (200, 200, 200), 20)
            crop = temp_img[y: y + h, x: x + w]
            cv2.imwrite('temp/preprocess/cropped.png', crop)
    return crop


def sort_contours(cnts, method="left-to-right"):
    # initialize the reverse flag and sort index
    reverse = False
    i = 0
    # handle if we need to sort in reverse
    if method == "right-to-left" or method == "bottom-to-top":
        reverse = True
    # handle if we are sorting against the y-coordinate rather than
    # the x-coordinate of the bounding box
    if method == "top-to-bottom" or method == "bottom-to-top":
        i = 1
    # construct the list of bounding boxes and sort them from top to
    # bottom
```

```
    boundingBoxes = [cv2.boundingRect(c) for c in cnts]
    (cnts, boundingBoxes) = zip(*sorted(zip(cnts, boundingBoxes), key=lambda b: b[1][i], reverse=reverse))
    # return the list of sorted contours and bounding boxes
    return cnts


def horizontal_seg(img):
    ad = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 55, 30)

    # Denoise
    im = cv2.fastNlMeansDenoising(ad, None, 20, 15, 3)
    im = cv2.GaussianBlur(im, (5, 5), 0)

    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, ksize=(17, 1))
    ad = cv2.erode(im, kernel)

    # threshold image
    ret, threshed_img = cv2.threshold(ad, 127, 255, cv2.THRESH_BINARY)

    # find contours and get the external one
    contours, heir = cv2.findContours(threshed_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # Sorting contours from top-to-bottom
    contours = sort_contours(contours, "top-to-bottom")
    count = 0
    # with each contour, draw boundingRect
    for c in contours:
        if 2000 < cv2.contourArea(c) < 5000:
            count += 1
            # get the bounding rect
            x, y, w, h = cv2.boundingRect(c)
            crop = img[y: y + h, x: x + w]
            cv2.imwrite('temp/preprocess/h_seg' + str(count) + '.png', crop)


def vertical_seg():
    # Removing older segmented images
    files = glob.glob('temp\\segmentation\\*.png')
    for file in files:
        os.remove(file)

    check = os.path.isfile("temp\\preprocess\\h_seg1.png")
    if check == True:
```

```
        # Segmentation for 1st image
        img = cv2.imread('temp\\preprocess\\h_seg1.png', 0)

        temp_img = cv2.resize(img, dsize=(180, 30), interpolation=cv2.INTER_CUBIC)

        ad = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 55, 20)  # (55, 1)

        # Erosion
        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, ksize=(5, 1))
        im = cv2.erode(ad, kernel)

        # Adding border
        row, col = im.shape
        im = cv2.rectangle(im, (0, 0), (col, row), (255, 255, 255), 6)

        im = cv2.resize(im, dsize=(180, 30), interpolation=cv2.INTER_CUBIC)

        # threshold image
        ret, threshed_img = cv2.threshold(im, 127, 255, cv2.THRESH_BINARY)

        # find contours and get the external one
        contours, hier = cv2.findContours(threshed_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        contours = sort_contours(contours)

        # with each contour, draw boundingRect in green
        count = 0
        for c in contours:
            if 200 < cv2.contourArea(c) < 1500:
                count += 1
                # get the bounding rect
                x, y, w, h = cv2.boundingRect(c)
                seg = temp_img[y: y + h, x: x + w]
                cv2.imwrite('temp/segmentation/seg' + str(count) + '.png', seg)

    check = os.path.isfile("temp\\preprocess\\h_seg2.png")
    if check == True:
        # Segmentation for 2nd image
        img = cv2.imread('temp\\preprocess\\h_seg2.png', 0)
        temp_img = cv2.resize(img, dsize=(180, 30), interpolation=cv2.INTER_CUBIC)

        ad = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 55, 25)  # (55, 1)
```

```

        ad = cv2.resize(ad, None, fx=2.5, fy=2.5, interpolation=cv2.INTER_CUBIC)

        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, ksize=(1, 1))
        im = cv2.erode(ad, kernel)

        # Adding border
        row, col = im.shape
        im = cv2.rectangle(im, (0, 0), (col, row), (255, 255, 255), 6)

        im = cv2.resize(im, dsize=(180, 30), interpolation=cv2.INTER_CUBIC)

        # threshold image
        ret, threshed_img = cv2.threshold(im, 127, 255, cv2.THRESH_BINARY)

        # find contours and get the external one
        contours, hier = cv2.findContours(threshed_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        contours = sort_contours(contours)

        # with each contour, draw boundingRect in green
        for c in contours:
            if 180 < cv2.contourArea(c) < 800:
                count += 1
                # get the bounding rect
                x, y, w, h = cv2.boundingRect(c)
                seg = temp_img[y: y + h, x: x + w]
                cv2.imwrite('temp/segmentation/seg' + str(count) + '.png', seg)


def recognition():
    # Loading segmented images
    images = glob.glob('temp\\segmentation\\*.png')
    # sorting the list by their name
    images = natsort.natsorted(images)
    images = [cv2.imread(im, 0) for im in images]
    images = [torch.Tensor(cv2.resize(img, dsize=(100, 30), interpolation=cv2.INTER_CUBIC)) for img in images]

    string = ''
    for img in images:
        predictions = segmentationModel.forward(img.view(-1, 100*30))
        prediction = torch.argmax(predictions, dim=1).numpy()
        if prediction[0] == 10:
            string = string + 'Dhaka'
```

```
        elif prediction[0] == 11:
            string = string + 'Metro'
        elif prediction[0] == 12:
            string = string + 'Ga-'
        else:
            string = string + str(prediction[0])
    # Current Date and Time
    now = datetime.now()
    now = now.strftime("%d-%b-%Y %H:%M:%S")
    append_list_as_row('temp/segmentation/numberPlateList.csv', [string,
   now])
    return string


def process_img2string(img, t):
    # Resize the image with interpolation
    cv2.imwrite("temp/number_plate_detected" + str(t) + ".png", img)
    # gray conversion
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Region of Interest selection
    img = pre_process(img)
    # Horizontally segmentation using image processing
    horizontal_seg(img)
    # Vertical Segmentation of those image
    vertical_seg()
    # Recognition of segmented  images
    string = recognition()
    return string


te = 0
with detection_graph.as_default():
    with tf.compat.v1.Session(graph=detection_graph) as sess:
        while True:
            ret, image_np = cap.read()
            # Expand dimensions since the model expects images to have
   shape: [1, None, None, 3]
            image_np_expanded = np.expand_dims(image_np, axis=0)
            image_tensor = detection_graph.get_tensor_by_name('image_tensor
   :0')
            # Each box represents a part of the image where a particular
   object was detected.
            boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
            # Each score represent how level of confidence for each of the
   objects.
            # Score is shown on the result image, together with the class
```

```
label.
        scores = detection_graph.get_tensor_by_name('detection_scores:0')
        classes = detection_graph.get_tensor_by_name('detection_classes:0')
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')
        # Actual detection.
        out = sess.run(
            [boxes, scores, classes, num_detections],
            feed_dict={image_tensor: image_np_expanded})

        rows = image_np.shape[0]
        cols = image_np.shape[1]

        # Visualization of the results of a detection.
        vis_util.visualize_boxes_and_labels_on_image_array(
            image_np,
            np.squeeze(out[0][0]),
            np.squeeze(out[2][0]).astype(np.int32),
            np.squeeze(out[1][0]),
            category_index,
            use_normalized_coordinates=True,
            line_thickness=4)
        num_detections = int(out[3][0])

        for i in range(num_detections):
            classId = int(out[2][0][i])
            score = float(out[1][0][i])
            bbox = [float(v) for v in out[0][0][i]]

            if score > 0.9 and classId == 2:
                # Creating a box around the detected number plate
                x = int(bbox[1] * cols)
                y = int(bbox[0] * rows)
                right = int(bbox[3] * cols)
                bottom = int(bbox[2] * rows)
                # Extract the detected number plate
                tmp = image_np[y: bottom, x: right]
                te = te + 1
                text = process_img2string(tmp, te)
                text_height = .8
                cv2.rectangle(image_np, (x, y), (right, bottom), (125, 255, 51), thickness=2)
                cv2.putText(image_np, text, (x, bottom + 19),
                            cv2.FONT_HERSHEY_SIMPLEX, text_height, (125, 255, 51), 2)
```

```
                    cv2.imwrite('temp/full_image.png', image_np)

            # print(category_index[2]['name'])
            cv2.imshow('object detection', cv2.resize(image_np, (800, 600)))
            if cv2.waitKey(25) & 0xFF == ord('q'):
                cv2.destroyAllWindows()
                break
```

Listing A.1: **Number Plate Detection, Segmentation and Recognition code**