Real Time Bangladeshi License Plate Detection 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 thesispresented 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 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 stuffs.

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) created a common standard for vehicle license plate. The size and aspect ratio of all license plates are same.

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. There is a wide variation among the license plate patterns. It is also difficult in segmenting Bangla characters. The successes of previous works, for the most part have been restricted to correct detection and recognition of license plates whose pictures are taken from the front or the rear of vehicles with slight angular variations. As a result, most Bangla automatic license plate recognition (ALPR) systems in apply struggle when the license plates are skewed and not practical in real time detection. In this paper, we tried to address the issues. We note that our projected methodology can simply be generalized and applied to non-Bangla license plates as well.

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 attached to a motor vehicle for official recognition purposes. Automatic Number Plate Recognition (ANPR) is an automatic system which capable of reading vehicle license plates without human interference through the use of high speed image, detection of characters within the given images, verification of the character patterns as being those from a vehicle license plate, character recognition 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, road signs, street lights, highway overpasses or buildings etc. Camera software is made capable of recognizing the patterns that make the license plates and deciphering the letters and numbers into a digital configuration. Then, the information is compared in real-time to make a list of plate numbers that belong to a set of vehicles of interest. If the system detects a match, an alarm rings or an alert message is sent to the dispatcher or other assigned 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. Multiple cameras are mounted on the vehicle. As the vehicle moves, it takes the pictures of the license plates from other vehicles around 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.

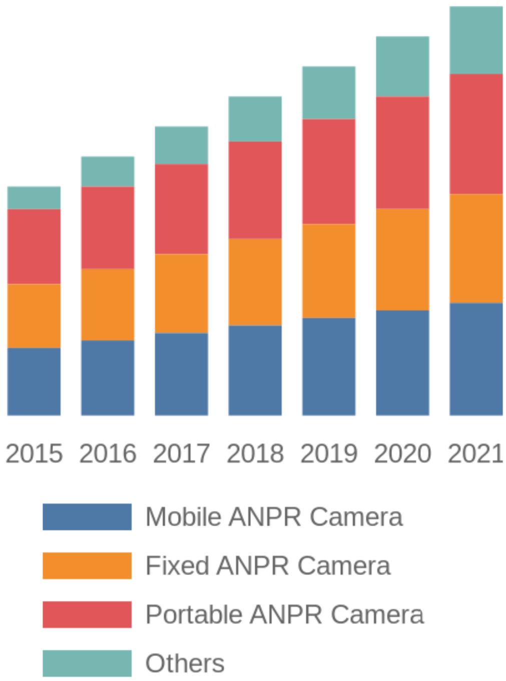

Figure 1.3: Use of Different ANPR Camera Over Years

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 accurate. 3. Coverage is extended. 4. If modern technology is used then it can be cost efficient. 5. Very useful on arterial routes and motorways.	1. Limited to a particular location. 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 needed. 2. Can be deployed at specific locations. 3. Can be moved without cost to follow a hot spot or a problem in an area 4. Response real time 5. Provides officers the ability to identify a vehicle which has matched against a hotlist which they may not otherwise have stopped 6. Can receive hits from the fixed ANPR cameras and respond in real time	1. Vulnerable to damage 2. Less reliable at scanning license plates. 3. Connected to cars which can break down, need servicing etc 4. Dependent on officers on the ground to set up and keep an eye on the system.

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 linked to CCTV (Local Authority, private use)	1. Can be used 24/7. 2. Camera coverage is extended with limited extra costs. 3. Can track vehicles. 4. Can be used in car parks, shopping centres, petrol stations etc. 5. Improved chance of getting picture of the driver.	1. City environment, less particular to travelling criminality. 2. In case no suitable agreements in place, disagreement of interest between local authority and policing objectives. 3. Ineffective positioning of cameras (too high, focused on pedestrian area etc). 4. Need of suitable legislation.

1.2 Technological Difficulties

In spite of the fact that, the thoughts behind ANPR are basic, putting them into practise is more challenging. It would be wrong to say that all ANPR systems can be utilized within the same way or can create the same results. A few systems work with lower specification cameras creating lower quality pictures and less exact reads. More up to date ANPR

cameras have improved specifications, with infrared capabilities to enable reading number plates and taking photos in poor light conditions or at night.It is accepted that the in general read rates for ANPR is 90% to 94%, in perfect conditions and supported by high-quality advanced systems. The older ANPR systems are notably untrustworthy (with performance rates between 60% and 80%), being acutely criticized for misreading number plates and creating 'hits' on guiltless drivers. In spite of the fact that ANPR technology has developed impressively over the last few years, concerns about the accuracy and trustworthiness of ANPR systems remain.
There are a numerous possible difficulties that the system must be able to cope with. Some of which are:

1. The image or video of poor resolution because the plate is too far away or from the use of a low-quality camera.
2. Blurry images due to motion blur.
3. Poor lighting, low contrast, reflection or shadows.
4. An object obscuring the plate or a portion of the plate like dirt, tow bar etc.
5. Using different fonts, popular for vanity plates. Some countries do not allow such plates which eliminates the problem.
6. Lack of coordination between countries. Cars from different countries can have the same number but different design of the plate etc.

Some of the above mentioned problems can be solved within the software, but it is primarily left to the hardware side of the system to work out solutions to these problems. Increasing the height of the camera may avoid problems with objects such as: other vehicles obscuring the plate but it introduces other problems like adjusting the increased skew of the plate. Few small-scale systems allow few errors in the license plate.

1.3 Does ANPR bring us near to a total surveillance society?

Some people stress that for CCTV surveillance to be 'total', it has to have the power to spot (anybody, anywhere, anytime) and classify individuals. People need to be cautious of being watched and there has to be a surety of response from authorities to acts of non-conformity.
Being digital and automated, ANPR systems have an increased ability to reserve and

examine information, an increased ability to detect suspects without having to watch for behavioural patterns.

In most cases, the population subject to CCTV observation (open road systems) is unknown to the observers, so they cannot efficiently distinguish and classify individuals in open space. Most CCTV frameworks cannot however routinely connect a person's picture to a database.

ANPR cameras are connected to databases comprising data on the complete enrolled driving population. In any case, stand-alone ANPR frameworks have their drawbacks over CCTV. Firstly, they are restricted to the driving population; they cannot 'watch' everything and everyone. Not only is ANPR restricted to vehicles, but the data on the known driver of the vehicle of interest can be wrong, as the driver may not be the enrolled attendant of the vehicle or the databases utilized in association with ANPR may not be up to date at the time when the framework cross-checks the data regarding the vehicle registration number. ANPR frameworks can match, examine and spread individual information at high speeds. These are a few of the reasons why ANPR might bring us closer to a 'maximum surveillance society. One of the most drawbacks of both CCTV and ANPR frameworks is the degree to which these surveillance frameworks produce a definitive response to non-conformity. Most of the CCTV frameworks are not observed on a regular basis and CCTV staff have other responsibilities or do not observe all the cameras all the time.

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
Reduced number of operators; reduced 'operator' bias (automatic detection of suspects)	High number of operators; highly skilled; danger of 'operator' bias (operator decides 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 severe consequences, including failure. In real-time advanced image processing, the normal processing time per sample must be less than the examining period, which is the corresponding sampling rate. A system with a real-time limitation can not be considered fruitful in case it produces the right action or the right answer after a certain deadline. In real time systems time or temporal correctness is as necessary as logical correctness of a program. The rightness of a real-time system is based on the rightness of the output and timeliness.
Real time systems are categorized in 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 deadline must be met with correct response or the system will lead to a total failure. An assurance of always meeting the hard deadline is needed. Examples include air traffic control, vehicle subsystems control, nuclear power plant control etc.

1.4.2 Soft Real Time System

The soft real time systems perform task almost in the defined deadline. They do not assure a hard deadline. Task can be performed even after the time has passed. Examples include multimedia transmission and reception, networking, cellular networks, web sites, services 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

Several State Police Forces and the Department of Justice use both fixed and mobile ANPR systems. In 2005, New South Wales Police Force Highway Patrol were the first to use fixed ANPR camera system. They began to use a mobile ANPR system with three infrared cameras fitted to Highway Patrol fleet in 2009. The system identifies unregistered, stolen vehicles, disqualified or suspended drivers.

United States

According to a 2012 report, nearly 71% of all US police departments use some form of ANPR system. Mobile ANPR is becoming a remarkable component of municipal predictive policing strategies and intelligence gathering. It is also used for recovery of stolen vehicles, identification of wanted criminals and revenue collection from individuals who are troublesome on city or state taxes or fines, or monitoring for ”Amber Alerts”.

Canada

Federal, provincial, and municipal police services across Canada use ANPR software. It is also used on toll routes and parking enforcement agencies.

Belgium

The city of Mechelen uses an ANPR system since September 2011. It is used to scan all cars crossing the city. Cars listed on ’black lists’ generate an alarm in the dispatching room, so they can be intercepted by a patrol.

Sweden

ANPR is tested by the Swedish Police at nine different locations in Sweden.

United Kingdom

Vehicle movements are recorded by nearly 8000 cameras capturing between 25 and 30 million ANPR ‘read’ records everyday. These records are stored for up to two years. It can be accessed, analysed and used as evidence by UK law enforcement agencies.

1.5.2 For Electronic toll collection

Ontario

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

Portugal

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

South Africa

In South Africa, ANPR is used for the etoll fee collection. Car owners driving into or out of the city must pay a charge. The number of tolls passed relies upon the distance travelled on the specific freeway.

Sweden

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

1.5.3 For Vehicle Speed Control

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

1.6 Study of Bangladeshi Vehicle License Plate

Bangladesh Road Transport Authority which is known as BRTA, issues vehicle license plates for motor vehicles in Bangladesh. The license plates in Bangladesh use the Bengali alphabet and numerals. The general format of license plates in Bangladesh is ”city - vehicle class letter and number - vehicle number”. An example is given below.

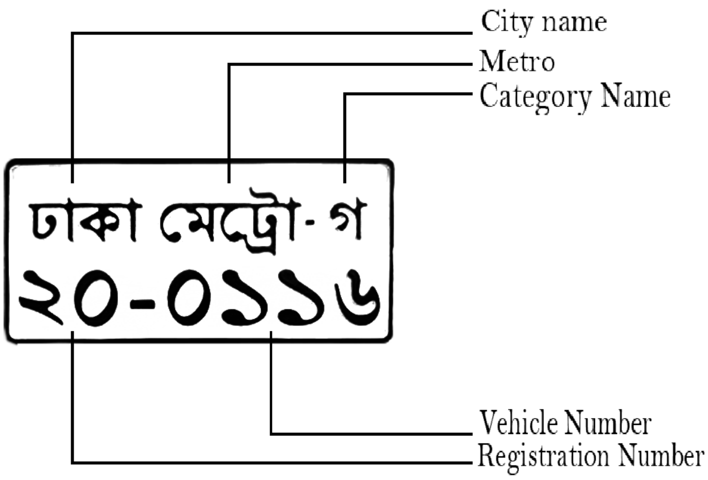

Figure 1.4: A sample of BRTA Standard License Plate

The license plates are placed in both the front and back of the vehicle. The letters and numerals permitted 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 private 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

Continued on next page

Table 1.3 – continued from previous page

Letters	Identities of different vehicles
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
ভ	Motor car (extra-large, 2001 cc and above)
ম	Light private 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 seen that the public and private license plates have the same format and they have two rows. Government and military license plates have only one row. The background color of government owned vehicles are yellow and the foreground is black. It has one Bangla character indicating registration type which is followed by a five digit registration number. License plates of military vehicles have an arrow sign or the abbreviation of the name of the military force at the beginning. Then, it is 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 offices of BRTA that are functioning around the country. These offices are listed in Fig. 1.6.

ঢাকা মেট্রো (উত্তর) Dhaka Metro (North)	ঢাকা মেট্রো (দক্ষিন) Dhaka Metro (South)
ফরিদপুর (Faridpur)	ময়মনসিংহ (Mymensingh)
গাজীপুর (Gazipur)	নারায়নগঞ্জ (Narayanganj)
টাংগাইল (Tangail)	মানিকগঞ্জ (Manikganj)
চট্টো মেট্রো (Chittagong)	ফেনী (Feni)
রাঙামাটি (Rangamati)	নোয়াখালী (Noakhali)
কুমিল্লা (Comilla)	কক্সবাজার (Cox's Bazar)
বি- বাড়িয়া (Brahmanbaria)	রাজশাহী (Rajshahi)
নওগা (Naogaon)	বগুড়া (Bogra)
রংপুর (Rangpur)	দিনাজপুর (Dinajpur)
পাবনা (Pabna)	সিরাজগঞ্জ (Sirajganj)
খুলনা (Khulna)	ঝিনাইদাহ (Jhenaidah)
কুষ্টিয়া (Kushtia)	যশোর (Jessore)
মৌলভীবাজার (Maulvi Bazar)	বরিশাল (Barisal)
সিলেট (Sylhet)	

Figure 1.6: Vehicle Registration Area

License plates are exceptionally distinctive from one nation to another nation. Because of this reason, it is not compelling to use an ANPR framework created in one country to another country without particular adjustments.

There are a few interesting characteristics of Bangladeshi license plates which are composed in Bangla language. Most of the languages in the world have disjoint characters in a word but, the characters in a Bangla word are often joined at the top by a horizontal line. That line is called Matra. Moreover, a few Bangla characters are comprised of two or more isolated regions. There can be a few disjoint parts in a character either at the top or at the bottom.

These extraordinary properties of Bangla characters make it exceptionally difficult to fragment. Unlike the other nations, a lot of varieties can be seen among the license plate designs in Bangladesh which are shown in Fig.1.5.

A huge number of Bangladeshi license plates have two rows where the first row carries the registration area and its type and the second row carries the registration number. There can be an additional row at the top or at the bottom of the license plate containing a few additional information. This additional information also makes it more 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 like vehicle theft, traffic management, parking management, identifying stolen vehicle etc. 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.

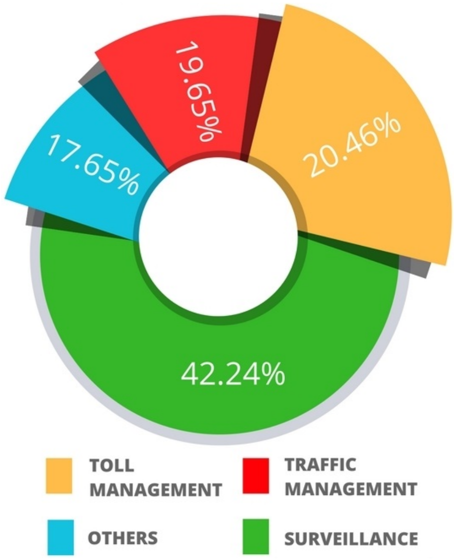

Figure 1.7: Global ANPR Market by Applications in 2016

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 1.8: 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 travelers 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.

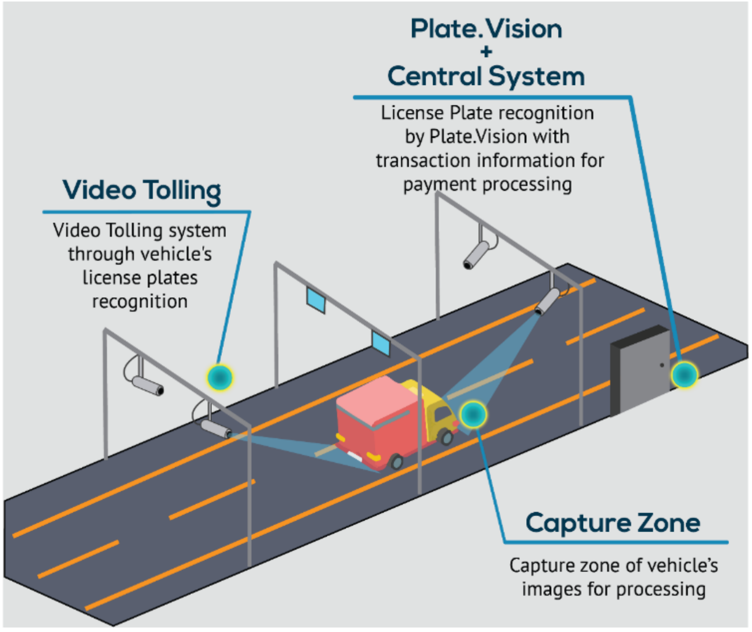

Figure 1.9: ANPR for automatic toll collection

1.7.3 Importance in Traffic Management

In Bangladesh, the traffic condition is disorganized. As traffic jam causes a huge loss financially, it is important to improve the traffic management system. ANPR can help to ensure mobility. It can measure vehicle speed. Travelling above the speed limit dramatically reduces a driver's chance of stopping if something unexpected happens. It can accurately determine the average speed of vehicles travelling on a section of road by identifying their license plate at both ends of a journey. When vehicles pass through different road signs, the license plates are displayed with a safety message.

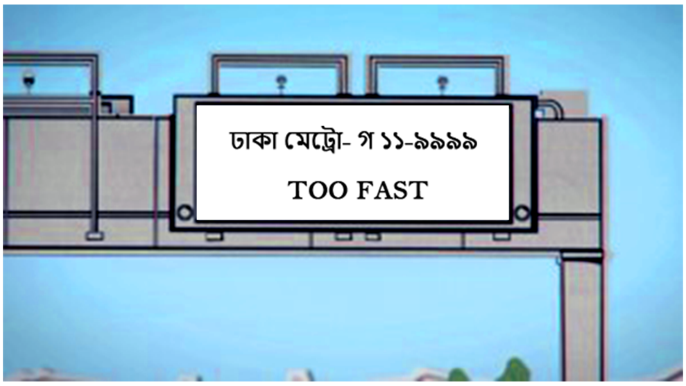

Figure 1.10: Safety message with license plate if a vehicle is over speed

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. So by using ANPR, it can be easily checked in database if the vehicle is registered or not.

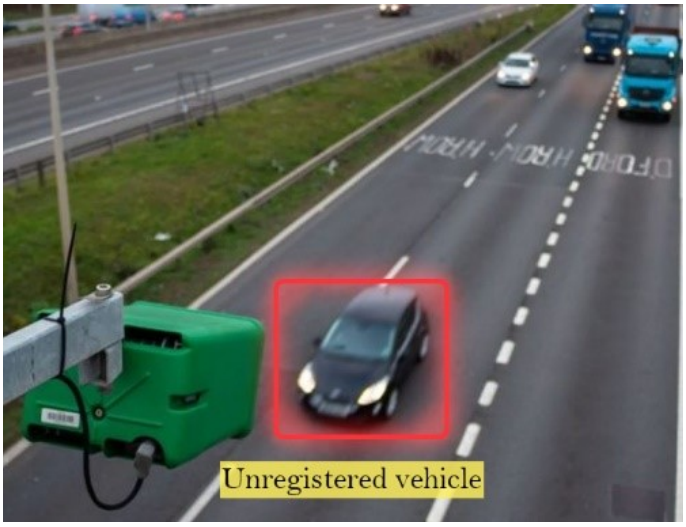

Figure 1.11: Detecting unregistered vehicle

Sometimes it becomes very difficult to identify for the traffic police to identify vehicle owner who has violated the traffic rules or drive too fast. Therefore, it becomes impossible to catch and punish them because the traffic police might not be able to retrieve license number from the moving vehicle because of the speed. Therefore, Automatic Number Plate Recognition (ANPR) system can be implemented as one of the solutions to this kind of problems.

1.7.4 Importance in Parking Management

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

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 ensures 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.

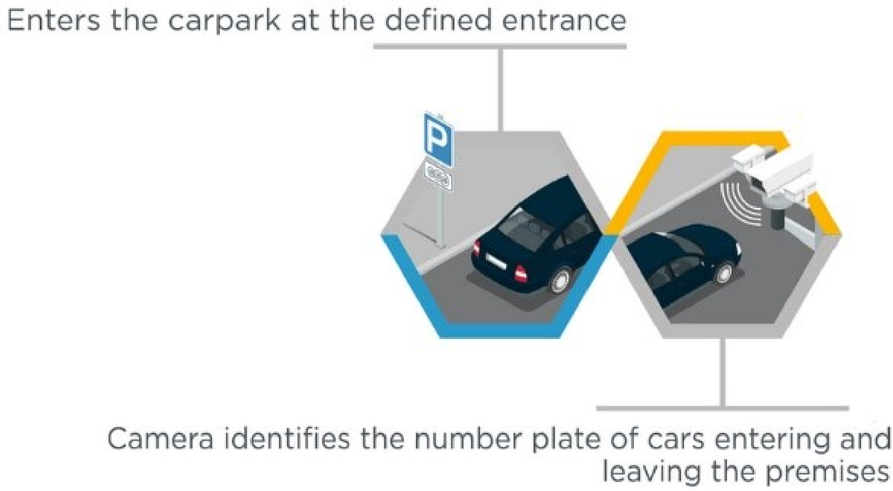

Figure 1.12: ANPR in parking management

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. This can bring in extra money for the company.

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 Provides Evidence

ANPR systems can provide the details regarding when someone 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 congestion by charging for vehicle access to the central business district at peak hours. Traffic congestion not only causes financial losses, however additionally has a destructive impact on road safety. ANPR can assist to enhance movability and traffic 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 applications of license plate recognition, different techniques have been developed for these three stages. In this chapter, we present a survey of the techniques that are used in three stages ANPR in different prior works.

2.2 Methods Used in Related Works

1. **Mask R-CNN:**
   Mask R-CNN is a deep neural network focused to solve instance segmentation problem in machine learning or computer vision. In other words, it can split up different objects in a image or a video. If an image is given, it gives the object bounding boxes, classes and masks.

   In [1], they used it for Image segmentation. Their proposed method of pre-processing license plate images has two steps: performing instance segmentation and transforming the segmented license plates into uniform rectangular views. They trained the model for 100 epochs with the batch size set equal to 100. For the rest of the hyper parameters, used the same values.

2. **Harris Method and Shi-Tomasi Method:**
   The idea of the Harris method is to detect points based on the intensity variation in a local neighborhood: a small region around the properties should show a large intensity change when compared with windows shifted in any direction.
   J. Shi and C. Tomasi made a small modification to it which shows better results

compared to Harris Corner Detector.

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

3. **YOLOv3:**
The full form of YOLOv3 is You Only Look Once version 3. It is a real-time object detection algorithm that recognizes specific objects in videos, live feeds, or images. YOLO is a Convolutional Neural Network (CNN) which performs object detection. CNN's are primarily classifier-based systems which is used to process input images as structured arrays of data and recognizes patterns between them. YOLO is much faster than other networks and still maintains accuracy. Another benefit of using it is that it can generalize very well. So, it performs extraordinarily well on data that it has never been trained. It also can filter background noises from the actual data very effectively. Another advantage of using YOLO is that it can detect and localize objects in an image simultaneously using only one convolutional network which remarkably decrease segmentation and recognition time as well.

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 segmented license plates.

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

4. **Bounding Box:**
A bounding box is an imaginary rectangle that serves as a point of reference for object location and makes a collision box for that object. Information annotators draw these rectangles over pictures, sketching out the object of interest inside each picture by characterizing its X and Y coordinates. This makes it simpler for machine learning algorithms to discover what they're searching for, decide collision ways, and conserves valuable computing assets. Bounding boxes are one of the most well-known image annotation strategies in deep learning. Compared to other picture preparing strategies, this strategy can diminish costs and increment annotation effectiveness. To localize an object in an image, the computer needs to know what it is and where it is. An annotator will draw bounding boxes around other vehicles and label them. This helps arrange a calculation to recognize what vehicles look like.

For segmentation, bonding box on filtered binary image is used in [5]. For recognition of segmented characters and words ,they established separate and extendable databases of binary images of the Bangla numbers 0-9; the words “Dhaka”, “Chotto” as samples of City Name; the word “Metro”; and the letters “Ka”, “Kha” and “Ga” as samples of Category Name.

In [6], the contours of characters are detected, Finally, using the bounding box parameters the characters are extracted for the recognition process.

Bounding boxes are formed around each of the connected components in the image and the boxes are colored green in [7].

5. **Median Filtering:**

   Median filtering is a a non-linear advanced filtering method, frequently utilized to expel noise from an image. Such noise lessening may be an ordinary pre-processing step to improve the results of later processing (for example, edge detection on a picture). Median filtering is very broadly utilized in advanced image processing because, under certain conditions, it preserves edges while removing noise. The rule of the median filter is to replace the gray level of each pixel with the median of the gray levels in a neighborhood of the pixels, rather than utilizing the average operation. For median filtering, we indicate the kernel size, list the pixel values secured by the kernel, and determine the median level.

   In [5], the proposed system has three basic steps which are: noise elimination and conversion to binary image, segmentation of words and characters, recognition of the segmented words and characters. The image acted upon with median filtering to reduce noises.

   2D median filtering is done in [7], [8] on the image for the removal of noise.

6. **Morphological Operations:**

   Morphological Operations are a wide set of image processing operations that prepare digital pictures based on their shapes. In a morphological operation, each picture pixel is comparing to the value of another pixel in its neighborhood. By choosing the shape and size of the neighborhood pixel, you can build a morphological operation that’s delicate to particular shapes in the input picture.

   Morphological operation like subtraction of the eroded and diluted versions detect the edges of words and single characters in [5].

   In [7], morphological operations are performed on the picture. It tests a picture with a little shape or template known as structuring component. The structuring component is situated at all possible areas within the picture and it is compared with the corresponding neighboring pixels. A few morphological operations test whether the structuring component 'fits' inside the neighborhood, whereas others, test whether it 'hits' or 'intersects' the neighborhood.

After the segmentation in [9], there may still exist noises within the picture which is not perfect. These noises can be of numerous sorts, such as little gaps or/and bulges of the target candidate regions. These noises can be evacuated utilizing mathematical morphology operation.

7. **Morphological Erosion:**
   Erosion is one of the two fundamental operations in morphological image processing from which all other morphological operations are based. It is originally defined for binary images, later being extended to gray scale images. Erosion decreases the size of the objects, removes the small irregularities, reduces the brightness of the bright objects and is used prior in Opening operation.

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

8. **Morphological Dilation:**
   Dilation is another fundamental operation in morphological image processing. 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].

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.

10. **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.

11. **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.

12. **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.

13. **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].

14. **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].

15. **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].

16. **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.

17. **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.

18. **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.

19. **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].

20. **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.

21. **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].

22. **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].

23. **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.

24. **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]

25. **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.

26. **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.

27. **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.

28. **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 ("hidden") 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 Image 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 network with this dataset to recognize the single 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 • MATLAB functions are used for character recognition	90% accuracy for close range

[14]	Did not need dataset	• HAAR Feature based Classifier to detect license plate • Class letter extractor with a proposed method • Convolution Neural Network for recognizing class letters	• 96.92% accuracy for number plate 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 projections with threshold to segment Bangla characters and digits efficiently	• Average detection rate of their algorithm is 93% • Segmentation rate 98.1% • Recognition rate 88.8%
[9]	Did not need dataset	• HSI color model • Geometrical properties • Candidate Regions • Intensity Histogram	The rate of success of the license plate detection algorithm is 85%.
[17]	Used own dataset	Tesseract OCR machine on processed image for License plate detection	The final output of the license plates was found to be very close to the actual 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% • Feature extraction
[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 ALPR system for Saudi Arabian license plate

In [8], the research design is divided into four main steps which are: Image Acquisition, Image Pre-processing, LP Detection, Character Segmentation and Character Recognition. In preprocessing part, the main goal is to prepare images for license plate detection using the proposed RBF (Radial Basis Function) network. It follows the given steps:

1. converts the gray-scale image into binary image.
2. performs edge detection using Sobel's mask operator.
3. performs morphological operation using dilation.
4. fills the interior gaps in order to obtain a closed shape using 'flood fill' algorithm.
5. applies the filtering task.
6. performs noise removal operation.
7. the contour of the selected area is used to map it on the original gray-scale image in order to get an outline across this area.

License plate detection part starts with the pre-processed image mentioned in the above section. A threshold value is fixed in the form of minimum and maximum values in order to obtain the license plate only and remove other very small or very large objects which were outside the threshold range. s. During training phase, the identified objects were classified as the "Plate" (denoted as "1") or "No Plate", (denoted as "0") manually based on the shape of the identified white area. Two RBF NNs are used for learning of the width side and the length side of the image. Before testing any new image using RBF NN, the image has to go through the same image pre-processing steps mentioned in previous section and the object detection process using threshold values.

The next step is license plate extraction. The license plate is extracted from the image by specifying its top left and bottom right corners. Then, the border across the license plates are removed so that the characters written on it can be recognized in the next step. The x and y coordinates of top left and bottom right corners of the LP are used to crop the license plate from whole input image.

The next part is character segmentation. Character segmentation is the process of extracting the characters and numbers from the license plate image. After removing the plate borders in the preceding step, this step starts with removing the noise from the plate. The process used here for character segmentation is based on thresholding and Connected Component Analysis (CCA). In binary image processing, CCA is a necessary process that examines and labels the pixels of a binary image into components based on pixel connectivity. On the extracted LP from the preceding step, following steps are performed:

1. Otsu's thresholding method is performed to produce binary image.
2. Median filter to remove noises.
3. Any pixels that connected with the border are removed using morphological reconstruction.
4. Connected Component Analysis (CCA) is done to label the component.
5. An algorithm is defined to choose whether the component belongs to number, strip or Arabic word.

The final step is character recognition. In this research, it is done by feature extraction. An image is partitioned into a sequence of horizontal "scan lines" using the raster scanning. Each scan line consists of connected pixels and a number emerges in a 4×2 size matrix. This study also has two features called ratio of size and ratio of foreground and background pixels. These features help to determine non-number image and Arabic word because strip and Arabic word has difference on those two features compared to the numbers. This study has total 10 number of features, which are 8 features of pixels existence and 2 features of ratio.
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 ALPR system for Vietnamese license plate

In [15], There are four steps which are: Pre-processing, License plate detection, Character segmentation, and Optical Character Recognition (OCR).

In preprocessing stage, the edge features are enriched. The algorithms used in this step are graying, normalizing and histogram equalization. After making the image grayscale, Sobel filters is used to extract the edging image. Then thresholds the image to a binary image. Local adaptive thresholding algorithm is used for binarizing.

Next step is license plate detection. For this, the combination of the Hough Transform and Contour algorithm is used which produces higher accuracy and faster speed so it can be used in real time systems. From the extracted edging image, the contour algorithm is applied to detect closed boundaries of the objects. These contour lines are then transformed to Hough coordinate. As there are few (black) pixels in the contour lines, the transformation of these points to Hough coordinate needs much less calculation. So, the speed of the algorithm is upgraded significantly without any accuracy loss.

Evaluating plate-candidates algorithm bases on two major steps. The two steps are: evaluating the ratios between the heights and the widths of the candidates and using horizontal intercepts to count the number of cut-objects in the candidates. In this work,implemented two horizontal cuts at 1/3 and 2/3 of plate-candidate's height are implemented.

The next step is segmentation. The character images are passed to the OCR module for recognizing. They used a horizontal projection to detect and segment rows in 2 row plates. They searched for the minimum values in vertical projection. The minimum positions which provide cut pieces satisfied all predefined constraints are considered as the points for character segmentation. The HMM (Hidden Markov Model) model is used for character recognition.

They took images in two camera positions which are:

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

2. Random locations where rotated angles: 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 ALPR system for Chinese license plate

[18] is a work on Chinese license plates. Compared with the license plates of other countries, the types of Chinese license plates are more diverse. The current plates are of the 1992 standard. It consists of the one-character provincial abbreviation, a letter of the alphabet and five numbers or letters of the alphabet. Previously, all license plates had used the five-number designation.

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

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

They used a database with 190 photos containing license plate images. These pictures have different backgrounds, angles and light conditions. Every image contains one to two detection targets. The system is trained in four steps. The first two steps train the region proposal and detection networks which are used in Faster R-CNN. The final two steps merges the networks from the first two steps in a way that a single network is generated for detection. The learning rate for the initial two steps is set higher than the last two steps as the last two steps are fine-tuning steps.

The first step is to train a RPN used in Faster R-CNN. Then, they trained a fast R-CNN network using the RPN from step 1. To unite the two networks, they used interchangeable training and re-training RPN using weight sharing with Fast R-CNN.

To completely evaluate the system, they used average precision to calculate the capability to make correct classifications and the capability to find all relevant objects.

Figure 2.3: Chinese License Plate

2.3.4 ALPR system for Brazilian license plate

Brazilian license plate is specified by 3-letter and 4-digit sequence above which are the state code and municipality. The license plate design is being slowly being eliminated and is being replaced by the new Brazilian Mercosur design. Although there is no termination date determined for the old ones.

In [4], They are using DL methods known to be computationally expensive, avoiding small objects becomes even more important when one looks for a real time application. They used a YOLO based network as these are said to be the fastest networks even faster than Faster RCNN that does object detection and recognition applying region proposals without needing image pyramid and sliding windows. In tests, the FAST-YOLO network was able to perform a 20-class detection and classification in 800×600 images in less than 5.5ms1 or around 180 frames per second(FPS).

For the detection and recognition of characters, they built a new network inspired on the YOLO architecture, with basic practical differences to accommodate 35 classes (0-9, A-Z except the letter O, which is detected jointly with the number 0) and outputs a feature map having the same aspect ratio of license plate where height is one-third times smaller than width.
The vehicle front view and its license plate are detected using a single classifier organized in a cascaded manner. The first layer detects the front view from the input image, and the second layer extracts the license plate from the detected front view image. To attain a better settlement between accuracy and running times, the classifier is based on the FAST-YOLO architecture. This network was built and trained to handle 20 different classes of objects and runs at 200 FPS. We hypothesized that the FAST-YOLO network

customized for 2 classes could have the ability to help both tasks in a single network when executed in a cascaded way.

In license plate detection step, first pass involves the whole image and looks only for front views. Any license plate found is rejected. The detected front views are then cropped and delivered to the same network and only the output related to license plate is used. If numerous license plates are found, only the one with the maximum probability is kept.

For character detection and recognition, first the number of max pooling layers needed to bring down from five to three in order to keep the fine output coherency by avoiding many dimensionality reductions. Second, to maintain the network depth equivalent to FAST-YOLO and yet being allowed to use as much transfer learning as possible. Then the first eleven layers of YOLO network is used, stopping on the twelfth layer, since it contains the fourth max pooling in that network. At last, four more layers were attached and trained from scratch for the betterment of non-linearity.

Brazilian license plate is formed by three letters followed by four numbers. Two heuristic rules to filter the results are used. The rules are:

1. If more than seven characters are identified, only the seven most probable are kept.
2. the first three characters are supposed to be letters and the following four digits.

If a letter is detected in the license plates related to digits, it is switched by the digit that presented the largest occurrence in the confusion matrix acquired with the training data. The same process is used when a digit is detected in the license plates related to letters.

The system successfully detected and recognized all seven characters of a license plate in 63.18% of the test set and 97.39% when considering at least five correct characters. Taking the segmentation and recognition of each character individually, the system was able to segment 99% of the characters, and correctly recognize 93% of them.

Figure 2.4: Brazilian License Plate

Chapter 3

Methodology

3.1 Proposed Work Flow

The block diagram for 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.

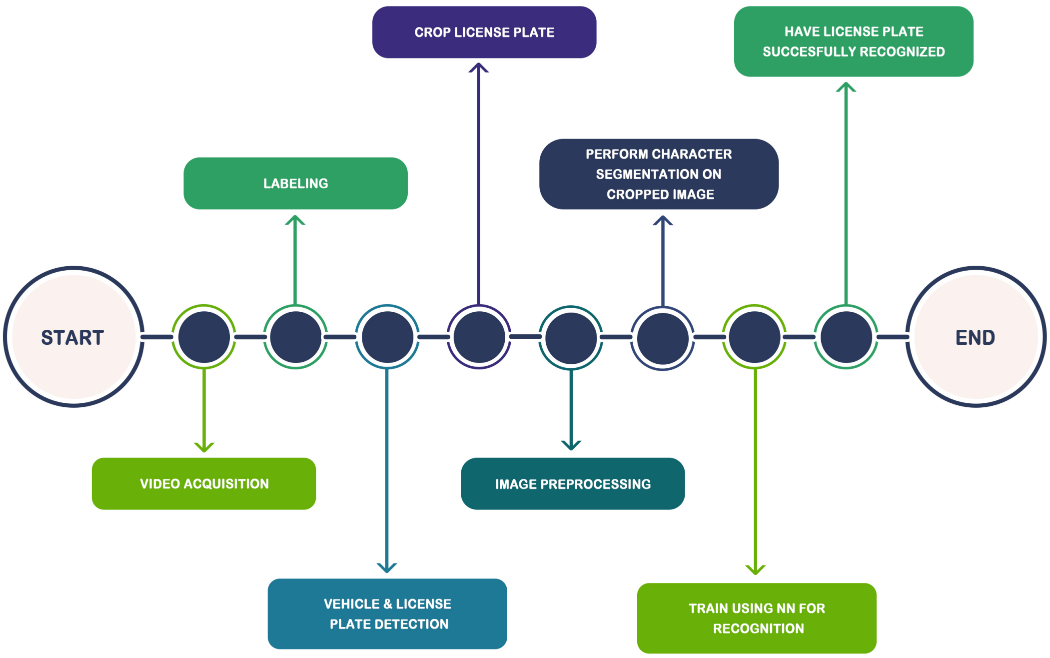

Figure 3.1: Work Flow

3.1.1 Video Acquisition

In our work, we used a phone to capture few videos and still images which were taken randomly from different angels of the camera on running vehicles on the road. The videos were taken in color mode. The quality of the videos were different depending on the surrounding environment.

Figure 3.2: Video Acquisition

3.1.2 Labeling

Data labeling is the process of detecting and tagging data samples. This work can be done manually but is usually performed or assisted by software. It is an important part for supervised learning. Both input and output data are labeled for classification to provide a learning basis for future processing.

We trained our system to identify vehicles and license plates in images. We provided the system with multiple images of various types of vehicles from which it learned the common features of each. So it was able to correctly identify the vehicles and license

plates in unlabeled images.
To label and annotate the training and test images, LabelImg annotation tool was used which is an open-source graphical image annotation tool. We did the labeling of the objects from images as car, bike, license plate.
The annotations were saved as XML files. The XML files were then converted into CSV files. The CSV files were converted into tfrecord format. Finally, the tfrecord files were used as input data to train the models.

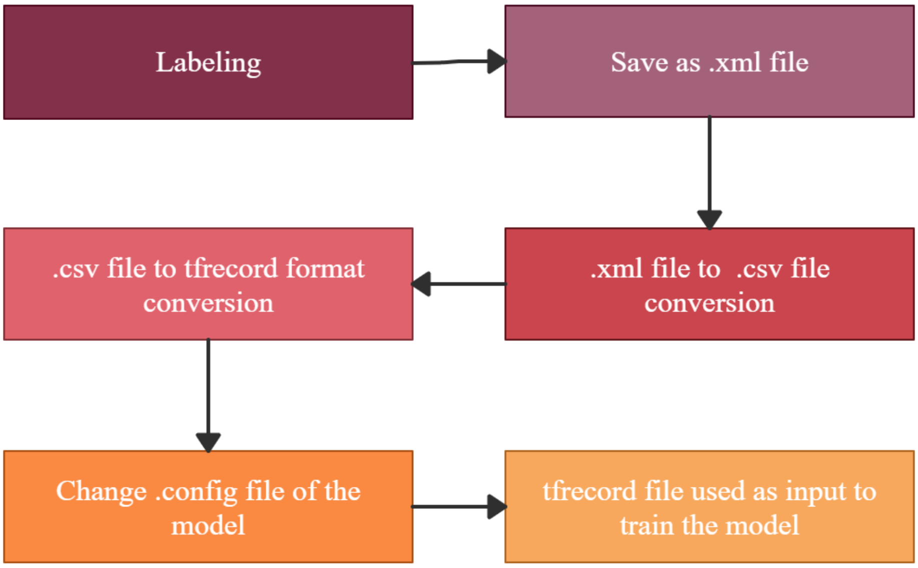

Figure 3.3: Data Processing for Training the System to Detect LP

The proposed model was tested using few videos and still images which were taken randomly from different angels of the camera on running vehicles on the road. The model could be tested with some other test samples, but due to having its good performance in each test in the license plate detection and the recognition of characters and numbers on the license plate 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.

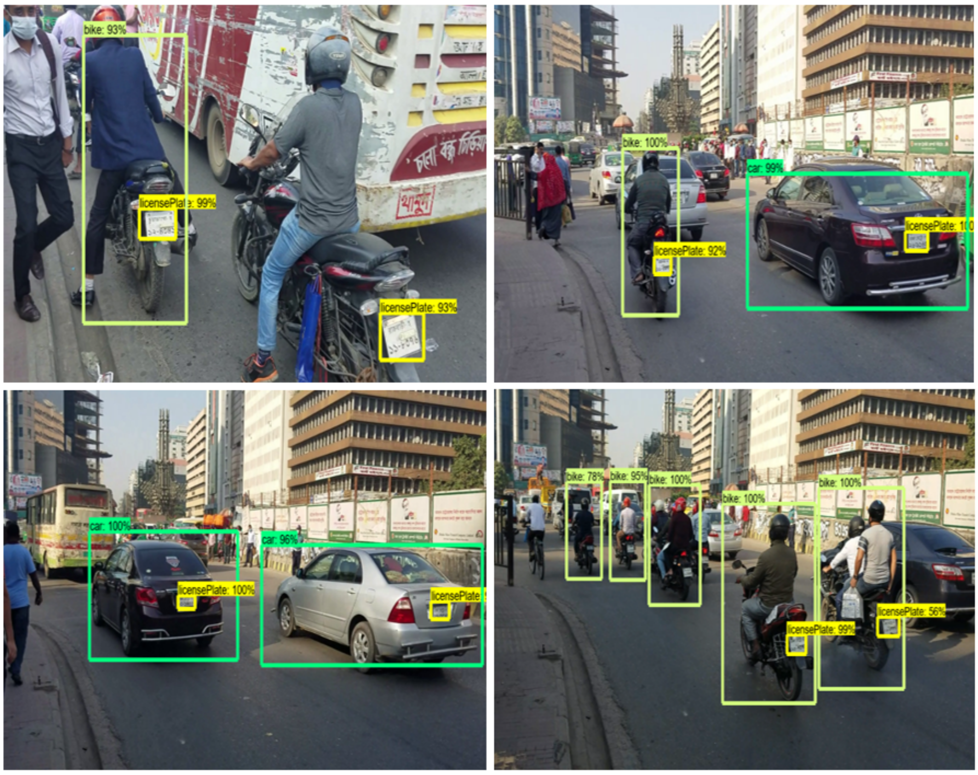

Figure 3.4: Vehicle and License Plate Detection

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.

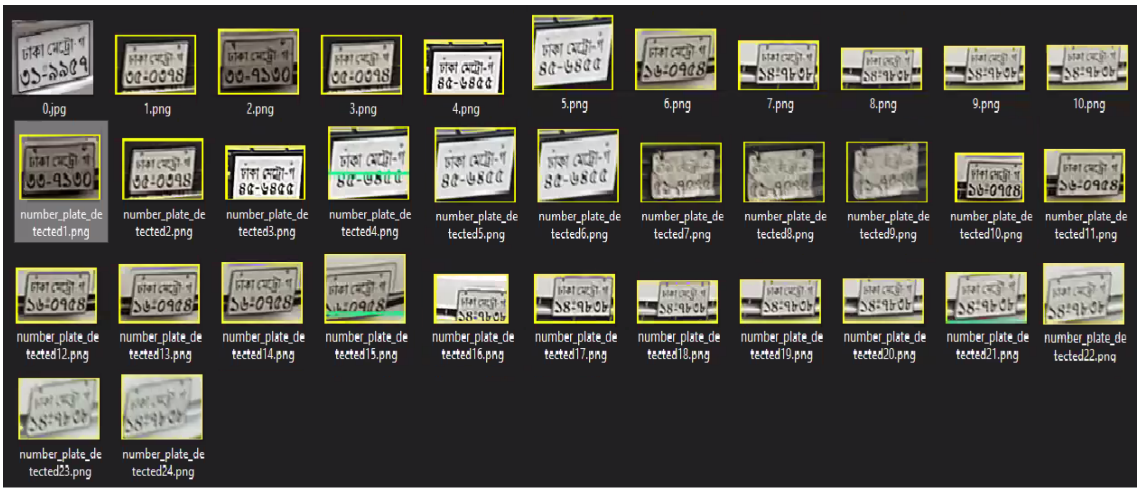

Figure 3.5: License Plates after Cropping

3.1.5 Image Pre-processing

Image pre-processing is the title for operations on pictures at the lowest level of abstraction whose aim is an advancement of the picture information that suppresses undesired distortions or upgrades a few picture features vital for advanced handling. It does not increment picture data content. These operations do not increment picture data content but they diminish it in case entropy is a data measure.

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

Gray-scale Conversion

Using cvtColor function an image is converted from one color space to another. In case of a change to-from RGB color space, the order of the channels ought to be indicated explicitly (RGB or BGR). The default color format in OpenCV is frequently referred to as RGB but it is really BGR (the bytes are turned around). So the primary byte in a standard (24-bit) color picture will be an 8-bit Blue component, the second byte will be Green, and the third byte will be Red. The fourth, fifth, and sixth bytes would at that point be the second pixel (Blue, then Green, at that point Red), and so on.

RGB $\leftrightarrow GRAY(CV\_BGR2GRAY)$:

$$\text{RGB[A] to Gray:} \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$\text{Gray to RGB[A]:} \quad R \leftarrow Y, G \leftarrow Y, B \leftarrow Y, A \leftarrow \max(ChannelRange)$$

Figure 3.6: Conversion to Gray-scale

A natural way to convert a color picture 3D array to a gray-scale 2D array is, for each pixel, to take the average of the red, green, and blue pixel values to get the gray-scale value. This combines the lightness or luminance contributed by each color band into a reasonable 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 looks approximately ten times brighter than blue. Through numerous repetitions of carefully designed experiments, psychologists have figured out how distinctive we see the luminance of red, green, and blue to be. They have given us a diverse set of weights for our channel averaging to get total luminance.

Sharpening

Upgrading the high-frequency components of a picture leads to an enhancement in the visual quality. Image sharpening refers to any upgrade procedure that highlights edges and fine details in a picture. Image sharpening comprises of including to the initial picture a signal that is proportional to a high-pass filtered form of the first picture.

The first picture is first sifted by a high-pass filter that extracts the high-frequency components, and after that, a scaled version of the high-pass channel output is included in the first picture, in this way creating a sharpened picture of the initial. The homogeneous regions of the signal, i.e., where the signal is consistent, stay unaltered. The sharpening operation can be represented by:

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

Here,

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

F is high pass filter.

γ is the tuning parameter which is greater that or equal zero.

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

The main point in the effective sharpening process lies in the choice of the high-pass filtering operation. Traditionally, linear channels have been utilized to execute the high-pass channel, however, linear techniques can lead to unsatisfactory results if the initial picture is adulterated with noise. A trade-off between noise attenuation and edge highlighting can be obtained if a weighted middle channel with appropriated weights is used. The weight we used is:

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

Denoising

We performed picture denoising utilizing a Non-local Means Denoising algorithm with a few computational optimizations. Noise anticipated being a Gaussian white noise. We applied it on our gray-scale images.

Gaussian blur is also used. It is a type of image-blurring filter that uses a Gaussian function which also expresses the normal distribution in statistics for calculating the transformation to apply to each pixel in the picture. The formula of a Gaussian function in one dimension is:

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

Here,

σ is the standard deviation of the Gaussian distribution.

Adaptive threshold algorithm is used also.

Binarization

Binarization is done using Otsu's thresholding method. The algorithm comprehensively looks for the threshold that minimizes the intra-class change, characterized as a weighted sum of variances of the two classes.

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

Here,

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

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

Dilation

We applied dilation to our binary images. It steadily broadens the boundaries of regions of foreground pixels and areas of foreground pixels develop in size while gaps inside those regions become smaller.

Erosion

Erosion is used in binary images to erode away the boundaries of regions of foreground pixels. Thus, areas of foreground pixels reduce in size, and holes within those areas become larger.

3.1.6 Character Segmentation

Bangla characters in Bangladeshi license plates are exceptionally troublesome to fragment since of their various complexities. To deal with these complexities, an algorithm is presented that uses horizontal and vertical projections with thresholds to fragment Bangla characters and digits effectively.

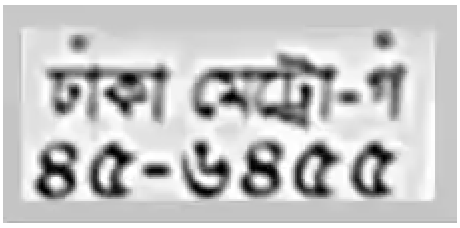

Figure 3.7: Before Segmentation

The segmentation process has two steps which are:

1. **Separating the rows:**
   Bangladeshi number plates contain two lines of text. Horizontal projection is done by calculating total number of row pixels which sections the rows in two different lines. The row with the least values of horizontal pixels are the beginning or the conclusion of a row in the plate.

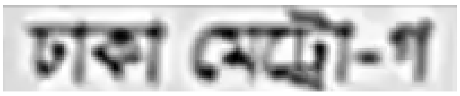

Figure 3.8: After Horizontal Segmentation 1st Row

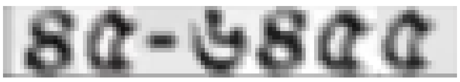

Figure 3.9: After Horizontal Segmentation 2nd Row

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

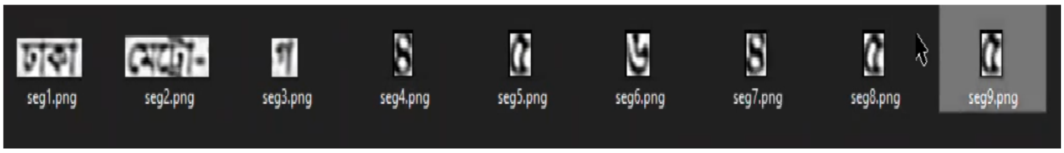

Figure 3.10: Segmented Bangla Words and Digits

3.1.7 Train System using Neural Network for Recognition

We trained our system using neural network. A neural network is a series of algorithms that endeavors to recognize fundamental connections in a set of information through a process that imitates the way the human brain works. In this sense, neural networks refer to systems of neurons, either natural or artificial in nature. Neural networks can adjust to changing input, so the network creates the finest possible result without requiring updating the output criteria.

Neural Network

A neural network works comparably to the human brain's neural network. A "neuron" in a neural network is a scientific function that collects and classifies data agreeing to a particular design. The network bears a solid likeness to factual strategies such as curve fitting and regression analysis.

A neural network contains layers of interconnected junctions. Each junction is a perceptron and is comparative to numerous linear regression. The perceptron feeds the signal delivered by a multiple linear regression into an activation work that will be nonlinear. In a multi-layered perceptron (MLP), perceptrons are organized in interconnected layers. There are three types of layers which are:

1. **Input Layer:** The input layer collects input patterns.
2. **Output Layer:** The output layer has classifications or output signals to which

input designs may map. For instance, the patterns may include a list of quantities for specialized indicators about security.

3. **Hidden Layer:** Hidden layers fine-tune the input weightings until the neural network's edge of mistake is negligible. It is hypothesized that hidden layers extrapolate notable features in the input information that have predictive control with respect to the outputs. This depicts feature extraction, which finishes a utility comparative to factual procedures such as vital component analysis.

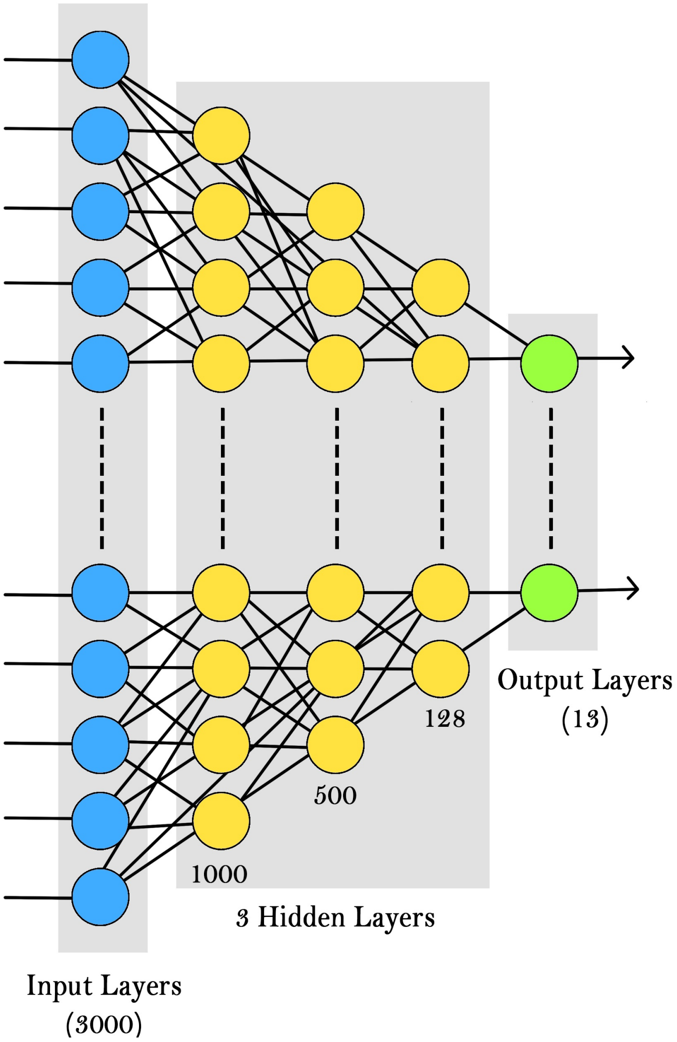

Figure 3.11: Neural Network of Our System

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 sort of activation function based on a ReLU, but it incorporates a little slope for negative values rather than a flat slope. The slope coefficient is decided before training, i.e. it is not learnt 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 characteristic ”S”-shaped curve or sigmoid curve. A common case of a sigmoid function is the logistic function. It helps in lessening the time required for making models, there is a major disadvantage of data loss due to the derivative having a short-range.
It is characterized by the equation:

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

Name	Equation	Derivative (with respect to x)	Range
Logistic (a.k.a. Sigmoid or Soft step)	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$
Leaky rectified linear unit (Leaky ReLU)	$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$

Figure 3.12: Sigmoid and Leaky ReLU Functions Comparison

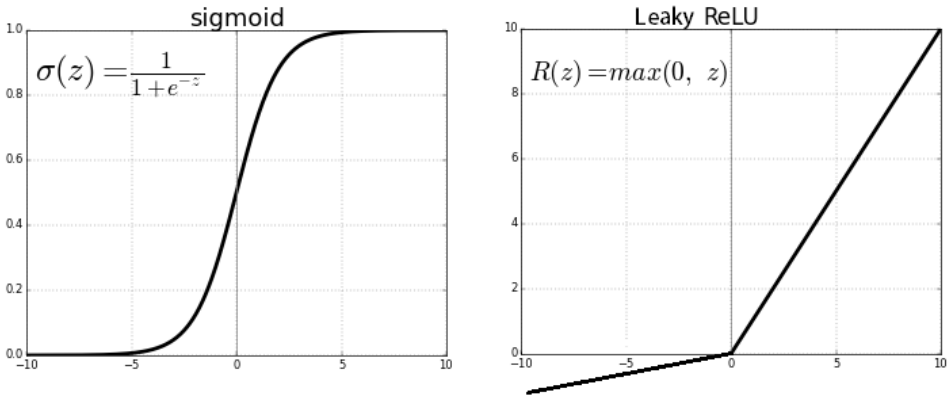

Figure 3.13: Sigmoid and Leaky ReLU Functions Graph Plot

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.

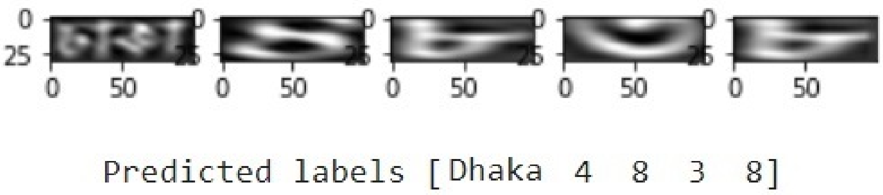

Figure 3.14: Prediction Using NN

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. 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.

	A	B	C	D
1	Date and Time	License plate		
2	7/3/2021 3:31	DHAKA METRO GA 456455		
3	7/3/2021 8:03	DHAKA METRO GA 350374		
4	7/3/2021 8:15	DHAKA METRO GA 160754		
5	7/3/2021 8:34	DHAKA METRO GA 337130		
6	7/3/2021 8:57	DHAKA METRO GA 147838		

Figure 3.15: 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.

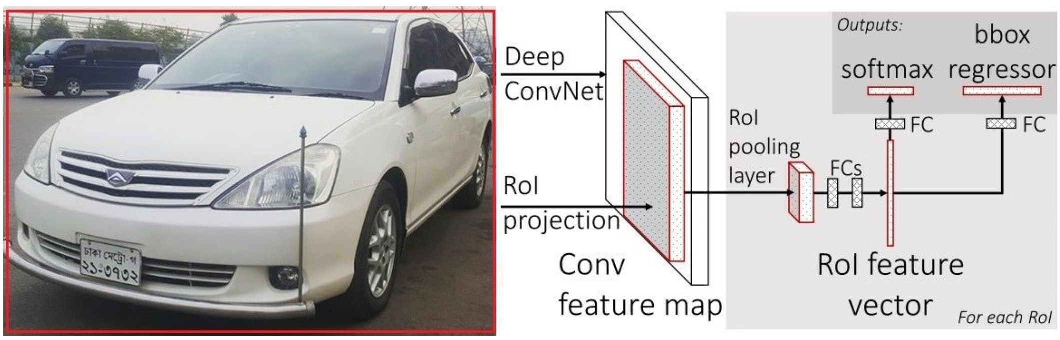

Figure 3.16: Faster R-CNN

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.

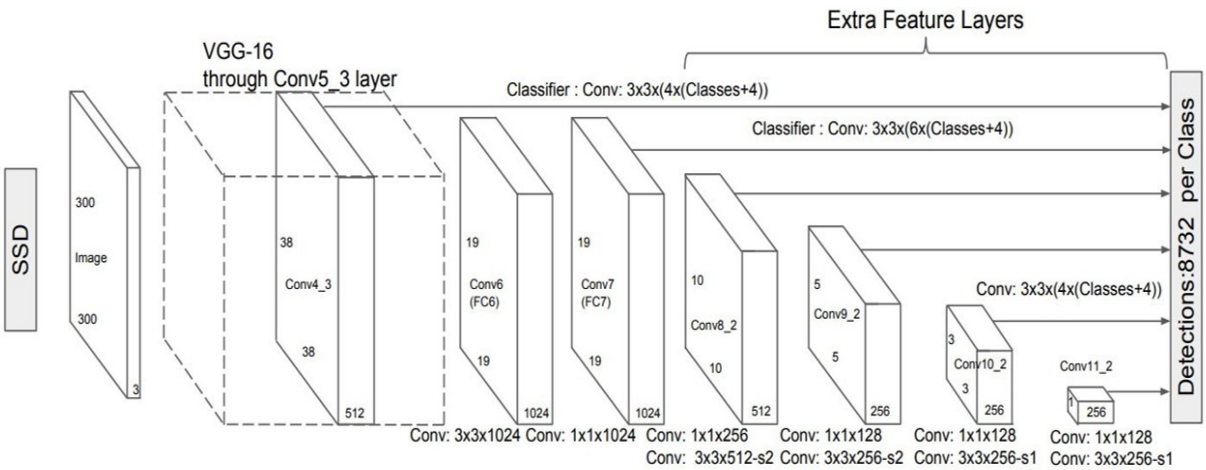

Figure 3.17: SSD Mobilenet V1 COCO

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.

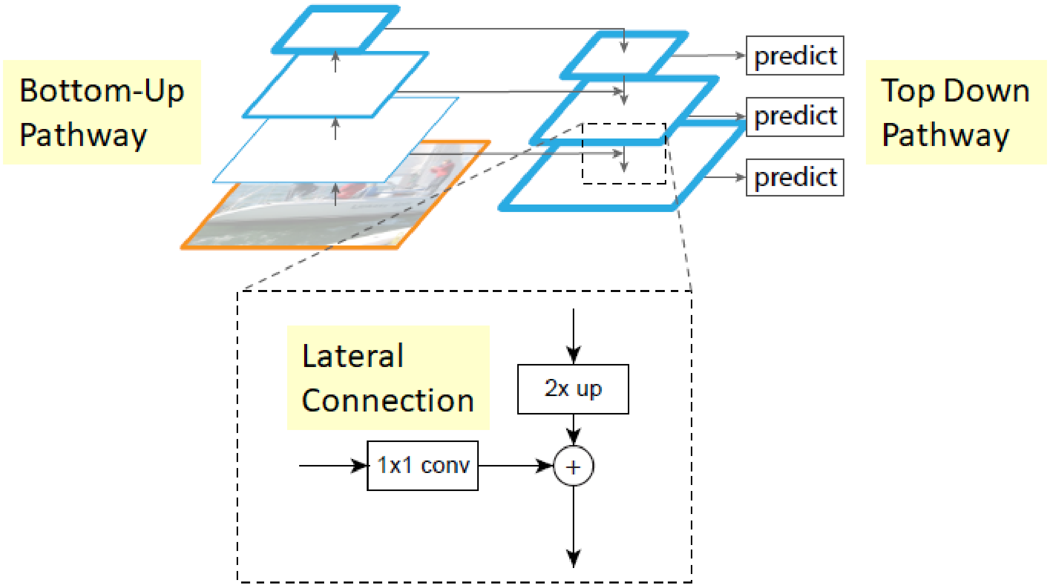

Figure 3.18: SSD Mobilenet V1 FPN COCO

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 accuracy. The model size of it is smaller than other high-accuracy models but it can only achieve 5 frames per second (fps) on a high-end embedded CPU. It still do not meet the requirement of real-time processing.

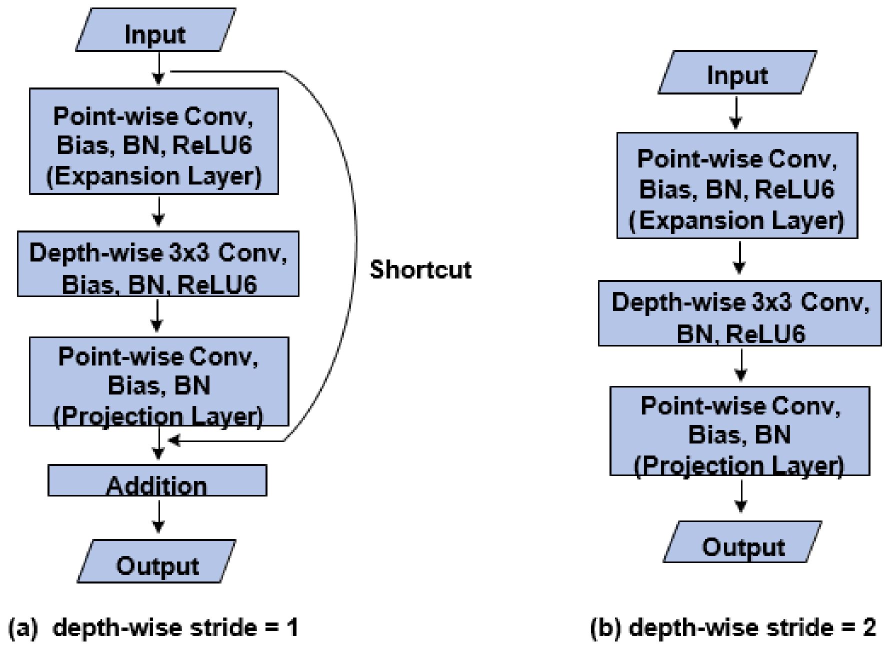

Figure 3.19: The structure of bottleneck residual block

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 but without any non-linearity. It is claimed that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.

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

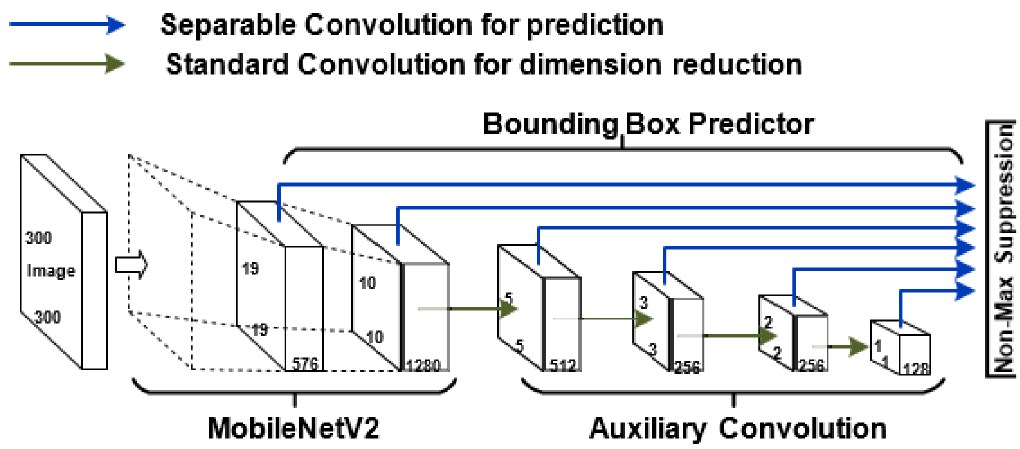

Figure 3.20: The network architecture of SSDlite Mobilenet V2

called base network. It is usually a truncated classification network such as VGG-16. It is followed by a set of auxiliary convolutional layers which enable features extraction at several scales and decrease the input size of each following layer. The bounding box predictor is a small group 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 3.1: Training parameters for the models which has been used in vehicle and license plate detection

Model	Batch Size	Initial 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.

4.1 Localization Graphs

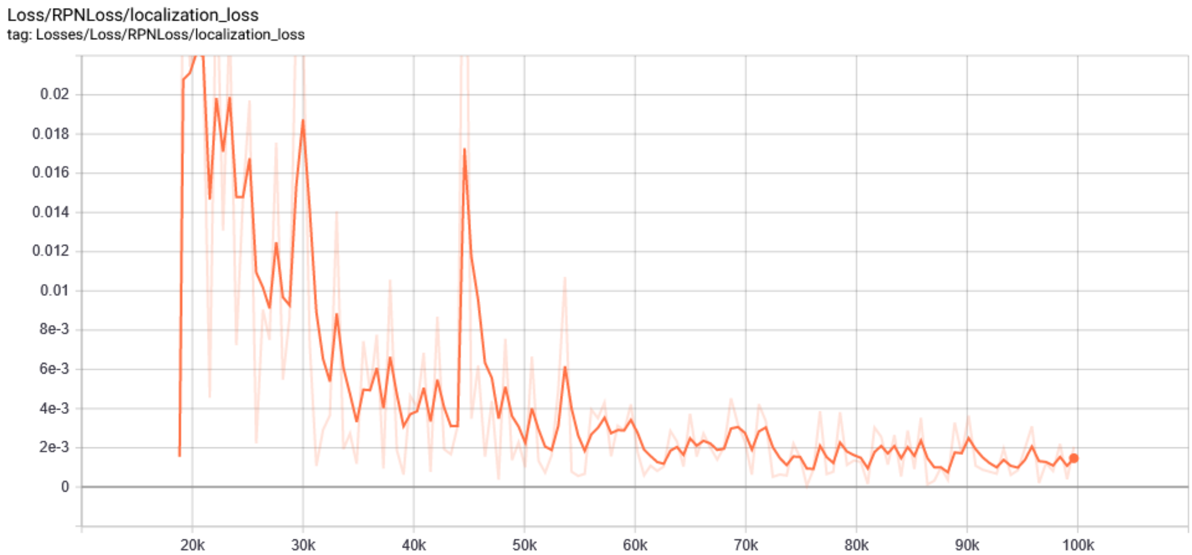

Figure 4.1: Localization Graph of Faster RCNN

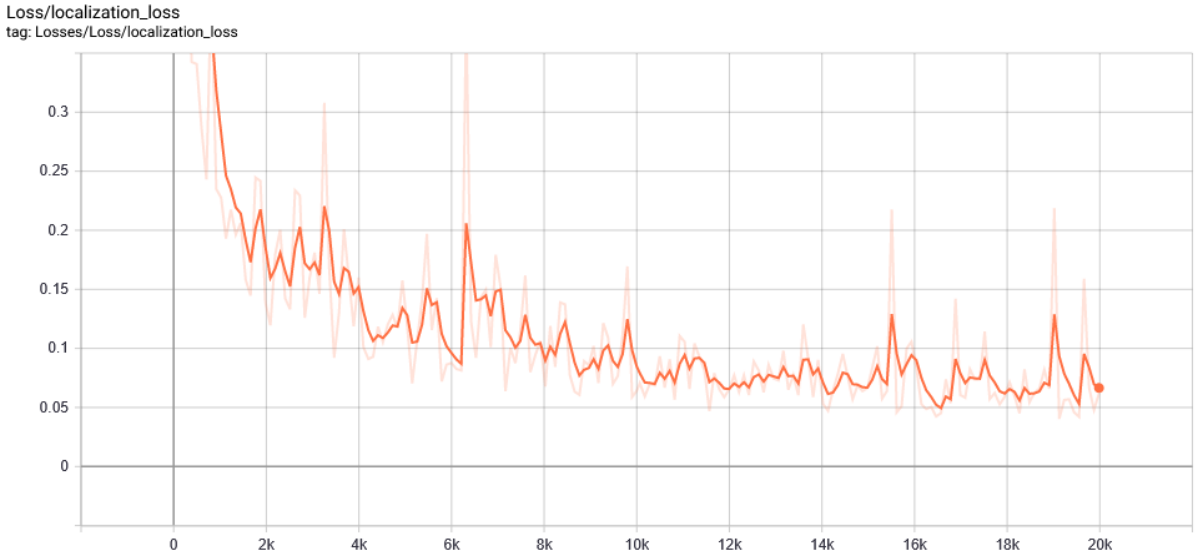

Figure 4.2: Localization Graph of SSD Mobilenet V1 COCO

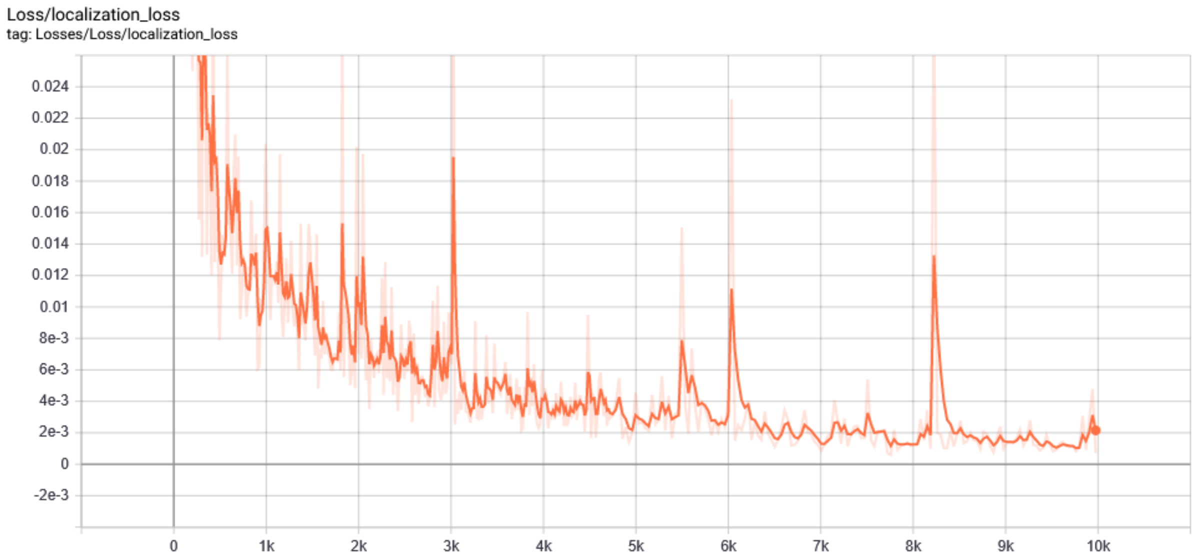

Figure 4.3: Localization Graph of SSD Mobilenet V1 FPN COCO

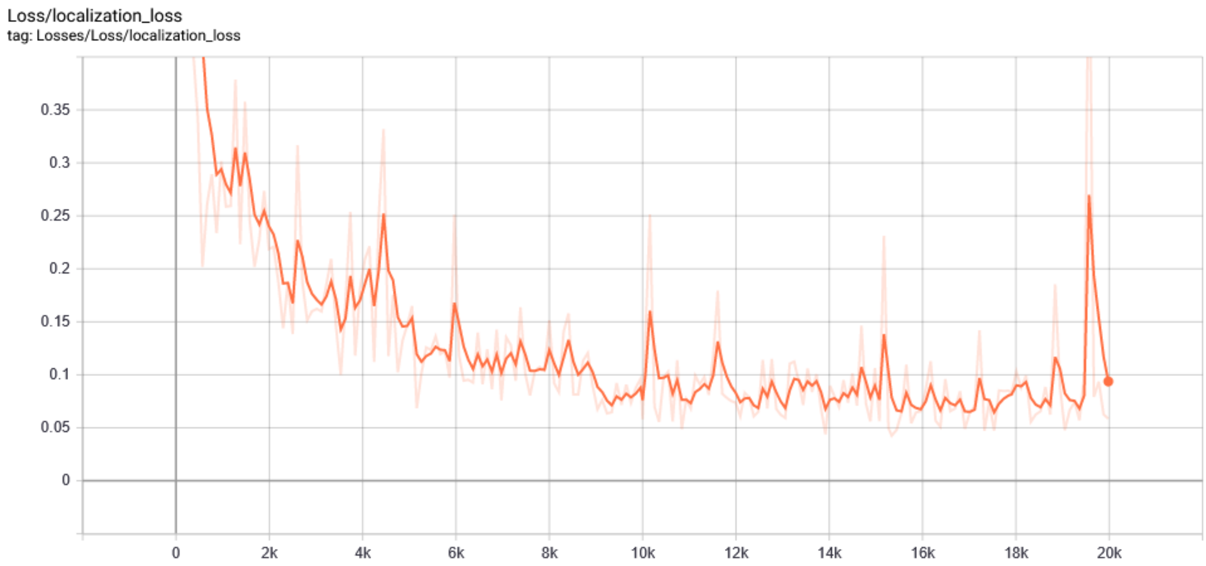

Figure 4.4: Localization Graph of SSDlite Mobilenet V2 COCO

4.2 Classification Graphs

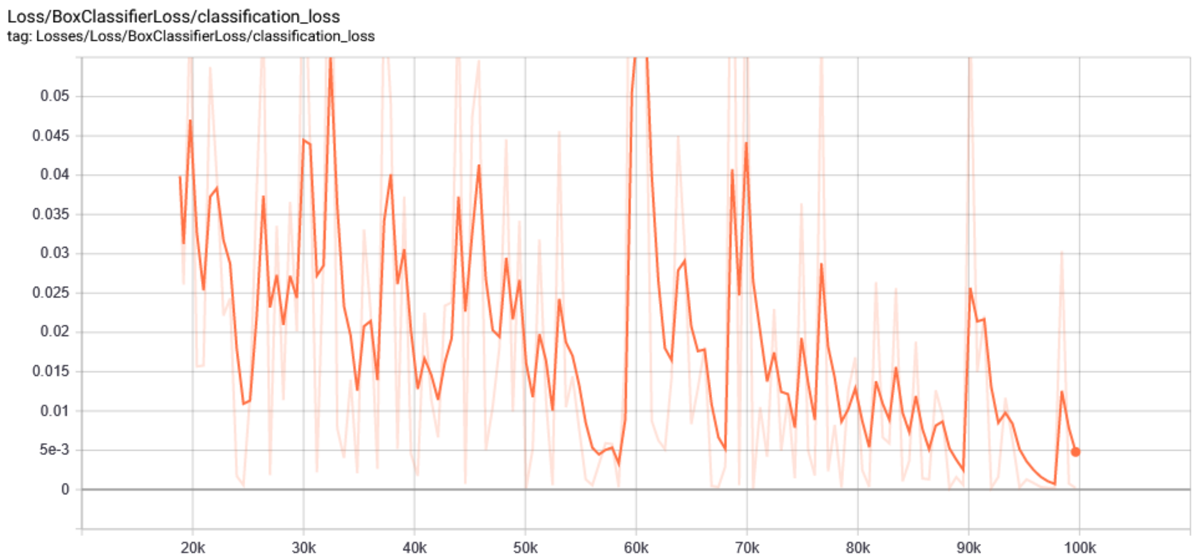

Figure 4.5: Classification Graph of Faster RCNN

Figure 4.6: Classification Graph of SSD Mobilenet V1 COCO

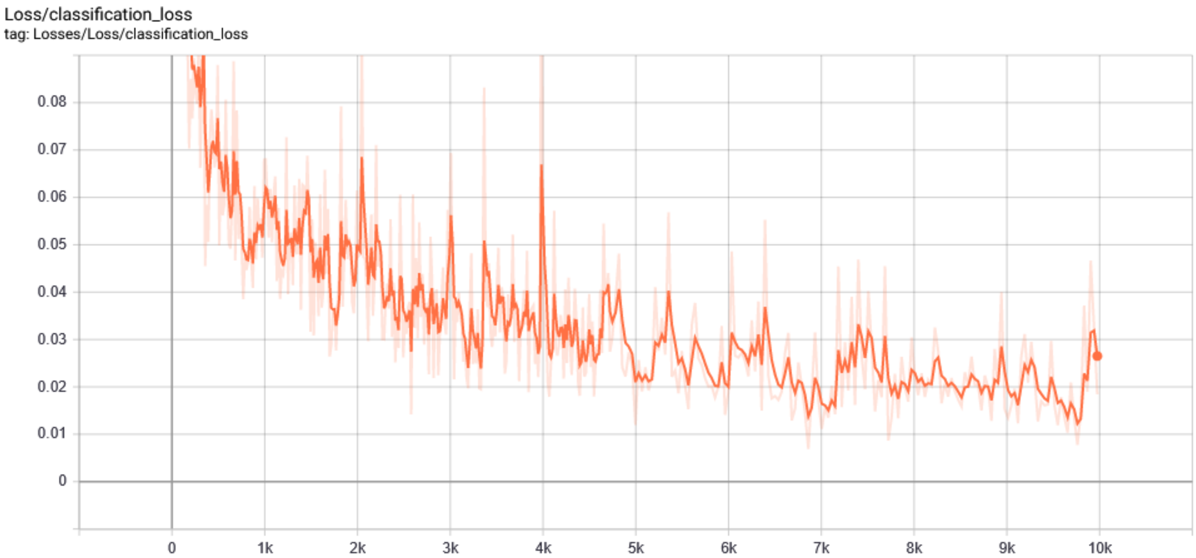

Figure 4.7: Classification Graph of SSD Mobilenet V1 FPN COCO

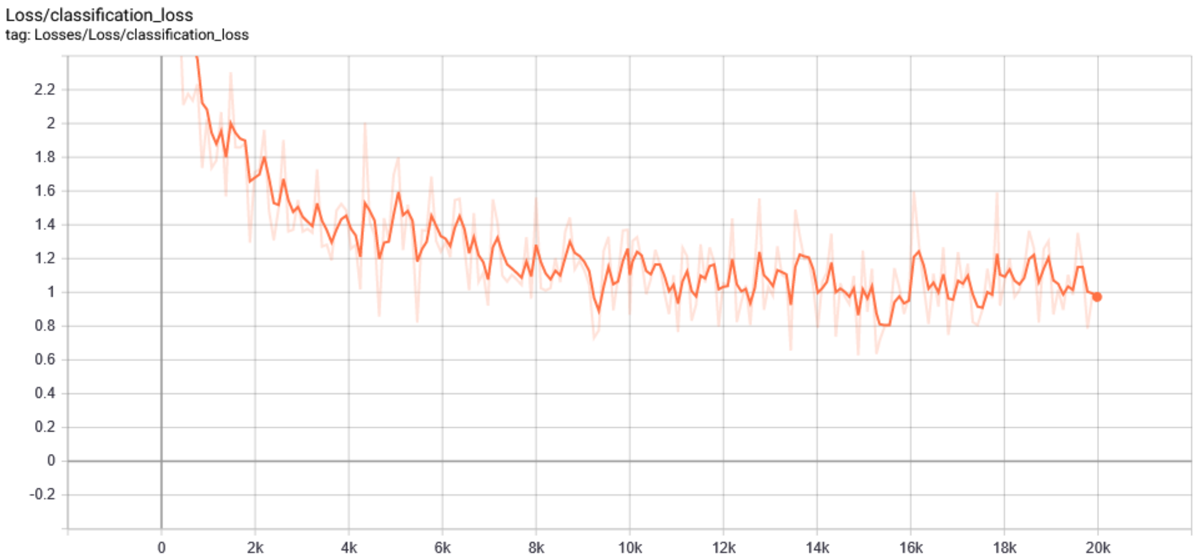

Figure 4.8: Classification Graph of SSDlite Mobilenet V2 COCO

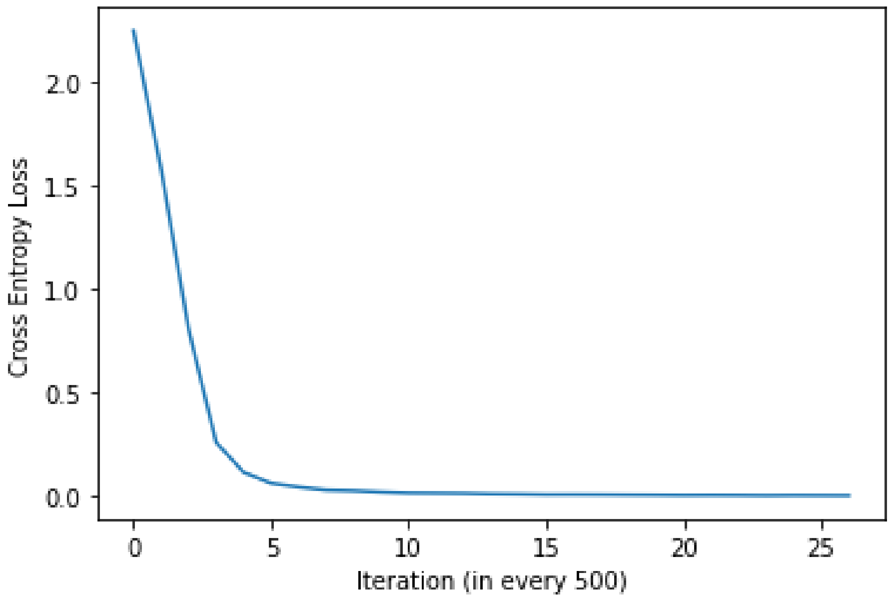

Figure 4.9: Loss Graph Of Recognition

In this system, mobile camera is used to capture few videos at 5 to 10 meters separations from camera to vehicle with the distinctive background. These algorithms are applied on videos of Bangladeshi commercial vehicles snapped beneath distinctive light conditions, such as night, sunny, and cloudy. The parameters that are utilized in the sifting operation to distinguish ROI from the candidate list appear as the experimental result 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 presence of complex background and exceedingly variable license plate designs in the input, the normal detection rate of our algorithm is 92.6%, segmentation rate 80.3% and recognition rate 88.8%. It demonstrates that the performance of our algorithm is better 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 proposed system in real-time applications more effective and efficient way, algorithms can be implemented in hard wire and parallel appliances, which need a lot of research 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 that can fit in police vehicles, allowed law enforcement officers 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 certain limits on parameters like the speed of the vehicle, script on the vehicle number plate, skew within the picture which can be expelled by improving the algorithms further.
- ANPR solutions 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 six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import zipfile

from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image

from utils import label_map_util
from utils import visualization_utils as vis_util
import cv2
import glob

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

# In[4]:

# 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

# 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='')

# In[7]:

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)


# # Detection
# Size, in inches, of the output images.
IMAGE_SIZE = (12, 8)


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
```

```python
    # 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)
            # draw a green rectangle to visualize the bounding rect
            # img = cv2.rectangle(img, (x, y), (x + w, y + h), (255, 255, 255), 2)
            crop = img[y: y + h, x: x + w]
            cv2.imwrite('temp/preprocess/h_seg' + str(count) + '.png', crop)
    # cv2.imshow("output", img)
    # print(len(contours))
```

```
    # print(count)
    # cv2.waitKey(0)


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)
        # cv2.imshow("original", img)
        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)
        # cv2.imshow("adaptive", ad)

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

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

        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)
                # draw a green rectangle to visualize the bounding rect
                # temp_img = cv2.rectangle(temp_img, (x, y), (x + w, y + h), (0, 255, 0), 2)
                seg = temp_img[y: y + h, x: x + w]
                cv2.imwrite('temp/segmentation/seg' + str(count) + '.png', seg)
        # cv2.imshow('contours1', temp_img)

    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)
        # cv2.imshow("adaptive", ad)

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

        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, ksize=(1, 1))
        im = cv2.erode(ad, kernel)
        # cv2.imshow("erosion", im)

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

        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)
                # draw a green rectangle to visualize the bounding rect
                # temp_img = cv2.rectangle(temp_img, (x, y), (x + w, y + h), (0, 255, 0), 2)
                seg = temp_img[y: y+h, x: x+w]
                cv2.imwrite('temp/segmentation/seg' + str(count) + '.png', seg)
        # cv2.imshow('contours', temp_img)
        # cv2.waitKey(0)


# In[10]:
def cleanup_image(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()
    # Template matching

    string = "Zim"
    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 = cleanup_image(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**