BANGLA LICENSE PLATE RECOGNITION USING IMAGE PROCESSING AND ADAPTIVE NEURO-FUZZY INFERENCE TECHNIQUE

ABU JAR MD. MINHUZ UDDIN AHMED

MASTER OF SCIENCE IN CIVIL ENGINEERING (TRANSPORTATION)

DEPARTMENT OF CIVIL ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA, BANGLADESH
AUGUST, 2021

BANGLA LICENSE PLATE RECOGNITION USING IMAGE PROCESSING AND ADAPTIVE NEURO-FUZZY INFERENCE TECHNIQUE

by
ABU JAR MD. MINHUZ UDDIN AHMED
Student ID No.: 0417042410

A thesis submitted to the Department of Civil Engineering, Bangladesh University of Engineering and Technology, Dhaka in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE IN CIVIL ENGINEERING (TRANSPORTATION)

DEPARTMENT OF CIVIL ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA, BANGLADESH
AUGUST, 2021

The thesis titled, **"Bangla license plate recognition using image processing and adaptive neuro-fuzzy inference technique"**, ID No. **0417042410**, Session: "**April 2017**" has been accepted as satisfactory in partial fulfilment of the requirement for the degree of Master of Science in Civil Engineering (Transportation) on August 03, 2021.

BOARD OF EXAMINERS

Dr. Md. Shamsul Hoque Professor CE, BUET, Dhaka	Chairman (Supervisor)
Dr. Md. Delwar Hossain Professor and Head CE, BUET, Dhaka	Member (Ex-Officio)
Dr. Mohammad Neaz Murshed Associate Professor CE, BUET, Dhaka	Member
Dr. Md. Shahid Mamun Professor Civil Engineering, AUST, Dhaka	Member (External)

DECLARATION

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it contains no material which has been accepted or submitted for the award of any other degree.

I also declare that, to the best of my knowledge and belief, this thesis contains no material previously published or written by any other person except where due reference is made in the text of the thesis.

August 03, 2021

ABU JAR MD. MINHUZ UDDIN AHMED
Student ID: 0417042410
Date: 03.08.2021

ACKNOWLEDGEMENT

First and above all, praise is to almighty ALLAH for providing the opportunity and granting the capability to proceed successfully to complete this thesis.

The current form of this thesis appearance has been made possible due to the assistance and guidance of several persons. A special gratitude should be given to Dr. Md. Shamsul Hoque, Professor, Department of Civil Engineering, Bangladesh University of Engineering and Technology (BUET) for giving his consistent supervision and pertinent services throughout the thesis.

In addition to that, profound gratitude should be expressed towards Dr. Md. Hadiuzzaman, Professor, Department of Civil Engineering, Bangladesh University of Engineering and Technology (BUET) for his logistic support especially in the field of artificial intelligence while conducting this research.

Special honor to be given towards Mr. Rashadul Alom and Mr. Haron Ali, Camerapersons collecting the photographs of the license plates taken for the ALPR model development and validation.

Last but not the least tribute should be given to Mr. Asgar Ahmed, Sergeant, Dhaka Metropolitan Traffic Police for providing his kind support while collecting photographs from the nominated locations in desired manner for the purpose of research.

ABSTRACT

Automatic License Plate Recognition (ALPR) system is widely used all over the world for traffic vehicle detections. But Bangladesh as one of the populous countries of the world having increasing vehicle demands day-by-day, there is no well-structured ALPR system developed for the detection of vehicles where all the license plates are written in Bangla. Especially, Bangladesh is such a country where congestion is a perennial problem in its capital city Dhaka bearing a huge wastage of money causing incremental traffic delay. Similar problem is observed in all aspects like toll collection, implementation of traffic regulation etc.

Therefore, an effort has been made to develop an ALPR model using MATLAB which could be used to recognize characters from the license plate images through image processing (IP) and Adaptive Neuro-Fuzzy Inference System (ANFIS). As a founding stone of the ALPR system in Bangladesh, 20 frequently used vital characters in the Bangladeshi license plates have been acknowledged to develop the model. Additionally, during the development of the model, some basic assumptions have been made like connected words in Bangla have been treated as distinct characters. Geometric features for individual character have been distinguished using different variable input values which have been used to train ANFIS and later on detecting those characters from the license plate images. There are two basic operations of the ALPR model, first one to detect the license plate region and the second one is to recognize characters through extracting them from that region. ALPR is performed from the stored images in the directory of the program during this research which has been recommended to be converted into real-time captured image detection model for the future studies. For detection of the license plate region, shape, area and background intensity of the license plate have been incorporated to the model. Validation has been done for this model using confusion matrices.

It was observed that for both license plate status and character recognitions, more than 90% accuracy has been attained. Basic operations of the model have also been found satisfactory while executing the program in MATLAB. Relatively greater errors have been identified for the unknown characters as some of those have been recognized as the known characters trained by ANFIS. It can be expected that if all characters are trained through ANFIS, this problem can be mitigated up to a great extent. Also, some complexities have been found during the detection of the license plate region for white colored vehicles as it matches with the background intensity condition of the license plates. Albeit these minor issues, all other operations have been found prominently viable. Therefore, the model can be expected to be a foundation of Bangladeshi ALPR system.

LIST OF CONTENTS

CHAPTER 3: METHODOLOGY AND DATA COLLECTION

CHAPTER 4: DATA ANALYSIS AND MODEL DEVELOPMENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full form
AI	Artificial Intelligence
ALPR	Automatic License Plate Recognition
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
ANPR	Automatic Number Plate Recognition
AQM	Active Queue Management
ATM	Active Traffic Management
AVI	Automatic Vehicle Identification
BRTA	Bangladesh Road Transport Authority
CCA	Connected Component Analysis
CNN	Convolutional Neural Network
CPR	Car Plate Recognition
DMP	Dhaka Metropolitan Police
FCM	Fuzzy c-means
GUI	Graphical User Interface
IP	Image Processing
ITS	Intelligent Transportation System
LDC	Least Developed Country
LPR	License Plate Recognition
MF	Membership Function
MI	Machine Intelligence
ML	Machine Learning
NN	Neural Network
OCR	Optical Character Recognition
OD	Origin-Destination
PNN	Probabilistic Neural Network
PSDB	Police Scientific Development Branch
RFID	Radio-frequency Identification
RGB	Red, Green and Blue
RNN	Recurrent Neural Network
RR	Retro-Reflective
SNN	Simulated Neural Network
uint8	unsigned 8-bit integer
UK	United Kingdom
UN	United Nations
USA	United States of America
WWII	World War II

CHAPTER 1
INTRODUCTION

1.1. Background

On the eve of November 2019, the new traffic act *i.e.,* “Road Transport Act – 2018” has entered into force in the fullness of time [1]. Despite embracing the argument of whether it should have been enforced before or after proper campaigns [2], the concern is whether it can be implemented in a competent manner or not. It is quite evident that the road safety movement initiated by the young generations of Bangladesh paved the way to implant this act anyway [3]. After massive protest throughout the country by the students for couple of weeks, Cabinet of Bangladesh Govt. finally approved this act *i.e.,* “Road Transport Act – 2018” in August 6, 2018 [4].

Although the act was well accepted by the passenger and pedestrian groups of the country, statistics faced up to the regulation authority. It has been started with the fact that Dhaka is the most densely populated city in the world [5]. Not only this but also Bangladesh is persistently holding the place within the world’s most 15 densely populated countries [6]. From this point of view, it can be easily predicted that how enormous amount of vehicle should be there in the roadway of Bangladesh. One report of Bangladesh Road Transport Authority (BRTA) [7] shows that there are 3, 35, 660 private passenger cars, 21, 45, 659 motor cycles, 1, 35, 081 trucks and many other types of vehicle. Moreover, as one of the top 5 fast growing countries of the world [8], the number of vehicles is expected to be increased more in the near future. So simply the question arises, “How to implement the traffic act over such an enormous number of vehicles?” It is quite obvious that the regulating authority will be bewildered while handling these activities.

The number of vehicles might not be a matter of concern if the available roadways in Dhaka city would have been satisfactory enough for those vehicles. It was noticed that due to the massive emergence of private cars, available roadways have become incapable of supporting the traffic loads. A news report of 2020 [9] it says, “Private cars have taken up more than 50% of available road space in Dhaka. A car can accommodate two people in a 10-square-feet space, while a public transport can accommodate 15 people on the same space.” The summary of the scenario is, Dhaka city as well as Bangladesh is still far behind in developing and implementing proper public transport policy [10].

The above-mentioned private cars and others giving rise to chronic congestion in the roadway. To be more specific according to a report of 2019 [11] due to traffic jams, 3.8 million working hours is lost every day. It might be the scenario of a normal day. But if focused on the traffic condition during Ramadan *i.e.,* a holy month of the Muslim majority people in Bangladesh, the situation becomes terrible. A news report referring to the World Bank's report mentioned, "…in the last 10 years, the average traffic speed in Dhaka has dropped from 21 kilometers per hour (kmph) to 7 kmph, and by 2035, the speed might drop to 4 kmph-which is slower than the walking speed."[12] Despite exploring whether those data denominate time mean speeds or space mean speeds, the implication is abominable.

As the protest of the students mentioned earlier in [1] rose peak, the Dhaka Metropolitan Police (DMP) immediately announced traffic week and demanded, "To reinforce discipline on roads and highways, we'll observe the Traffic Week across the country from Monday."[13] Interestingly, if the consequences is seen in writing, it indicates that 4, 484 cases had been filed in just one day during the traffic week [14]. If it is compared to the previously mentioned average traffic speed of Dhaka [12], one can guess whether it is actually utilitarian by guaranteeing the usual flow of the traffic stream, or not. It should be also kept in mind that; a flow breakdown can cause incremental delay over the highway. So, the visual 'one second delay' of a particular vehicle is not a second only in collective manner. The essence for conducting any documentation *i.e.,* filing cases against a vehicle is the license plate. It is consequently the most frequent thing which has to be identified on the roadway which needs a matching algorithm using digital technology. Although, it has been decided that every vehicle must use digital number plates and Radio-frequency Identification (RFID) tags from year 2016 [15]. But barely it came into effect, private car and bike users find almost no usefulness of this Retro-Reflective (RR) digital number plates at all [16]; let alone the public transports *e.g.,* busses.

Coincidentally at the same time, a report holds attention that Hundreds of vehicles are stuck on the highway adding to people's suffering in regular manner [17]. If further entered into the news, it clearly depicts the common scenario due to toll collection of toll plaza in Bangladesh. Though one of the major purposes of toll roads is to give dedicated service to the vehicles in exchange of fare commonly known as toll tax. But collection of toll tax has become so significant in time killing that time saving turned optional by means. Beyond doubt, license plate number entry is again a frequent task of toll tax collection. Same scenario is tangible for the parking entrances and exits of the public places *e.g.,* hotels, restaurants, shopping malls etc. Even if RFID is seen to be acting properly overnight, the private organizations or institutions might get little or almost no benefit from it in the near future. So, the need of a license plate recognition system by means of any technique seems crucial at this moment.

Before extending the discussion to the legitimate comparisons of the usage of license plate recognition systems among different regions of the world, the implication of recognizing license plate number should be divulged. Although it will be explicated in the upcoming chapters *e.g.,* Literature Review, Methodology and Data Collection etc.; “Automated License Plate Recognition (ALPR) is a technology that uses optical character recognition (OCR) to automatically read license plate characters.” [18] For recognition, it should have an attribute similar to human being which can identify alphanumeric values. “Fuzzy logic is capable of defining non-numeric values are often used to facilitate the expression of rules and facts.” defined by Zadeh [19]. This fuzzy logic is created according to the human like reasoning architecture commonly known as Artificial Neural Network (ANN). Adaptation of fuzzy logic and ANN to a hybrid intelligent system is known as Adaptive Neuro-Fuzzy Inference System (ANFIS). This intelligent system can be used to recognize the license plate numbers of the vehicles from their images via processing.

Nowadays, license plate detection has become far-flung not only among the developed countries but also the developing countries of the world. It is mostly used for the Origin-Destination (OD) Survey in traffic engineering. Tracsis Traffic and Data Services, one of the renowned United Kingdom (UK) based traffic service company published an overview about Automatic Number Plate Recognition (ANPR) where it mentioned, “During 2019 Tracsis collected data from 29,000,000 number plates.” [20] It is seen that it solely indicates the utmost importance of this technology in field of traffic engineering. In addition to that for the security reasons, this number plate recognition system is becoming highly cherished in the developed countries *e.g.,* United States of America (USA). To clarify this scenario this statement would have been enough: “Automatic license plate recognition (LPR) technology is becoming popular because it significantly reduces the time to arrest criminals. There is an increased need for advanced security measures on the roads, in schools, parking lots, tunnels, private businesses, etc.” [21] Bangladesh seemingly not far behind from being a developing followed by a developed country. United Nations (UN) has already recognized that Bangladesh has been becoming a developing country from Least Developed Country (LDC) soon. According to their statement, “… Bangladesh, the largest least developed country (LDC) in terms of population and economic size, looks likely to leave the LDC category by 2024.” [22]

In the meantime, Bangla which is the state language of Bangladesh has already placed the 7th place in terms of population. According to the report based on a study: “visual-content provider Visual Capitalist ranked Bangla in the top ten of the lists with 265 million native and non-native speakers across the globe.” [23] Emphatically, license plate numbers are designated using Bangla language and those will have to be recognized accordingly. Meanwhile, another problem is apparent *i.e.*, the license plate number poorly transparent. According to the study report of ICE Business Times, “The bus fleet of Dhaka is not very old … but poor maintenance, inadequate technology, unavailability of required facilities are the main causes for their deplorable condition.”[24] It is quite unfortunate that while the developed counties are being concerned with poor visibility due to fog [25], Bangladesh is suffering from the same problem due to lack of maintenance.

Keeping eye on all these factors as discussed above, it can be thoroughly expected that a proper development of automatic license plate recognition system in Bangla will have a colligation in mitigating the traffic related problems in Bangladesh. Thus, the scope and objectives of this technology based on the rationale will have to be drawn specifically which have been held forth in the upcoming discourses.

1.2. Objectives of the study

A noetic feeling is aroused from the statement “But there are limitations of roads and human skill to manage traffic jam.” [26] Combining this doctrine along with the steering of the background leads to an idea of developing Artificial Intelligence (AI) based solution which might cater for identifying vehicles as well as providing provisions of security.

As discussed before, ANFIS model will be developed thereby procedure of which will be discussed in the upcoming chapters namely, Methodology and Data Collection, Data Analysis and Model Development, and Model Validation. This model will be able to recognize license plate numbers of the registered vehicles of Bangladesh which are written in Bangla.

In addition to the model development, an effort will be made to compare the accuracy of the model for different visibility conditions. As stated in the background that, Bangladesh is suffering from poor maintenance vulnerability [24]. Therefore, the license plates are also seen to have faded numbers on it most of the times, although nowadays RR registration numbers are being used.

Meanwhile, the essence of the research has been established with proper justifications. Henceforth, major objectives of the research should also be established thoroughly. The specific objectives of this research are:

i. To formulate techniques for image processing of the Bangladeshi vehicles where non-lane based heterogeneous condition prevails

ii. To develop license plate recognition model using Adaptive Neuro-Fuzzy Inference System

iii. To train, calibrate and validate the proposed model for detecting even slightly faded and ill maintained license plates written in Bangla

iv. To vindicate performance of ALPR model for rotated and skewed license plates with respect to the horizontal line of symmetry.

The objectives stated above are the predominant intents to be acquired by means of this research. In addition to fulfilling these objectives, some observations will be made after the analysis of data which will be resolved later on in the chapters *e.g.*, conclusions and recommendations etc. later on.

Whatever those observations will be, everything will be done according to the guidelines of the Methodology and Data Collection chapter and comprehending the scope of the study which will be stated bit by bit. It can be expected from the view of policy maker that it will be utilized to undertake necessary measures and make the best use of this model. Furthermore, it would provide a stable foundation to uphold the study of AI in traffic management within Bangladesh.

1.3. Scope of the study

"By looking at the questions the kids are asking, we learn the scope of what needs to be done." [27] Correspondingly, after the discussions made in the background and objectives of the study, the scope of the study can be outlined. The study should be undertaken based on the availability of the resources on particular area of the concerned domain. Although it can be perceived colloquially, but it should be enlisted for aiding the arguments which will be made in the upcoming chapters.

It is well visible that: "The amount of available data for the evaluation of an innovative technology varies widely." [28] In case of AI application it turns out to be more variable because of human behavior, local trend, language etc. Hence, there are some local aspects which will have to be adopted. Keeping all these factors in mind, scopes are conscripted below:

i. The discussions made in the research regarding license plate will be generally applicable to the registered vehicles of Bangladesh unless specified. To be more specific, it will adopt the language, pattern, provision etc. of the Bangladeshi license plates and anything similar to the ordinances of the People’s Republic of Bangladesh pertaining to this.

ii. The qualitative measures *e.g.*, accuracy, efficiency etc. of the study in the data analysis will be referring to the Bangladeshi license plates, techniques and the software used for ANFIS modeling in this study mentioned in the Methodology and Data collection chapter correspondingly.

iii. The qualitative measures which will be obtained from this study will be dependent upon user inputs, photo resolutions, capture angles etc. as well. These factors will be discussed in the Methodology and Data Collection chapter.

iv. Calibration and validation of the model will be made in accordance with the discussions made in the Data Analysis and Model Development chapters.

The scopes drawn above enunciate the applicability of the research and encapsulate the whole study depicting interrelationships of different chapters of this research. Methodology and discussions will be expatiated based on these presaged scopes in those chapters.

1.4. Organization of the thesis

The overall thesis will be amassed with total six chapters to illustrate the methodology for achieving the aforementioned objectives coping with the scope of the study. These six chapters will be organized as follows:

Chapter 1	depicting contextual overview in a nutshell
Chapter 2	reviewing available relevant literatures on ALPR, fuzzy logic, ANFIS and software used for those applications; finally establishing the necessity of detailed study for ALPR in Bangla

Chapter 3 describing the methodology used for the thesis sorting out the best one for fulfilling the objectives and data collection procedure for this purpose as well

Chapter 4 dealing with various aspects of data analysis and depicting the model development bit by bit

Chapter 5 validating the proposed models using validation data and by performing some statistical analysis for the proposed model

Chapter 6 bringing the entire thesis to an end through suggesting recommendations for the future researches

CHAPTER 2
LITERATURE REVIEW

2.1. Introduction

On the cease of 1893, license plate was first issued in France making it the first country to do so followed by Germany in 1896 [29]. After that beginning, development of license plate has not been turned back by the human civilization. Then after a long period when rise of automobile industry had begun to boost towards peak, ANPR system had been developed. "Surprising to most people, the history of ANPR stretches into the last century as it was invented in 1976 in the UK at what was then known as the Police Scientific Development Branch (PSDB) (now titled Home Office Scientific Development Branch) and early systems were developed for use from 1979" [30]. In the meantime, a powerful tool of AI, ANFIS technique was developed in the early 1990 according to the Jang, 1991 [31]. As concluded from the Introduction chapter, this ANFIS will be used to develop the model of license plate recognition. Therefore, a brief overview on how the LPR techniques have emerged and how it coincides with the ANFIS modeling should be discussed. This chapter has been embellished with all the literature on gradual development of LPR and ANFIS. Moreover, constructive criticism will be made regarding application of LPR and ANFIS in Bangladesh. Thereby the importunity of this research will be once again formalized in the overview.

2.2. License plate

License in English refers to "an official document, card, etc., that gives you permission to do, use, or have something." [32]. It also refers to "freedom to act however you want to" [32]. Interestingly, the usage of license word itself is provided by a freedom to use however people wants to in case of license plate. It is because in British English it is known as number plate; whereas in American English it is called license plate but in Canadian English it is spelled as licence plate [29]. License plate can be defined as: "A sign on a road vehicle that shows its registration number" [33]. Similar to the word license, license plate is also called number plate or licence plate. It is usually attached to the front or back side of the vehicle, depending upon vehicle type and local jurisdictions. In case of Bangladesh, RR license plates have been in use for most of the vehicle [see Figure 2.1] especially in private cars and motorbikes. Available researches regarding license plates will be depicted later on in this chapter.

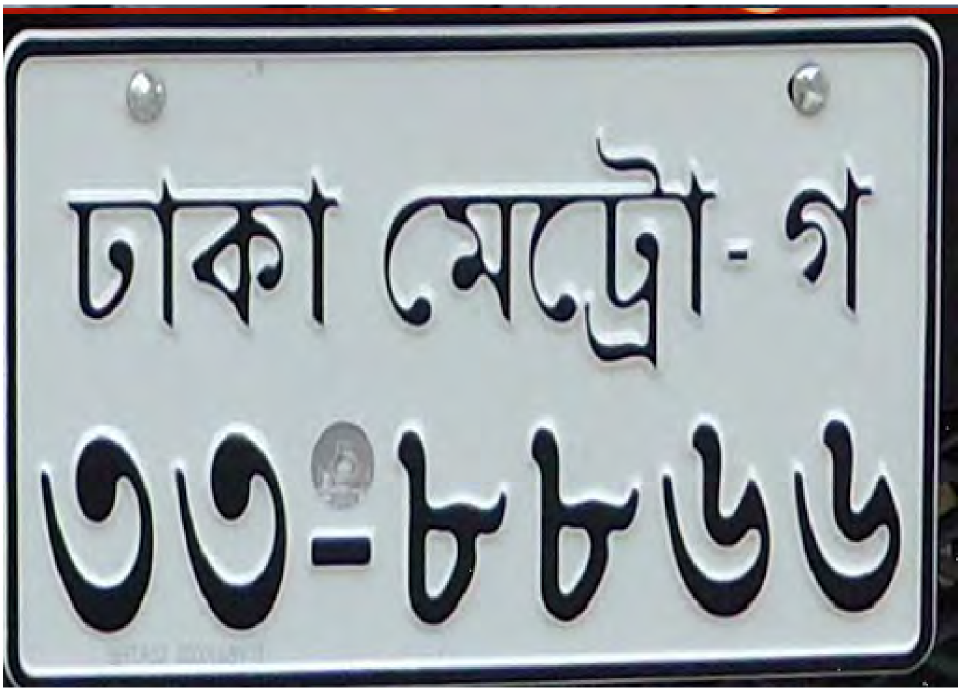


Figure 2.1: An example of license plate used in Bangladesh at present [34]

2.2.1. Unveiling of license plate in the world

As mentioned at the beginning of this chapter that license plate was initially introduced by France [29]; it soon began as a legal practice within the western world. Germany made their way in 1896. Not so far behind from that, US launched their first license plate in 1901. New York became the first state of US requiring license plate. But then "the plates weren't the long alphanumeric combinations drivers would recognize from today." [35] At that time owners were expected to provide numbers of their own to recognize their vehicles. So, it can be easily perceived that no restriction was there for providing designating license plate of their own. But if the phenomenon of state issued license plate is considered, the first one would be Massachusetts. It was the first state to provide porcelain license plate in 1903 [36]. After that, West Virginia produced the first metal license plate made of embossed metal in 1906 [37]. Since then, usage of metal had been developed until the metal shortage caused by World War II (WWII) in 1944. At that time states of US were forced to use alternative materials for license plates namely embossed fiberboard [see Figure 2.2], cardboard and soybean-based plastic [38]. Standardization process in license plates was taken into consideration in 1957 when automobile manufacturers imposed uniform dimensions for the license plates after meeting with organizations pertaining to the standardizations [39]. At that time standard size of the license plates in US states were set at 6 inch × 12 inch. Thereby the gradual improvements of the license plate system have been made thoroughly not only in US but also throughout the world.

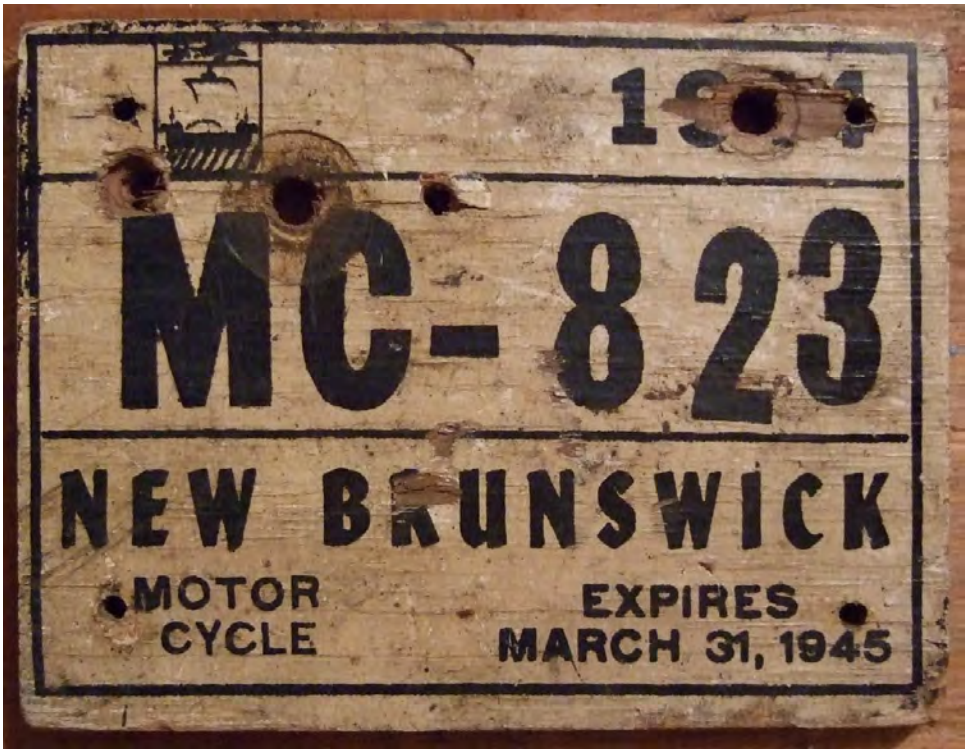


Figure 2.2: An example of license plate used in New Brunswick, US in 1944 [40]

2.2.2. Unveiling of license plate in Bangladesh

As discussed earlier, it is seen that the universal history of license plate dates back to the ninetieth century. But People's Republic of Bangladesh have made their journey begun with current version in 1973 just after its liberation [41]. BRTA is the authorized institution that issues the license plates in Bangladesh. As discussed earlier about the dignity of license plates are made in English or other languages in the western world. But as mentioned in the introductory chapter that as the official and mother language of Bangladesh, Bangla being 7th populous language at present and the world witnessed to go through Bangla license plates from 1973 [see Figure 2.3]. Since then to now on, not only the government license plates [see Figure 2.3(a) and 2.3(c)], but also the international organization license plates [see Figure 2.3(b)] are holding registration numbers written using Bangla numerals and alphabets.

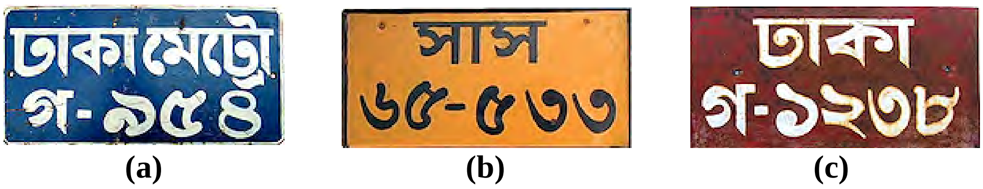


Figure 2.3: Samples of license plate used in Bangladesh in 1973 for (a) Government corporation, (b) International organization and (c) Government vehicle [42]

The situation at present is lot more different than that of eightieth decade. Bangladesh with the recent economic growth as discussed in the Introduction chapter, massive upraise of vehicle ownership is evident. The same situation encountered by western world nearly when Bangladesh started to use license plate as mentioned earlier in case of UK in 1976 [30]. Besides, license plate numbering attributes have been changed a lot as well [see Figure 2.1]. As mentioned in the scope of the study that the study will be applicable to the license plates of Bangladesh, those particular attributes will be enlisted in the Methodology and Data Collection chapter. Prior to that, it will recollect all the researches made after on the previous license plates, incapacitation of those will be made in this chapter later on.

2.2.3. Automatic license plate recognition

Dispute of spelling of license has been apprised in the beginning of this chapter, the situation becomes more complex in case of the automatic recognition of license plates. As mentioned in the beginning as well that the automatic recognition of license plate was formulated in UK [30], they have nominated the name ANPR for this considering number plate instead of license plate following their usual convention. But due to the variety of nomination, it has been already in use with different terminology *e.g.,* Automatic License Plate Recognition (ALPR), License Plate Recognition (LPR), Car Plate Recognition (CPR), Automatic Vehicle Identification (AVI) etc. Whatever the terminology is, it is recognized as the "heart" of any Intelligent Transportation System (ITS) [43]. The formal definition can be taken as "software used to recognize the number plates automatically by performing sophisticated optical character recognition on images to read [see Figure 2.4] the license plates of vehicles." [44]

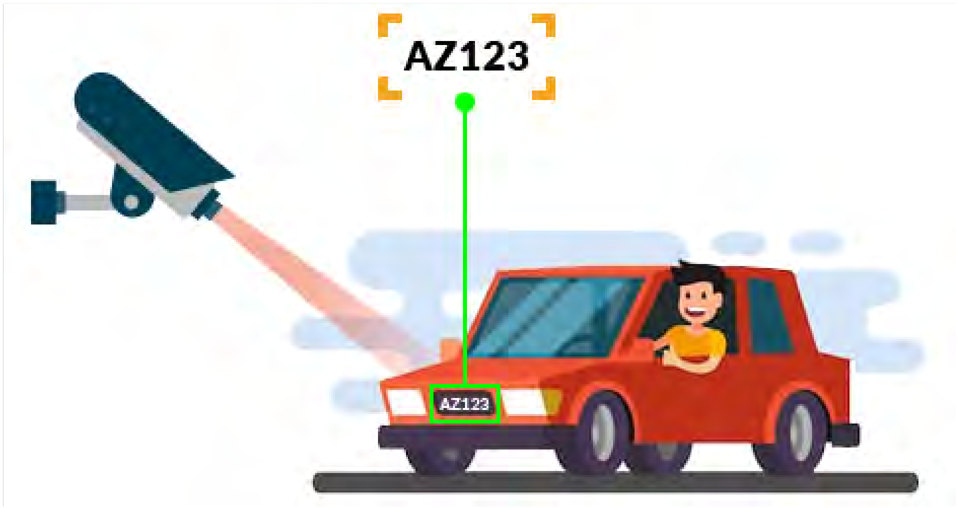


Figure 2.4: Pictorial process of ALPR technique as a whole [45]

In year 2019, solely according to the report of Bangladesh Police and BRTA, there is an occurrence of death in every two hours in Bangladesh [46]. And the main reasons behind this includes weak traffic management, uncontrolled speed of vehicles on roads etc. [47] Aside from this, ALPR is an algorithm used to monitor the vehicles' average speed and thereby identify the vehicles that exceed the speed limit [44].

It is quite unambiguous that the essences of mitigating the problems mentioned in the Introduction chapter of this research are mostly the reactive measures. But from the discussion at present it is comprehensible that ALPR can be used for proactive measures as well *e.g.*, monitoring speed of the vehicles as well as the license plate numbers. In undeniably truth that it is impossible for human being to hold on a speed radar gun throughout the day to monitor that is used for measuring instant speed of any vehicle [48]. Secondly, the judgment of human being will be somewhat erroneous for different factors which are complex rather than a software especially when number of vehicles passing through a section in every moment is beyond human aptitude to quantitate. Thirdly and most importantly all other misconducts are easy to prove whereas the over speed phenomena are almost impossible to establish without any recording-based technology.

While talking about recognizing vehicle and keeping its record, there are apparently two options widely used nowadays namely, ALPR and RFID. Comparison between ALPR and RFID, consequently the rationale of using ALPR in context of Bangladesh has been elaborated later on. The activities of ALPR can be divided into two major parts. First one is to capture close images of the vehicle license plates. The second one is the appropriate algorithm to process those license plate images and recognize the characters using OCR technique. First one mentioned is perceptibly the issue regarding camera and its resolution power. Second one is the concern of AI.

2.2.4. ALPR VS. RFID in context of Bangladesh

From the discussion made in the Introduction chapter it was quite transparent that though government of the people's republic of Bangladesh made ordinance not to ply any vehicle to move on the roadway without RR license plate and RFID tag [15], this process is still ongoing and car owners are not still yet to get visible benefits.

On the contrary, there is a vast number of vehicles seen on the roadway moving without RR license plates even at the outset of the year 2020 [49]. This clearly indicates that those vehicles do not contain any RFID tags as well. Apart from this, some analytical discussions must be made between RFID and ALPR in context of Bangladesh to establish the rationale of using ALPR technique.

It is inevitable statement that accuracy of RFID is greater in comparison with ALPR technique. But this RFID tag is issued by a government agency *e.g.*, BRTA in context of Bangladesh, data of which is kept private and generally private agencies do not get access of those.

Therefore, almost all the time an RFID detector will have to be connected to the central server which will identify the vehicles by its frequency. On the other hand, although ALPR technique can be run offline to collect license plate numbers, in case of fake license plates it often fails to identify. Differences in the basis of provisioning costs, environmental impact etc. are shown in Table 2.1 as well. All comparisons will give result in a specific overview on these two techniques.

Table 2.1: Comparison between ALPR and RFID from Application Point of View

Basis of comparison	ALPR	RFID
Tools	Uses cameras based on OCR	Uses RFID tags as well as a reader
Provisioning cost	Lower provisioning cost [50]	Relatively high provisioning cost
Accuracy dependency	Dependent on a software algorithm and voting logic to determine likelihood of the OCR	Incorporates a method to prevent errors in data transmission between tag and reader [50]

Contd. Table 2.1

Basis of comparison	**ALPR**	**RFID**
Environmental impact	Fog, sun or vehicle glare; camera covered with dirt might affect [51]	Can function well even in adverse weather and dirt
Internet connection	Continuous internet connection is not a necessity	Continuous internet connection is a must
Handling access	Can be handled by any public or private agencies	Limited handling access due to sensitive information
Invalidity	Only if a vehicle has no license plate	Vehicle having no RFID tag despite having license plate
Tools per lane	Generally, one tool per lane	One tool handles two lanes

Comparative analysis between ALPR and RFID shows that both of those have some advantages over another. Especially in high secured area RFID might be suitable keeping an eye on the fact that it would instantly identify fake registrations. On the other hand, ALPR can be widely applicable in generalized perspective by all sort of agencies pretermitting the requirements of internet connection. But nowadays most of the experts recommend hybrid approach where ALPR and RFID solutions can be combined to good effect at the same entrance to prevent unauthorized vehicles and overcome the inaccuracies with ALPR [50]. Bangladesh government also followed this provision. It is also seen in Table 2.1 that the performance of ALPR is greatly dependent on the software algorithm. Thus, a rationale has been established to create appropriate algorithm to extract features of the license plates and recognize thereby.

2.3. Artificial intelligence

As discussed previously that successful operation of ALPR system entails OCR algorithm which must be undertaken using AI. Also, it is necessary to determine the suitable choice from AI containing multivariate options. AI is also known as Machine Intelligence (MI), "is a branch of computer science that aims to imbue software with the ability to analyze its environment using either predetermined rules and search algorithms, or pattern recognizing machine learning models, and then make decisions based on those analyses."[52] There is a misconception that AI is a system, but rather it is a method which is implemented in the system [53]. The inclusive aim of AI research is to establish technology that allows computers and machines to function properly in an intelligent manner. But the general problem of simulating intelligence has been broken down into sub-problems *e.g.*, reasoning, problem solving, knowledge representation, planning, machine learning, perception etc. [54]

4.2.1. Machine learning

As mentioned previously that AI problems can be further divided into sub-problems. One of those sub-problems is Machine Learning (ML). The formal definition lies in: "a technique that figures out the "model" out of "data"." [55] The data referred here literally means information such as documents, audio, images, etc. Besides, the model mentioned in the definition is the final product of ML. If focused on the term that contains learning indicates the data that this process used during learning. The process can be further simplified as to take training data then ML and generate model [see Figure 2.5]. After model has been generated one can get output after giving some input value to this model.

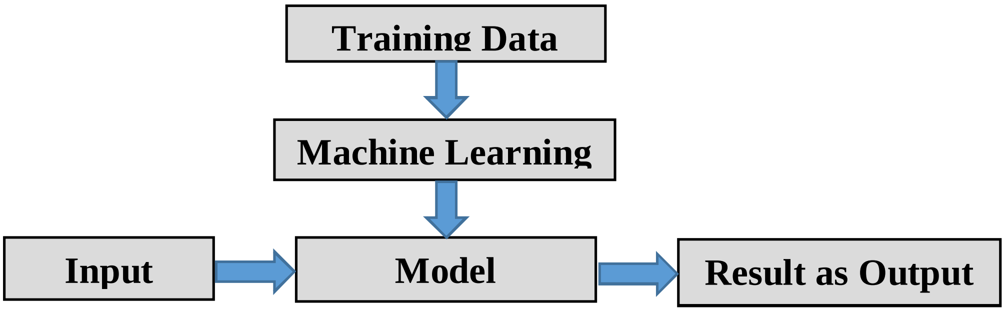


Figure 2.5: Exclusive process of machine learning

Learning techniques of ML can be further divided into three major parts namely, supervised learning, unsupervised learning and reinforcement learning. The first category *i.e.*, supervised learning process is somewhat similar to the human learning process. Always input as well as correct output is rendered to train machine in this case. On the other hand, no output is mentioned in case of unsupervised learning. In addition to that, another category where grades and some outputs are given with inputs, that is called reinforcement learning. Referring to the research scope that the algorithm will have to recognize the characters, so there is no chance of admitting unsupervised or reinforcement learnings. Variation in different ML technique lies in the process it uses to learn from the training data to generate model. Hence, Neural Network (NN) is widely used as the model for ML which will be discussed subsequently.

2.3.2. Artificial neural network

As mentioned from the beginning of the research that the model should have some scope to adopt human like reasoning, ANN could be a perfect match for that. NN in case of artificial neurons called ANN or Simulated Neural Network (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation [56]. As this is a type of ML, the same process is valid for ANN as well [see Figure 2.5]. Only the difference is “Model” in the process will be replaced by “Neural Network” in this case. Similarities between NN and human brain are depicted in the Table 2.2. It is observed that the mechanism of brain is imitated by the neural network.

Table 2.2: Similarities between Brain and Neural Network

Basis of similarity	Brain	Neural Network
Unit	Neuron	Node
Transmission of data	Connection of Neuron	Connection weight

Depending upon the transmission of data mentioned in the Table 2, variety of NNs can be created. Additionally, depending upon the number of layers existing in the NN, it can be further classified into two types namely, single layered NN, multi-layered NN. Single layered NN consists of only inputs and outputs; in between these two there are multiple combinations possible. On the other hand, in multi layered NN, there should be at least one hidden layer in between inputs and outputs. In brief, it can be said that if there is absence of any hidden layer in between inputs and outputs [see Figure 2.6], it will turn into single layered NN, otherwise multi layered.

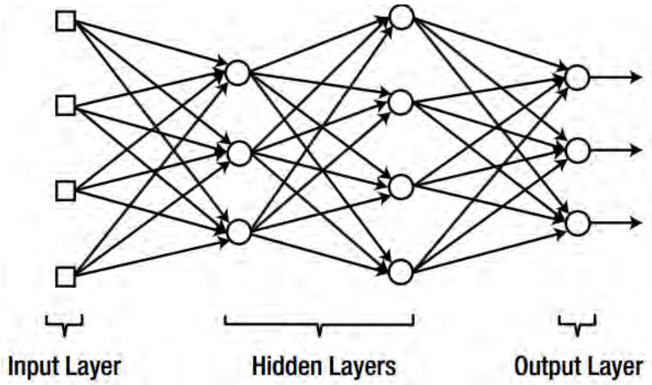


Figure 2.6: A layered structure of nodes [55]

In addition to the types as mentioned above, NN can be further divided into two types based on classification methods namely, binary classification, multiclass classification [see Figure 2.7]. If each input of the NN can be classified into two categories, then it is called binary classification; if it is more than two then called multiclass classification.

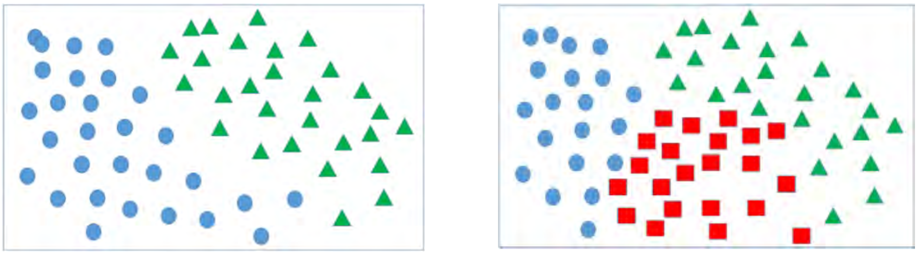


Figure 2.7: Types of NN; (a) binary classification and (b) multiclass classification

2.3.3. Fuzzy inference system

In the Introduction chapter for referring decision making according to the human like reasoning, illustration of fuzzy logic has been introduced. To construe further, it can be said that it is a computational intelligent technique that computes output based on the fuzzy inference rules and present inputs [57]. The intelligent technique mentioned is based on fuzzy logic as described earlier [19]. FIS is advanced by five functional blocks [see Figure 2.8] namely, rule base, database, decision making unit, Fuzzyfication inference unit, Defuzzification inference unit.

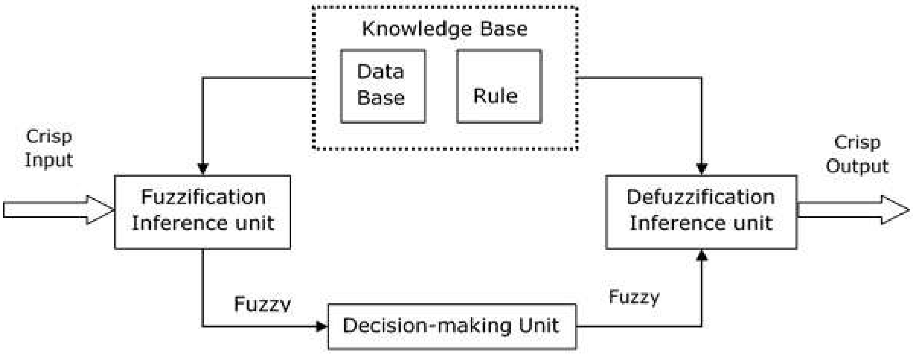


Figure 2.8: Pictorial process of FIS technique as a whole [58]

Fuzzification inference unit is used to bear the inputs to the decision-making domain using database as well as rule base and finally create fuzzy outputs. Also, these contribute to the Defuzzification inference unit to generate non fuzzy outputs afterwards [59]. Based on the rules applied and output obtained in the FIS, it can be further divided into two major parts namely, Mamdani FIS and Takagi-Sugeno FIS. The main difference between these two types would be the output feature, where Takagi-Sugeno FIS output membership functions (MFs) are either linear or constant. Rules *i.e.*, MFs of FIS have to be defined by the user himself, according to which the inputs are processed to get desired output. MFs will be illustrated in the Methodology and Data Collection chapter later on.

2.3.4. Adaptive neuro fuzzy inference system

As discussed earlier about ANN and FIS, it is seen that ANN gives a structure of nodes similar to the NN but no definite pattern or algorithm for modeling is defined. On the other hand, in FIS techniques for classification and creating outputs are defined by the user. But in real-life scenario *e.g.,* for detection of image attributes, it is very tough task for defining all the membership functions and parameters for decision making. Therefore, a hybrid learning algorithm ANFIS is derived which uses both FIS and ANN to create models introduced by Jang [60]. MF parameters of ANFIS are tuned (adjusted) using either a back-propagation algorithm alone or in combination with a least squares type of method as depicted by Billah et al. [61]. As parameters are tuned using ANFIS, therefore the only need is to define a pattern for the algorithm, remaining task will be performed by the inference system.

As discussed for ANN, multilayered ANN is involved in ANFIS modeling where FIS acts as the hidden layers [see Figure 2.6] and in combination with ANN generates the final output [see Figure 2.9]. As Sugeno type is a more compact and computationally efficient representation than a Mamdani system, it gives itself to the use of adaptive techniques for constructing fuzzy models [62]. In line with this it can be anticipated that this Sugeno type of ANFIS would be appropriate for the analysis of image attributes of the characters to be recognized which will be discussed in the Methodology and Data Collection chapter. ANFIS performance is operated by stepwise algorithm of which the structure comprises of five layers namely, Fuzzification, Fuzzy AND, Normalization, Defuzzification and output layer [see Figure 2.9]. The first step of ANFIS is the identification of the input and output variables [57]. For illustration, first order Sugeno model and two typical if-then fuzzy rules with a set of two variables x and y can be considered [63]. Output of the model is obtained as f which uses variables in combination with three coefficients namely, p, q and r.

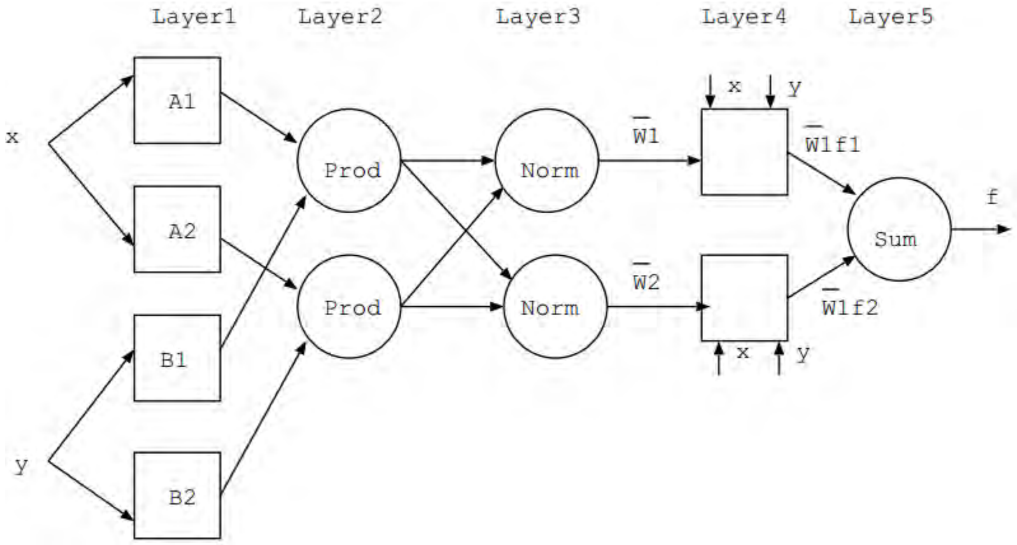


Figure 2.9: ANFIS architecture involving all layers for two inputs x and y

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$ (2.1)

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$ (2.2)

All nodes in in first layer which is also known as input layer are adaptive. For layer 1, $O_{1,i}$ is the output of the ith node of this layer. The out put can be stated as

$$O_{1,i} = \mu A_i(x) \text{ for } i = 1,2 \text{ or} \quad (2.3)$$

$$O_{1,i} = \mu B_{i-2}(y) \text{ for } i = 3,4 \quad (2.4)$$

x(or y) is the input node i and A_i (or B_{i-2}) is a linguistic label associated with this node. Therefore, $O_{1.i}$ is the membership grade of fuzzy set (A_1, A_2, B_1, B_2). Typical MF can be defined as

$$\mu_A(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b_i}} \quad (2.5)$$

Here, a_i, b_i *and* c_i is the parameter set which are referred to as premise parameters.

In layer 2, every node is a fixed node labeled as Prod. The output is the product of all the incoming signals. Thereby the output will be

$$O_{2,i} = \omega_i = \mu A_i(x) \,.\, \mu B_i(y) \text{ for } i = 1,2 \quad (2.6)$$

Each node i in Equation (2.6) represents the fire strength of the rule. Any other T-norm operator can be used that can perform the AND operator.

In layer 3, every node is a fixed node labeled as Norm. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rule's firing strengths. The outputs are called normalized firing strengths and output of ith node is which represented by

$$O_{3,i} = \overline{\omega_i} = \frac{\omega_i}{\omega_1+\omega_2} \text{ for } i = 1,2 \tag{2.7}$$

Here, $\overline{\omega_i}$ is called the normalized firing strength that will be used in the next layer.

In layer 4, every node is an adaptive node with a node function

$$O_{4,i} = \overline{\omega_i} f_i = \overline{\omega_i}(p_i x + q_i y + r_i) \text{ for } i = 1,2 \tag{2.8}$$

$\{p_i, q_i, r_i\}$ is called the parameter set of this node. These are denoted as consequent parameters.

In layer 5, only one output is obtained therefore only a single node that is a fixed node labelled as sum is generated. It computes the summation of all incoming signals as

$$O_{5,i} = \sum_i \overline{\omega_i} f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \text{ for } i = 1,2 \tag{2.9}$$

2.4. Image processing

As designated earlier in the chapter that for ALPR there are some issues regarding image quality, therefore processing of the image is a prerequisite. Beyond this, there is a requirement of any image to be processed in a convenient manner for AI to be applied on that. Thus, Image Processing (IP) can be defined as a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it [64]. IP is basically performed through three major steps namely, importing image, manipulating image through analyzing and exporting output image [see Figure 2.10].

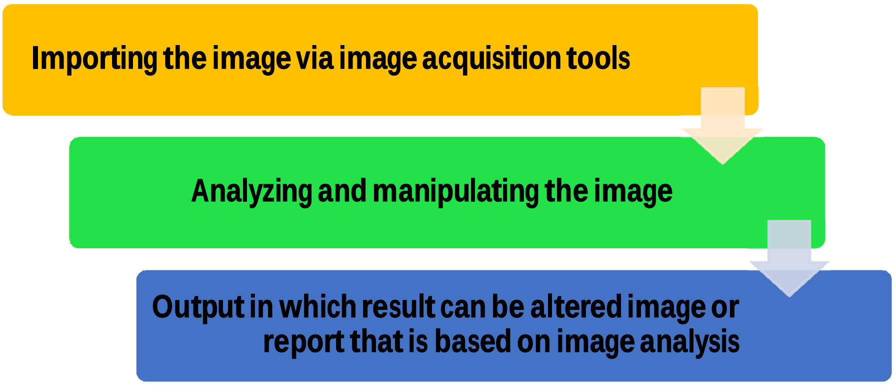

Importing the image via image acquisition tools
Analyzing and manipulating the image
Output in which result can be altered image or report that is based on image analysis

Figure 2.10: Basic steps to operate IP

There are two types of method used for IP namely, analogue IP, digital IP. Analogue IP is performed in case of printouts and photographs. On the other hand, digital IP is performed on digital images through computer [see Figure 2.11]. It is manifest that LPR of any vehicle must be done from the digital license plate image of that vehicle. IP has become one of the most burgeoning technologies in recent times of which modern trend will be discussed through reviewing recent literatures in this chapter later on. Based on the discussion, methodology will be set for extracting necessary features for the research being conducted.

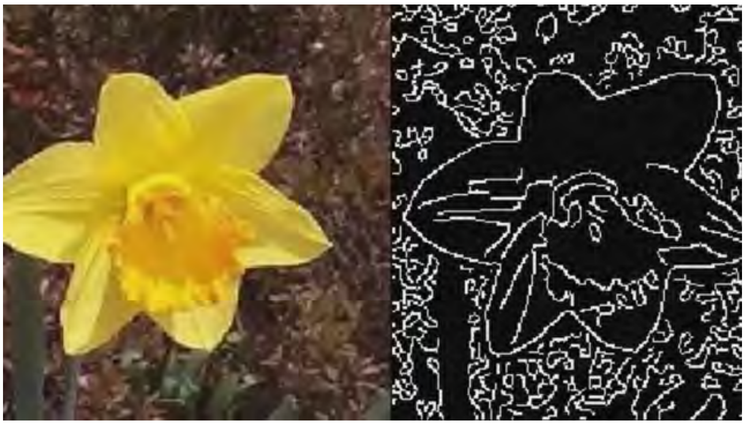


Figure 2.11: Example of digital IP [65]

2.5. Relevant ANFIS researches available globally

From the discussion made so far in this chapter, it can be understood that application of AI especially ANFIS would have potential application in Transportation Engineering especially Traffic Engineering if it is dealt with human like reasoning. The demonstration would be undertaken now with proper examples. After the discussion, some scopes for adapting these researches with LPR system in an integrated manner will be concluded.

One of the most popular field of traffic engineering study is the car following phenomenon. There are several theories and models regarding car following accordingly. Fuzzy logic-based models are included in a category of one of those theories. ANFIS modeling successfully on car following phenomenon for heterogeneous traffic was conducted by Imran [66]. Further approach was made to model real-life traffic based on instantaneous reaction delay by Khodayari et al. [67]. But there were some problems in classification algorithm. Further approach was made by Moghadam et al. [68] aiding ANFIS through Classification and Regression Tree (CART) algorithm to model real-life traffic.

After car following theories, the most popular field of traffic engineering study is the lane changing phenomenon. Alike car following phenomenon, lane changing is a decision-making process by human brain as well. Therefore, successful applications of ANFIS have been done for lane changing models. Ma [69] might be the one among very few founding applicants by whom integrating car following and lane changing behavior using FIS was performed. However, main limitation was the applicability of the model in heterogeneous and non-lane based condition. Afterwards, fruitful approach of ANFIS for lane changing decisions in heterogeneous condition which is a complex phenomenon was made by Sarker [70]. However, car following approach was upgraded by Ghaffari et al. [71] from their earlier one [67] by adapting lane changing behavior using anticipation and evaluation idea afterwards.

As discussed above, it is observed that whether the car following or lane changing everything depends upon the driver behavior itself. Therefore, it is conjectured that tracing driver behavior might act as an aid for the existing models of car following and lane changing phenomena. Keeping an eye on this fact, an approach was made by Ghaffari et al. [72] for overtaking maneuver behavior modeling using ANFIS. This research however was not considering the common behavior of the drivers in normal condition. An approach for mitigating this limitation was made by Feng et al. [73] where driving style was modeled using real driving data. Yet again there was not enough attributes to trace driving behavior for non-lane based heterogeneous traffics.

Another major and probably the most welcomed research items nowadays are based on road accident. For road accidents not driver behavior but also a number of other factors are responsible. Hence, combination of all the factors should be done to make proper approaches towards accidents. Fuzzy neural network was used by Zheng et al. [74] for predicting road accident frequencies. However, adaptation was not fair enough for that model. After that an approach was made by Hosseinpour et al. [75] for predicting accidents using ANFIS. Besides, data-based modeling of vehicle crash was performed by Zhao et al. [76]. Not only ANFIS based prediction but also detection approach by analyzing and tracking vehicular movement was made by Arul [77].

Microscopic parameters can be combined to trace macroscopic parameters *e.g.,* driving behavior in a row. Relationships among macroscopic parameters *i.e.,* fundamental diagrams of traffic flow can also be evaluated using ANFIS of which approach was made by Hadiuzzaman et al. [78]. Not only this but also complete transportation system can be decorated commonly as ITS through the road accidents, vehicle tracking etc. as discussed earlier. One efficacious paradigm was performed by Taghavifard et al. [79] who established an intelligent system for prioritizing emergency service for the injured person in an accident.

Not only road accident but also other communication network problems can be solved using ITS via FIS, the approach was made by Askerbeyli et al. [80]. Moreover, effects FIS on asynchronous transfer mode was also attempted by Askerbeyli et al. [81]. One of the most important activity for the vehicles in the roadway *i.e.,* sign detection along with recognizing was tested using ANFIS by Billah et al. [82]. Though it has less applicability in Bangladesh due to manual traffic signaling provision. Needless to say, all these approaches are made on the basis of homogeneous non-lane based condition where abnormalities are mostly absent.

According to the discussion made on traffic signal control in [82], more researches are available relating it to the traffic congestion. This topic is an utmost concern of Bangladesh as described in the Introduction chapter. Bingham [83] might be the first one by whom FIS for traffic signal control was applied. Though it was the introductory item of FIS pertaining to the traffic signal, incident detector was not included in it. Fuzzy discrete dynamic system was applied by Adamski et al. [84] for traffic congestion and incident detector. Being advanced on how the process of traffic signal acts, fuzzy based intelligent traffic control system was performed by Sadeghi [85]. These researches were however made on the basis on any particular model. Multi model criteria for estimating traffic congestion was used by Shankar et al. [86] from traffic flow information based on fuzzy logic. Although it was estimating congestion in quite compressive manner, there was not provision for management of traffics. Therefore, a successful approach was made by Kusumawardani [87] in utilizing Active Queue Management (AQM) and ANFIS as intranet traffic control. However, more realistic approach was made by Udofia et al. [88] to create dynamic traffic signal phase sequencing for an isolated intersection using ANFIS. One of the major limitations of that study was the poor performance for multilane behavior. An overcoming approach was made by Lai et al. [89] for controlling traffic flow in multilane isolated intersection using ANFIS. Along with these researches, the applicability of fuzzy logic controller in a single intersection quested by Dereli et al. [90] where normal conditions do not prevail.

In addition to the research arenas discussed above, there are plenty of other topics related to traffic engineering available where ANFIS has been used in successful manner so far. Due the traffic congestion and signaling vehicles face delay, which might cost a lot. Therefore, modeling of delay was performed using ANFIS in signalized intersection by Gokdag et al. [91]. Also, the crossing behavior of the pedestrian in uncontrolled intersection which influence a lot in overall transportation network was trialed by Amin et al. [92] using ANFIS. Last but not the least important phenomenon in transportation network is vehicular exhaust emission especially in the urban roadway. This phenomenon was taken into consideration by Jain [93].

Summing up all the reviews from the discussion above, it can be well understood that ANFIS has been successfully implemented in transportation engineering especially in ITS related tasks over time. But at the same time, it is seen that none of the researchers used LPR system to integrate their research scopes eventually. Moreover, applications of ANFIS on LPR system should be reviewed to get an overview on the present condition of advancements which will be discussed after this. Scopes for assimilating license plate numbers in the ITS will be discussed in the conclusion chapter in this research.

2.6. Relevant researches available on LPR globally

From the discussion made earlier in this chapter it is well visible that there is hardly any field of transportation engineering and ITS where AI especially ANFIS has not been applied so far. Yet LPR was not considered in the above discussions. Hence, during this discussion relevant researches conducted on LPR will be reviewed and necessity of the research for Bangla LPR will be concluded in conventional manner.

As discussed at the very beginning of this chapter in [30] that ALPR was invented in 1976 due the urgency of traffic management and the gradual rise of automobile industries after WWII. Several researches have been conducted so far for making it more effective to implement. An approach was made by Martinsky [94] to establish some rational algorithmic and mathematical principles of ALPR. There all the principles were considering the license plate to be written in English. An approach was made by Khalifa et al. [95] using ANN for detecting Malaysian

LPR. This is to be noted that Malaysian vehicles are number using English numerals. Attempt was made by Nagare [96] for LPR using neural network in context of India. Also, the existing license plates in India are also written in English.

However, a big problem in ALPR as discussed earlier in [50] that the accuracy of detection. Same thing faced in UK and this sort of real-life problems in detecting ALPR was studied by Rhead et al. [97]. Later on, a review was studied by Gaikwad et al. [98] on the ANPR system considering some limitations. Finally, approach was made by Biswas et al. [99] for the enhancement of ALPR that was based on ANN. All the researches discussed so far was acquitted unless the fact that those approaches were made for the standstill images of the license plates. Therefore, an efficient algorithm for real-time license plate localization was studied by Reji et al. [100]. With the development of LPR technologies throughout the world, an attempt was made for ALPR designated in Arabic fonts by El-Harby et al. [101].

All the ALPR researches conducted discussed so far based on some IP algorithms which varies from region to region according to the convenience of researchers' approaches. One noteworthy fact for the researches mentioned earlier was the software used for LPR. It can be observed that almost all the researchers have used MATLAB, a renowned toolkit for IP and AI algorithms. Rationale of using MATLAB will be further illustrated in the Methodology and Data Collection chapter later on. To give more examples of ALPR using MATLAB, research work of Ibrahim et al. [102] can be shown where they experimented data using MATLAB in case of Malaysia as a review of previous research work in [95]. There are research works available using OCR in MATLAB by Simin et al. [103]. One thing to be remembered here that, MATLAB provides built-in OCR system for English fonts.

In addition to the above researches, tracking of vehicle number plates using MATLAB was studied by Manisha et al. [104] whereas neural network classifier and KMEAN were used by Kaur et al. [105] via MATLAB for ANPR. Even in Nigeria there are relevant researches conducted using MATLAB by Chibuisi et al. [106] as well as Amusan et al. [107]. Also, in Nigerian vehicle license plates English fonts are used as well similar to the Indian license plates conducted by Tiwari et al. [108]. A smart technique was proposed by Chhikara et al. [109] for the accurate identification of license plates using MATLAB. In most of the researches there are complexities pertaining to the algorithm used in MATLAB. Therefore, an attempt was made by Stefanovic et al. [110]. Besides, other countries of the world are choosing MATLAB for ALPR of their vehicle license plates written in their native language and format *e.g.*, Iraqi ALPR studied by Omran et al. [111].

2.7. LPR researches in context of Bangladesh

From the discussion made earlier, it is quite clear that ALPR technique has been studied so far throughout the world where popular toolkit for that is MATLAB. Research items in Bangladesh are inextricably linked with those researches by some means. Although, some researches regarding ALPR especially Bangla license plates have been conducted so far, almost all the researches have some shortcomings either in license plate perspective or in algorithms used for the detection.

License plate detection methodology in Bangla was proposed by Azam [112], where IP was done thoroughly. But the method remained deprived of being used due the advent of RR digital license plate in Bangladesh as discussed in the very beginning of the chapter. On the contrary that study considered the analogue license plate format [see Figure 2.12]. Moreover, in that research a methodology was established to identify the license plate not to read the texts.

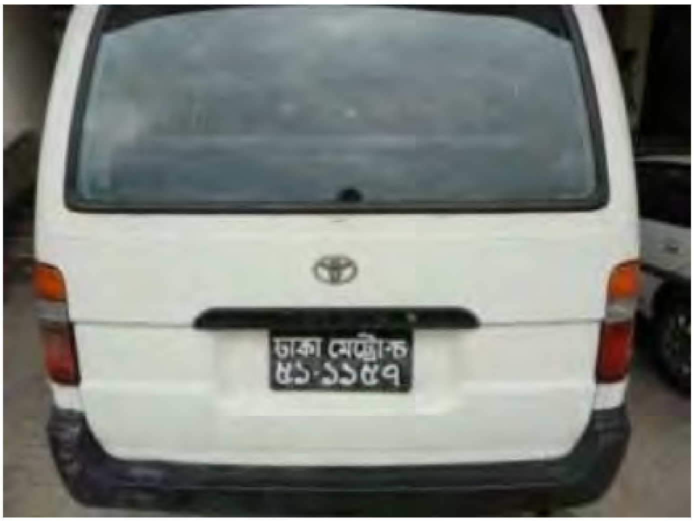


Figure 2.12: Analogue license plate used in BD before the advent of RR digital license plate [112]

Earlier, a Bangla OCR system for the printed characters using neural network approach was proposed by Alam et al. [113]. The paper was however a guideline for scrutinized analysis of Bangla characters. But the main problem was the inability to accept variations due to skewness or fading by some means. Away from this, nowadays some Bangla text recognition studies can be found where deep Convolutional Neural Network (CNN) were used. Deep learning for handwritten Bangla digit recognition was proposed by Alom et al. [114]. One of the attributes of deep learning is stated as no algorithm is required to specify what to do with the data. The problem arises at the same time when model is being trained, because no specific directions are provided for training [115]. At the same time, a large and unbiased dataset was prerequisite for the study of Bangla digits conducted by Shawon et al. [116].

Besides, most recent study conducted by Hoq et al. [117] perceived that ANN performed best. ANPR was also investigated by Shahed et al. [118] using MATLAB but not ANFIS. One good sign of avoiding complexities in that research was to use Connected Component Analysis (CCA). Though in Bangla language, a word is also considered as a connected component if CCA is conducted, by it has a good scope in ALPR system. More advantages of CCA will be discussed in the Methodology and Data Collection chapter later on. However, that research used 2D correlation coefficient matching with the images stored in the database for the identification of characters. For a particular input character, the 2D correlation coefficient of which image stored in the database will give least value will be the output of that character. The main limitation with this method is that it will by any means always give some output values stored in the database. If the desired character is not stored in the database, this algorithm will not inform the user. Rather it will be giving wrong outputs all the time. Besides, if the characters are titled by some means, then it might start performing abnormally. So, there must be some borderlines using which if sufficient match of the characters does not apply then it will not result in anomalous outputs.

Therefore, a well-defined supervised leaning algorithm *e.g.,* ANFIS must be taken into consideration for the desired outcome. ANFIS was used for character recognition quite successfully by Billah et al. [61]. But that study was conducted for English texts, not Bangla. Also, algorithm should be able to acclimatize the abnormalities in images due to shockwave [119] within the non-lane based traffic in Bangladesh. Henceforth, it can be concluded that there is an upmost necessity of a research to be conducted for ALPR in Bangla using ANFIS.

2.8. Overview

During inauguration of the chapter, it was pledged to arrange necessary information towards the gradual development of LPR techniques and ANFIS that have been established by this time. Origination of IP techniques has been explored as well. Moreover, a recapitulation on researches pertaining to ANFIS and ALPR conducted so far has been highlighted too. It was found that a wide range of researches using ANFIS for not only LPR but also every field of transportation is available. But at the same time there are very rare items available for recognizing license plates written in Bangla language. Beyond these, the limitations of the available researches for Bangla text recognition are also brought to light. Thereby the exigency of an algorithm for LPR in Bangla language using ANFIS has been established precisely that will be started from the next chapter.

CHAPTER 3
METHODOLOGY AND DATA COLLECTION

3.1. Introduction

The necessity of the ALPR system in Bangla was ascertained by the discussions made in the earlier chapters, which has paved the way for methodology and data collection. Therefore, specific methodology for conducting the research will be discussed in this chapter. Besides, appropriate software will be finalized for the research to be conducted after discussing pros and cons of the available software. It will be considered that both IP and AI algorithm will have to be used using single software. Therefore, whichever software will be chosen that should go in line with both of these applications. Also, data collection process of the vehicle license plates will be discussed after illustrating all possible methods in this chapter.

3.2. Review of AI and available software

It was seen from the discussions of AI in the previous chapter that ML is a subset of AI. For further getting into AI, specifically ANN was taken into account earlier. From the analogy now on, all other popular aspects of AI will be overviewed and relative advantages of one over another will be called into attention as well. After that, the finalization of the desired procedure will be done. It was mentioned in the introductory chapter that an algorithm will have to be developed which will accord with both IP and AI. However, IP techniques will be illustrated in the Data Analysis and Model Development chapter later on. But determination of the appropriate software will be done in this chapter after analyzing pros and cons of the available toolkits based on the convenience of applying IP and nominated AI at the same time.

3.2.1. Benefits of ANFIS

In the meantime, wide applicability of ANFIS has been enunciated in the earlier chapter. “Fuzzy logic has been accepted as an emerging technique during the past decades because of a wide range of successful applications, such as in industrial processes and automotive applications,” [66] statement by Imran still seems to be obvious after one decade. So far major aspects of ANFIS application have been prominent in the field of car following models, lane changing models, driver’s behavior, road accidents, evaluation of Microscopic and Macroscopic parameters of traffics, ITS, signal control, road sign detection, passenger movement, energy consumption etc. including ALPR studies. In all those cases, ANFIS was not used to quantify subtle behavior, rather ANFIS was modeled to generate human like reasoning for decision making. Needless of saying that human like reasoning infers anything to be judgmental with overlapping regions of decisions rather than being so much absolute for any particular value. Worthy of adding that fuzzy logic was developed to accept those variations in the judgments from person to person. This concept was combined with ANN to create FIS which further modified to a hybrid learning approach namely ANFIS later on.

Plenty of benefits of using ANFIS especially for ALPR system can be drawn. Especially for supervised learning, it can be used as a powerful tool. Some exclusive features of ANFIS in regard to ALPR system can be stated as follows:

i. The use of fuzzy logic will be able to interpret the vagueness of perception [66] of various inputs associated with license plates from the point of view of the observer *i.e.*, the cop, the driver, the manufacturer of the license plate etc. Thereby it will be able to represent the human like detection which is prevalent in realistic events.

ii. Non-numeric values can be utilized as both inputs and outputs in case of ANFIS. Nonetheless, only outputs will be considered as non-numerical for this particular study which will be specified in the Data Analysis and Model Development chapter later on.

iii. A specific range of ANFIS output values can be set as any desired output of ALPR, other values than that, ANFIS outputs can be identified as unknown texts. It aids this supervised learning method as an advantage over the study conducted by Shahed et al. [118] where 2D correlation coefficient was used, therefore wrong output might be detected in case of unknown texts as well.

iv. Various options of MFs are available for ANFIS and capability to adapt the parameters automatically can be performed by ANFIS as well.

v. One of the important advantages of ANFIS is that it would take relatively less computing time compared to other available AI algorithms illustrated later on.

All the advantages drawn above acclaims ANFIS to be used for developing an ALPR system. Besides, comparative study of the remaining popular AI processes will glorify this decision afterwards.

3.2.2. Problems with other AI architectures

As stated in the Literature Review chapter about AI, the focus was given into ANN and ANFIS afterwards. But there are other types of AI as well. In addition to ANN, there is a choice of Recurrent Neural Network (RNN) which caters for the sequential information in the input data. It is basically an improvement of ANN, sometimes regarded under the ANN as a subcategory. In the architecture of RNN [see Figure 3.1], there are recurrent connections in the hidden layers. To be more specific, RNN is generally used to solve problems regarding time series data, text data, audio data etc. [120] and so on. Especially when next output is dependent or at least influenced by the previous output, then RNN performs at its best. ALPR in Bangla study is dependent upon IP, therefore it has less chance to deal with the RNN architecture. If the circumstances were any letter to be identified within a text which occurs after another, then RNN might have worked fine. Although, in case of present study RNN can be performed as well but due to the constraints of scope discussed in the Introduction chapter, it cannot be recommended here. Yet, scope for application of RNN into ALPR will be pointed out later on in the Conclusions and Recommendations chapter.

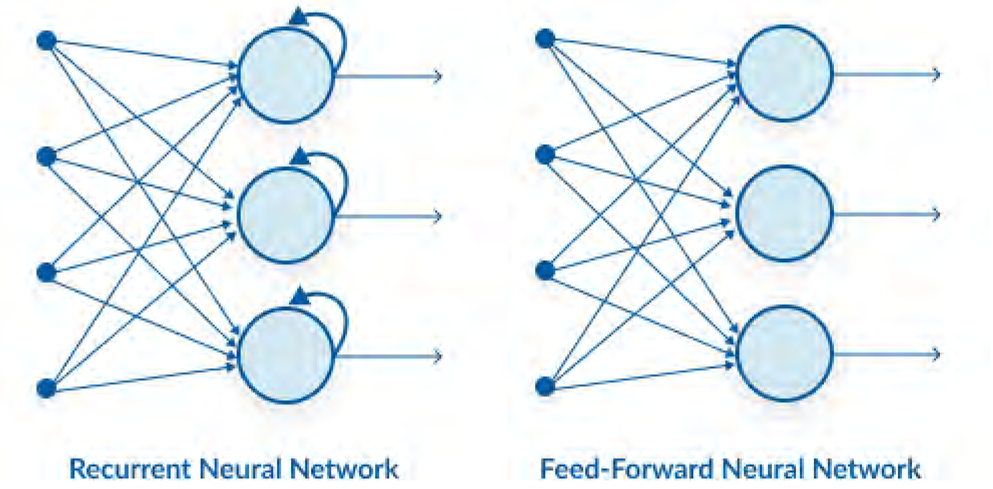


Figure 3.1: Comparison between RNN and conventional Feed-Forward ANN [120]

Furthermore, there is another algorithm available beyond ANN namely CNN, which has already been highlighted in [114] earlier in the Literature Review chapter. It is basically a subcategory of Deep Neural Network (DNN) which is commonly known as deep learning nowadays. Alike RNN, it is too considered as a modification of ANN and regarded under the ANN as a subcategory in some cases. Currently, algorithms regarding CNNs are all in the deep learning community right now [120]. But it is also inevitable that though CNN has been being used in different domains and applications, it is more prevalent in IP [see

Figure 3.2] and video processing projects. It has an advantage over other tools, as filters or feature extractions can be learnt using CNN automatically without mentioning it explicitly. Therefore, operation of feature extractions in CNN is integrated within the network. Therefore, at a time it is an advantage and sometimes a disadvantage as mentioned earlier in [115].

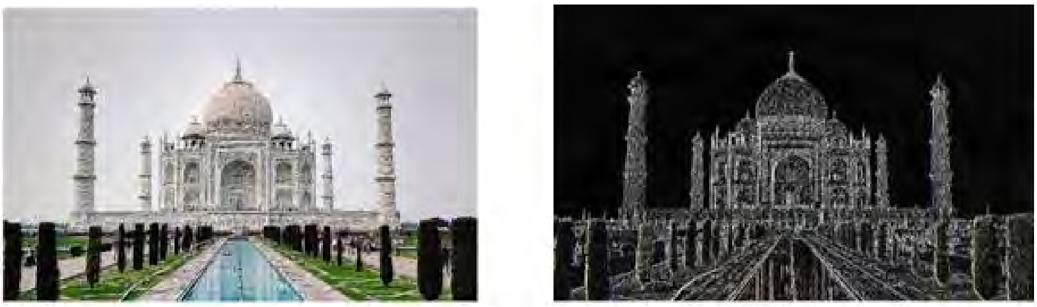


Figure 3.2: Output of CNN applied on an image [120]

As per the accuracy pursuit of CNN architecture, it is suitable for face recognition of human or quantifying subtle variations lying in different images. But it is no match for ALPR where identification of texts irrespective of the textures or fonts is expected. CNN might be the suitable one to perform if anyone wishes to identify handwritten variations of each license plate. But from the discussions made so far from the beginning of this research, it is well defined that the objective is to develop a model which will be able to recognize the numbers in the license plates not the textures or font styles as it was intended in [114]. On the contrary, it might be contended by the logic that there should not be any objection against identification of the texture in addition to the basic operation of ALPR. The counter of this rationality would be the prerequisites of substantial computing power, time as well as a large amount of labeled data [121]. Henceforth, CNN cannot be suitable for the model which will satisfy the need to record real-time data of numerous vehicles in the highway during ALPR.

Another popular algorithm of AI can be found as Probabilistic Neural Network (PNN). It is based on Bayes theory and was developed by Specht in 1990 [122]. It has an additional pattern layer in which input patterns are mapped in a number of class levels. Based on the best response among the class levels, the output of PNN is obtained. Albeit some advantages it has few disadvantages such as slow execution of the network [123] because of several layers and heavy memory requirements etc. Additionally, there is a problem similar to the [118] as where the output cannot exceed the predefined classes *i.e.*, any unknown characters will be adapted within one of the predefined classes as well. Therefore, application of PNN is found in the studies of Mowrin [57] and Malik [124] where there were less possibilities for the outputs to go beyond the predefined classes.

All of the comparisons of analogous AI algorithms can be summarized as a whole in the Table 3.1. It would reflect the fitness of the desired algorithm to be used in this research.

Table 3.1: Comparison among Analogous AI Algorithms

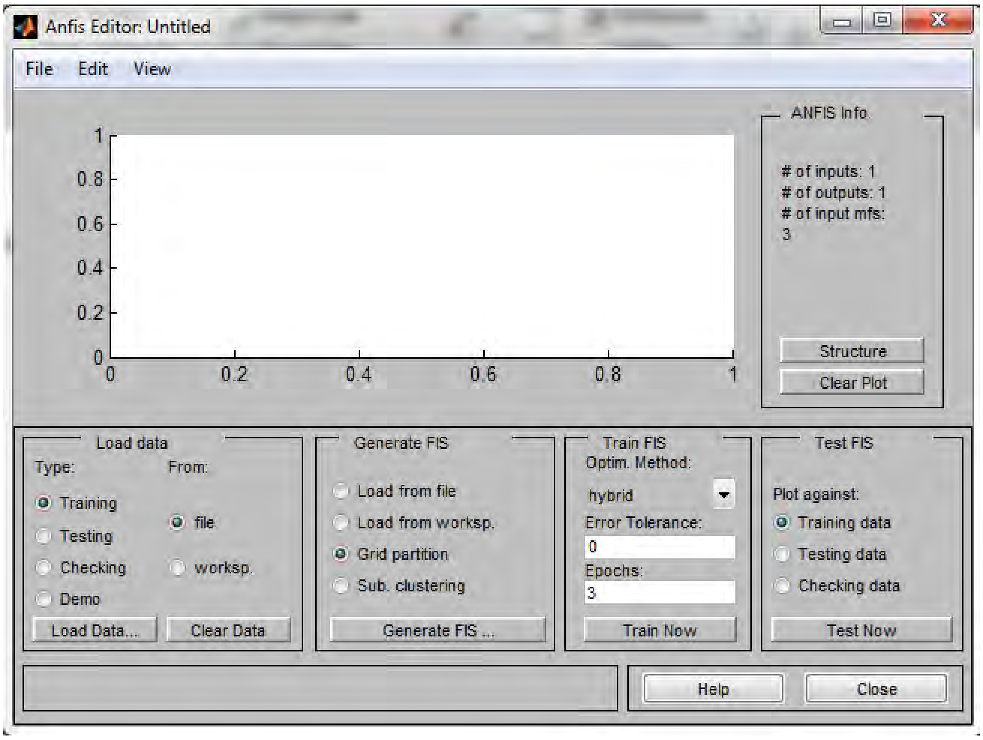
Basis of comparison	RNN	CNN	PNN	ANFIS
Sequential information	Yes	No	No	No
Execution time	Medium	Very High	High	Medium
Highly scrutinized data process	No	Yes	No	No
Binary features distinguishing	No	No	Yes	Yes
Image Processing	No	Yes	No	Yes
Unknown character detection	No	Yes	No	Yes
Data requirements	High	Very High	Less	Less

From the above comparison, it can be concluded that for the purpose of ALPR system, ANFIS is the best match. Hence, ANFIS will be used for the study.

3.2.3. Benefits of MATLAB

From the discussions so far, it is established that ANFIS will be used for the development of ALPR system. Additionally, IP will have to be operated to extract features as a prerequisite of ALPR through ANFIS. Keeping an eye on these facts, MATLAB software developed by MathWorks can be referred to be used for the overall model development. An excellent environment is provided by this software to develop ANFIS models. Moreover, it has a wide range of dedicated applications and aspects of using as a programming language. Not only ANFIS but also other algorithms of AI can be developed using MATLAB as well.

One of the important reasons behind this is the dedicated application within this software named “Neuro-Fuzzy Design” caters for ANFIS modeling. A user-friendly Graphical User Interface (GUI) [see Figure 3.3] is provided by this application also known as “ANFIS Editor”, using which ANFIS can be applied in a comprehensive manner. Besides, convention of implementing programming is still available for any feature. Moreover, MATLAB has a provision of accessing a code via online accounts, therefore no necessity of carrying pc or modules from one place to another. Once the modules are connected with the database, using online access it can be retrieved from any part of the world. Furthermore, “it includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook” commended by the developers itself [125]. In addition to that, smartphone application of MATLAB is also offered for free by MathWorks.



# of inputs: 1
# of outputs: 1
# of input mfs:
3

Load data Type: ● Training; From: ● file

Generate FIS: ● Grid partition

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 3

Plot against: ● Training data

Figure 3.3: GUI of ANFIS Editor in MATLAB

Apart from the ANFIS Editor, MATLAB has wide acceptability throughout the world for its built-in functions regarding IP. Multidimensional IP tasks such as image segmentation, image enhancement, noise reduction, geometric transformations, and image registration using deep learning as well as traditional image processing techniques etc. can be performed using MATLAB. Most importantly it would reduce complexities due to coding and would also reduce computing time of the device itself. Albeit a paid package, plenty of advantages can be diagrammed for MATLAB which will be thoroughly done after the reviewing shortcomings of the other available software afterwards.

3.2.4. Problems with other software

Considering the scope of the research, some other software can be brought to light for discussion. Due to the purpose of avoiding complexities, two popular software nowadays namely Python and R in terms of IP, AI application and statistical modeling among the engineers will be discussed here.

Python is a renowned programming software which was ranked as the number one programming site in 2019 [126] by the ranking site PopularitY of Programming Language index (PYPL). It was evident that Python alone occupied almost 30 percent share of the all available programming languages. One of the important reasons behind this worldwide popularity is that it has first class integration with C/C++ and therefore seamlessly offload the CPU heavy tasks to C/C++. As it is open source software, it has limitless modules made by different users as well to aid different programming tasks. This is in a sense, a major limitation of Python as well because it has to be dependent on third party frameworks [127] and libraries thereby. Moreover, it might be simple structured to use in any fields, but it is vulnerable to ANFIS modelling and image processing tasks. It is because there is no GUI ANFIS Editor available for Python, also there is fewer built-in functions for IP tasks.

Another famous programming language R, is a powerful statistical software. The main usage of this software is apparent in data science applications. Another benefit of R is that it can be paired with many other programming languages like Python, C, C++, Java etc. Various ML algorithms such as ANN can be performed using R as well. But at the same time, it should be mentioned that alike Python, R cannot provide any built-in GUI for ANFIS Editor as MATLAB does. Also, it is much slower than MATLAB and Python [128]. IP can be operated through R accordingly but again almost no built-in functions provided by R for IP.

In addition to the above facts, syntax of R is implicitly tough to be performed and also fewer provisions for accessing data from remote area through internet. All the discussions made so far can be tabulated in the Table 3.2 from judgmental aspects of suitability of the programming language to be used in the research.

Table 3.2: Comparison among Analogous Software for ANFIS and IP

Basis of comparison	Python	R	MATLAB
GUI of ANFIS Editor	No	No	Yes
Built-in Functions for Image Processing	Dependent on third party packages	Dependent on third party packages	Rich in built-in functions
Premium Access	Open source	Open source	Paid Software
Complexities in syntax	Complex	Complex	Comprehensive
Server host connectivity	No	No	Yes

Contd. Table 3.2

Basis of comparison	Python	R	MATLAB
Execution time	Less	High [128]	Less to Medium
Statistical Library	More dependent on third party packages	Rich and less dependent on third party packages	Relatively less efficient compared to R

It can be concluded from all aforementioned comparisons that MATLAB suits best for conducting ANFIS research for developing ALPR system. Also, it was observed in the Literature Review chapter that numerous researches including ALPR studies have been conducted so far using MATLAB. By this time, version of MATLAB R2020a is available as a premium package. But for fulfilling the objectives as mentioned in the initial chapters, any version after R2006a can be used. If higher versions are used, it would increase nothing but the time for execution. Moreover, one thing should be mentioned that from MATLAB R2019b, it warns that ANFIS Editor will be later on removed [129] by splitting it into “mamfis” and “sugfis” due to the extension of usage. These are two types of FIS *i.e.*, Mamdani type and Sugeno Type as mentioned in the Literature Review chapter. However, it will not create any complexity in executing the program as it would remain suggesting and bypassing the fact. But to avoid unnecessary warnings and time consumptions, MATLAB R2012b will be used to undertake the study and develop the model later on. Therefore, mention of MATLAB in this research will simply indicate MATLAB R2012b, unless specified.

3.3. Data collection methods

For the development of a model according to the objectives of the research and availing the scope of the study, collection of data must be done. In this particular research, the data would be the image samples of the license plates being used in Bangladesh. Before discussing the methods of data collection, convention of the license plates in Bangladesh must be illustrated. In Bangladesh, license plates are provided by BRTA which is the authorized government institution. As mentioned in both the Introduction and the Literature Review chapters that government has decided not to comply with analogue license plates [15] recently, RR license plates have become very common nowadays in Bangladesh albeit some exceptions. License plate orientation in Bangladesh can be defined as a rectangular flat metal plate [see Figure 3.4] containing combination of letters, symbols and numerals.

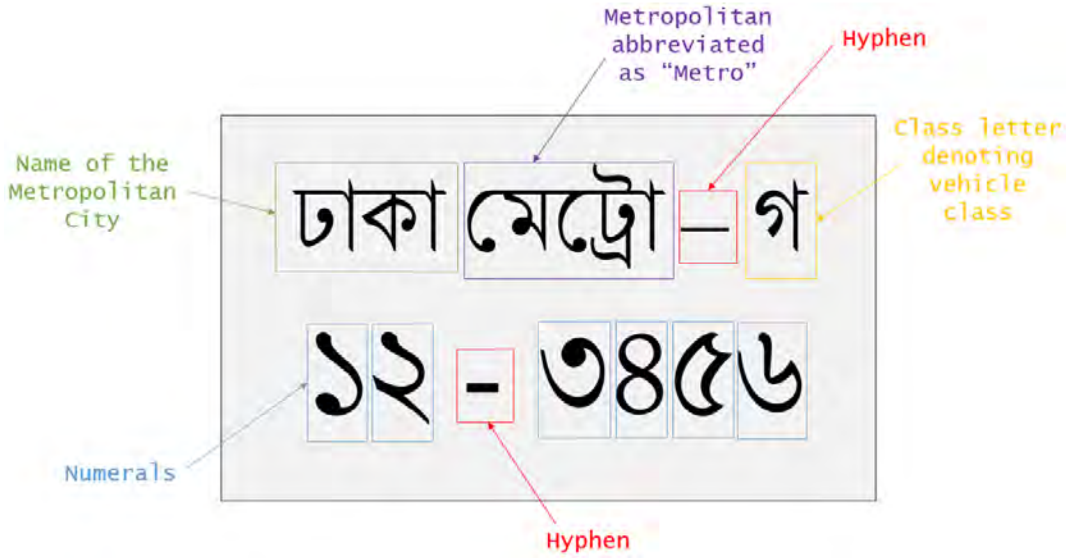

Metropolitan abbreviated as "Metro"
Hyphen
Class letter denoting vehicle class
Name of the Metropolitan City
ঢাকা মেট্রো – গ
১২ - ৩৪৫৬
Numerals
Hyphen

Figure 3.4: License plate orientation according to the guidelines of BRTA

There might be some confidential formats and designations used to numerate the license plates by BRTA. But some usual conventions are accumulated for ALPR considerations to develop model later on in the Data Analysis and Model Development chapter. Yet, violations of this rules are apparent in the existing scenario as mentioned earlier in [16]. Also, there are exceptions for diplomatic vehicles and others. The usual size of the rectangular license plate is 524 mm × 112 mm. The background color of the license plate is white with a black border visible around it. Within the license plate there are two rows of texts [see Figure 3.4] for representing the information about the vehicle.

First row of the vehicle is rendered with the name of the metropolitan city from the left under which the vehicle has been registered. After that metropolitan is abbreviated as "Metro" followed by a hyphen. Finally, at the rightmost side of the first row is filled with a Bangla alphabet representing the vehicle class. Second row of the license plate is used to represent the number of the vehicle therefore comprises total six numerals. First two numerals followed by the other four numerals separated with a hyphen in between these two. The registration number under the vehicle class is represented by the first two numerals whereas the vehicle number is represented by the remaining four numerals [41]. All these texts should be provided by RR according to the guidelines in [16] though many violations in real-life are apparent. Certainly, a particular font style along with a font size can be found after analyzing the license plates provided by the BRTA.

From the discussion made above, there are three possible ways to generate sample data for the research. First one is to consider the overall format (including font style and size) and regulations of BRTA to generate pseudo license plates using software. One benefit can be drawn from this method would be the privilege of generating unlimited data for analysis. But at the same time, a perfection of the license plates would be led by this method different from the real-life scenario, therefore most of the cases it would not work properly. It is due to the imperfection and lethargic attitude towards the regulations by the people of Bangladesh. Also, there are weather related issues found as well. Moreover, the research has almost no sensitivity towards the font style used in the license plates as ANFIS does not require definite position of orientation and angle of the characters as it is required for CNN studies discussed earlier.

Another method and a sophisticated one would be the collection of video data using a video recording device *i.e.,* a camera. The video data will be collected from some predefined location of the roadway facing the front side of the vehicles. So that license plates of the vehicles approaching towards camera can be focused and images of the license plates can be extracted afterwards. Advantage of this method is the coincidence with the real-life scenario where this algorithm is intended to be applied. But at the same time, it could create complications during collecting data, as elevated platform is required for that. Also, the desired images might not be found all the time due to the non-lane based behavior of the vehicles and the hindrance generated from the least gap acceptances. Also very high resolution camera, sometimes infrared camera is used in real-life ALPR system. This would be costly enough for the study to be conducted, though it would be relatively less time consuming for collecting data.

Another method is the conventional method of capturing images of different vehicles from the desired sites. Site selection procedure would follow some criteria which will be discussed later on. Albeit a very time-consuming method, the license plates used in the real-life can be represented truly here. Also, the desired quality and number of data can be warranted thereby. Also, relatively less resolution camera can be used for this method as images will be taken from a distance which will be fair enough. In addition to that, this method will be cost effective in an optimized manner.

Keeping an eye on all these issues, it can be concluded that standstill image captures using medium resolution camera will be the appropriate option for this research. Therefore, this conventional method will be used to collect data, specific considerations of which will be discussed later on.

3.4. Site selection for data collection

It is of utmost importance to select appropriate sites for collection of successful data for the analysis and model development. As established from the earlier discussions that standstill image captures will be taken as data, suitable sites with some predefined criteria will be established as well. Therefore, criteria will be enunciated followed by conduction of a reconnaissance survey for achieving those and finalization thereby.

3.4.1. Criteria for site selection

For successful extraction of license plate images from the captured images suitable sites should be selected. Some basic considerations of the model development should also be pointed out beforehand. Otherwise data will not be representative in manner aiding the algorithm of the model. Henceforth, nature of the vehicle types and license plates is diagramed in a tabulated manner in the Table 3.3. Based on the considerations drawn there, in quest of those vehicle classes and types, the suitable sites should be selected which will be discussed afterwards.

Table 3.3: License Plate Features Covered in the Research

Features	Specifications for the research
Language of the license plates	Bangla
Text denotation method on the plates	Both handwritten and printed
Font style considered	Any usual font style
Background of the license plate	White background with black borders
Vehicle type	Private cars and some motor bikes
Vehicle class	Medium (from 1001 to 2000 cc); denotation letters: “খ” (Kha), “গ” (Ga) and “হ” (Ha)
Metropolitan cities covered	“ঢাকা” (Dhaka), “চট্টগ্রাম” (Chittagong), “সিলেট” (Sylhet), “খুলনা” (Khulna) and “রাজশাহী” (Rajshahi)
Vehicle color	Any usual color
Condition of the license plates	Transparent as newly registered and also the slightly faded plates
License plate location	Only front sided license plates of the vehicles

From the illustration of the features of the license plates mentioned above, it can be concluded that motor cars commonly known as the private cars in Bangladesh will be considered for the model development in this research. This also specifies the scope of the study predicted earlier in the Introduction chapter. Now, based on the considerations made in Table 3.3, site selection criteria are enlisted below. According to these criteria, reconnaissance survey will be undertaken thereby.

i. The site should contain ample space to collect data *i.e.*, the images captured by the photographers.

ii. The site should be such that capturing images would not create any hampering in standard vehicle movements.

iii. The site should comprise enough data according to the aforementioned features and specifications in Table 3.3.

iv. The volume of traffic should not be considerably high enough so that the vision of the camera would not be able to capture the license plates.

v. Enough illumination should be rendered at the image capturing site so that the license plates are fairly detectable in the captured images.

vi. The site should be such in manner that would contain both the newly registered vehicles and the older vehicles. Therefore, the vulnerable or faded license plates will be included in the data.

vii. Lateral movements of vehicles such as overtaking and lane changing should be minimum at the site.

viii. The roadside of the site should be free from various side frictions such as pedestrian or vendor activities.

From the criteria set above, it is worthy of saying that hardly any site can be found in Bangladesh governing those criteria. Therefore, from practical point of view it might be necessary to compromise few criteria which will be decided after the reconnaissance survey is undertaken.

3.4.2. Reconnaissance survey

To accomplish the criteria pinched based on the license plate considerations, a reconnaissance survey was carried out. For the reconnaissance survey, primarily a number of sites were chosen in Dhaka city to check whether those satisfy the criteria or not. After thorough study from the survey, some potential sites in the Table 3.4 were sorted out on the basis of those criteria. Among those potential sites, decision will be made for final selection later on.

Table 3.4: Potential Sites Nominated from the Reconnaissance Survey

Site Location	Consideration behind nomination
Bangladesh Secretariat	Government registered vehicles will be found mostly
Kuril foot over bridge	Well elevated foot over bridge on the roadway which contains a large amount of motor cars
Motijheel bank hub	Most of the vehicles are parked during working hour outside on the roadway
Car parking of Bashundhara city complex	A renowned recreation place as well as shopping mall where cars can be found standstill
BRTA office	BRTA manufactured license plates can be found immediately in a comprehensive manner

One of the supplementary benefits of the reconnaissance survey would be the selection of probable image capturing locations at the time of reconnaissance survey conducted physically. Another benefit of this survey is to identify potential hazards while collecting data from practical point of view. It is because although some potentials and problems might be distinguished from macroscopic point of view, specific time and space dependent issues can only be identified after visual inspection of the site.

3.4.3. Final selection

On the basis of the reconnaissance survey, some potential sites were nominated in the Table 3.4 from which final selection will be made here. For that a comparative discussion is needed among those sites. First option shown there is the Bangladesh Secretariat which is situated in Paltan, a long-standing building from the pre-liberation period of Bangladesh. One of the main advantages of this site is almost all the vehicles here are from government officials. Therefore, it can be expected that regulations of RR license plates are properly followed those vehicles. Also it can be expected that, the features of the license plates mentioned in the Table 3.3 will be fulfilled there as well. In the meantime, a problem will be the sanctuary problem and everytime any car entering the secretariat might be hampered due to this data collection. Moreover, most of the cars existing there might be the newest cars rather than older cars. Therefore, some other sites could be recommended if found fulfilling more criteria.

The second option is the Kuril foot over bridge. The main reason behind nomitating this site is the well elevated foot over bridge from which data can be collected in continouos manner without hampering the movement of the vehicles. Also, it is expected that a large number of desired license plates mentioned earlier in the Table 3.3 will be found there as well. Also, movement of the vehicles in that roadway is in better level of service than most other roadways in Dhaka. On the contrary, there are some problems with this site as well. There are always side frictions in the adjacent region of that site. For this reason, speed of the vehicles reduces within the vicinity of that site reducing the headway of the vehicles. Eventually obstruction might be generated before the license plates because of the front vehicles. Moreover, requirement of high resolution camera will also be generated due to the elevated platform. Lane changing phenomenon might also occur during movement, it might lead to further complexities. Therefore in quest of a site, fulfilling further criteria might be quested.

The third option for the data collection site might be the Motijheel bank hub. One of the significant reason behind chosing this site is the ease of access to the vehicles. A large number of vehicles especially the cars as mentioned earlier in the Table 3.3 can be seen standstill outside on the roadway during the working hours. It is because of the scarcity of the parking space in this zone. The collection of data can be done in well manner without hampering the usual vehicle movements. Also combination of both the recently registered license plates and the license plates registered few years ago can be found there. Therefore, slightly faded license plated can also be found there. Also, no problem with the illumination will be there because of the daylight during working hours. As images will be captured from the adjacent locations, no issue regarding camera resolution will be provoked. The only issue there might be the efforts of the photographer. It is because a large area might have to be covered [see Figure 3.5] in approximation of 1.5 km^2. Despite this issue, other criteria are expected to be fulfilled in a comprehensive manner.

Fourth option for the data collection site is the car parking area of the Bashundhara city complex, a well renowned shopping mall in Dhaka city allowing a large number of vehicles there. It will provide facilities to collect data of different types of license plates according to the considerations. Also it would not hamper the movement of the usual traffic because it will be parked at the parking lot. Therefore, no issue regarding lateral movement or roadside frictions as well. It would mitigate efforts for collecting data by roaming around a huge area to some extents as discussed earlier in case of Motijheel bank hub. The problem lying in this site would be the lack of illumination, therefore quality of the images will not be according to the desired criteria. In regard of this issue, comparison with other sites must be done to chose the best one.

The final site nominated initially in the list is the BRTA office. In this case, data collection will be conducted through direct access to the license plate manufacturer of the BRTA authority. Therefore, original license plate features will be obtained. Eventually no objection will be aroused about data collection at all. Therefore, this might be the best option apparently. But there is another problem, the license plates will be found here will be RR license plates and all of those will be the brand new license plates. Although, some ancient samples might also be found on request. But according to the objective of the research, real-life scenario might not be represented properly followed by the model afterwards.

Keeping an eye on all these issues, finally Motijheel bank hub has been selected. In this regard, photographer will have to give more efforts roaming around the entire region [see Figure 3.5] during the working hours. Considering the features of the license plate as mentioned in the Table 3.3, image will have to be captured. All other criteria mentioned earlier for data collection are expected to be fulfilled in a vigorous manner. Starting from the Shapla Chattar, roadways towards western direction and southern direction will be covered in search of proper vehicles. It is expected that sufficient data of license plates can be accumulated through this site. Qualitative and quantitative features of the collected data will be discussed later on.

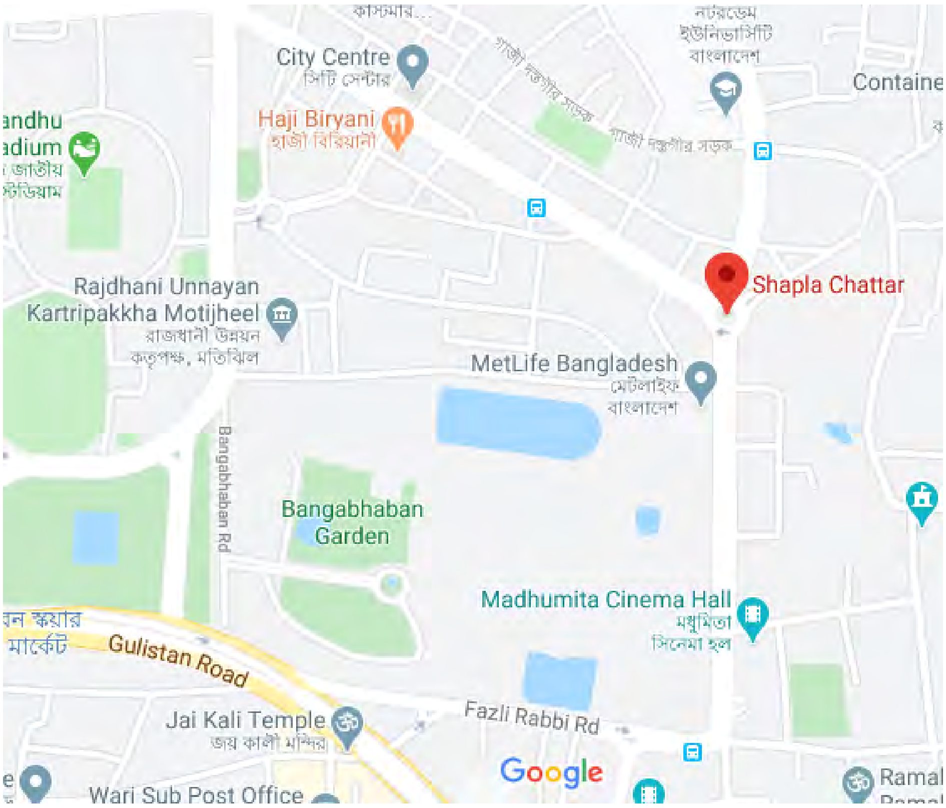


Figure 3.5: Satellite image of the Motijheel bank hub [130]

3.5. Collection of data

In order to develop the model and calibrate the parameters used in ANFIS modeling, collection of data has been done. As mentioned earlier that in this research, data would be the images of the license plates containing the features mentioned earlier in the Table 3.3. In this section, various details of the images of license plates collected at the finalized site declared earlier will be discussed. In addition to that, some auxiliary data *i.e.*, the images of license plates have been collected discussed later on. From the images of the license plates, characters such as letters, numbers, symbols etc. will be extracted and analyzed to train ANFIS modeler of MATLAB. Discussion on the extraction and analysis considerations will be made in the Model Development and Calibration chapter later on.

3.5.1. Standstill image captures

At the selected site *i.e.*, the Motijheel bank hub, extraction of raw data was carried out by the image capturing of different vehicles. During collecting data, some basic directions were given to the photographers about the selection of the vehicles, capturing angles and regions of the vehicles to be included with license plates in the capture. Directions were not too complex for the photographers which might limit number of captures. It was assumed that the site will contain representative amount of the desired vehicle types. Therefore, even wild collection of photographs from the site might produce representative samples for the model development as well. A camera having a resolution power of at 12 mega pixels was used for data collection. The captured images were in Red, Green and Blue (RGB) format.

One day might not cover the desired number of data, therefore the data collection was performed in the two consecutive working days. Although it was not necessary for the ANFIS modeling to collect numerous data as mentioned earlier, increased number of data would add to the value of the study. Time range for data collection was set as 11 AM to 1 AM, considering the working hours. Also, two days was fixed following the weather forecasts so that no unwanted situation might arise. As mentioned during reconnaissance survey that, during working hour proper illumination can be ensured, it worked quite perfectly for the site. It almost hampered no movement as vehicles are already parked outside on the roadway.

As the site region is an open environment, therefore possibility of coinciding same vehicle license plates was very high. Two photographers might have done the same thing in one day but due to the high probability of coinciding data, same photographer was appointed for the two consecutive working days. Question might arise in this regard that, there are possibilities for the same vehicles parked in the same place in the two consecutive days. The justified reply would be that, there are rooms for the officials to park their vehicles within their official parking lots. Example of 37 storied City center [131] in Dhaka can be provided which is situated in the concerned site which can accommodate enough vehicles in their dedicated parking lot. Therefore, it can be stated undoubtedly that, mostly the cars from the clients or the outsiders can be identified in the roadway. So, there is less probability of collecting same data in the two consecutive days. Even though, due to human errors same data might be collected in the two consecutive days. A process was applied for distinguishing those data from the unique ones discussed in the Data Analysis and Model Development chapter. Also, through data analysis, matching to the features mentioned earlier in the Table 3.3 was identified.

License plates from the front side of the cars were captured according to the consideration of the vehicle types. Also, same license plates were captured from at least three different angles which might aid in the assumptions of the model development discussed later on. Some approximations were made to be applied before analyzing data as well. One of those approximations is to convert those images with weather effects using photo editing tools manually to accept vulnerability due to adverse weather. It might aid to robust the adaptability of the model for the future studies. Also, the original license plate designations are to be adjusted later on after each image which will be used for calibration and validation after the model development.

3.5.2. Auxiliary data

In addition to the main source of data in the form of image captures as mentioned earlier, some auxiliary data were also collected. These data were mainly required for camera calibration *i.e.*, choosing the best position for image captures. Auxiliary data were also collected apart from the primary data beyond the features mentioned in the Table 3.3. It was because ANFIS performance was to be checked against unknown character inputs as well.

In addition to that, images of some computer-generated fonts have been collected as auxiliary data to match the actually collected data with those fonts. After comparing those auxiliary images with the collected data, similar font style might be identified which will be used for the parameter definition in the ANFIS modeling described afterwards in the Data Analysis and Model Development chapter.

3.6. Overview

Staring from the beginning of this chapter, potential methods and toolkits have been compared with one another and the best suited ones are identified for the model development. It was found that ANFIS modeling would be justified to lead the research. Also, this algorithm for ALPR to be done using MATLAB software as it provides convenient environment for both IP and ANFIS modeling. Later on, different data collection methods have been discussed in this chapter from which the appropriate one has been chosen considering all constraints. After that, criteria for site selection have been discussed based on which the reconnaissance survey was done followed by the finalization of the data collection site. Finally, details of the collection of both primary and auxiliary data have been illustrated there using which the research will be carried on for the data analysis and model development.

CHAPTER 4
DATA ANALYSIS AND MODEL DEVELOPMENT

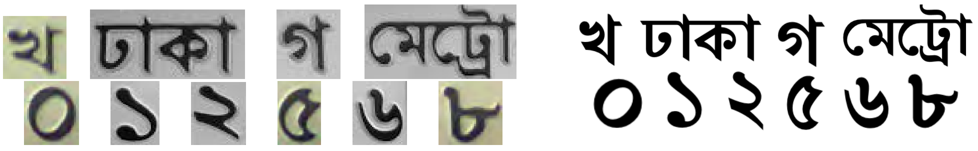
4.1. Introduction

As discussed in the earlier chapters about various limitations of other AI architectures and software, finally ANFIS as AI architecture and MATLAB as software for developing ALPR system have been justified. Taking the features into account as mentioned earlier in the Table 3.3, license plates will be distinguished and analyzed afterwards in this chapter. Moreover, starting from a raw data *i.e.,* captured image, the procedure for IP with different built-in functions of MATLAB will be illustrated as well.

In this chapter, usage and performance details of different functions of MATLAB will be demonstrated along with proper examples. Also, MFs and other parameters of ANFIS within MATLAB will be explored as well. Furthermore, features of different Bangla characters, numerals and words will be reconnoitered using which the model will be developed and programming codes will be written afterwards. Model development on the basis of features will be self-explanatory for the calibration process of which validation will be done in the Model Validation chapter later on. Finally, the flow diagram of the complete model will be depicted within one diagram to illustrate the process in a comprehensive manner.

4.2. Representation of characters obtained from collected data

After the collected data is done as discussed in the earlier chapter, some characters have been extracted to train the model. The characters have been extracted manually for comparing these to the available computer-generated font styles collected as auxiliary data as mentioned in the Methodology and Data Collection chapter. Fortuitously, similarities of those extracted characters with one computer generated font style was found [see Figure 4.1].



(a) (b)

Figure 4.1: Comparison between (a) license plate extracted and (b) computer generated characters

It was observed that, almost every font of license plate [see Figure 4.1(a)] has a good match over the computer generated [see Figure 4.1(b)] font style. An indication is obtained from it that the computer-generated fonts can be used for the training of the model which will be one of the major fragments of the model development. Total 20 characters have been distinguished from the collected data according to the specification in the Table 3.3 and some auxiliary data as mentioned in the Table 4.1. Efficient features from the geometry of these characters will be quantified for the ANFIS modeling.

Table 4.1: Finally Selected Characters for ALPR Model Development

Character type	List of characters	Counts
Letters	“গ” (Ga), “খ” (Kha), “হ” (Ha)	3
Numerals	Original: ০,১,২,৩,৪,৫,৬,৭,৮,৯	10
	In English: 0,1,2,3,4,5,6,7,8,9	
Words	Original: ঢাকা, সিলেট, বরিশাল, রাজশাহী, মেট্রো, খুলনা, চট্টগ্রাম	7
	In English: Dhaka, Sylhet, Barisal, Rajshahi, Metro, Khulna, Chittagong	
Total		20

It is seen that, three letters Ga, Kha and Ha have been finalized for the ALPR, which also represents the lion’s share of the private cars in Bangladesh as well as the motorbikes. Also, most of the major divisional metropolitan city names were considered for the model development. Last but not the least important type of characters *i.e.*, the numerals have been taken for the ALPR as well. All the exclusive features of these characters will be discussed later on. Apparently, there are three types of character as prescribed in the Table 4.1, but for the analytical purpose based on the geometry of those characters, two major types were distinguished namely, single-character and multi-character. Parameters for these divisions will be discussed along with their features in this chapter as well.

4.2.2. Single-character and multi-character types

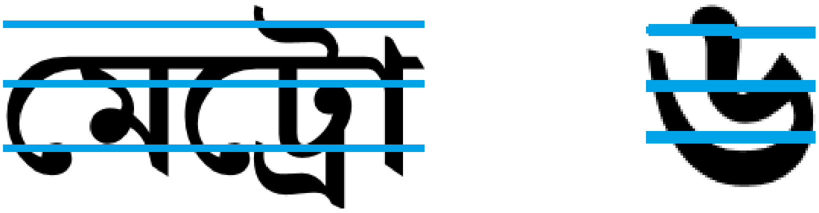
As mentioned earlier that, although there are three different types, those will be divided into two major divisions for the analytical purpose. Two types are single-character type and multi-character type. It can be found quite different from the general OCR algorithm as OCR usually caters for the single-character types. The reason behind this is, unlike common OCR algorithm as for English language, letters within the Bangla words are not in separated form. Rather it can be seen that words are generally combination of different letters attached together with a dash above which is called ‘Matra’ in Bangla. Examples of ‘Dhaka’ and ‘Metro’ can be provided for convenience [see Figure 4.1]. An algorithm for separating those letters from the word could have been suggested. But that approach would create complexities in ALPR system especially using ANFIS which uses simple distinguishing attributes. Besides, there are very few numbers of metropolitan areas registered for the vehicles, therefore considering the total word as one character might not be a problem for ALPR. In addition to that, separation of letters might lead the ALPR system to higher inaccuracy rates at it is found for some approaches discussed in the Literature Review chapter. Therefore, words are categorized as multi-character type whereas numerals and letters are individually categorized as single-character type.

There will be basically two parameters for these two types which will be used in ANFIS. First one named as $out1$ is used to denote the single-character types, therefore while dealing with single-character types, that parameter will give unique non-zero values for different single characters. There will be another parameter named as $out2$ for denoting the multi-character types which will always result in zero for the single-character types and give unique non-zero values for different multi characters. These two parameters can be represented as:

$$\begin{aligned} out1 &= 0 \; for\ multi\ characters \\ &\neq 0 \; for\ single\ characters \end{aligned} \tag{4.1}$$

$$\begin{aligned} out2 &= 0 \; for\ single\ characters \\ &\neq 0 \; for\ multi\ characters \end{aligned} \tag{4.2}$$

Returns of $out1$ and $out2$ for single-character and multi-character types will be discussed later on while discussing exclusive features of each character separately. Another question might seem rational in this stage that what would be the determiner of the single-character and multi-character types. For this, a customized function is used in MATLAB named as ‘separatedcharacter’ [see Appendix-A] to get average number of separate vertical lines throughout the character from top to bottom [see Figure 4.2]. It was found that this value can be used as a determiner of single-character type and multi-character type.



(a) (b)

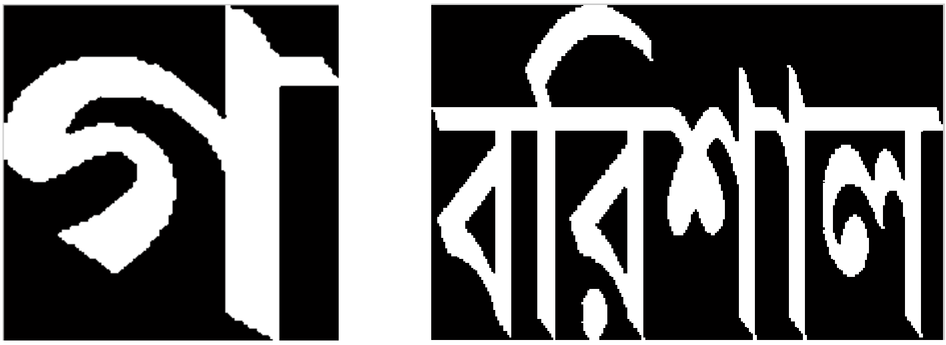
Figure 4.2: Comparison between (a) multi-character and (b) single-character types

From the geometric analysis, it was found that an average value of 2.25 vertical lines suits best for distinguishing single and multi-character types. All the single-character types *i.e.*, the numerals and the letters used in the research have an average vertical line of less than 2.25 [see Figure 4.2(b)]. On the other hand, greater or equal value of average vertical lines *i.e.*, output of separatedcharacter function indicates multi-character types [see Figure 4.2(a)]. All these operations will be undertaken after converting any image into a binary image using built-in functions depicted later on. Eventually, equations 4.1 and 4.2 can be reformed as:

$$\begin{aligned} out1 &= 0 \; for \; seperatedcharacter \geq 2.25 \\ &\neq 0 \; for \; seperatedcharacter < 2.25 \end{aligned} \tag{4.3a}$$

$$\begin{aligned} out2 &= 0 \; for \; seperatedcharacter < 2.25 \\ &\neq 0 \; for \; seperatedcharacter \geq 2.25 \end{aligned} \tag{4.4a}$$

One thing is to be referred here that the images will have to be resized to a specific rectangular boundary so that proper attributes of the characters can be extracted from their geometry. As multi-character types are greater in width, their reference width must be greater as well [see Figure 4.3]. In case of single-character types, size can be assumed as square. Keeping eye on all these things, multi-character types will be converted into 256 px × 512 px rectangle [see Figure 4.3(b)], whereas the single-character types will be converted into 256 px × 256 px square [see Figure 4.3(a)]. It can be easily realized from the multi-character type image that further squeezing might violate the feature qualities from the geometry. Again, it should be remembered that all operations will be done after converting these characters into binary images. Before resizing, the images will have to be cropped keeping the maximum extent throughout [see Figure 4.3]. Therefore, within the resized image, towards all directions there should have at least one touching point each.



(a) (b)

Figure 4.3: Resized images of (a) single-character and (b) multi-character types

Another significant thing which has to be considered during character recognition is the weight of the font. Based on the font weight, variations in geometric properties can be discernable. Therefore, it is necessary to restrict the weight of the fonts in such a manner so that the characters will be converted into desired font weight similar to the reference characters. This phenomenon can be interpreted through the number of non-zero elements in the binary image of a character which will be depicted later on during MATLAB IP-function illustrations. In this research, percentage of non-zero elements within the image has been used. Restricting the value of percentage of non-zero elements to a specific one can be time consuming as well. Therefore, a range of percentage of non-zero elements has been determined for the research from 30% to 40%. If it is seen that any character obtained from ALPR feature extraction does not fall within this category then morphological operations are conducted until this condition is satisfied which will be demonstrated later on as well.

4.2.3. Features of the single-character types

As mentioned earlier that numerals and letters will be treated as single characters, determiner of single-character and multi-character types was also interpreted earlier. Once, it is distinguished whether the character is single or multi type, further distinguishing will have to be done based on the geometry of those characters designated as exclusive features. Based on these features, $out1$ and $out2$ will be further decorated with unique non-zero values which will be finally used to designate final character output in ANFIS modeling. Here, all the exclusive features will be illustrated in details. First of all, single characters will be resized to 256 px × 256 px square images. After that, recognition of geometric features will be conducted. While recognizing, searching for the features will be done by the program according to the order as described in this section. Although all images will be resized to the specific dimensions, percentage has been used keeping an eye on the ease of the perception.

First to start with is the Bangla numeral of ‘0’, which has a significant hole at the middle of the character [see Figure 4.4] which is an exclusive feature. This geometric feature can be applied to identify numeral ‘0’ during ALPR operation. The region can be identified with a rectangular portion starting from the 30% width to the 70% width and starting from the 30% height to the 70% height of the resized character.

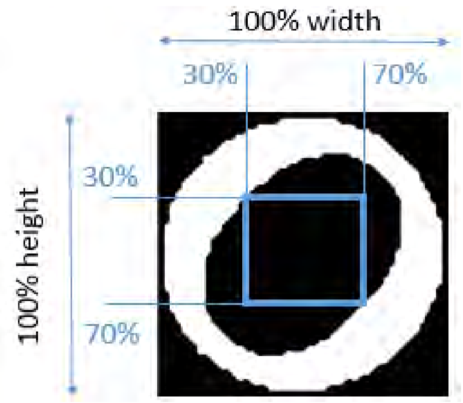


Figure 4.4: Geometric feature of the Bangla numeral '0' for ALPR system

For safety purpose, percentage of non-zero elements of this strip denoted as $Q1$ has been set to be less than 12% instead of 0% as tolerance for the binary output $A1$. The feature extracted can be classified as below.

$$\begin{aligned} A1 &= 1\ for\ Q1 < 0.12 \\ &\neq 0\ for\ Q1 \geq 0.12 \end{aligned} \tag{4.5}$$

There are two features for the Bangla numeral of '1' which can be used to recognize this character. First one is the middle left portion and the second one is the top right corner of numeral '1' [see Figure 4.5]. A rectangle starting from the 0% width to the 30% width and starting from the 30% height to the 70% height has been set for the first feature denoted as $Q2$. A triangular portion has been defined covering the area of the top right portion starting from the 50% width up to the 50% height [see Figure 4.5] denoted as $Q3$.

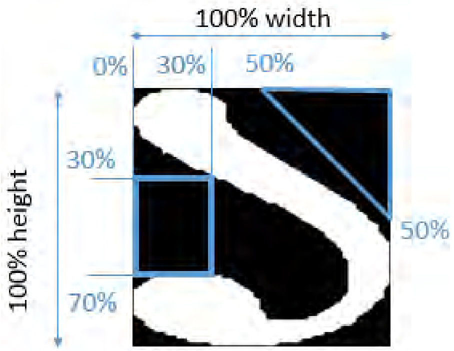


Figure 4.5: Geometric features of the Bangla numeral '1' for ALPR system

Percentage of non-zero elements for both of these two features has been set as 12% instead of 0% as tolerance for the binary outputs $A2$ and $A3$ respectively. The features hence can be classified as:

$$\begin{aligned} A2 &= 1 \; for \; Q2 < 0.12 \\ &\neq 0 \; for \; Q2 \geq 0.12 \end{aligned} \tag{4.6}$$

$$\begin{aligned} A3 &= 1 \; for \; Q3 < 0.12 \\ &\neq 0 \; for \; Q3 \geq 0.12 \end{aligned} \tag{4.7}$$

For the Bangla numeral of '2' there are two features for identifying this character. First one is identical with the previously mentioned $Q3$ for numeral '1'. Second one is a new feature denoted as $Q4$ depicting a triangular area in the bottom left side of '2' [see Figure 4.6]. Although $Q3$ might add some non-zero elements within the region, but as a tolerance level has been set, it is expected that the character would accord to this feature.

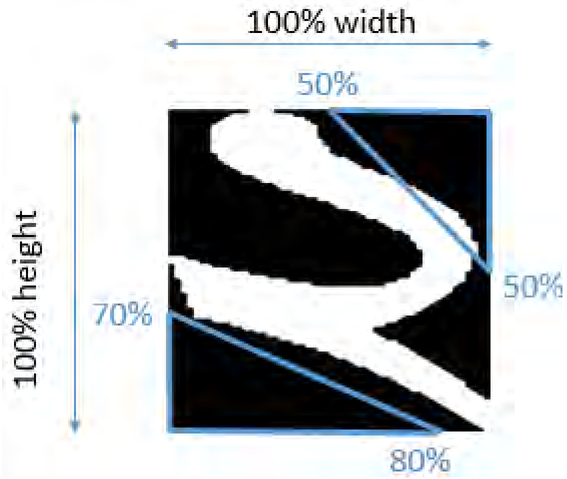


Figure 4.6: Geometric features of the Bangla numeral '2' for ALPR system

Percentage of non-zero elements for both of these two features has been set as 12% instead of 0% as tolerance for the binary outputs $A3$ and $A4$ respectively. The classification thereby can be defined as the feature $Q3$ follows classification of $A3$ according to the equation (4.7) and the feature $Q4$ as below.

$$\begin{aligned} A4 &= 1 \; for \; Q4 < 0.12 \\ &\neq 0 \; for \; Q4 \geq 0.12 \end{aligned} \tag{4.8}$$

For classifying the Bangla numeral of ‘3’, three exclusive features from its geometry can be extracted. First one is the rectangular blank space on the top left side of ‘3’ [see Figure 4.7]. It can be denoted as $Q5$, starting from the 0% width to the 35% width and starting from the 0% height to the 15% height. The second one is another rectangular blank space on the top left side. It can be denoted as $Q6$, starting from the 15% width to the 30% width and starting from the 0% height to the 30% height. The third one is middle rectangular portion remained blank within the character of ‘3’. It can be denoted as $Q7$, starting from the 35% width to the 75% width and starting from the 45% height to the 70% height. The feature $Q7$ is different from $Q1$ in such a manner that $Q1$ covered a huge region along the height of the character, numeral ‘3’ is slightly conservative in this regard.

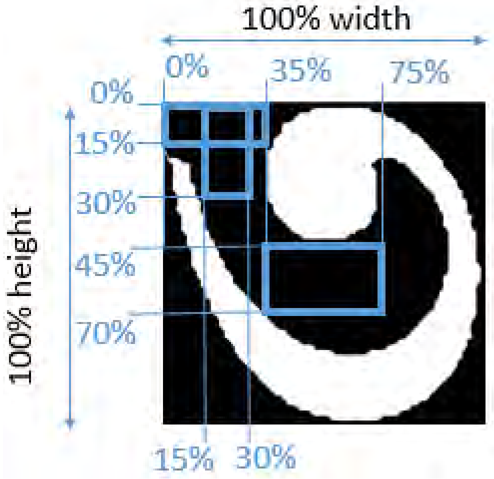


Figure 4.7: Geometric features of the Bangla numeral ‘3’ for ALPR system

Percentage of non-zero elements for $Q5$ has been set to 8% as tolerance level considering very few chances of any non-zero element in that region, whereas it was set 15% for $Q6$ and $Q7$ as tolerance level considering critical region [see Figure 4.7] and greater possibility of having non-zero elements within the regions bounded by those features. Binary outputs of $Q5, Q6$ and $Q7$ are $A5, A6$ and $A7$ respectively. The features hence can be classified as:

$$\begin{aligned} A5 &= 1\ for\ Q5 < 0.08 \\ &\neq 0\ for\ Q5 \geq 0.08 \end{aligned} \tag{4.9}$$

$$\begin{aligned} A6 &= 1\ for\ Q6 < 0.15 \\ &\neq 0\ for\ Q6 \geq 0.15 \end{aligned} \tag{4.10}$$

$$\begin{aligned} A7 &= 1\ for\ Q7 < 0.15 \\ &\neq 0\ for\ Q7 \geq 0.15 \end{aligned} \tag{4.11}$$

For classifying the Bangla numeral of ‘4’, two exclusive features have been identified. First one is the bottom middle portion of ‘4’, as it can be understood from the geometry that numeral ‘4’ contains a symmetrical shape along both width and height [see Figure 4.8]. Only the bottom middle blank portion has been pointed out using a rectangular boundary denoted as $Q8$ starting from the 45% width to the 55% width and starting from the 65% height to the 75% height. Another exclusive feature which is denoted as $Q9$ starting from the 25% width to the 75% width and starting from the 45% height to the 55% height, is the obvious portion containing non-zero elements almost throughout that region.

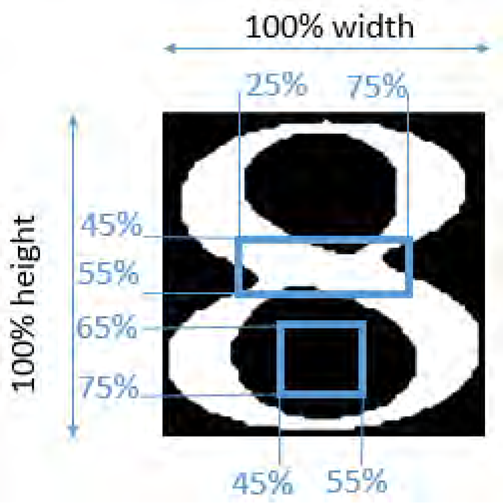


Figure 4.8: Geometric features of the Bangla numeral ‘4’ for ALPR system

Percentage of non-zero elements for $Q8$ has been set to 15% considering inclusion of some non-zero elements within this region as tolerance for binary output $A8$. On the other hand, for $Q9$ as this region should abound by non-zero elements, percentage of non-zero elements has been set to 80% allowing provision for some blank space as tolerance for binary output $A9$. The features therefore can be classified as:

$$A8 = 1 \; for \; Q8 < 0.15 \qquad (4.12)$$
$$\neq 0 \; for \; Q8 \geq 0.15$$
$$A9 = 1 \; for \; Q9 \geq 0.80 \qquad (4.13)$$
$$\neq 0 \; for \; Q9 < 0.80$$

For identifying the Bangla numeral of ‘5’, one exclusive feature has been distinguished. It is the blank triangular portion on the right side of the character ‘5’ [see Figure 4.9] having base starting from the 35% height to the 100% height and vertex lying on 75% width. This feature has been denoted as $Q10$.

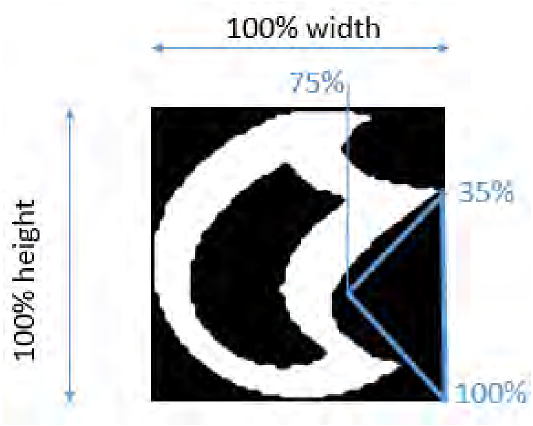


Figure 4.9: Geometric feature of the Bangla numeral '5' for ALPR system

Percentage of non-zero elements of $Q10$ has been set to 9% as tolerance level as a binary output $A10$. This feature thereby can be classified as:

$$\begin{aligned} A10 &= 1\ for\ Q10 < 0.09 \\ &\neq 0\ for\ Q10 \geq 0.09 \end{aligned} \tag{4.14}$$

There is one exclusive feature for the Bangla numeral of '6'. It is the rectangular blank portion on the top right side [see Figure 4.10] of the character '6' denoted as $Q11$. The region can be defined as the rectangle starting from the 75% width to the 100% width and starting from the 0% height to the 25% height.

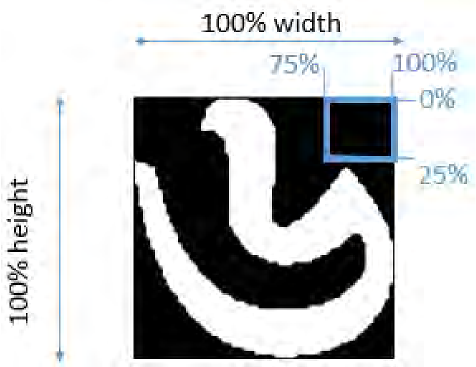


Figure 4.10: Geometric feature of the Bangla numeral '6' for ALPR system

Percentage of non-zero elements within $Q11$ region has been set to 5% tolerance level only considering lower probability of having non-zero elements there. A binary output $A11$ is given rise by that condition and thereby it can be classified as:

$$\begin{aligned} A11 &= 1\ for\ Q11 < 0.05 \\ &\neq 0\ for\ Q11 \geq 0.05 \end{aligned} \tag{4.15}$$

For recognizing Bangla numeral ‘7’, one exclusive feature has been identified. It is the bottom left blank rectangular portion [see Figure 4.11] denoted as $Q12$ starting from the 0% width to the 75% width and starting from the 70% height to the 100% height of character ‘7’.

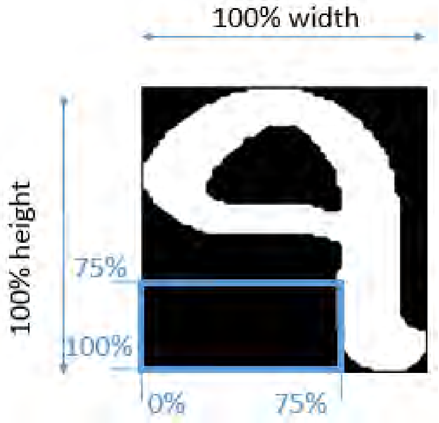


Figure 4.11: Geometric feature of the Bangla numeral ‘7’ for ALPR system

Percentage of non-zero elements within $Q12$ region has been set to 12% tolerance level taking a binary output $A12$. Therefore, the classification feature can be expressed as:

$$\begin{aligned} A12 &= 1\ for\ Q12 < 0.12 \\ &\neq 0\ for\ Q12 \geq 0.12 \end{aligned} \tag{4.16}$$

Some exclusive features from the geometry of the Bangla numeral of ‘8’ can be considered. First one which is denoted as $Q13$ corresponds to the wide rectangular blank portion on the top right side [see Figure 4.12] of the character ‘8’ starting from the 25% width to the 100% width and starting from the 0% height to the 30% height. Second one which is denoted as $Q14$ corresponds to the rectangular blank portion available on the bottom right side of the character starting from the 80% width to the 100% width and starting from the 70% height to the 100% height.

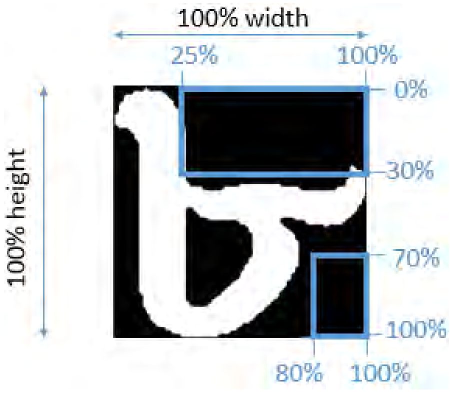


Figure 4.12: Geometric features of the Bangla numeral ‘8’ for ALPR system

Percentage of non-zero elements within $Q13$ region has been set to 12% tolerance level from which a binary output is generated, denoted as $A13$. On the hand, this value within $Q14$ has been set to 8% tolerance level considering less possibility of having any non-zero elements there and it generates a binary output denoted as $A14$. These can be expressed as:

$$\begin{aligned} A13 &= 1\ for\ Q13 < 0.12 \\ &\neq 0\ for\ Q13 \geq 0.12 \end{aligned} \tag{4.17}$$

$$\begin{aligned} A14 &= 1\ for\ Q14 < 0.08 \\ &\neq 0\ for\ Q14 \geq 0.08 \end{aligned} \tag{4.18}$$

For recognizing of the Bangla numeral of '9', one exclusive feature in addition to a feature of Bangla numeral '1' has been defined. Feature similar to the Bangla numeral of '1' is $Q3$ which is also true for the Bangla numeral of '2' as well discussed earlier. For distinguishing '9' from those, the exclusive feature is the number of average vertical lines within the rectangular region starting from 75% height of the character image to the end [see Figure 4.13] denoted as $Q15$.

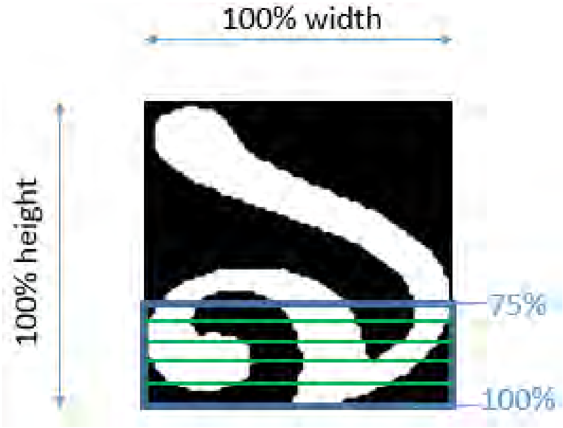


Figure 4.13: Geometric feature of the Bangla numeral '9' for ALPR system

From the observation it can be established that the number of average vertical lines within $Q15$ should be close to 3. On the other hand, in case of the numeral '1' and '2' it would be around 1. Still accepting variations, a tolerance level of 1.72 has been set for the number of average vertical lines which gives a binary output $A15$. The classification can be expressed as:

$$\begin{aligned} A15 &= 1\ for\ Q15 \geq 1.72 \\ &\neq 0\ for\ Q15 < 1.72 \end{aligned} \tag{4.19}$$

By this time geometric features of all the numerals have been discussed with their corresponding values. Only remaining single-character types are the three letters Ga, Kha and Ha which will be discussed. Most frequently found Bangla letter for the private cars is “Ga” which can be recognized through two exclusive features. First one is the rectangular blank portion of the bottom right side of the character ‘Ga’ starting from the 0% width to the 45% width and starting from the 80% height to the 100% height of the character [see Figure 4.14] denoted as $Q16$. The second one is required to differentiate between character ‘Ga’ and Bangla numeral ‘7’ because that numeral might also possess position approach towards $Q16$. The second feature denoted as $Q24$ is the top right shallow but wide rectangular blank portion [see Figure 4.14] of ‘Ga’ starting from the 0% width to the 60% width and starting from the 0% height to the 2% height.

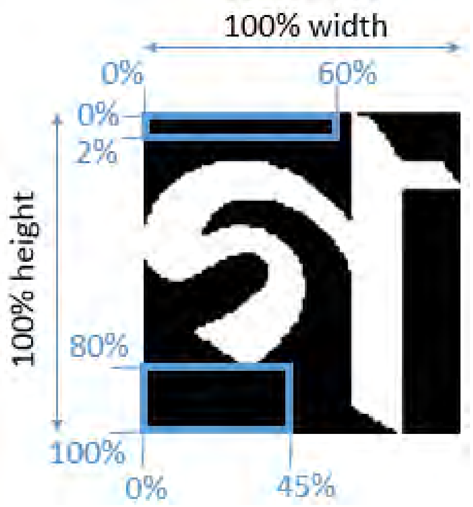


Figure 4.14: Geometric features of the Bangla letter ‘Ga’ for ALPR system

Percentage of non-zero elements for $Q16$ has been set to 12% tolerance level for which a binary output $A16$ can be obtained. On the other hand, this value for $Q24$ has been set to only 2% with a binary output $A24$ considering there is almost no possibility for non-zero elements to be inside this region. For classification of the character ‘Ga’, these two features can be expressed as:

$$A16 = 1 \; for \; Q16 < 0.12 \quad (4.20)$$
$$\neq 0 \; for \; Q16 \geq 0.12$$
$$A24 = 1 \; for \; Q24 < 0.02 \quad (4.21)$$
$$\neq 0 \; for \; Q24 \geq 0.02$$

To start with the Bangla letter ‘Kha’, it should be noted that it possesses almost same geometric features as ‘Ga’ mentioned earlier. Especially $Q16$ matches all the time with ‘Kha’ and ‘Ga’. Thus, in order to distinguish between them one exclusive feature of 'Kha' has been set. It is not only a narrow but also a deep rectangular portion of middle bottom side of 'Ga' and 'Kha' [see Figure 4.15] denoted as $Q17$ starting from the 45% width to the 65% width and starting from the 65% height to the 85% height.

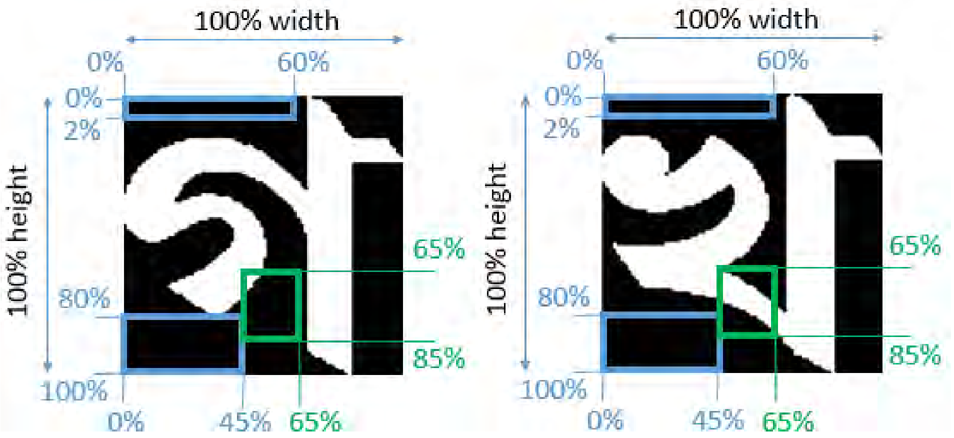


Figure 4.15: Geometric features of the Bangla letter 'Kha' for ALPR system

Percentage of non-zero elements for $Q16$ can be set similar to that has been already mentioned earlier during feature orientation of 'Ga' shown in equations (4.20) and (4.21). Percentage of non-zero elements for $Q17$ has been set to 25% tolerance level considering sensitive region and possibility of variations and binary output of $A17$ for this. Hence it can be expressed as:

$$\begin{aligned} A17 &= 1 \; for \; Q17 < 0.25 \\ &\neq 0 \; for \; Q17 \geq 0.25 \end{aligned} \tag{4.22}$$

The only remainder of single-character type is the Bangla letter 'Ha' which is used for the motorbikes in Bangladesh. There is an obvious similarity of this character with Bangla numeral '2' [see Figure 4.16] on the $Q4$ region. Even $Q3$ region is found satisfactory for 'Ha' character along with numerals '1' and '2' in many cases. The only difference is that the character 'Ha' contains 'Matra' as it was discussed earlier during the discussion of words. This is a horizontal line on the top of the Bangla characters. Therefore, that geometric feature has been set to distinguish between 'Ha' and 'Kha'. It is denoted as $Q23$ depicting a rectangular boundary covering the total width of the character and starting from the 0% height to the 2% height [see Figure 4.16].

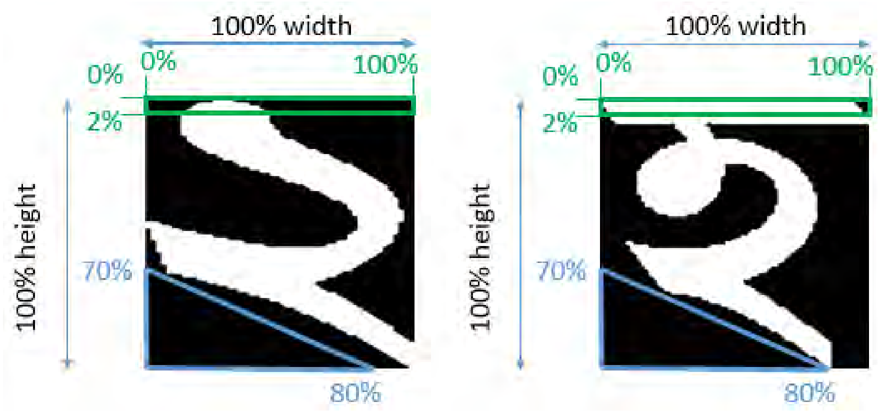


Figure 4.16: Geometric features of the Bangla letter ‘Ha’ for ALPR system

Percentage of non-zero elements within $Q23$ has been set to 40% tolerance level considering that it would easily separate numerals ‘1’ and ‘2’ from the identification. A binary output can be obtained from this tolerance level denoted as $A23$. Therefore keeping $Q4$ feature for output $A4$ as mentioned in equation (4.8), $Q23$ can be classified as:

$$\begin{aligned} A23 &= 1 \; for \; Q23 \geq 0.4 \\ &\neq 0 \; for \; Q23 < 0.4 \end{aligned} \qquad (4.23)$$

In the meantime, geometric features of all single-character types have been discussed with proper illustrations. All those features can be combined together to get the desired outputs. As it was also mentioned earlier that the algorithm will search for the features according to the order as depicted, hence it should be kept in mind that features of successor characters must not satisfy predecessor characters by any means. Otherwise, false recognition of those characters might be made by the algorithm. Keeping all these facts in mind, equation (4.3a) can be reformed for single-character types as below.

From equation (4.3a) $for \; seperatedcharacter < 2.25$ &

$$\begin{gathered} if \; A1 = 1 \\ out1 = 1, denotes \; Bangla \; numeral \; '0' \end{gathered} \qquad (4.3b)$$

$$\begin{gathered} elseif \; A2 = 1, A3 = 1 \\ out1 = 2, denotes \; Bangla \; numeral \; '1' \end{gathered} \qquad (4.3c)$$

$$\begin{gathered} elseif \; A3 = 1, A4 = 1, A23 = 0, A17 = 0 \\ out1 = 3, denotes \; Bangla \; numeral \; '2' \end{gathered} \qquad (4.3d)$$

$$elseif\ A5 = 1, A6 = 1, A7 = 1, A16 = 0$$
$$out1 = 4, denotes\ Bangla\ numeral\ '3' \tag{4.3e}$$

$$elseif\ A8 = 1, A9 = 1, A12 = 0, A13 = 0, A14 = 0$$
$$out1 = 5, denotes\ Bangla\ numeral\ '4' \tag{4.3f}$$

$$elseif\ A10 = 1, A23 = 0, A13 = 0, A16 = 0$$
$$out1 = 6, denotes\ Bangla\ numeral\ '5' \tag{4.3g}$$

$$elseif\ A10 = 0, A11 = 1, A13 = 0, A15 = 0, A21 = 0, A12 = 0$$
$$out1 = 7, denotes\ Bangla\ numeral\ '6' \tag{4.3h}$$

$$elseif\ A12 = 1, A23 = 0, A24 = 0$$
$$out1 = 8, denotes\ Bangla\ numeral\ '7' \tag{4.3i}$$

$$elseif\ A13 = 1, A14 = 1$$
$$out1 = 9, denotes\ Bangla\ numeral\ '8' \tag{4.3j}$$

$$elseif\ A3 = 1, A15 = 1, A21 = 0$$
$$out1 = 10, denotes\ Bangla\ numeral\ '9' \tag{4.3k}$$

$$elseif\ A16 = 1, A17 = 1, A21 = 0, A24 = 1$$
$$out1 = 11, denotes\ Bangla\ letter\ 'Ga' \tag{4.3l}$$

$$elseif\ A16 = 1, A17 = 0, A21 = 0, A23 = 0$$
$$out1 = 12, denotes\ Bangla\ letter\ 'Kha' \tag{4.3m}$$

$$elseif\ A4 = 1, A23 = 1$$
$$out1 = 13, denotes\ Bangla\ letter\ 'Ha' \tag{4.3n}$$

All equations mentioned above are the detection algorithms of single-character types *i.e.*, having $out1 = 0$ value. Combination of all these will be used in the MALTAB programming along with the features of multi-character types as an algorithm for ALPR which will be discussed later on.

4.2.4. Features of the multi-character types

According to the general guidelines mentioned earlier, multi-character types are basically the Bangla words and those will be resized to 256 px × 512 px of binary images. After that, detection of the geometric features will be done accordingly.

Therefore, detailed illustrations of those features are necessary which will be made from now on.

First multi-character type is the Bangla word ‘Sylhet’ which is used to denote Sylhet metropolitan area of Bangladesh. One exclusive feature has been distinguished from the geometry of the resized character ‘Sylhet’. It is the top middle blank rectangular portion [see Figure 4.17] denoted as $Q18$ starting from the 35% width to the 65% width and starting from the 0% height to the 15% height.

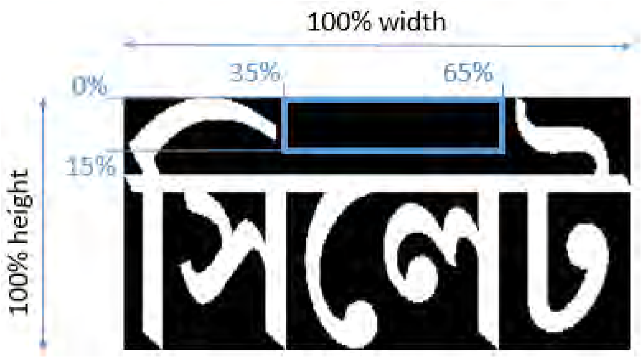


Figure 4.17: Geometric feature of the Bangla word ‘Sylhet’ for ALPR system

Percentage of non-zero elements within $Q18$ region has been set to 5% tolerance level only admitting lower possibility of having non-zero elements there from which a binary output $A18$ can be obtained. Therefore, it can be classified as:

$$\begin{aligned} A18 &= 1\ for\ Q18 \leq 0.05 \\ &\neq 0\ for\ Q18 > 0.05 \end{aligned} \tag{4.24}$$

Next multi-character type for the ALPR model is the Bangla word ‘Rajshahi’. One exclusive feature has been set for the character ‘Rajshahi’ for distinguishing it from the character ‘Sylhet’. It is because binary output of $Q18$ *i.e.*, $A18$ is similar for both these two characters. So, the exclusive feature for ‘Rajshahi’ is the top left blank rectangular portion [see Figure 4.18] denoted as $Q19$ starting from the 0% width to the 45% width and starting from the 0% height to the 15% height. This portion can be utilized to distinguish between the ‘Rajshahi’ and the ‘Sylhet’.

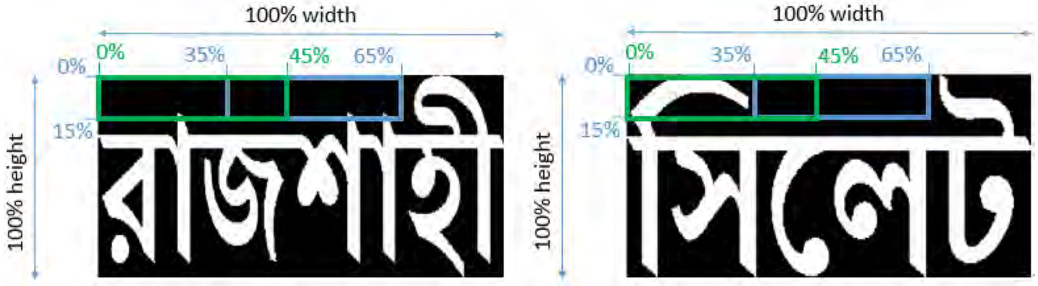


Figure 4.18: Geometric features of the Bangla word ‘Rajshahi’ for ALPR system

Percentage of non-zero elements within $Q19$ has been set to 10% tolerance level considering the necessity of distinguishing between those characters which will generate a binary output denoted as $A19$. Therefore $A18$ being same for ‘Rajshahi’ and ‘Sylhet’, output $A19$ can be classified as:

$$\begin{aligned} A19 &= 1 \; for \; Q19 \leq 0.1 \\ &\neq 0 \; for \; Q19 > 0.1 \end{aligned} \tag{4.25}$$

Another Bangla word ‘Barisal’ which is used to denote Barisal metropolitan can be identified with one exclusive feature. It is the top right blank rectangular portion [see Figure 4.19] denoted as $Q20$ starting from the 45% width to the 85% width and starting from the 0% height to the 15% height. Though it could have been extended up to 100% width, but still flexibility has been left over there.

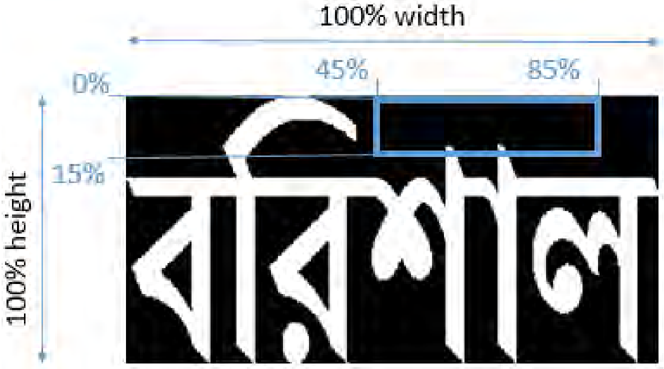
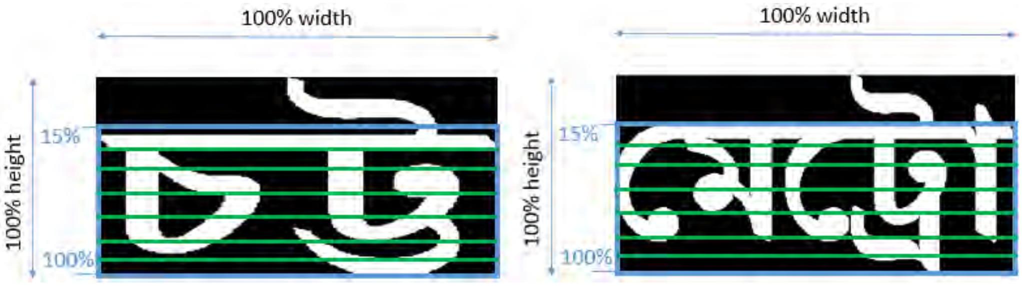


Figure 4.19: Geometric feature of the Bangla word ‘Barisal’ for ALPR system

Percentage of non-zero elements for the region bounded by $Q20$ has been set to 10% from which a binary output denoted as $A20$ can be obtained. The classification thereby can be expressed as:

$$\begin{aligned} A20 &= 1 \; for \; Q20 \leq 0.1 \\ &\neq 0 \; for \; Q20 > 0.1 \end{aligned} \tag{4.26}$$

For the word ‘Chotto’ which is used for the representation of Chittagong metropolitan has one similar binary output $A19$ in the region $Q19$ to that of the word ‘Rajshahi’ depicted earlier. Also, the word ‘Metro’ which is used to represent metropolitan, has similar circumstance for $A19$. Also, it should be noted that the character ‘Metro’ is the only word which will be found in each and every license plate. Therefore, further division of average number of separate vertical lines has been utilized as an exclusive feature of the ‘Chotto’ and the ‘Metro’ to distinguish those from ‘Rajshahi’. The region is defined as the full width of the character and starting from the 15% to the 100% height [see Figure 4.20] denoted as $Q21$. It is found that average number of vertical lines for the character ‘Chotto’ lies between 2.4 and 3; whereas for the character ‘Metro’, it lies between 3.90 and 6.15. This value can be a well distinguisher as it lies well above 6.5 for ‘Rajshahi’.



(a) (b)

Figure 4.20: Geometric features of the Bangla words (a) ‘Chotto’ and (b) ‘Metro’ for ALPR system

The value of average number of vertical lines within the region bounded by $Q21$ has been set to four ranges from which four outputs can be obtained denoted as $A21$. Three of which have been already mentioned earlier. If no match is found for the value within those ranges, then null output having value 0 is obtained. All of these can be classified as:

$$\begin{aligned} A21 &= 1 \; for \; 3.9 < Q21 < 6.15 \\ &= 2 \; for \; 2.4 < Q21 < 3 \\ &= 3 \; for \; Q21 > 6.5 \\ &= 0 \; for \; other \; values \; of \; Q21 \end{aligned} \tag{4.27}$$

Two exclusive geometric features of the words ‘Khulna’ and ‘Dhaka’ can be identified using which these two multi-character types can be distinguished. First one is to distinguish these two characters, other one is same for these two characters which is mainly used to distinguish these two from all other characters.

First feature denoted as $Q22$ is the blank rectangular portion [see Figure 4.21] of the bottom right side of 'Khulna' and 'Dhaka' starting from the 35% width to the 100% width and starting from the 92% height to the 100% height. The second geometric feature is a wide rectangular region having full width and starting from the 5% height to the 25% height denoted as $Q25$. This portion of 'Khulna' and 'Dhaka' is filled with enough non-zero elements in comparison with the other multi-character types as mentioned so far.

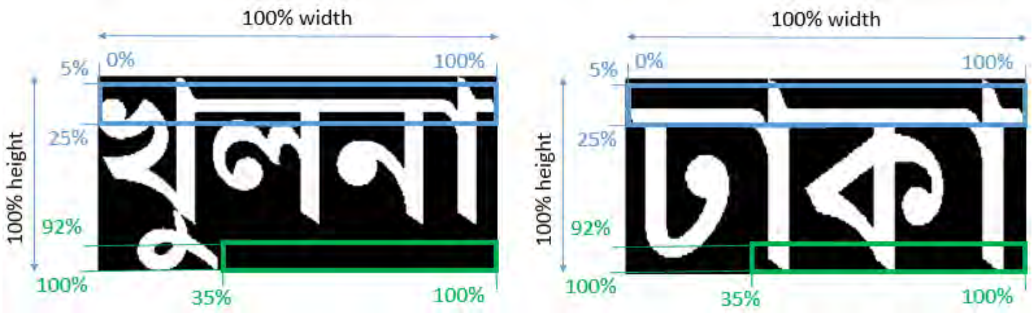


Figure 4.21: Geometric features of the Bangla words 'Khulna' and 'Dhaka' for ALPR system

Percentage of non-zero elements within the region bounded by $Q22$ is set to 5% tolerance level from which a binary output can be obtained denoted as $A22$. This output can distinguish between characters 'Dhaka' and 'Khulna'. On the other hand, the value of percentage of non-zero elements within $Q25$ has been set to 15% tolerance level assuming that 'Khulna' and 'Dhaka' would always result in a greater value than that. From this another binary output can be obtained denoted as $A25$. These can be classified as:

$$\begin{aligned} A22 &= 1 \; for \; Q22 \leq 0.05 \\ &\neq 0 \; for \; Q22 > 0.05 \end{aligned} \tag{4.28}$$

$$\begin{aligned} A25 &= 1 \; for \; Q25 \geq 0.15 \\ &\neq 0 \; for \; Q25 < 0.15 \end{aligned} \tag{4.29}$$

In the meantime, all geometric features of all multi-character types covered in this research have been illustrated with relevant figures. All the features can be combined to generate the final output of $out2$. While generating the algorithm it should be kept in mind that the operation for searching features will follow the order. Therefore, if the features are not found similar to that of the first one then it will go for the second one. Hence to avoid wrong outputs and to distinguish properly, some features of the successor characters have been imported to the predecessors to transfer searching algorithm to the next characters. Final conditions for recognizing those multi-characters are enlisted below.

From equation (4.4a) $for\ seperatedcharacter \geq 2.25\ \&$

$$\begin{gathered} if\ A18 = 1, A19 = 0 \\ out2 = 1, denotes\ Bangla\ word\ 'Sylhet' \end{gathered} \tag{4.4b}$$

$$\begin{gathered} elseif\ A18 = 1, A19 = 1, A21 = 3 \\ out2 = 2, denotes\ Bangla\ word\ 'Rajshahi' \end{gathered} \tag{4.4b}$$

$$\begin{gathered} elseif\ A20 = 1, A21 = 3 \\ out2 = 3, denotes\ Bangla\ word\ \ 'Barisal' \end{gathered} \tag{4.4c}$$

$$\begin{gathered} elseif\ A19 = 1, A21 = 2 \\ out2 = 4, denotes\ Bangla\ word\ 'Chotto' \end{gathered} \tag{4.4d}$$

$$\begin{gathered} elseif\ A19 = 1, A21 = 1, A20 = 0 \\ out2 = 5, denotes\ Bangla\ word\ \ 'Metro' \end{gathered} \tag{4.4e}$$

$$\begin{gathered} elseif\ A22 = 1, A25 = 1 \\ out2 = 6, denotes\ Bangla\ word\ 'Khulna' \end{gathered} \tag{4.4f}$$

$$\begin{gathered} elseif\ A22 = 0, A25 = 1 \\ out2 = 7, denotes\ Bangla\ word\ 'Dhaka' \end{gathered} \tag{4.4g}$$

All the equations mentioned above *i.e.*, the equations (4.4b) to (4.4g) will be combined with the equation (4.4a); on the other hand, the earlier mentioned equations (4.3b) to (4.3n) will be combined with equation (4.3a). These equations will be considered as the basic inputs for the ANFIS model established later on.

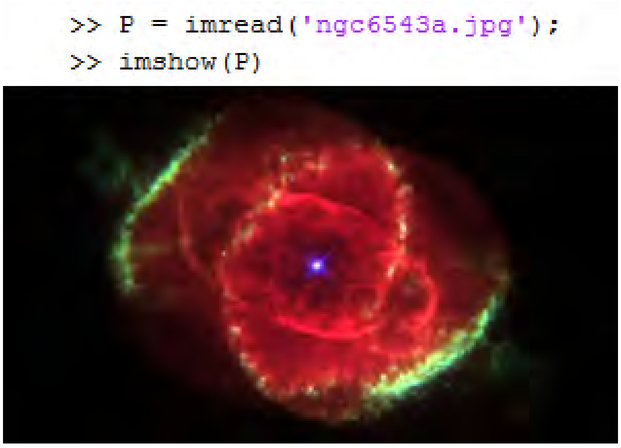
4.3. IP operations in MATLAB

From the discussions made above, it can be understood that a lion’s share of this research will be pertaining to the IP system for the collected RGB images. There are multi-dimensional features and variations available in MATLAB for IP. Therefore, discussion of the chosen features which will be used for the model development, is made in this chapter. These features will be used in the extraction of character images starting from the beginning of the raw data. After that, those characters will be used for the ANFIS model development discussed later on.

There are mainly two sub-models of the complete model. First one is relevant to the training data set for ANFIS and IP as a prerequisite of that. Training model as denoted, will be used to take different characters as inputs and after some pre-processing, it would collect features as variables and then using ANFIS operation over that to generate outputs. Recognizing model is denoted for the second model, will be more complex in manner as to extract different characters from the captured RGB images initially. After that, the IP system is similar to that of the Training model. Then the operation of ANFIS will be applied. Finally, outputs will be organized together to interpret the data and display the ALPR results for the users. In this section, relevant IP functions and operations for both Training and Recognizing models will be discussed in details.

4.3.1. MATLAB functions: imread and imshow

For starting with any image in MATLAB, the function for reading the image is a must. That function is denoted as ‘imread’ in MATLAB. Inclusion of imread(filename) in the command window of the MATLAB will read the image file named ‘filename’ within the current folder of MATLAB. But usual practice is the file extension to be included after the filename. It should be noted that, the filename image will be of type unsigned 8-bit integer (uint8) as pixel values of images can never be greater than $2^8 – 1$. Calling of ‘imread’ function for an image in MATLAB will automatically create a uint8 matrix for that image. The image can be either RGB, grayscale or binary in manner. If the image is RGB, then a three-dimensional matrix will be created through reading of that image in MATLAB using ‘imread’ containing two-dimensional pixel values for three channels *i.e.,* red, green and blue. If the image is grayscale, then a two-dimensional grayscale uint8 matrix will be created by reading of that image in MATLAB using ‘imread’. If the image is binary, then a two-dimensional matrix will be created by reading of that image in MATLAB using ‘imread’ containing binary values only *i.e.,* 0 or 1. Another vital IP function in MATLAB is the ‘imshow’ used for displaying any image for the users in a new window. If any variable is assigned to the image-reads using ‘imread’ by MATLAB or any images generated by plotting in MATLAB, then ‘imshow’ can be used. Therefore, ‘imread’ and ‘imshow’ can be used simultaneously in MATLAB [see Figure 4.22].

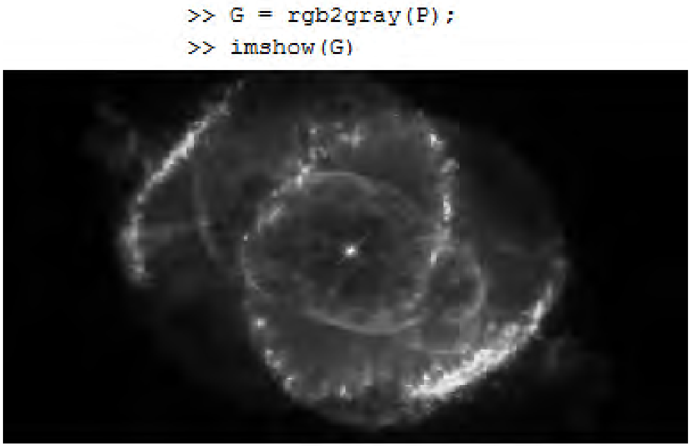


```
>> P = imread('ngc6543a.jpg');
>> imshow(P)
```

Figure 4.22: Illustration of the MATLAB functions imread and imshow

4.3.2. MATLAB functions: rgb2gray, graythresh and im2bw

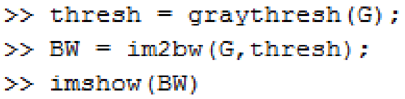
It is obvious that conversion of images from one color type to another color type will be necessary during IP. Therefore, three functions ‘rgb2gray’, ‘graythresh’ and ‘im2bw’ are used. In MATLAB, if an RGB image is to be converted into a grayscale image, ‘rgb2gray’ function can be used directly as rgb2gray(image) where ‘image’ is the three-dimensional matrix of the RGB image which has been converted into grayscale image. The RGB image read done previously [see Figure 4.1] can be converted into grayscale image in similar manner [see Figure 4.23].



```
>> G = rgb2gray(P);
>> imshow(G)
```

Figure 4.23: Illustration of the MATLAB function rgb2gray

For converting the grayscale image or RGB into binary image, function ‘im2bw’ is used. Level of luminance is required for converting into binary image within ‘im2bw’ function which can be obtained by a well renowned method provided by Otsu [132]. Providentially, MATLAB provides this operation as a function named ‘graythresh’ by which an image can be analyzed and threshold value indicating the desired level of luminance for ‘im2bw’ can be obtained. Obtaining the level of luminance and converting the previously diagramed grayscale image [see Figure 4.23] into binary image can be done accordingly [see Figure 4.24]. It can be seen that ‘thresh’ variable has been evaluated using ‘graythresh’ function initially. The function ‘graythresh’ was applied on the grayscale image. It could be applied on the RGB image as well. In that case different output might have been obtained due to the difference in level of luminance obtained for that image.



```
>> thresh = graythresh(G);
>> BW = im2bw(G,thresh);
>> imshow(BW)
```

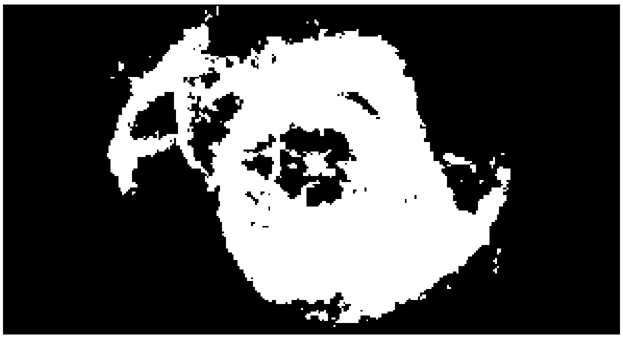


Figure 4.24: Illustration of the MATLAB functions graythresh and im2bw

4.3.3. MATLAB functions: imresize and imrotate

Two vital MATLAB IP functions are ‘imresize’ and ‘imrotate’. These two functions are frequently used in any IP operation especially for developing any ALPR system. It is because all the time, characters have to be recognized by the ALPR system based on the reference of some other characters as mentioned earlier in this chapter. If proper interference cannot be applied between the reference and the testing characters, anomalous results might be obtained. MATLAB provides two built-in functions ‘imresize’ and ‘imrotate’ for resizing and rotating operations respectively. The resizing function can be used as imresize(image, [A B]) syntax [see Figure 4.25]. Here, the image to be resized is added first, after that A and B are two integer-pixel inputs of the image corresponding to the new image height and width respectively.

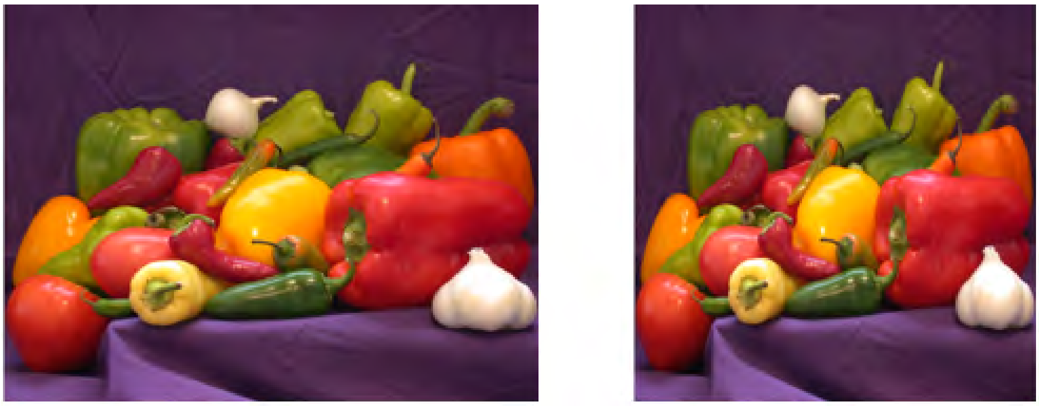


Figure 4.25: Illustration of the MATLAB function imresize

For rotating the original image as read in the previous example, 'imrotate' function can be used as imrotate(image,C) syntax [see Figure 4.26]. In that function, counter clock wise rotation will be made with a rotation value of C degree. It can be seen that the image has been rotated within a frame of equal size *i.e.*, width and height of the original image. Also, it can be seen that the outer portion of the image in the frame is marked as black color. However, it should be noted that, both 'imresize' and 'imrotate' functions can be applied to all images irrespective of whether the original image is RGB, grayscale or binary image.

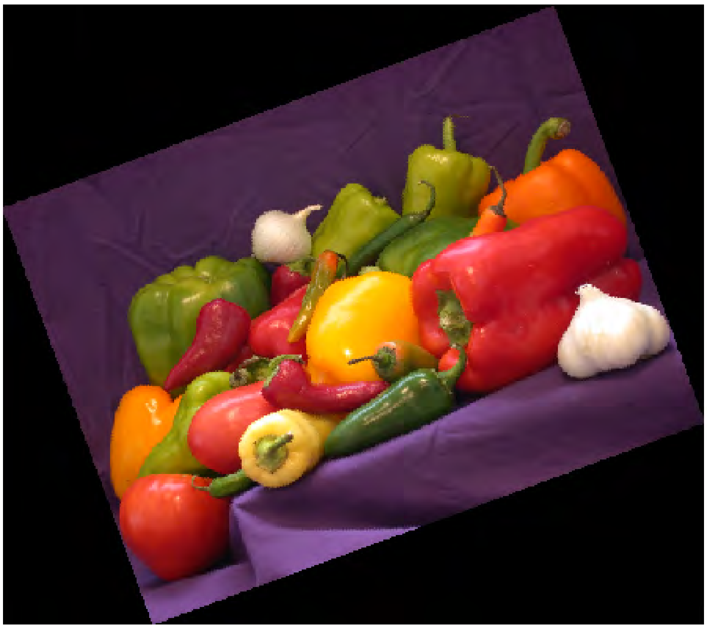


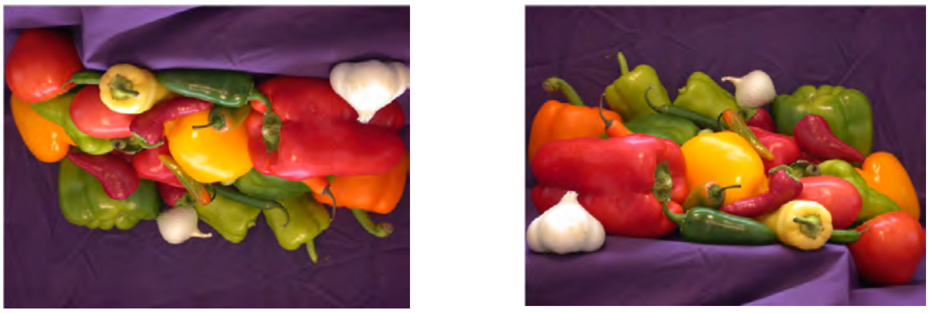
Figure 4.26: Illustration of the MATLAB function imrotate

4.3.4. MATLAB function flipdim

For flipping any image to either horizontal or vertical direction, a built-in function of MATLAB named 'flipdim' can be used. This function can be used as flipdim(image,F) syntax of which example has been shown [see Figure 4.27] for the original image used before [see Figure 4.25]. F has been used to denote whether the flip will be vertical or horizontal. If value of F is used as 1, then the image is flipped with respect to vertical axis [see Figure 4.27a]. On the other hand, value of F equals to 2 indicates a flip in the original image with respect to the horizontal axis [see Figure 4.27b]. This function can be applied on all color types of image *i.e.*, RGB, grayscale or binary.

```
>> VF = flipdim(original_image,1);
>> imshow(VF)
```

```
>> HF = flipdim(original_image,2);
>> imshow(HF)
```



(a) (b)

Figure 4.27: Illustration of the MATLAB function flipdim for (a) vertical flip and (b) horizontal flip

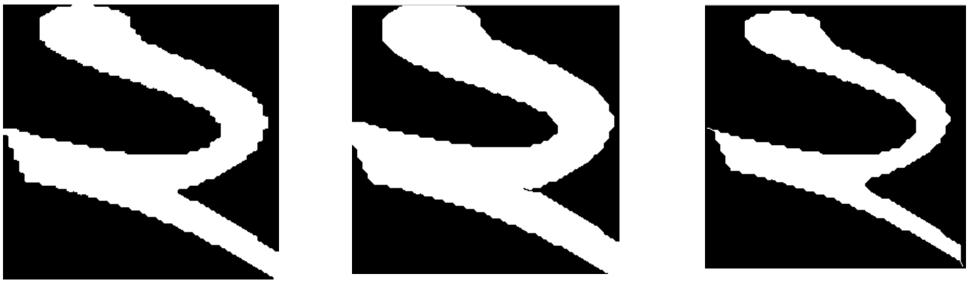
4.3.5. MATLAB functions: bwmorph and nnz

In order to conduct any type of morphological operations in binary images, a built-in function named ‘bwmorph’ is used in MATLAB. Various morphological operations can be conducted using this function. Two significant of those are to perform ‘thick’ or ‘thin’ operations. The syntax of the function is bwmoprh(bwimage,M,P) [see Figure 4.28] where ‘bwimage’ indicates the binary image, M being the morphological operations either ‘thick’ or ‘thin’ and P being the multiplier of the operation.

```
>> I = imread('2.png');
>> imshow(I)
```

```
>> J = bwmorph(I,‘thick’,5)
>> imshow(J)
```

```
>> K = bwmorph(I,‘thin’,5)
>> imshow(K)
```



(a) (b) (c)

Figure 4.28: Illustration of the MATLAB function bwmorph applied on (a) original image to convert it into (b) thick and (c) thin images

First the information about the skeleton of the original image [see Figure 4.28a] is gathered by it. After that, if operation ‘thick’ is applied with a multiplier then it converts the original binary image into a thicker one [see Figure 4.28b] by increasing thickness of the white pixels. If operation ‘thin’ is applied with a multiplier then it converts the original binary image into a thinner one [see Figure 4.28c] by decreasing the white pixels.

As mentioned earlier that binary image is represented as a two dimensional matrix having either 0 or 1 value within the matrix. The white pixels of the binary image are represented by value 1 whereas the black pixels of the binary image are represented by value 0. To count the number of non-zero pixels or elements *i.e.*, white pixels within the binary image, a built-in function ‘nnz’ can be used in MATLAB. The syntax of the function is nnz(bwimage) [see Figure 4.29] where binary image has been represented by ‘bwimage’. The most vital role is played by this function for this particular research as almost all of the geometric features of the characters are extracted through its usage.

```
>> I = imread(‘2.png’);
>> imshow(I)
>> nnz(I)
ans = 22050
```

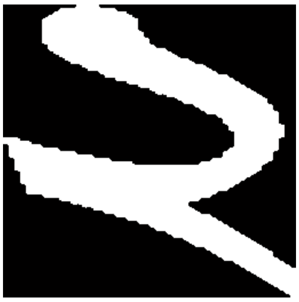


Figure 4.29: Illustration of the MATLAB function nnz

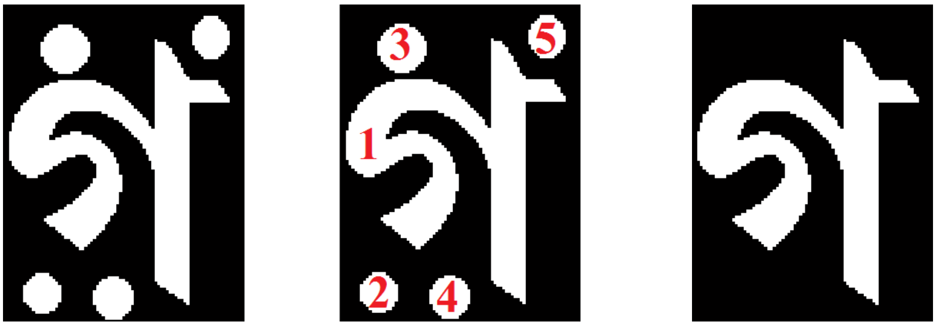
4.3.6. MATLAB functions: bwlabel, regionprops and bwareaopen

For the ALPR process some important operations are conducted using the MATLAB functions ‘bwlabel’, ‘regionprops’ and ‘bwareaopen’. In order to distinguish different objects in a binary image, ‘bwlabel’ function can be used. Different objects are labeled using this function. For example, if a binary image is taken with five disconnected objects [see Figure 4.30] within that image, then ‘bwlabel’ function will be distinguishing all those different objects by labeling them. In that case, connected non-zero elements will be treated as one object. On the other hand, ‘regionprops’ function will give three properties: ‘area’, ‘centroid’ and ‘boundingbox’ of all the different objects within that binary image. The syntax for this function is regionprops(bwimage) [see Figure 4.30a] in which ‘bwimage’ is the binary image. For the image used, five different objects have been identified of which three properties of each have been stored. By the ‘Area’, the area of the white pixels or non-zero elements of an individual object are referred. Coordinates of centroid for each object are stored in ‘Centroid’. Two coordinates of the top right corner and the width as well as the height of the rectangular bounding area for each object are stored in the ‘BoundingBox’. The syntax of ‘bwlabel’ function is bwlabel(bwimage) which has to be defined with a matrix containing two variables. Different objects are labeled with different numbers in the image matrix [see Figure 4.30b]. It is defined by the first variable which is actually another matrix. For the ease of perception, a generalized image has been shown there. Numbers of different objects are stored in the second variable. Another function named ‘bwareaopen’ is used to eliminate all open areas less than any specified pixel within a binary image. The syntax for this function is bwareaopen(bwimage,P) [see Figure 4.30c]. Binary image is denoted by ‘bwimage’ and limit of pixels is denoted by P which is an integer number.

```
>> I = imread(‘Ga.png’);
>> regionprops(I)
ans =
5x1 struct array with
fields:
    Area
    Centroid
    BoundingBox
```

```
>> [L N] = bwlabel(I);
>> N
N = 5
```

```
>> J = bwareaopen(I,300);
>> imshow(J)
```



(a) (b) (c)

Figure 4.30: Illustration of the MATLAB functions (a) regionprops, (b) bwlabel and (c) bwareaopen

If focused on the ‘bwlabel’ operation, it can be seen that MATLAB labels different objects prioritizing the leftmost object first, accordingly others in descending order irrespective of their vertical positions. If two objects have same horizontal coordinate of their leftmost element, then object lying on the top will be prioritized while labeling. But in case of the ALPR operation it should be noted that the first row will have to be evaluated first followed by the second row.

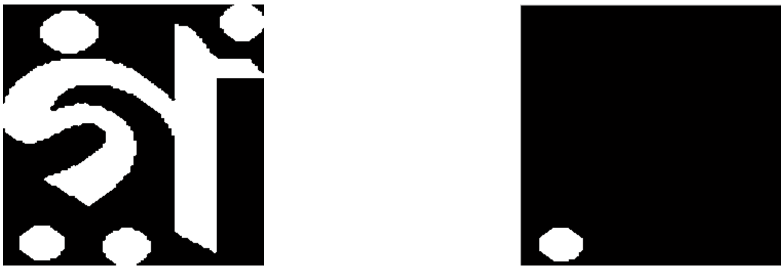
Therefore, some modifications in the algorithm will be required which will be discussed later on.

4.3.7. MATLAB functions: bwboundaries and poly2mask

One important task in ALPR system is to detect the license plate first and for that an important operation is to be conducted which is known as boundary detection. The plate should be identified based on the rectangular dimension criteria. MATLAB provides a built-in function ‘bwboundaries’ using which this operation can be done with some modifications. The syntax of this function is bwboundaries(bwimage) [see Figure 4.31] using which all layers of ‘bwimage’ are distinguished. A cell is created where the numbers of different layers are included each containing all the polygonal coordinates for depicting the layer. It is different from ‘bwlabel’ in a sense that labels are created by ‘bwlabel’ for each object whereas the coordinates are stored using ‘bwboundaries’. For the example made earlier in case of label [see Figure 4.30b], five objects were distinguished of which coordinates are stored in ‘bwboundaries’ in different cells [see Figure 4.31a]. Using those coordinates or any other coordinates, individual object can be shown through white pixels by another built-in function of MATLAB ‘poly2mask’ [see Figure 4.31b] within a layer of black pixels with the predefined size. The syntax of the function is poly2mask(x,y,i,j) where x and y are the coordinates of masking; whereas i and j are the size representatives of the image.

```
>> bbnd = bwboundaries(I)
bbnd =
    [1310x2 double]
    [ 117x2 double]
    [ 146x2 double]
    [ 129x2 double]
    [ 124x2 double]
```

```
>> coordx = bbnd{2,1}(1:end,2:2);
>> coordy = bbnd{2,1}(1:end,1:1);
>> size(I)
ans =
   256   256
>> pymk = poly2mask(coordx, coordy,256, 256);
>> imshow(pymk)
```



(a) (b)

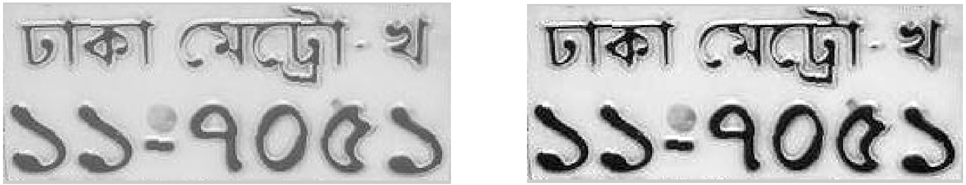
Figure 4.31: Illustration of the MATLAB functions (a) bwboundaries and (b) poly2mask

4.3.8. MATLAB other IP functions

There are some other IP functions in MATLAB, vital roles can be played by those for ALPR. One of those is the 'imadjust' function. Intensity of an image can be adjusted using the intensity range. If no intensity range is selected, by default the bottom 1% and the top 1% of all pixel values are saturated. First of all, the image should be converted into a grayscale image [see Figure 4.32a] for the function application. After obtaining the grayscale image, the syntax imadjust(gsimage) can be used to adjust grayscale image named 'gsimage' [see Figure 4.32b].

```
>> i = imread('rgb_plate.jpg');
>> i1 = rgb2bw(i);
>> imshow(i1)
```

```
>> i2 = imadjust(i1);
>> imshow(i2)
```



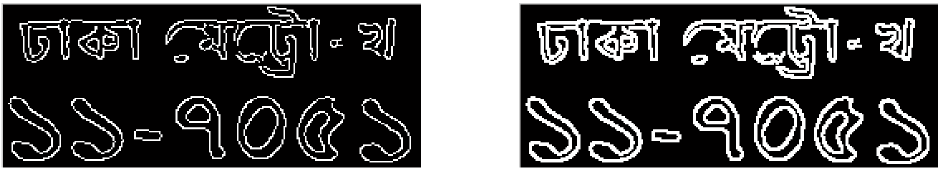
(a) (b)

Figure 4.32: Illustration of the MATLAB function imadjust over (a) a grayscale image to convert it into (b) an intensity adjusted image

For identification of possible edges within the image, built-in function 'edge' can be used. The syntax of this function is edge(gsimage, 'M', T) where 'gsimage' is the grayscale image which will be converted into edged image. The method is defined by M and a threshold value is defined by T. There are different methods for edge detection among which the method proposed by Canny et al. [133] can be used being popular. This method can be automatically applied [see Figure 4.33a] by MATLAB through writing 'canny' inside the function 'edge'. A binary output is obtained from this function containing value of 1s at the edges keeping 0s on the remaining portions. Another function 'imdilate' is used to dilate the edged image [see Figure 4.33b]. Dilation should be conducted creating structural elements around the edges of the image. It is basically a two-dimensional, morphological, structural element created within the binary image. Different shapes of 2-D structural element are available like: diamond, disk, octagon, line, rectangle, square etc. Square shape is used according to the syntax of strel('square',P) [see Figure 4.33b] within which the pixel value for the square shaped elements is denoted by P. After that the syntax imdilate(egimage,se) [see Figure 4.33b] is used to dilate the edged image named 'egimage'.

```
>> i3 = edge(i2,‘canny’,0.4);
>> imshow(i3)
```

```
>> se = strel(‘square’,2);
>> i4 = imdilate(i3,se);
>> imshow(i4)
```



(a) (b)

Figure 4.33: Illustration of the MATLAB functions (a) edge and (b) imdilate

One important benefit of the function ‘imdilate’ is that, if some edges are not connected due to poor image quality, application of ‘imdilate’ could connect them together. After that if holes are to be filled up within a binary image, a built-in function of MATLAB named ‘imfill’ [see Figure 4.34a] can be used. The syntax of the function is imfill(bwimage, ‘holes’) for filling holes within the binary image ‘bwimage’. However for explication, the same license plate image have been used as illustrated before [see Figure 4.32]. But in this research, all these IP functions will be used for filling holes of the area bounded by the rectangular boundary of the image. Basically, it is done to mask the region of the license plate to extract that region from the gross image so that characters can be extracted afterwards.

```
>> i4 = imfill(i3,'holes');
>> imshow(i4)
```

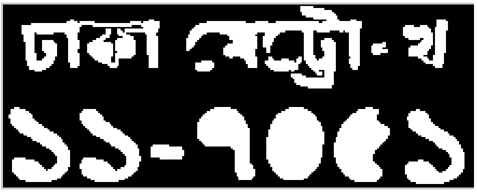
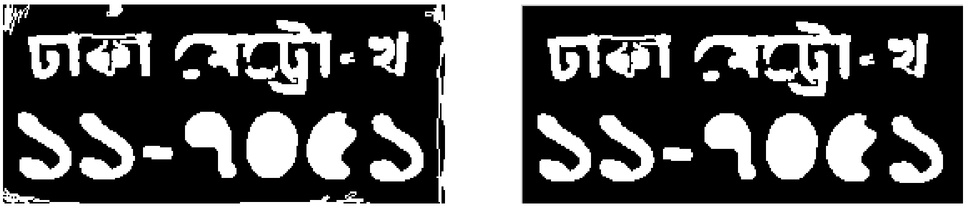


Figure 4.34: Illustration of the MATLAB function imfill

Another important function of IP in MATLAB which can be used for ALPR system is ‘imclearborder’. This function can be used to suppress structures in an image that are lighter than their surroundings and that are connected to the image border. Both grayscale and binary images can be modified using this function. The syntax of this function is imclearborder(image) [see Figure 4.35] where the ‘image’ may be either grayscale or binary. If noises in borders of a binary license plate [see Figure 4.35a] image are to be removed, this function can be applied with good results. In that case, application of ‘imclearborder’ would clear the white pixels [see Figure 4.35b] around the border. While using this function, safety should be maintained as the characters somehow connected to the border due to lack of margin or padding, might be removed by this function application.

```
>> i = imread('rgb_lplate.png');       >> j = imclearborder(i);
>> i = im2bw(i);                       >> imshow(j)
>> imshow(i)
```



(a) (b)

Figure 4.35: Illustration of the MATLAB function imclearborder to remove border pixels of (a) an image to (b) a cleared one

In the meantime, all relevant IP functions which can be used in MATLAB for developing ALPR model have been illustrated with proper examples. All those functions will be used to extract characters and their geometric features which will be used in the ANFIS model development discussed later on.

4.4. ANFIS operations in MATLAB

As all the relevant procedures for extracting character properties through IP have been discussed so far, those properties can be used for the ANFIS modeling now with proper justifications. As discussed in the Methodology and Data Collection chapter that MATLAB has a dedicated GUI of ANFIS editor, ANFIS can be operated easily using that editor. ANFIS editor can be called through ‘anfisedit’ command in MATLAB. A pop-up window [see Figure 3.3] is generated by this command. Four major portions can be discussed within the ANFIS editor namely, Load data, Generate FIS, Train FIS, Test FIS. All the procedures with proper explanations will be done from now on.

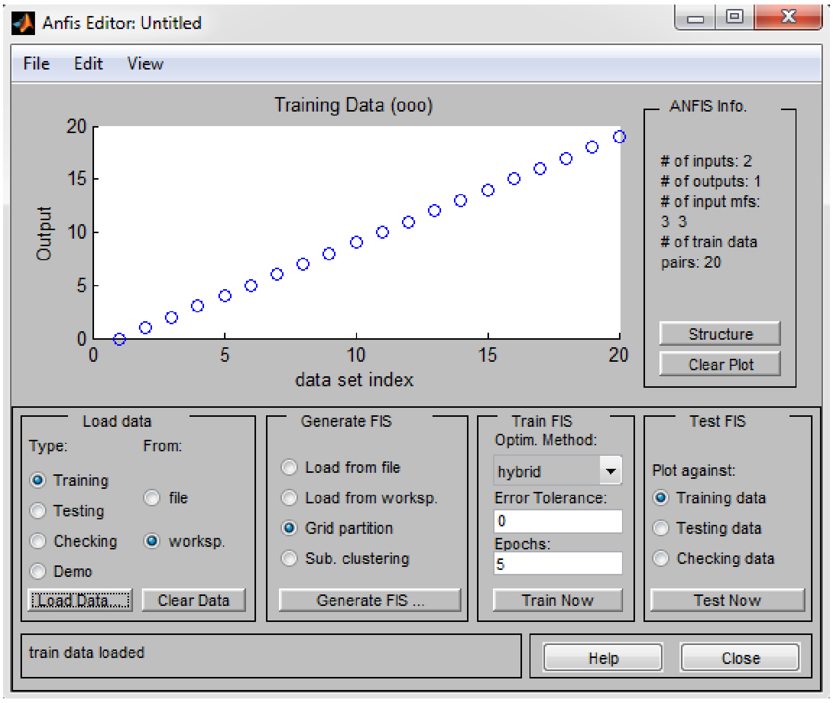
4.4.1. ANFIS editor: load data

First step in ANFIS modeling is the ‘Load data’. A data table should be generated as mentioned in the Table 4.2 containing the input vector(s) of the ANFIS and the output vector. Data type might be training, testing, checking or demo. Data can be loaded from any matrix created within MATLAB or data can be imported from any exterior file as well. Such a matrix should be created that would contain input vector(s) for the initial column(s) and the last column of the matrix would be filled up with output vector. No additional row or column should be included within that matrix. As for the example given in the Table 4.2, first three columns except the first row should be the desired matrix for the ANFIS model. It was also discussed earlier that numeric outputs will be designated to interpret non-numeric characters in ANFIS. One recognized character is set after each ANFIS output.

Table 4.2: Data for ALPR Model Development

$out1$	$out2$	**ANFIS output**	**Recognized character**
1	0	0	“০” (0)
2	0	1	“১” (1)
3	0	2	“২” (2)
4	0	3	“৩” (3)
5	0	4	“৪” (4)
6	0	5	“৫” (5)
7	0	6	“৬” (6)
8	0	7	“৭” (7)
9	0	8	“৮” (8)
10	0	9	“৯” (9)
11	0	10	“গ” (Ga)
12	0	11	“খ” (Kha)
13	0	12	“হ” (Ha)
0	1	13	“সিলেট” (Sylhet)
0	2	14	“রাজশাহী” (Rajshahi)
0	3	15	“বরিশাল” (Barisal)
0	4	16	“চট্টগ্রাম” (Chittagong)
0	5	17	“মেট্রো” (Metro)
0	6	18	“খুলনা” (Khulna)
0	7	19	“ঢাকা” (Dhaka)

All data mentioned above in the Table 4.2 are combined to generate a matrix which have been called into ANFIS editor afterwards. For input matrix, ‘Load data’ from workshop has been selected from which Output vs. data set index plot [see Figure 4.36] has been obtained. The plot can be cleared by clicking on the ‘Clear Plot’ situated in the right side of the ANFIS editor. Also, the data can be cleared by using ‘Clear Data’ situated in the bottom portion of the ANFIS editor.



Anfis Editor: Untitled

ANFIS Info.

# of inputs: 2
# of outputs: 1
# of input mfs:
3 3
# of train data
pairs: 20

Load data
Type: ● Training ○ Testing ○ Checking ○ Demo
From: ○ file ● worksp.

Generate FIS
○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Train FIS
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 5

Test FIS
Plot against: ● Training data ○ Testing data ○ Checking data

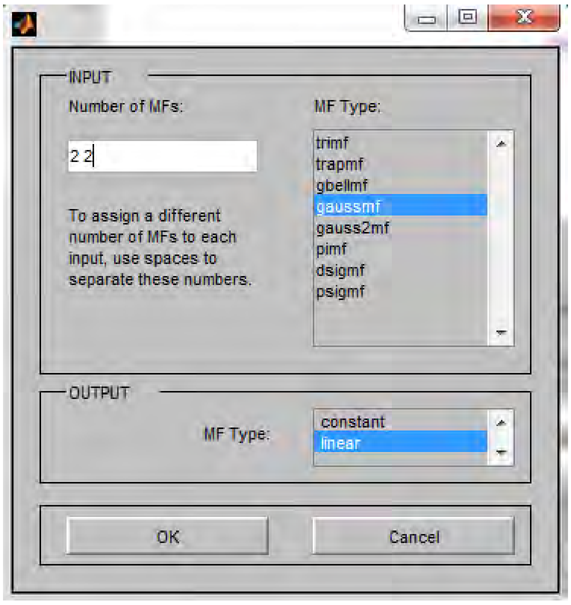
train data loaded

Figure 4.36: Output vs. data set index in ANFIS editor

4.4.2. ANFIS editor: generate FIS

Second portion of the ANFIS editor is governed by FIS generation. There are four options within ANFIS editor for generating FIS. FIS file can be generated from any file that has been already saved in the directory or that can also be called from workshop if saved in the command. Other two options are 'Grid partition' and 'Sub. Clustering' indicating subtracting clustering. Subtracting clustering has an advantage of reducing number of rules which might cost operating time in MATLAB. Also, there is another option in ANFIS which can be obtained using a built-in function named 'genfis3' which generates a FIS using fuzzy c-means (FCM) clustering by extracting a set of rules that models the data behavior. However, this FCM clustering is not available in ANFIS editor rather it should have been performed using command lines therefore not desired.

As the extent of the inputs and output is very small, 'Grid partition' can be used without affecting much running times. After selecting 'Grid partition', if the button 'Generate FIS' is clicked, then it opens up another pop-up window [see Figure 4.37] showing the selection options for MF type and its number. There are different types of MF available in the ANFIS editor namely: 'trimf', 'trapmf', 'gbellmf', 'gaussmf', 'gaussmf', 'pimf', 'dsigmf', 'psigmf'. All these are indicators of the shapes of different MFs. They might be triangular, trapezoidal, Gaussian curve etc. in shape. Number of MFs can also be selected for each input type. Minimum two MFs should be defined per input. Finally, there is an option for selecting output type of MF, whether it should be constant or linear. Scope is extended for the output to adjust variability by selecting linear whereas that is restricted by using constant.



INPUT

Number of MFs: 2 2

To assign a different number of MFs to each input, use spaces to separate these numbers.

MF Type: gaussmf

OUTPUT

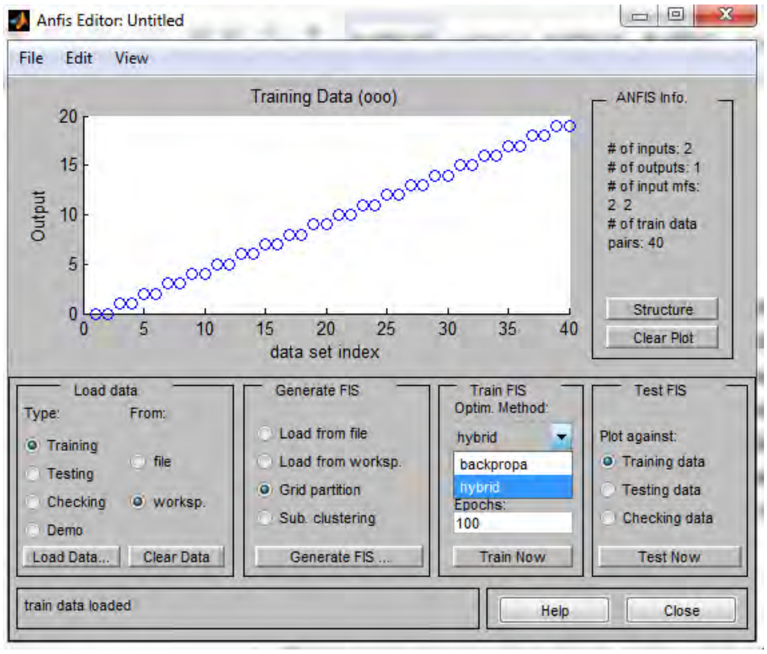
MF Type: linear

Figure 4.37: Defining MF type and number in ANFIS editor

Whatever the type of MF and the number are used, all these information are shown at the top right corner of the ANFIS editor [see Figure 4.36]. As the number of MFs per input is increased, both operating time and number of rules for ANFIS are increased. Therefore, keeping number of MFs as small as possible is desired. Different types of MF will be used to judge the best suited FIS which will be used for the ALPR system later on.

4.4.3. ANFIS editor: train FIS

Third portion of ANFIS editor deals with the training parameters of ANFIS which should be used if training data set is being used. Also, it can be used to modify learning while testing set is used in this editor after the training is done. There are two optimization algorithms available for training FIS [see Figure 4.38] namely, hybrid optimization and back propagation. From different studies as made by Sarangi et al. [134], it is evident that the hybrid algorithm performs better than the back propagation as back propagation is trapped by local minima. Therefore, hybrid algorithm has been chosen for the optimization in the research as well. Also, there are options for error tolerance and epochs used in training the FIS. A range is allowed for errors in error tolerance, if no range is fixed then it is automatically set to zero. Number of epochs are represented by the epochs in ANFIS editor. Epoch is basically a measure of the number of times all of the training vectors that are used once to update the weights. Errors can be reduced by using higher epoch size, but at the same time operating time might be increased too. After clicking the button ‘Train Now’ training of the FIS is started and errors are displayed in the bottom portion of the ANFIS editor. Command descriptions are continuously updated on that portion.



# of inputs: 2
# of outputs: 1
# of input mfs:
2 2
# of train data
pairs: 40

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid
backpropa
hybrid
Epochs:
100

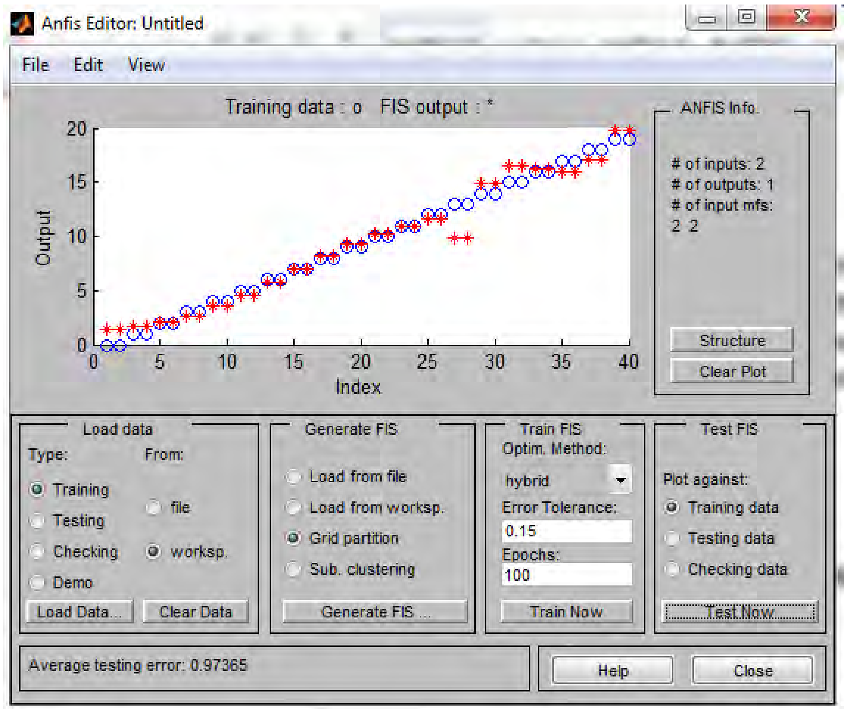
Plot against: ● Training data ○ Testing data ○ Checking data

train data loaded

Figure 4.38: Optimization methods for training FIS in ANFIS editor

4.4.4. ANFIS editor: test FIS

The final portion in ANFIS editor is the ‘Test FIS’ using which testing of the generated FIS can be done. There are three options [see Figure 4.39] in the test FIS portion: plot against ‘Training data’, ‘Testing data’ and ‘Checking data’. Using these three options, ‘Test FIS’ data can be plotted against those data as it was introduced in ‘Load data’ portion initially.



ANFIS Info.: # of inputs: 2; # of outputs: 1; # of input mfs: 2 2

Load data — Type: ● Training ○ Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS: ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0.15; Epochs: 100

Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Average testing error: 0.97365

Figure 4.39: Testing FIS by plotting output values in ANFIS editor

It can be seen from the plot that both training data and FIS output have been plotted for mutual comparison. Training data have been plotted as ‘Training data’ option was selected in the ‘Load data’ option initially. From the plot, performance of the FIS output can be visualized comprehensively. After testing FIS, it can be saved in the directory if the performance is satisfactory so that it can be utilized afterwards as discussed in the generate FIS section.

FIS formatted file can be called using ‘readfis’ function [see Figure 4.40] in any command window if the file is situated in the current directory. Also, using ANFIS model new output can be obtained if inputs are provided within another function named ‘evalfis’.

```
>> fis = readfis('Training_FIS');
>> input = [0 1; 1 0; 3 0]
input =
     0     1
     1     0
     3     0
>> output = evalfis(input,fis)
output =
   12.9993
    0.0002
    1.9999
```

Figure 4.40: Reading FIS and evaluating FIS output through MATLAB functions readfis and evalfis

Thereby MATLAB ANFIS editor and other relevant functions can be used to utilize the parameters containing geometric features of different characters discussed so far. Thus, combination of both IP and ANFIS can be utilized for the ALPR model as a whole.

4.5. Model development using IP and ANFIS algorithms

So far, all the MATLAB functions relevant to the ALPR system development for both IP and ANFIS operations have been illustrated with examples. Geometric features and rational parameters have also been announced from the analysis of the data. Now combination of those operations and features will be made to get the desired model into action. As it was mentioned earlier in this chapter that two sub-models will be used to create the complete model of ALPR. First one is the Training model and the second one being the Recognizing model. Both of these two sub-models will be diagramed to fulfill the complete ALPR model using MATLAB.

4.5.1. Assumptions made during model development

From the widely discussed literature review and methodology in the earlier chapters, some basic assumptions of the license plates are summarized based on which the algorithm for ALPR has been developed:

i. An image does not contain more than one license plate
ii. A license plate consumes more than 5% of the total image area
iii. License plates are rectangular where the height to width ratio is close to 0.5
iv. Background color of the license plate is white
v. License plates comprising of two rows of characters
vi. Two rows are evenly distributed concerning the horizontal line of symmetry
vii. Each connected component has been treated as one character
viii. Hyphens, dots etc. have been ignored considering it unnecessary
ix. Each character area is not less than 30% of the average of all proper character areas
x. Letters are assumed to be oriented from left to right (e.g., English, Bangla etc.) which indicates Left to Right (LTR) writing system
xi. Marginal rotation indicates a rotation not exceeding ± 45o of the actual horizontal alignment
xii. There are 9 desired characters in the license plate in total

xiii. No character, i.e., the connected component in the license plate would contain less than 5% area of the actual license plate area

Based on these assumptions, the RR digital license plate of Bangladesh has been taken as reference [see Figure 3.4] as it contains two rows, usually found with a marginal rotation seen in real-life scenarios. It contains a combination of the metropolitan city name, vehicle class number, and numerals including hyphens.

4.5.2. Development of the training model

Training model has been created to train the device about different geometric features with their corresponding outputs. This model is saved in a directory where all the training data *i.e.,* images of each character are stored in the computer device. A MATLAB extension file named ‘FeatureTraining.m’ [see Appendix-B] has been programmed which executes all the tasks in the model.

First of all, all the additional windows in the MATLAB except the editor and main window are closed. After that, all the variables stored within the workspace of MATLAB are cleared before running the new program. Also, the command window is cleared before the new program gets started. All these operations will be automatically done once the program is run.

After the MATLAB command window is ready to execute the program, first of all user inputs are asked by the program. In this particular case, inputs are the image files each containing one individual character covered in this research. All relevant image formats are supported by the selection. It should be noted that, there must be at least two selections of the images done as user input. Otherwise, warning will be returned to the user in the command window for giving at least two inputs. Also, if no input is given, then input will be asked by the command window from the user as well.

Next step of the model is to convert the input images into binary images. For this conversion, the color of the existing images is checked. If images are found to be in RGB format then those are converted into grayscale images initially. After that, the threshold of the grayscale images is calculated for converting these images into binary images. If already images are in binary images, then no modification is done in this step.

Once the binary images have been obtained, those images are corrected using ‘bwareaopen’ function. Any noises or unnecessary dots within the images are removed by this operation. After that, the images are cropped up to their extents to get the original character regions. While performing this operation, if any rows of the images are found to be completely blank, then those rows are eliminated from those images to get modified images. Therefore, clean and stretched images are obtained by these operations in this step.

Resizing is the next step for the training model to obtain the clean and stretched images as it was required to extract geometric features from the resized images. A customized function ‘seperatedcharacter’ (see Appendix-A) is used to distinguish between single-character and multi-character types. Single-character type images are resized to 256 px × 256 px whereas multi-character types are resized to 256 px × 512 px dimensions.

After resizing the images according to the character types, characters in those images are corrected morphologically. If the thickness of the characters in the images are out of predefined range, then morphological correction is operated to convert the characters within that range. For the training model, this range has been set between 30% and 40% as percentage of the non-zero elements within the images.

After getting final images, extraction of different geometric features are done accordingly to obtain the input values for the ANFIS operation. Basically, percentages of the non-zero elements of different portions of the characters are calculated to get those features. Single-type and multi-type characters are the initial divisions for that.

Combining all the features, inputs for the ANFIS operation are evaluated according to the order mentioned earlier in this chapter. A matrix is obtained from the program after execution containing all the necessary input values for the ANFIS operation. Values of this matrix containing inputs regarding different characters are trained in the ANFIS editor afterwards. Different MF types and numbers are used to give trials using various epochs to get the optimized ANFIS model. Training of ANFIS is saved as ‘Training_FIS.fis’ file in the directory of recognizing model which will be discussed later on. Data for this training FIS model are rendered from the data discussed earlier in this chapter in the Table 4.2. This training FIS will be used for the matching of the characters in the ALPR system. A complete process for the training model that has been illustrated can be shown in a flow diagram [see Figure 4.41].

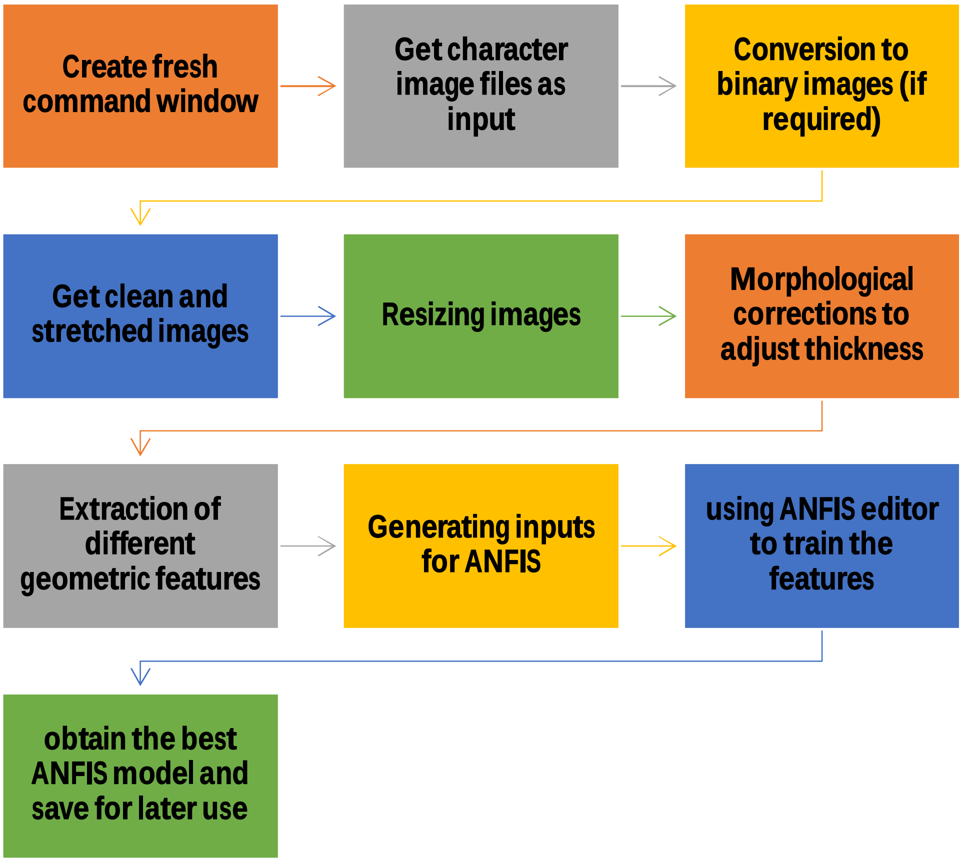

Create fresh command window
Get character image files as input
Conversion to binary images (if required)
Get clean and stretched images
Resizing images
Morphological corrections to adjust thickness
Extraction of different geometric features
Generating inputs for ANFIS
using ANFIS editor to train the features
obtain the best ANFIS model and save for later use

Figure 4.41: Complete process of the proposed training model

Different constant type MFs varying with numbers as mentioned earlier in this chapter have been used to observe the performance of ANFIS for those MFs. For the ease of calculation, epochs of 10, 50 and 100 have been used for each MF type. Also, there are two input vectors for ANFIS as discussed earlier in this chapter, number of MFs for these two input vectors are taken as [2,2], [2,3], [3,2] and [3,3] for observing the performance. All those epoch errors have been tabulated in the Table 4.3 to compare the best ANFIS MF type for the ALPR model. From this comparison, the best MF would be taken for the model based on the relative number of epochs and cost of operating time.

Table 4.3: Epoch Errors for Different MF Numbers and Types in ANFIS

		Epoch Error (× 10^{-4})			
	Nos. of MFs	**[2,2]**	**[2,3]**	**[3,2]**	**[3,3]**
Type of MFs	**Epochs**				
trimf	10	17.90	58.24	24.89	203.30
	50	0.46	58.24	0.64	54.36
	100	**0.15**	58.24	0.15	54.36

Contd. Table 4.3

		Epoch Error (× 10^{-4})			
	Nos. of MFs	**[2,2]**	**[2,3]**	**[3,2]**	**[3,3]**
Type of MFs	**Epochs**				
trapmf	10	1435.20	476.24	1171.00	463.82
	50	1362.00	0.01	1118.30	0.01
	100	1266.60	0.01	1054.70	0.01
gbellmf	10	1251.00	336.29	1015.60	330.39
	50	1122.10	117.74	921.96	117.48
	100	921.06	19.16	796.69	19.16
gaussmf	10	1100.80	413.36	910.28	400.80
	50	1041.60	257.82	862.51	254.72
	100	974.96	14.67	811.63	14.69
gauss2mf	10	1396.80	280.18	1144.80	277.57
	50	1256.70	0.02	1036.70	0.02
	100	924.88	0.02	883.30	0.02
pimf	10	1502.00	259.61	1217.90	257.65
	50	1430.80	0.01	1162.00	0.01
	100	1333.10	0.01	1091.10	0.01
dsigmf	10	1497.20	324.97	1195.30	320.83
	50	1387.30	70.67	1115.30	70.63
	100	1255.90	7.64	1013.30	7.64
psigmf	10	1497.20	324.97	1195.30	320.83
	50	1387.30	70.67	1115.30	70.63
	100	1255.90	7.64	1013.30	7.64

From the performances of different MFs, it can be observed that 'trimf' alone dictates the ANFIS while only 2 MFs are used for each input. Epoch error of 0.15x10^{-4} has been observed for 100 epochs in 'trimf' which is very small relative to all other MFs and their corresponding epochs. Lesser error than 'trimf' can be found for increased number of MFs in 'trapmf'' and 'pimf' as 0.01x10^{-4}, if the second input 3 MFs keeping the first one as 2 MFs even for 50 epochs. Similar result is found by increasing the first input as 3 MFs as well. Epoch error of 0.02x10^{-4} can be obtained for 'gausss2mf' in [2,3] and [3,3] inputs which is close to the relevant comparisons but still not less than the 'trimf' for [2,2]. As the difference of 'trimf' with both 'trapmf' and 'pimf' are not too high compared to the increase of the number of MFs, 'trimf' of 100 epochs can be nominated for the training model. It is also because just one increase in the number of MF could be costly in the sense of operating time compared to the number epochs as well. In addition to that, it is mentioned from the very beginning of the chapter that ALPR is a complex operation where time efficiency is an important factor.

Therefore, the output of MF ‘trimf’ with 100 epochs have been selected as the training model outcome which will be used to recognize characters in the recognizing model discussed later on.

4.5.3. Development of the recognizing model

After the process of training model is completed and the ‘Training_FIS.fis’ file is obtained, the next sub-model named as recognizing model can be initiated. This model is the desired ALPR model which will be used by the clients of this model. The main program will be stored in the immediate parental directory of the training model as mentioned earlier having file name ‘RecognizeLicensePlate.m’ [see Appendix-C] indicating MATLAB file extension. In addition to that, all the images of the license plates that are to be recognized and the relevant functions of recognizing model should be kept in the same folder as well.

First of all, similar to the training model, all the additional windows in the MATLAB except the editor and main window are closed. After that, all the variables stored within the workspace of MATLAB are cleared before running the new program. Also, the command window is cleared before the new program gets started. These operations will be automatically handled once the run option is selected by the user for the program.

The next step is the asking for a raw image of any license plate in the form of user input. If no file is selected, the program will ask for adding an RGB image file of license plate in the command window. If multiple images are selected, the last selected one will be considered for the sample during execution.

Once the raw image of license plate has been taken as input, the program will send it to a customized function named ‘newrefine1’ [see Appendix-D], which will first convert the RGB image into a binary one. Next, the binary image will be divided into different layers by the help of ‘bwboundaries’ function of the MATLAB. These bounded regions will be used to identify possible license plate region within the image in the next step.

For the identification of the license plate boundary among all the boundaries, three conditions in total are used. First one is the width to height ratio of the bounded region, second one is the rectangularity of the region and third one is the background color of the region. As mentioned earlier about the license plate used in Bangladesh have a recommended dimension, therefore the width to height ratio can be a recognizing factor and that is why it is used. Secondly, the area bounded by the region is checked whether the maximum and the minimum coordinates correspond to the similar area of the original area bounded by the region. Thirdly, the background color is checked as it is known that the license plate is a plate having white like color background as mentioned earlier in this chapter. All these conditions are used to detect the actual license plate region successfully to move this to the next step. While performing this step, three customized functions namely, 'boundary_property.m' [see Appendix-E], 'polygonarea.m' [see Appendix-F] and 'nmax.m' [see Appendix-G] have been used.

Once the actual region of the license plate is obtained, the alignment of the license plate with respect to the image perspective is calculated. It is done by taking the top left point and top right point of the bounded region. Considering rectangular shape of the license plate, rotation angle of the plate is calculated. This rotation is applied on the image to obtain the original horizontal alignment of the license plate. It should be noted that after the actual license plate region has been determined, the rotation is applied on the RGB image keeping only the license plate portion eliminating other regions using the coordinates of the bounded region. Also, it should be mentioned that the license plate region is bounded by a linewidth so that the operations in the next steps become easier.

Once the rotated image has been obtained, it is sent to another customized function in MATLAB named 'newrefine2.m' [see Appendix-H] to modify the image further. Within this function, first the RGB image is converted into a grayscale image which in turns gets converted into an edged image having the function 'edge' applied. Eventually the edged image gets converted into binary image. Then eliminating any possibility of broken edge lines along the border of the license plate, structural elements are created and the image is dilated. Once the region is completely bounded by the edge border, 'imfill' function is used to completely create a void space having white pixels within the actual license plate region keeping the remaining portion as black. For avoiding any additional white pixels outside of that region, 'bwareaopen' function is used. In addition to that, any thin but long connected layers of white pixels are removed using the customized program. Finally, the region of the license plate boundary is modified through eliminating any existing white pixels around the border. For this operation, the built-in function 'imclearborder' is used. Operation of the function 'newrefine2' is completed by this step result of which is transferred to the next step in the original 'RecognizeLicensePlate.m' program.

Once the computer-generated initial corrections have been applied, the modified image is sent to the user by showing the modified binary image with the cropped license plate region. It is asked whether further bounding should be applied on the license plate region or not. If four corners are selected by the user, then it takes selected portion as input, otherwise the previously modified image is transferred to the next step.

Once the correctly oriented binary image of the license plate containing the characters are obtained, identification of individual character from the license plate is conducted. For that, assuming the characters being connected objects, function 'regionprops' is used. To avoid any disconnected object from the original connected objects which might be the taken as individual character by mistake, an algorithm of overlapping ratio is used in the form of a customized function 'overlapratio.m' [see Appendix-I].

Once all the characters are separated, it should be seen whether the characters are sorted according to its order or not. As discussed earlier in this chapter, MATLAB by default labels the objects according the preference of their horizontal position. But in case of Bangladeshi license plates, it was shown that two rows of characters are made where first row should be read first followed by the second one. Therefore, reordering of the characters is done in this step using some programming algorithms where the license plate is divided into two portions about the horizontal axis passing through middle of the image. It is assumed that the rows of the license plate are equally spaced along the height with proper extents. Any object lying above that middle axis would be ordered first starting from the left to the right. After that, the same operation will be done in the second half and these objects will be following the order next to the previous objects.

While discussing about the license plate convention in Bangladesh, it was seen that [see Figure 4.1], license plates in Bangladesh have at least two hyphens along with other characters. But those hyphens are of bare usage. Therefore, those hyphens should be ignored including any other unnecessary dots still existing within the license plate images. As those unnecessary dots and hyphens are already ordered, those should be eliminated with the geometric properties programmed in the recognizing model.

After clearing all unnecessary dots and hyphens, the finalized characters are obtained which can be sent to the customized function 'extractfeature.m' [see Appendix-J] to extract geometric features from those characters. This function is similar to the training model and 'separatedfunction.m' function is also used here. It is the same function which was used in the training model. The only difference is the location of this function is in the directory of the recognizing model folder, whereas the previous one's location was in the directory of training model folder.

Once the characters are sent to the 'extracfeature' function, all operations are similar to that of the training model until the input vectors containing geometric features are obtained. Starting of this function is done with the cleaning and stretching of the images of the characters. After that, the image is resized according to the single-character and multi-character types. Needless of mentioning that, 'separatedcharacter.m' file is used to undertake this operation. To keep line with the training model and the characters, morphological corrections are applied to get the desired thickness of the characters after that. Once the characters are ready, geometric features are extracted from the resized character images which are the output of the 'extractfeature' function. It should be noted that, even if any license plate region is detected but no character can be extracted from that region, it would be treated as no license plate status. This concept will be used in the Model Validation chapter later on.

Once all inputs are obtained from the function, the algorithm is turned back into the original program. There the 'Training_FIS.fis' file is read for the ANFIS operation to detect the characters. After obtaining outputs of the characters from ANFIS operation, the characters are designated to its original names using some customized algorithms. Finally, the characters are oriented in text format in the command window and the model completes its execution.

The total process has been diagramed in a figure [see Figure 4.42] indicating all major steps in the sub-model named as recognizing model. From this model, the applicability and the ALPR efficiency could be validated for the Bangladeshi license plates.

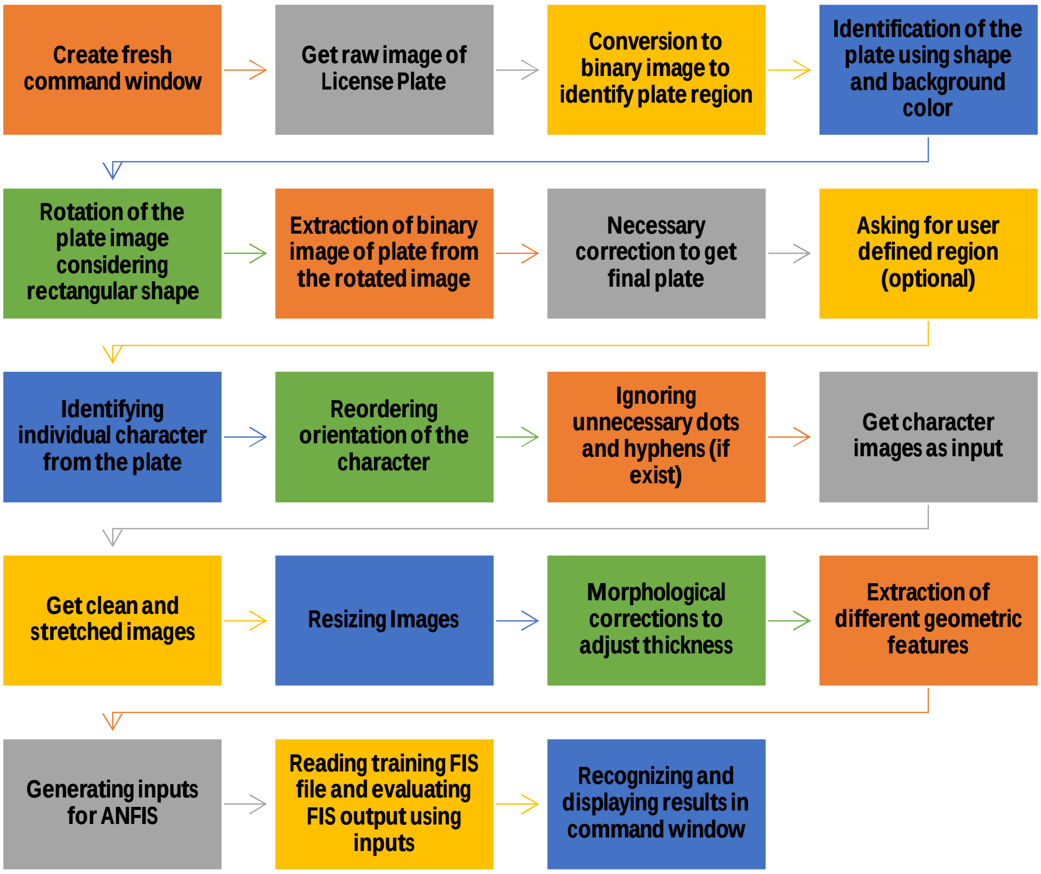

Create fresh command window
Get raw image of License Plate
Conversion to binary image to identify plate region
Identification of the plate using shape and background color
Rotation of the plate image considering rectangular shape
Extraction of binary image of plate from the rotated image
Necessary correction to get final plate
Asking for user defined region (optional)
Identifying individual character from the plate
Reordering orientation of the character
Ignoring unnecessary dots and hyphens (if exist)
Get character images as input
Get clean and stretched images
Resizing Images
Morphological corrections to adjust thickness
Extraction of different geometric features
Generating inputs for ANFIS
Reading training FIS file and evaluating FIS output using inputs
Recognizing and displaying results in command window

Figure 4.42: Complete process of the proposed recognizing model

4.6. Overview

From the beginning of this chapter, analysis of the collected image data of the license plates has been performed and geometric features of all characters are discussed in details. Also, different built-in functions of MATLAB regarding IP and ANFIS have been illustrated with proper examples. Finally, using help of those features and functions, a model has been developed containing two sub-models for training and recognizing respectively. During developing this model, best suit ANFIS MF has been set and that file has been saved for further usage. The complete model will be used for the ALPR system that is to be validated using raw images of license plates in Bangladesh. A set of collected data has been differentiated initially for the validation purpose and its usage will be discussed in the next chapter thoroughly.

CHAPTER 5
MODEL VALIDATION

5.1. Introduction

In the previous chapter, the ALPR model has been developed with the data collected for the model development. To vindicate the accuracy of detection by this model, some license plates are to be checked through it. As the data collected initially were divided into two parts which had a set of data for validation purpose, that set will be used here. After discussing the methodology of validating the model, it will be applied. Validation will be done for overall accuracy of detection, also the accuracy of all characters will be done individually. Model outcome for unknown characters will also be checked. Finally, some conclusions on the developed model will be made based on the performance of the model.

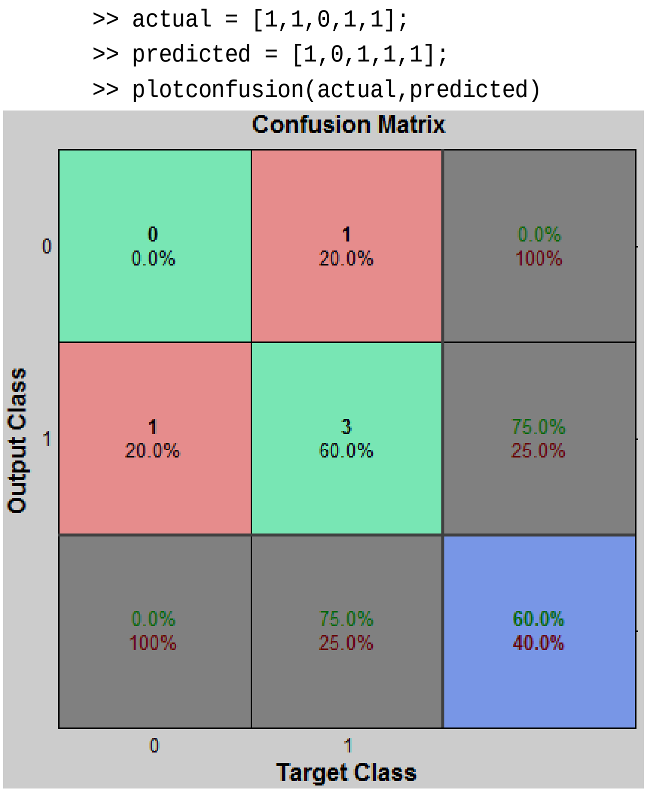
5.2. Validation methodology

Different ways are available for validating a model. The simplest and most common way is to check the accuracy measure of the data obtained from the model outcome and the actual outcome based on field data. It should be kept in mind that the data used for validation should be different from the data used in calibrating or developing the model. However, the conventional method of goodness of fit has been used in different researches as discussed earlier [128]. It has an advantage of simplicity to compare the results directly. But at the same time, it has just one way of interpretation. Therefore, lack of interpreting outcomes is persistent. Despite its simplicity, lack of credibility is another issue for this method being unable to predict long term stability and implications.

There are two major aspects of validating the model. First one is to check the ability to detect original license plate region by the model. It would be relatively less acute in manner but significant in the ALPR process. Because, unless the license plate is specified, no character can be extracted through the next algorithms of the model. The fundamental basis of the model will be malformed thereby. The second aspect is the ability to detect the individual character by the model. It could be modified by the application of some minor corrections within the model algorithms. Also, it would vary for different characters as well because of the variations in geometry. Validation of character detections can be further subdivided into single-character type and multi-character type as the basic principles for distinguishing these two categories were different from the beginning. Therefore, a meticulous analysis on these two categories would verify the philosophy.

Confusion matrix can shed light on the robustness of any prediction [135]. Unlike accuracy, it is not a metric to evaluate a model. This matrix can be used to generate multiple metrics which could give various information about the robustness of any prediction. Deeper can be revealed by this matrix than accuracy measure as it can show both correct and incorrect predictions of each class as well. Additionally, in case of binary classification tasks, a simple confusion matrix of 2x2 can be used. There might be an argument on using this matrix for evaluating different classes to validate the model because ALPR is an operation which has barely any significance with the class prediction. Rather ALPR is governed by the overall accuracy of the system. Simple riposte to this reasoning would be the purpose of model improvement for further application in future. Although the total model is accused of wrong outputs from ALPR if either output is malfunctioning, identification of the performances in different classes would give clear information about the algorithm. Improvement in specific algorithm might be a determiner of higher accuracy of the model with lesser efforts in the long run.

Both the actual and the predicted results are placed in the confusion matrix [see Figure 5.1] against each other for different classes. It is obtained using a function named ‘plotconfusion’ in MATLAB. The syntax of this function is plotconfusion(target,output) where the target result vector is indicated by ‘target’ and predicted output vector having the same length by ‘output’. Target classes are plotted horizontally whereas the output classes are plotted vertically. Within every cell of the matrix, the number of matches is provided, below which the percentage of those matches with respect to the total data are rendered as well. All true outcomes are shown using green backgrounds, which are basically the diagonal elements of the matrix. The false outcomes irrespective of false positives or false negatives are shown using red backgrounds. End rows and columns are rendered with percentages of correct and wrong predictions written in green and red texts respectively. For each class, these percentages are shown in gray backgrounds whereas the total percentage of all data are shown as the last element of the matrix in blue background. From these matrix, class specific outcomes and the performance of the model can be evaluated in comprehensive manner.



```
>> actual = [1,1,0,1,1];
>> predicted = [1,0,1,1,1];
>> plotconfusion(actual,predicted)
```

	0	1	
0	0 0.0%	1 20.0%	0.0% 100%
1	1 20.0%	3 60.0%	75.0% 25.0%
	0.0% 100%	75.0% 25.0%	60.0% 40.0%

Figure 5.1: Illustration of a binary class confusion matrix

Application of this confusion matrix to validate character recognition has been observable within different researches regarding ALPR and OCR conducted so far, some of which have been discussed earlier [116] in the Literature Review chapter as well. After the discussion it can be recommended that, this confusion matrix could be used for the validation of the ALPR model developed in the Data Analysis and Model Development chapter. For this, a validation model will have to be developed as well that has been discussed afterwards.

Before all other validations, the model will have to be checked by running with license plate images to see whether it is performing its basic tasks or not. After that, the confusion matrix can be generated to see the performance efficiency of the program, therefore the validation model will be generated accordingly.

5.3. Development of validation model

As discussed earlier in this chapter that confusion matrices will be generated to validate the ALPR model, the model should be generated using MATLAB as well. MATLAB has a built-in function ‘plotconfusion’ as it was mentioned earlier too. But this function is generally applied for binary classification. If multiclass classification is to be conducted using this function, then matrix greater than usual two classes *i.e.*, three or more class containing matrix should be created. In that case, indices of the matrix are also changed due to the increase of class numbers. This fact is handled by another function named ‘sub2ind’ which is used to give unique index numbers for each index in the multiclass matrix.

For evaluating license plate detection performance, a binary class confusion matrix would be a suitable one, indicating whether there is any license plate for detection or not. The model could be given a flexibility that, even if it recognizes a license plate without being able to extract any character that could also be considered as no license plate at all as mentioned in the Table 5.1. Therefore, this can be defined as ‘0’ class. It is based on the concept of the Model discussed in the Data Analysis and Model Development chapter. Except this, two generalized cases are possible. One could be there is no license plate available in the image which is denoted as ‘0’ class. Another one is the existence of license plate as well as the extraction of the characters regardless of detection accuracy. This case is denoted as class ‘1’ for the confusion matrix. These two classes can be regarded as the state of license plate. Therefore, there will be actual license plate states and also predicted license plate states containing the classes according to the cases in the Table 5.1 mentioned earlier.

Table 5.1: Binary Class Confusion Matrix Cases with Corresponding Class Values for Detecting License Plate State

Case name defining class	Corresponding value indicating the class
No license plate is available in the image	0
Rectangular license plate has been identified but no character found within it	0
Rectangular license plate is obtained in the image and characters detected within it	1

Once binary class confusion matrix of the license plate detection is obtained then the character detection accuracy can be evaluated as well. Therefore, multiclass confusion matrix will have to be created as mentioned earlier in this section. As discussed in the beginning of this chapter that character detection can be further divided into two sub-divisions based on character types, therefore evaluation of those types will be done individually.

For classification of single-character types, there are 13 characters covered in this research. Additionally, as mentioned earlier in the Methodology and Data Collection chapter that some data containing outside of the predefined character types will also be collected, those characters are to be checked here as well. Therefore, total 14 classes will be generated for the multiclass confusion matrix of single-character types as in the Table 5.2. For the ease of analysis, the variable $out1$ could be used for designating different characters in this validation. As $out1$ is numbered from 1 to 13 accordingly, additional one class number '14' can be added for the unknown single-type characters. Both unknown characters and defined multi-character types due to wrong identification are included in this class. In addition to that, any unnecessary dots still available in the license plate detected by the program is also included in that class.

Table 5.2: Multiclass Confusion Matrix Cases with Corresponding Class Values for Detecting Single-Character Types

Case/character name defining class	Corresponding value indicating the class
Character '০' (0)	1
Character '১' (1)	2
Character '২' (2)	3
Character '৩' (3)	4
Character '৪' (4)	5
Character '৫' (5)	6
Character '৬' (6)	7
Character '৭' (7)	8
Character '৮' (8)	9
Character '৯' (9)	10
Character 'গ' (Ga)	11
Character 'খ' (Kha)	12
Character 'হ' (Ha)	13
Unknown single-character types	14
Multi-character types detected by mistake	14
Unnecessary dots or noise not removed	14

Another multiclass confusion matrix approach is available for the multi-character types. Similar to the analysis of single-character types, the variable $out2$ could be used for designating different characters in this validation. As $out2$ is numbered from 1 to 7 accordingly, additional one class number ‘8’ can be added to that as in the Table 5.3 for the unknown multi-type characters. Both unknown characters and defined single-character types due to wrong identification are included in this class. In addition to that, any unnecessary dots still available in the license plates detected by the program is included in that class.

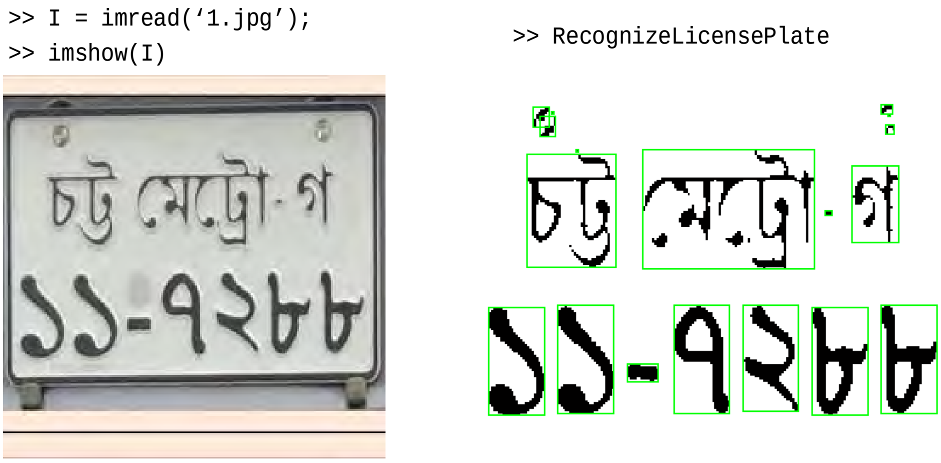
Table 5.3: Multiclass Confusion Matrix Cases with Corresponding Class Values for Detecting Multi-Character Types

Case/character name defining class	Corresponding value indicating the class
Character ‘সিলেট’ (Sylhet)	1
Character ‘রাজশাহী’ (Rajshahi)	2
Character ‘বরিশাল’ (Barisal)	3
Character ‘চট্টগ্রাম’ (Chittagong)	4
Character ‘মেট্রো’ (Metro)	5
Character ‘খুলনা’ (Khulna)	6
Character ‘ঢাকা’ (Dhaka)	7
Unknown multi-character types	8
Single-character types detected by mistake	8
Unnecessary dots or noise not removed	8

In order to generate confusion matrix for both binary class and multiclass later on, a customized function named ‘confusion_matrix’ [see Appendix-K] has been developed in MATLAB. If the classification is a binary one, then ‘plotconfusion’ is directly called via this function. If the classification is a multiclass, then it operates some pre-preprocessing before applying ‘plotconfusion’ function. Using this customized function validation of the developed ALPR model will be done afterwards.

5.4. Performance evaluation of the model

Before the application of the validation model, performance of the basic operation by the ALPR model should be observed. Therefore, the recognizing model as discussed in the Data Analysis and Model Development chapter has been run [see Figure 5.2] with the license plate images. These images are included within the validation data set of the collected data. It was seen that the basic operations by the program is quite satisfactory. Starting from a raw image of license plate [see Figure 5.2a], the model is seen to be executing the program successfully to get a modified binary image [see Figure 5.2b].



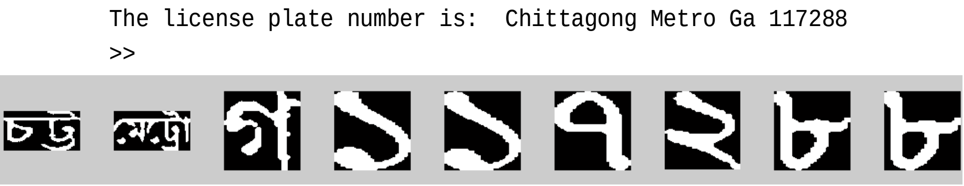
```
>> I = imread('1.jpg');
>> imshow(I)
```

```
>> RecognizeLicensePlate
```

(a) (b)

Figure 5.2: Execution of recognizing model on (a) a raw image of license plate to obtain (a) modified image

The recognizing model containing function named as 'RecognizeLicensePlate.m' file in the directory is seen to be identifying the license plate region and the characters successfully. As discussed earlier, even if any unnecessary dots and hyphens are existent in the modified license plate image, those have been eliminated through the algorithm to obtain the relevant characters only [see Figure 5.3]. It can be seen that the model has well performed to operate its desired tasks. Thus, the basic operation of the model has been validated except the efficiency validation discussed afterwards.



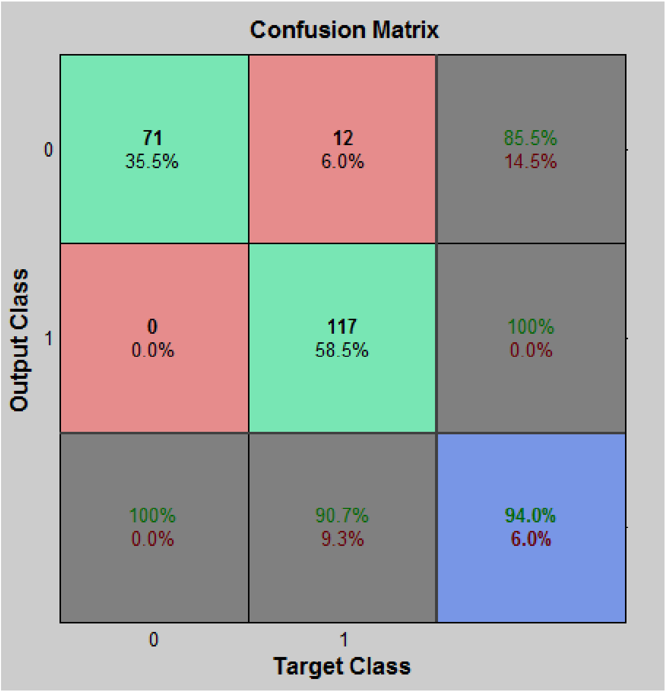
```
The license plate number is:  Chittagong Metro Ga 117288
>>
```

Figure 5.3: Extraction of license plate numbers by the ALPR model

In the meantime, the methodology for the validation has been illustrated. It is established that confusion matrix will be generated to validate the model. Therefore, development model for the confusion matrix creation has been done as well. Using the validation model, state of license plate, single-character type detection and multi-character type detection will be performed. Among these three, first one will be the binary class confusion matrix and other two being the multi-class confusion matrix. Validation data set that is distinguished from the collected data will be applied in this regard.

5.4.1. Accuracy of the license plate state detection

As discussed during the development of validation model that, license plate state detection will be a binary class confusion matrix, therefore all validation data have been classified manually. During classification, the images having no license plate are classified as ‘0’ and all the license plate containing images have been classified as ‘1’. These being the actual states of license plates, predicted states have been evaluated as well. For that, the ALPR model has been run for those images. Either no character was found even for the initially detected potential license plate or no license plate was detected, that outcome has been marked as ‘0’ class. In other cases, where there are potential characters found after detecting license plates, those are classified as ‘1’. All these data [see Appendix-L] containing the actual license plate states and the predicted license plate states have been rendered to the customized function ‘confusion_matrix’ to generate a binary confusion matrix [see Figure 5.4] for that.


Confusion Matrix

Output Class \ Target Class	0	1	
0	71 35.5%	12 6.0%	85.5% 14.5%
1	0 0.0%	117 58.5%	100% 0.0%
	100% 0.0%	90.7% 9.3%	94.0% 6.0%

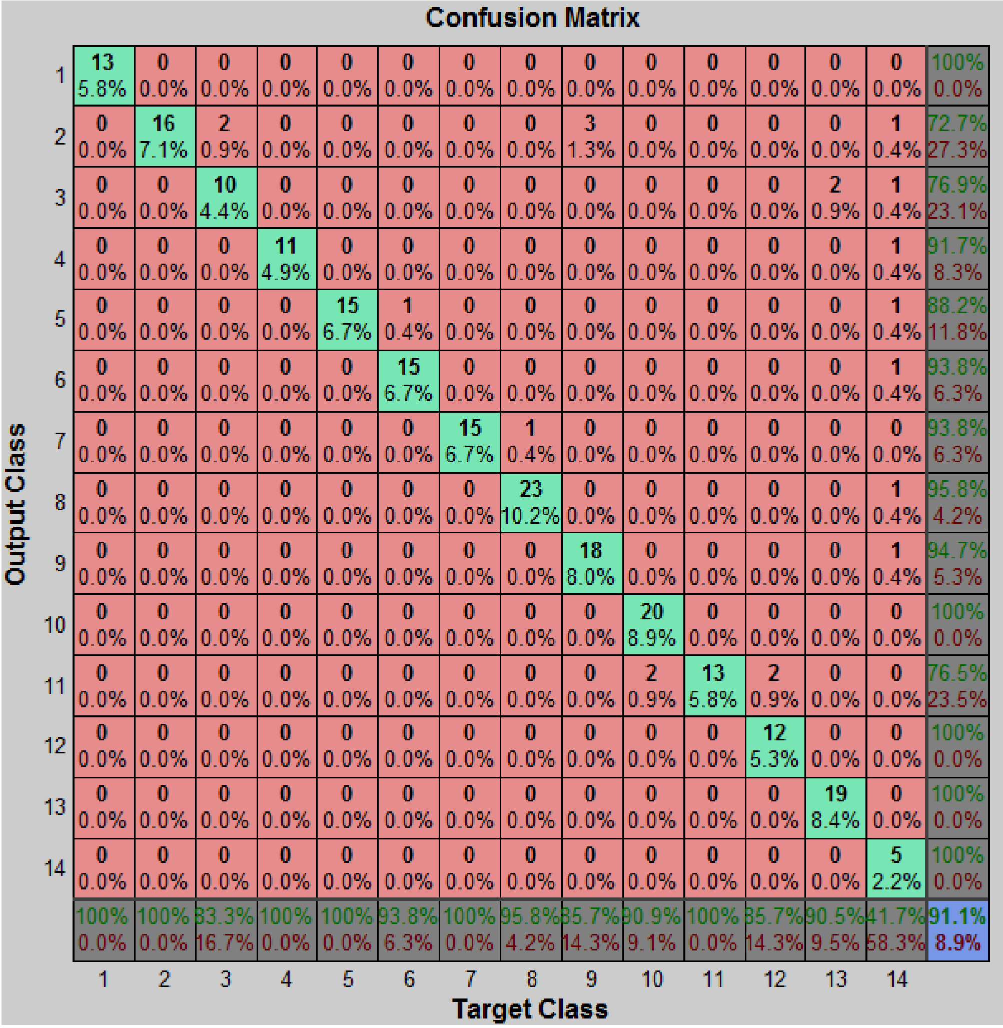
Figure 5.4: Binary class confusion matrix for license plate state detection

It can be seen from the confusion matrix that overall 94% accuracy is obtained for the license plate state detection. One important aspect of the confusion matrix is that, no error has been obtained for any license plate case. The implication is that if there is no license plate in the image, then there is almost no possibility for the model to identify any license plate from that image. Even if any rectangular portion has been identified initially; as no characters could be extracted from there, no license plate detection case prevailed. Therefore, all the 35.5% cases having no license plate were identified truly.

On the other hand, when there are available license plates in the images, license plate detections have been done in 90.7% of those cases. Somehow the model was unable to identify the remaining 9.3% cases of the license plates due to improper orientation or due to the vulnerable conditions of the license plates. In addition to that, it was significantly observed that, the cars having white color are prone to these misclassifications. It is due the algorithm that identifies the license plate based on the color intensity, therefore car backgrounds have dominated as the potential license plate regions. If these cases are kept aside, then it can be observed that the accuracy is raised up to 98% for the license plate detection cases. Above all, the 94% overall accuracy obtained by the ALPR model can be acknowledged as a fair one. ROC curve could have been plotted for the binary classification containing sensitivity and specificity parameters as well which caters for true and false positive as well as negative values within the classification. It was avoided as the main purpose was served alone by the confusion matrix.

5.4.2. Accuracy of the single-character type detection

As discussed during the development of validation model that single-type characters will be used to generate multiclass confusion matrix, classification of the data has been done accordingly. It should be noted that, classification of both single-character and multi-character types are done with a basic consideration that these are qualified in the license plate state detection initially. In other words, the character samples were taken only from the successful license plate state detection cases. After that, all the single-character types from the final ALPR model outputs have been differentiated. Also, the manual classification of those character types is done. Thereby all classifications [see Appendix-L] containing the original character values and the predicted character values have been used to generate multiclass confusion matrix [see Figure 5.5] using the customized function ‘confusion_matrix’.



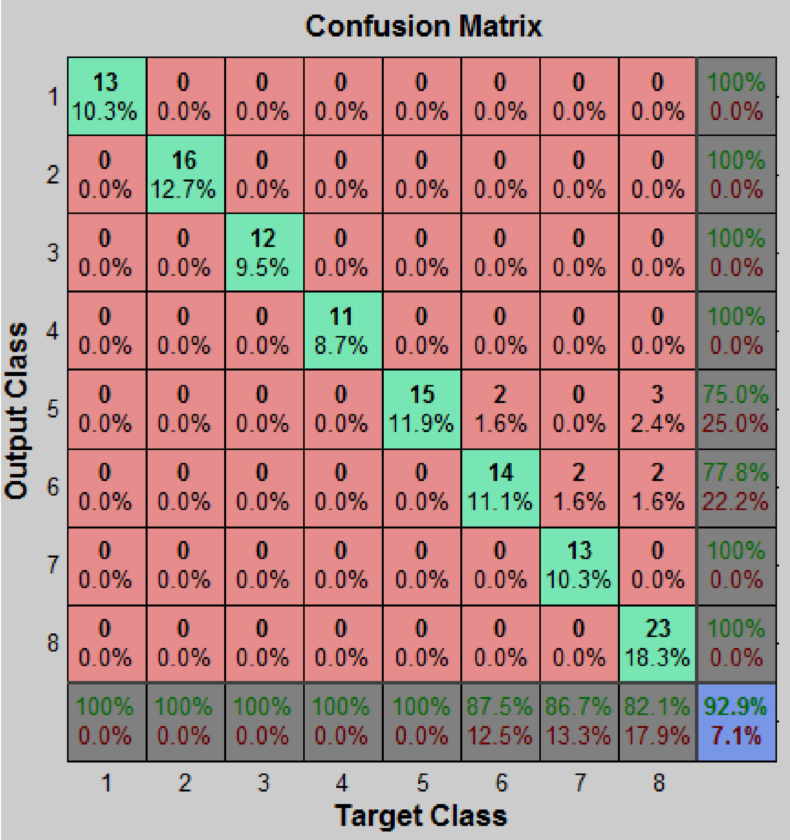
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	13 5.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	16 7.1%	2 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	72.7% 27.3%
3	0 0.0%	0 0.0%	10 4.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.9%	1 0.4%	76.9% 23.1%
4	0 0.0%	0 0.0%	0 0.0%	11 4.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	91.7% 8.3%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 6.7%	1 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	88.2% 11.8%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	93.8% 6.3%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 6.7%	1 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93.8% 6.3%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	23 10.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	95.8% 4.2%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 8.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	94.7% 5.3%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 8.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.9%	13 5.8%	2 0.9%	0 0.0%	0 0.0%	76.5% 23.5%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	12 5.3%	0 0.0%	0 0.0%	100% 0.0%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 8.4%	0 0.0%	100% 0.0%
14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 2.2%	100% 0.0%
	100% 0.0%	100% 0.0%	83.3% 16.7%	100% 0.0%	100% 0.0%	93.8% 6.3%	100% 0.0%	95.8% 4.2%	85.7% 14.3%	90.9% 9.1%	100% 0.0%	85.7% 14.3%	90.5% 9.5%	41.7% 58.3%	91.1% 8.9%

Figure 5.5: Multiclass confusion matrix for single-character type detection

It can be seen from the confusion matrix that overall 91.1% accuracy has been obtained for the single-character types. Among the misclassifications by the predicted model, it is observed that majority of those are generated due to the class ‘14’ alone. As it is known that for unknown character detections, class ‘14’ has been used, so wrong detection of those unknown characters as known characters might have been done. Except this fact, few misclassifications were also done by the ALPR program for the featured characters as well. Among those, significant characters are ‘2’, ‘9’ and ‘Kha. Additionally, it is also observed that characters ‘2’ and ‘9’ are frequently misclassified as ‘1’. It might be due to the similarities in geometry among them. Character ‘Kha’ might be misclassified due to the slight difference between ‘Ga’ and consequently ‘Kha’ has been misclassified as ‘Ga’ by the ALPR model. Albeit all these facts, especially if the misclassifications due to unknown characters are kept aside; the model is seen to be operating fair in terms of the featured ALPR algorithms.

5.4.3. Accuracy of the multi-character type detection

The last validation of the ALPR model is the multi-character type detection. This is another multi-class confusion matrix related analysis. Similar to the single-character types, multi-character types are classified accordingly as mentioned during the development of the validation model. In this classification as well, all the successful license plate state detection cases have been utilized. Multi-character types have been extracted from all ALPR model outputs. In addition to that, those character types have been designated manually as well. Hence, data containing both actual and predicted multi-character type classes [see Appendix-L] have been used to generate the multiclass confusion matrix [see Figure 5.6] using the customized 'confusion_matrix' function.



Confusion Matrix

Output Class	1	2	3	4	5	6	7	8	
1	13 10.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	16 12.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	12 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	11 8.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 11.9%	2 1.6%	0 0.0%	3 2.4%	75.0% 25.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 11.1%	2 1.6%	2 1.6%	77.8% 22.2%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	13 10.3%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	23 18.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	87.5% 12.5%	86.7% 13.3%	82.1% 17.9%	92.9% 7.1%

Target Class

Figure 5.6: Multiclass confusion matrix for multi-character type detection

From the confusion matrix of the multi-character type detection, it can be seen that an overall accuracy of 92.9% has been obtained. Similar to the single-character type detection, higher errors are possessed by the unknown character detections *i.e.*, target class '14' the share of which is 17.9%. Also, significant errors among others have been observed for the characters 'Dhaka' and 'Khulna'. Most of the times, these two characters are misclassified mutually or as 'Metro'.

Albeit these facts, accuracies greater than 85% have been found for all other cases including 100% for most of the characters. Therefore, the ALPR model can be accepted for the operation suggesting some scopes for the future researches.

5.5. Overview

Starting from the beginning of the chapter, validation criteria and methodology have been discussed. Based on the methodology, analyses for both binary class and multiclass confusion matrices have been taken into account. For the generation of the validation model, programming using the MATLAB software has been undertaken. Also, the distinguished data set for validation from the collected data has been utilized. Validation has been done not only for the license plate state detection but also for the character type detection in this chapter. By this validation, the proposed ALPR model has been vindicated to be operated to recognize the license plates in Bangladesh. Consequently, some observations during validation and model development have been acknowledged which will be concluded in the next chapter.

CHAPTER 6
CONCLUSIONS AND RECOMMENDATIONS

6.1. Introduction

The primary objective of this study was to develop of an Automatic License Plate Recognition (ALPR) system for the license plates of Bangladesh using image processing (IP) and Adaptive Neuro-Fuzzy Inference System (ANFIS). The outcome of the study was mainly an ALPR model containing two sub-models named as training model and recognizing model respectively, which have been validated by generating confusion matrices for license plate state and character type detections afterwards. In this section, the entire study has been summarized and all findings have been highlighted.

Initially, the current scenario of Bangladesh, suffering from traffic related problems has been diagramed. Necessity of an ALPR system in different sectors of Bangladesh has been established as well. Therefore, the objectives including the scope have been enlisted for the research.

Starting with the history of license plate usage in traffic operation, current scenario of the world in this field has been depicted. An assessment of all relevant literatures that are available has been conducted as well. It was found that the IP followed by the ANFIS has been widely used for this purpose. Also, it was found that there is barely any research available in this field in context of Bangladesh. Thereby, the rationale behind the study was proven.

Images of different vehicles have been captured as data from the nominated location according to the methodology. For this particular study, only private cars of specific types have been selected along with some motorbikes. These data have been divided into two parts: calibration or model development data, and validation data. Using the first set of data, geometric features of different characters used in license plates of Bangladesh have been distinguished. These features have been used to train the device in MATLAB program using ANFIS. Finally, validation of the ALPR model has been done by checking the performance through confusion matrices in detecting license plate states as well as the characters from the distinguished raw images kept for validation.

6.2. Main features of the model developed

While developing the ALPR model, a training model has been developed first for extracting geometric features and for training the device using ANFIS. Main inputs for ANFIS were two variables first one denoting the single-character type and the second one denoting the multi-character type. These two inputs have been used to train the output of the model using ANFIS editor in MATLAB. This training file was saved in the directory of recognizing model. The recognizing model has been created using MATLAB to take input from the directory and to generate desired ALPR outputs. The main features of the ALPR model developed can be listed as follows:

i. Two variables were used to represent the character type in the ALPR model namely, $out1$ and $out2$. These two input variables were used to generate the unique output having numeric values representing different characters used in the license plates of Bangladesh. All inputs and outputs are represented as integer numbers.

ii. While performing the ALPR operation, focus was specifically given for the Bangla texts as words are usually connected together with a dash over them known as ‘Matra’. Separating all characters individually was somewhat tough for this reason. Considering the ease of analysis and the purpose of detecting license plate characters, decision was made to treat those connected words as one character. Thus, those characters have been categorized as multi-character types keeping the residuals as single-character types.

iii. For all single-character types $out2$ is chosen as ‘0’ and for all multi-character types $out1$ is chosen as ‘0’. After that, total 20 characters have been selected for the ALPR model of which single-type and multi-character types contain 13 and 7 characters respectively.

iv. As there is no complexity regarding the set of rules in ANFIS, the algorithm can handle this issue using adaptations, therefore only the necessity was to check the performance of different Membership Functions (MFs). After comparative analysis, two similar triangular MFs have been used in MATLAB for each of the two inputs with 100 epochs. For training, hybrid optimization method has been utilized.

v. Rectangular shape is assumed for the detection of probable license plate region having a width to height ratio near 0.5. Also, the background color having the highest intensity among the probable plate regions obtained from the previous one has been considered as the actual license plate region.

vi. Even after refining images to get the actual license plate images, provision of user defined license plate region is kept open for the users. Therefore, user can customarily define the license plate region if the program is seen to have failed to refine the region properly.

vii. A reordering algorithm has been established to extract and show the characters in correct order. It was done because the priority had been given to the available leftmost object as predecessor character in MATLAB by default. On the other hand, among two rows, top one starting from left to right followed by the bottom one is seen from the orientation of the license plates in Bangladesh. Therefore, default algorithm of MATLAB has been modified through dividing the image into two parts by a horizontal line through the middle portion of the refined image.

viii. Different customized functions have been created using MATLAB to aid the operation of ALPR in both training and recognizing models.

Another model has been developed to validate the ALPR model known as the validation model. It was basically done through adopting a customized MATLAB function for generating both binary class and multiclass confusion matrices. The main features of this validation model can be enlisted as below.

i. Programs for both training and recognizing models were executed to see the performance of the basic tasks. All operations were found satisfactory throughout the algorithm.

ii. For generating binary class confusion matrix for the license plate state detection, it was checked whether there is any license plate or not. Based on the answer, two classes designated as ‘0’ and ‘1’ have been selected. Performance of the ALPR model was found satisfactory comparing the actual results with the predicted results listed manually.

iii. For generating multiclass confusion matrices for the single-character and the multi-character types, 14 and 8 classes have been distinguished respectively. There is one additional class for the unknown characters in both of these two classes in addition to the original character input values as depicted in the ALPR model development. Satisfactory results were found for detecting characters in the license plates especially the characters which were covered within the research.

The validation model can be applied any time for additional data to vindicate the consistency of the ALPR model. It is expected that every time it would give the similar result as obtained.

6.3. Constraints of the research

Although the study was fairly successful to recognize the characters from the license plate images, there are some constraints within this study. These constraints are enlisted below.

i. License plates regarding some private cars and motorbikes thereby total 20 characters were covered in this research. On the other hand, there are plenty of vehicle types available; at least more than 50 characters can be identified for developing ALPR in Bangladesh.

ii. Actual license plate area detection is still a complex one because of different circumstances. Even 94% accuracy was found during the validation of the model. At the same time, it was found that even if there are license plates, it was not detected by the ALPR model in 9.3% cases. An approach might be made to strengthen the algorithm to detect more license plates with higher accuracy. Especially problems with white colored vehicles should be mitigated.

iii. Ordering of characters from the refined license plate images is another loophole. Though fair results were obtained by the proposed ALPR model, existence of both rows of the license plate over the mid horizontal line might create additional complexities. More efficient algorithm to reorder the characters accordingly might be expected.

iv. Scarcity of enormous data is another constraint of this study, though it is expected that higher data would still govern the similar results of accuracy.

v. The proposed ALPR model is applicable for the standstill captured images only. There is no provision for real-time character recognitions from instantly captured images. Therefore, all the images for ALPR will have to be captured and saved in the model directory first. After that, the ALPR model will have to be run accordingly.

vi. Some higher accuracy algorithms have been omitted considering operating time consumption. Also, device features commonly used in Bangladesh have been considered. But it is true that, there is a sudden boom in the field of technology all over the world as well as Bangladesh. Therefore, higher accuracy algorithms could be used to compare their performances.

vii. Quality of the images were neither as high as it is desired for this type of studies nor any check was made for the performance comparison due to the image quality. It could be expected that higher quality images could increase the accuracy of the ALPR model by a wide margin.

6.4. Recommendations for future research

The ALPR model developed in this study could be used as a foundation stone for the detection of license plates in Bangladesh. Numerous research lines for future studies could be drawn from this model which were not acknowledged during the research due to limited scope and constraint issues. Also, some schemes were discovered during analyzing and validating the model. Henceforth, the recommendations for future research are listed below.

i. Number of characters for ALPR model could be increased to incorporate other modes of transport such as: truck, bus, van etc. even for non-motorized vehicles as well. In addition to that, there is a format for the license plates of diplomatic vehicles that is completely different from the usual license plates. Model for identifying and recognizing characters from those can be developed.

ii. The model developed in this study should be calibrated using better quality of images collected from various sites in Bangladesh. Therefore, a detailed study for evaluating performances can be conducted.

iii. In addition to the Bangla license plates used in Bangladesh, there are some unusual license plates still available in Bangladesh written using English texts and also some samples where characters are marked on the black background of plates. Therefore, addressing all of those traces could be done by developing models which are more consistent.

iv. Although analogy has been made for different relevant branches of neural networks (NNs), statistical comparison using the results obtained versus the operating time and other factors for all types of NN should be diagramed. It might give a clear indication of the justification for using different NN algorithms such as: Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) etc.

v. There is no provision for real-time license plate detections from instantly captured images. Also, this model could be modified further for real-time license plate recognition by video processing using MATLAB as it also provides dedicated functions for video processing.

vi. A database could be created as an extension of this ALPR model to store image data of different hours, days, weeks etc. by any entity to keep records which can be utilized in near future for multidimensional purposes such as: origin-destination survey, consumer survey, stolen vehicle detection etc.

REFERENCE

[1] *Road Transport Act 2018 – details, repercussions, reactions*. The Daily Star (2019). Retrieved January 19, 2020, from https://www.thedailystar.net/shift/news/road-transport-act-2018-details-repurcussions-reactions-1823818

[2] *Mobile court to enforce Road Transport Act-2018 from Monday*. Dhaka Tribune (2019). Retrieved January 19, 2020, from https://www.dhakatribune.com/bangladesh/rules-and-regulations/2019/11/18/mobile-court-to-enforce-road-transport-act-2018-from-monday

[3] *Road safety movement: aftermath…*. New Age (2018). Retrieved January 19, 2020, from http://www.newagebd.net/article/50142/road-safety-movement-aftermath

[4] *Cabinet approves Road Transport Act*. Dhaka Tribune (2018). Retrieved January 19, 2020, from https://www.indiatoday.in/education-today/grammar-vocabulary/story/difference-between-quarantine-isolation-lockdown-and-social-distancing-1658804-2020-03-23

[5] *Dhaka remains the world's most densely populated city*. Dhaka Tribune (2018). Retrieved August 22, 2019, from https://www.dhakatribune.com/bangladesh/dhaka/2018/10/14/dhaka-remains-the-world-s-most-densely-populated-city

[6] *World countries by population density 2019*. World Population Review (2019). Retrieved August 22, 2019, from http://worldpopulationreview.com/countries/countries-by-density

[7] BRTA (2018) *Number of registered motor vehicles in Bangladesh (Yearwise)*. Government of Bangladesh, Ministry of Road Transport and Bridges, Bangladesh Road Transport Authority. Retrieved August 22, 2019, from https://brta.portal.gov.bd/sites/default/files/files/brta.portal.gov.bd/monthly_report/d4d56177_644f_44f8_99c4_3417b3d7b0f4/MV_statistics-bangladesh-march-18.pdf

[8] World Bank Group (2019) *June 2019 global economic prospects heightened tensions, subdued investment. A World Bank Group flagship report*. World Bank Group. Retrieved February 20, 2020, from http://documents.worldbank.org/curated/en/541011559679035492/pdf/Global-Economic-Prospects-June-2019-Heightened-Tensions-Subdued-Investment.pdf

[9] *Can the number of private cars be controlled in Dhaka*? Dhaka Tribune (2017). Retrieved February 20, 2020, from

https://www.dhakatribune.com/bangladesh/dhaka/2017/11/11/traffic-dhaka-private-cars

[10] *Decent public transport service still elusive in Bangladesh capital*. New Age (2019). Retrieved March 20, 2020, from https://www.newagebd.net/article/93494/decent-public-transport-service-still-elusive-in-bangladesh-capital

[11] *How can we fix the world's worst traffic*? Dhaka Tribune (2019). Retrieved March 20, 2020, from http://dhakatribune.com/opinion/op-ed/2019/05/23/how-can-we-fix-the-world-s-worst-traffic

[12] *Worsening traffic congestion in Dhaka*. The Daily Star (2020). Retrieved March 20, 2020 from https://www.thedailystar.net/letters/news/worsening-traffic-congestion-dhaka-1749766

[13] *Traffic week from Sunday, announces DMP*. The Daily Sun (2018). Retrieved June 11, 2020 from https://www.daily-sun.com/post/327120/Traffic-Week-from-Sunday-announces-DMP-

[14] *DMP files 4,484 cases for traffic rules violation*. The Independent (2018). Retrieved June 11, 2020 from http://www.theindependentbd.com/printversion/details/161646

[15] *No vehicle without digital number-plate allowed on road from Jan 1: Quader*. The Daily Sun (2015). Retrieved June 11, 2020 from https://www.daily-sun.com/post/83177/No-vehicle-without-digital-numberplate-allowed-on-road-from-Jan-1:-Quader

[16] *Vehicle owners find no use of digital number plates*. The Daily Sun (2018). Retrieved June 11, 2020 from https://www.daily-sun.com/post/292031/Vehicle-owners-find-no-use-of-digital-number-plates

[17] *32km tailback on Dhaka-Chittagong highway*. Dhaka Tribune (2018). Retrieved June 11, 2020 from https://www.dhakatribune.com/bangladesh/nation/2018/08/16/hundreds-of-vehicles-are-stuck-on-the-highway-adding-to-people-s-suffering-especially-that-of-cattle-traders

[18] *Automated License Plate Recognition (ALPR)*. WhatIs.com (2011). Retrieved June 15, 2020 from https://whatis.techtarget.com/definition/Automated-License-Plate-Recognition-ALPR

[19] Zadeh, L. A. (1996) *Fuzzy Sets, Fuzzy Logic, Fuzzy Systems*. World Scientific Publishing Company, Singapore, ISBN 978-981-02-2421-9. Retrieved August 3, 2020 from https://doi.org/10.1142/9789814261302_0021

[20] *ANPR survey accuracy has transformed road planning*. Tracsis Traffic and Data Services (2020). Retrieved June 11, 2020 from https://tracsistraffic.com/anpr-survey-accuracy-has-transformed-road-planning/

[21] *The rise of automatic license plate recognition*. American City & County (2019). Retrieved June 14, 2020 from https://www.americancityandcounty.com/2019/05/01/the-rise-of-automatic-license-plate-recognition/

[22] UN (2018) *Leaving the LDCs category: Booming Bangladesh prepares to graduate*. Department of Economic and Social Affairs, United Nations. Retrieved June 14, 2020 from https://www.un.org/development/desa/en/news/policy/leaving-the-ldcs-category-booming-bangladesh-prepares-to-graduate.html

[23] *Bangla ranked at 7th among 100 most spoken languages worldwide*. Bangla Tribune (2020). Retrieved June 15, 2020 from https://www.dhakatribune.com/world/2020/02/17/bengali-ranked-at-7th-among-100-most-spoken-languages-worldwide

[24] *The problems of public transport in Dhaka*. ICE Business Times (2018). Retrieved June 15, 2020 from https://ibtbd.net/the-problems-of-public-transport-in-dhaka/

[25] *Inclement weather conditions: when visibility is low, drive slow*. Florida Department of Highway Safety and Motor Vehicles (2019). Retrieved June 15, 2020 from https://www.flhsmv.gov/safety-center/driving-safety/inclement-weather-conditions/low-visibility/

[26] *Traffic congestion in Dhaka should be reduced*. Daily Observer (2019). Retrieved June 14, 2020 from https://www.observerbd.com/details.php?id=186218

[27] *Buffy Sainte-Marie Quotes*. BrainyQuote.com (2020). Retrieved June 16, 2020, from https://www.brainyquote.com/quotes/buffy_saintemarie_341267

[28] Risk Reduction Engineering Laboratory (1991) *AWD technologies integrated AquaDetox/SVE technology: applications analysis report*. Office of Research and Development, U.S. Environmental Protection Agency. Retrieved August 2, 2020 from https://books.google.com.bd/books?id=fjldFhevpL4C

[29] *Vehicle registration plate*. Wikipedia, The Free Encyclopedia (2020). Retrieved June 20, 2020 from https://en.wikipedia.org/wiki/Vehicle_registration_plate

[30] *History of ANPR*. ANPR International Ltd., Registered in the UK (2020). Retrieved June 20, 2020 from http://www.anpr-international.com/history-of-anpr/

[31] Jang, J. R., “Fuzzy modeling using generalized neural networks and Kalman filter algorithm,” in *The ninth National Conference on Artificial Intelligence (AAAI-*

91), Anaheim Convention Center, Anaheim, California, 1991. https://www.aaai.org/Papers/AAAI/1991/AAAI91-119.pdf

[32] *License*. In Dictionary. Merriam-Webster.com since 1828 (2020). Retrieved June 20, 2020 from https://www.merriam-webster.com/dictionary/license

[33] *Licence Plate*. In Dictionary. Cambridge.org (2020). Retrieved June 20, 2020 from https://dictionary.cambridge.org/dictionary/english/licence-plate

[34] BRTA Kushtia (2020) *BRTA Online Aids*. Government of Bangladesh, Ministry of Road Transport and Bridges, Bangladesh Road Transport Authority. Retrieved June 20, 2020 from http://www.brta.kushtia.gov.bd/site/photogallery/7e2432be-2f5b-4d1e-9538-6fca1fbe7770/www.brta.kushtia

[35] Fabry, M. (2016) *This is why cars have license plates*. Time. Retrieved June 20, 2020 from https://time.com/4301055/license-plate-history

[36] Taylor, E. R. (2020) *Massachusetts archive - part 2*. Porcelain Plates.net, A Website for Porcelain License Plate Collectors & Enthusiasts (2020). Retrieved July 3, 2020 from http://porcelainplates.net/massachusetts_archive_2.html

[37] *West Virginia license plates*. Licenseplates.tv, The World's Largest Online Store for Custom License Plates (2013). Retrieved July 3, 2020 from https://www.licenseplates.tv/west-virginia-190.html

[38] *In the 1940s soybean plastics were futuristic; they still are*. The Plastics Collection, Syracuse University Library (2012). Retrieved July 3, 2020 from https://www.historicvehicle.org/licence-plates-a-timeline/

[39] *Wisconsin auto license plate history 1905-1993*. Wisconsin Department of Transportation (1993). Retrieved July 3, 2020 from https://books.google.com.bd/books?id=7LzViHwnZhUC

[40] Jerry Woddy (2011) *New Brunswick 1944 – wooden issue motorcycle plate*. Flickr, Connecting People through Photography. Retrieved July 3, 2020 from https://www.flickr.com/photos/woodysworld1778/5810708181/in/photostream

[41] *Vehicle registration plates of Bangladesh*. Wikipedia, The Free Encyclopedia (2020). Retrieved July 3, 2020 from https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Bangladesh

[42] Michael Kustermann (2014) *License plates of the world*. World License Plates, Bangladesh. Retrieved July 3, 2020 from http://www.worldlicenseplates.com/world/AS_BANG.html

[43] *What is ANPR? Why to use ANPR?* Adaptive Recognition Hungary (2010). Retrieved July 3, 2020 from http://www.arhungary.hu/101125/doc/arh_anpr12_flyer.pdf

[44] *ANPR in India*. Plate Recognizer, a subsidiary of ParkPow, Inc. (2020). Retrieved July 5, 2020 from https://platerecognizer.com/blog/anpr-for-india

[45] *Automated number plate recognition solution: what is ANPR?* Innominds, Powering The Digital Next (2020). Retrieved August 2, 2020 from https://www.innominds.com/connected-devices-and-iot/numix-video-analytics-solutions/anpr

[46] Akhtar, S. (2020) *Numbers of road accidents, fatalities shoot up in Bangladesh.* New Age. Retrieved July 5, 2020 from https://www.newagebd.net/article/99344/numbers-of-road-accidents-fatalities-shoot-up-in-bangladesh

[47] Kamruzzaman, M. (2020) *Bangladesh: alarming rise in road crashes.* Anadolu Agency. Retrieved July 5, 2020 from https://www.aa.com.tr/en/asia-pacific/bangladesh-alarming-rise-in-road-crashes/1692643

[48] *What is a radar gun?* Radar Gun, Powered by Weebly (2020). Retrieved July 5, 2020 from https://radargunstse.weebly.com/about.html

[49] Akhtar, S. (2020) *Vast vehicles run without RR number plates.* New Age. Retrieved July 5, 2020 from https://www.newagebd.net/article/96433/vast-vehicles-run-without-rr-number-plates

[50] Hrabina, M. (2018) *Automated vehicle identification for tolling and parking: RFID vs ANPR.* FEIG Electronics Inc. Retrieved July 7, 2020 from https://rfidreadernews.com/wp-content/uploads/2018/05/FEIG_AVI_WhitePaperFinal-1.pdf

[51] Kaur, H. (2019) *ANPR vs. RFID: which is better?* Park Wheels. Retrieved July 7, 2020 from https://parkwheels.co.in/anpr-vs-rfid-which-is-better/

[52] *What Does Artificial Intelligence (AI) Mean?* Techopedia Inc. (2020). Retrieved July 7, 2020 from https://www.techopedia.com/definition/190/artificial-intelligence-ai

[53] Sharma, A. (2019) *Difference between machine learning and artificial intelligence.* Geeks for Geeks. Retrieved July 7, 2020 from https://www.geeksforgeeks.org/difference-between-machine-learning-and-artificial-intelligence/

[54] *Artificial Intelligence.* Wikipedia, The Free Encyclopedia (2020). Retrieved July 7, 2020 from https://en.wikipedia.org/wiki/Artificial_intelligence#cite_note-Definition_of_AI-1

[55] Kim, P. (2017) *MATLAB deep learning: with machine learning, neural networks and artificial Intelligence.* Apress, Berkeley, CA. Retrieved July 3, 2020 from https://doi.org/10.1007/978-1-4842-2845-6

[56] *Neural Network*. Wikipedia, The Free Encyclopedia (2020). Retrieved July 3, 2020 from https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Bangladesh

[57] Mowrin, A. N. *Commuter Train Service Quality Prediction and User Attribute Ranking using Neural Network and Fuzzy Approach*, M.Sc. Engg. Thesis, Department of Civil Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2017. Retrieved July 28, 2020 from http://lib.buet.ac.bd:8080/xmlui/handle/123456789/4978

[58] *Characteristics of Fuzzy Inference System*. Tutorials Point, Simply Easy Learning (2020). Retrieved August 2, 2020 from https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_inference_system.htm

[59] Siler, W. and Buckley, J.J. (2004) *Fuzzy Expert Systems and Fuzzy Reasoning*. John Wiley & Sons, New Jersey, USA, ISBN 978-047-16-9850-0. Retrieved July 3, 2020 from https://onlinelibrary.wiley.com/doi/book/10.1002/0471698504

[60] Jang, J. -S. R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665 – 685, 1993. Retrieved July 28, 2020 from https://doi.org/10.1109/21.256541

[61] Billah, M., Waheed, S., and Hanifa, A., "An Optical Character Recognition System from Printed Text and Text Image using Adaptive Neuro Fuzzy Inference System." *International Journal of Computer Applications*, vol. 130, 2015a. Retrieved July 28, 2020 from https://www.semanticscholar.org/paper/An-Optical-Character-Recognition-System-from-Text-Billah-Waheed/726408f13f45b4d8f1901af1bfd3d89f2e23c92f?p2df

[62] Sugeno, M. (1985) *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., 655 Avenue of the Americas New York, NY, United States, ISBN 978-0-444-87829-8. Retrieved July 28, 2020 from https://dl.acm.org/doi/book/10.5555/537323

[63] Cruz, A. O. (2015) *ANFIS: Adaptive neuro-fuzzy inference systems*. Course Hero Retrieved July 28, 2020 from https://www.coursehero.com/file/47787879/anfispdf

[64] Anbarjafri, G. (2014) *Digital image processing: Introduction to image processing*. University of Tartu. Retrieved July 29, 2020 from https://sisu.ut.ee/imageprocessing/book/1

[65] Pingel, J. (2016) *Edge Detection with MATLAB*. MathWorks. Retrieved July 29, 2020 from https://www.mathworks.com/videos/edge-detection-with-matlab-119353.html

[66] Imran, M. A., *Neuro-Fuzzy Model for Car Following Behavior in Heterogeneous Road Traffic Condition*, M.Sc. Engg. Thesis, Department of Civil Engineering,

Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2009. Retrieved July 28, 2020 from http://lib.buet.ac.bd:8080/xmlui/handle/123456789/1609

[67] Khodayari, A., Ghaffari, A., Kazemi, R., and Manavizadeh, N., "ANFIS based modeling and prediction car following behavior in real traffic flow based on instantaneous reaction delay," in *13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, pp. 599-604, 2010. Retrieved July 29, 2020 from https://doi.org/10.1109/ITSC.2010.5625130

[68] Moghadam, M. P. A., Pahlavani. P., and Naseralavi, S., "Prediction of Car Following Behavior Based on the Instantaneous Reaction Time using An ANFIS-CART Based Model". *International Journal of Transportation Engineering*, vol. 4(2), pp. 109-126, 2016. Retrieved July 29, 2020 from http://www.ijte.ir/article_40536.html

[69] Ma, X., "Toward an Integrated Car-Following and Lane Changing Model by a Neural Fuzzy Approach". Helsinki Summer Workshop. Centre for Traffic Research, Stockholm, 2014. Retrieved July 29, 2020 from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.527.9124

[70] Sarkar, S., *Modeling of Lane Changing Decisions of Drivers for Heterogeneous Traffic Operation using Adaptive Neuro-Fuzzy Interface System (ANFIS)*, M.Sc. Engg. Thesis, Department of Civil Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2006. Retrieved July 28, 2020 from http://lib.buet.ac.bd:8080/xmlui/handle/123456789/3994

[71] Ghaffari, A., Khodayari, A. R., Kamali, A., and Tajdari, F., "A New Model of Car Following Behavior Based on Lane Change Effects using Anticipation and Evaluation Idea". *Iranian Journal of Mechanical Engineering Transactions of the ISME*, vol. 16(2), 2015. Retrieved July 29, 2020 from http://jmee.isme.ir/article_24902.html

[72] Ghaffari, A., Khodayari, A., Alimardani, F., and Sadati, H., "Overtaking maneuver behaviour modeling based on adaptive neuro-fuzzy inference system," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, Istanbul, pp. 387-392, 2012. Retrieved July 29, 2020 from https://doi.org/10.1109/ICVES.2012.6294269

[73] Feng, Y., Pickering, S., Chappell, E., Iravani, P., and Brace, C., "Driving style modelling with adaptive neuro-fuzzy inference system and real driving data," in *Stanton N. (eds.) Advances in Human Aspects of Transportation. AHFE 2018. Advances in Intelligent Systems and Computing*, Vol. 786. Springer, Cham, 2019. Retrieved July 29, 2020 from https://doi.org/10.1007/978-3-319-93885-1_43

[74] Zheng, L. B. and Meng, X., "An approach to predict road accident frequencies: application of fuzzy neural network," in *3rd International Conference on Road*

Safety and Simulation, Indianapolis, USA, 2011. Retrieved July 29, 2020 from http://onlinepubs.trb.org/onlinepubs/conferences/2011/rss/2/zheng,l.pdf

[75] Hosseinpour, M., Yahya A. S., Ghadiri, S. M., and Prasetijo, J., "Application of Adaptive Neuro-Fuzzy Inference System for Road Accident Prediction." *KSCE Journal of Civil Engineering*, vol. 17, no. 7, 2013. Retrieved July 29, 2020 from https://link.springer.com/article/10.1007/s12205-013-0036-3

[76] Zhao, L., Pawlus, W., Karimi, H. R., and Robbersmyr, K. G., "Data-Based Modeling of Vehicle Crash using Adaptive Neural-Fuzzy Inference System." *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 684-696, 2013. Retrieved July 29, 2020 from https://ieeexplore.ieee.org/document/6502245

[77] Arul, S. B., "ANFIS Based Road Accident Prediction and Detection by Analyzing and Tracking Vehicular Movement." *Journal of Chemical and Pharmaceutical Sciences*, vol. 9, no. 4, 2016. Retrieved July 29, 2020 from https://www.semanticscholar.org/paper/Anfis-based-road-accident-prediction-and-detection-Arul/5808bba665856de2e862c1c3129385b0214c3f44

[78] Hadiuzzaman, M., Siam, R. K., Hoque, N., Shimu, T. H., and Rahman, F., "Adaptive Neuro-Fuzzy Approach for Modeling Equilibrium Speed-Density Relationship." *Transportmetrica A: Transport Science, Transportation Research Board*, vol. 14, no. 9, pp. 784-808, 2018. Retrieved July 29, 2020 from https://trid.trb.org/view/1530299

[79] Taghavifard, M. T., Kalhori, S. R. N., Farazmand, P., and Farazmand K., "An Intelligent System for Prioritising Emergency Services Provided for People Injured in Road Traffic Accidents." *Mediterranean Journal of Social Sciences*, vol. 7, no. 1, 2016. Retrieved July 29, 2020 from https://www.mcser.org/journal/index.php/mjss/article/view/8758

[80] Askerbeyli, I. and Gedik, F. A., "Neuro-Fuzzy Approach for Solving Communication Network Problems." *International Journal of Electric & Computer Sciences IJECS-IJENS*, vol. 11, no. 1, 2011a. Retrieved July 30, 2020 from https://www.semanticscholar.org/paper/NEURO-FUZZY-APPROACH-FOR-SOLVING-COMMUNICATION-Askerbeyli-Gedik/c4428857a022dc057d7430d0a89e2da5050bced3

[81] Askerbeyli, I. and Gedik, F. A., "Effects of Fuzzy Logic Methods over ATM Networks." *International Journal of Recent Research and Applied Studies*, vol. 9, no. 2, 2011b. Retrieved July 30, 2020 from https://www.arpapress.com/Volumes/Vol9Issue2/IJRRAS_9_2_14.pdf

[82] Billah, M., Waheed, S., Ahmed, K., and Hanifa, A., "Real Time Traffic Sign Detection and Recognition using Adaptive Neuro Fuzzy Inference System." *Communications on Applied Electronics (CAE), Foundation of Computer Science FCS*, New York, USA, vol. 3, no. 1, 2015b. Retrieved July 30, 2020 from

https://pdfs.semanticscholar.org/2c40/3662d507cdf498201f7279e3212573a5aa98.pdf

[83] Bingham, E. (1998), *Neurofuzzy Traffic Signal Control*, M.Sc. Engg. Thesis, Department of Engineering Physics and Mathematics, Helsinki University of Technology, Espoo, Finland, 1998. Retrieved July 30, 2020 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.3378

[84] Adamski, A. and Habdank-Wojewodzki, S., "Traffic Congestion and Incident Detector Realized by Fuzzy Discrete Dynamic System." *Archives of Transport*, vol. 17, 2004. Retrieved July 30, 2020 from https://www.semanticscholar.org/paper/Traffic-congestion-and-incident-detector-realized-Adamski-Habdank-Wojew%C3%B3dzki/322fc36b655b1dad461acf88110c7d88dd17429f

[85] Sadeghi, H. M. M., *Fuzzy Intelligent Traffic Control System*, M.Sc. Engg. Thesis, Institute of Graduate Studies and Research, Department of Applied Mathematics and Computer Science, Eastern Mediterranean University, Gazimağusa, North Cyprus, 2010. Retrieved July 30, 2020 from http://i-rep.emu.edu.tr:8080/xmlui/bitstream/handle/11129/679/Sadeghi.pdf

[86] Shankar, H., Raju, P. L. N., and Rao, K. R. M., "Multi Model Criteria for the Estimation of Road Traffic Congestion from Traffic Flow Information Based on Fuzzy Logic." *Journal of Transportation Technologies*, vol. 2, no. 1, 2012. Retrieved July 30, 2020 from http://dx.doi.org/10.4236/jtts.2012.21006

[87] Kusumawardani, M., "Active Queue Management (AQM) and Adaptive Neuro Fuzzy Inference System (ANFIS) as Intranet Traffic Control." *Academic Research International*, vol. 4, no. 5, 2013.

[88] Udofia, K. M., Emagbetere, J. O., and Edeko, F. O., "Dynamic Traffic Signal Phase Sequencing for an Isolated Intersection using ANFIS." *Automation, Control and Intelligent Systems*, vol. 2, no. 2, 2014. Retreived July 30, 2020 from http://www.sciencepublishinggroup.com/journal/paperinfo?journalid=134&doi=10.11648/j.acis.20140202.12

[89] Lai, G. R., Sarkan, H. M., Rahman, R. Z. A., and Hassan, M. K., "Controlling Traffic Flow in Multilane-Isolated Intersection using ANFIS Approach Techniques." *Journal of Engineering Science and Technology*, vol. 10, no. 8, 2015. Retrieved July 30, 2020 from https://www.researchgate.net/publication/282886876_Controlling_traffic_flow_in_multilane-isolated_intersection_using_anfis_approach_techniques

[90] Dereli, T., Cetinkaya, C., and Celik, N., "Designing a Fuzzy Logic Controller for a Single Intersection a Case Study in Gaziantep." *Sigma Journal of Engineering and Natural Sciences*, vol. 36, no. 3, 2018. Retrieved July 30, 2020 from

http://eds.yildiz.edu.tr/AjaxTool/GetArticleByPublishedArticleId?PublishedArticleId=2667

[91] Gokdag, M., Hasiloglu, A. S., Karsli, N., Atalay, A., and Akbas, A., "Modeling of Vehicle Delays at Signalized Intersection with an Adaptive Neuro-Fuzzy (ANFIS)." *Journal of Scientific and Industrial Research*, vol. 66, no. 9, 2007. Retrieved July 30, 2020 from http://nopr.niscair.res.in/handle/123456789/1311

[92] Amin, H. J., Desai, R. N., and Patel, P. S., "Modelling the Crossing Behavior of Pedestrian at Uncontrolled Intersection in Case of Mixed Traffic using Adaptive Neuro Fuzzy Inference System." *Journal of Traffic and Logistics Engineering*, vol. 2, no. 4, 2015. Retrieved July 30, 2020 from http://www.jtle.net/index.php?m=content&c=index&a=show&catid=37&id=112

[93] Jain, S., *Neuro-Fuzzy Modelling of Vehicular Exhaust Emissions on Urban Roads*, PhD. Thesis, Department of Civil Engineering, IIT, Delhi, India, 2007. Retrieved July 30, 2020 from http://eprint.iitd.ac.in/handle/2074/6191

[94] Martinsky, O., *Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems*, B.Sc. Thesis, Department of Intelligent Systems, Brno University of Technology, Czech Republic, 2007. Retrieved July 30, 2020 from https://www.semanticscholar.org/paper/ALGORITHMIC-AND-MATHEMATICAL-PRINCIPLES-OF-NUMBER-Martinsky-Martinsky/1cdd8e949a681f6860948a6056f16ad73f6519b7

[95] Khalifa, O., Khan, S., Islam, R., and Suleiman, A., "Malaysian Vehicle License Plate Recognition." *The International Arab Journal of Information Technology*, vol. 4, no. 4, 2007. Retrieved July 31, 2020 from https://iajit.org/PDF/vol.4,no.4/10-Othman.pdf

[96] Nagare, A. P., "License Plate Character Recognition System using Neural Network." *International Journal of Computer Applications*, vol. 25, no. 10, 2011. Retrieved July 31, 2020 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.581&rep=rep1&type=pdf

[97] Rhead, M., Gurney, R., Ramalingam, S., and Cohen, N., "Accuracy of automatic number plate recognition (ANPR) and real world UK number plate problems," in *2012 IEEE International Carnahan Conference Security Technology (ICCST)*, Boston, MA, pp. 286-291, 2012. Retrieved July 31, 2020 from https://ieeexplore.ieee.org/document/6393574

[98] Gaikwad, D. Y., and Borole, P. B. (2014). *A review paper on automatic number plate recognition (ANPR) system*. International Journal of Innovative Research in Advanced Engineering (IJIRAE), Vol. 1, Issue 1.Retrieved August 2, 2020 from https://www.semanticscholar.org/paper/A-Review-Paper-on-Automatic-Number-Plate-(ANPR)-Gaikwad-Borole/9a8b05cc83f508afea246bf7ef17edf778f50ee6

[99] Biswas, A., and Gour, B., “An Enhancement of Number Plate Recognition Based on Artificial Neural Network.” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, ISSN: 0975-9646, 2016. Retrieved July 31, 2020 from http://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit2016070357.pdf

[100] Reji, P. L., and Dharun, V. S., “An Efficient Algorithm for Real Time License Plate Localization.” *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 24, 2016. Retrieved July 31, 2020 from http://www.arpnjournals.org/jeas/research_papers/rp_2016/jeas_1216_5570.pdf

[101] El-Harby, A. A., Behery, G. M., and Al-Dhawi, A. A., “Automatic Recognition System of Vehicle Plate for Recording Incoming Vehicles to the University Campus and the Outgoing.” *American Journal of Applied Sciences*, vol. 14, no. 4, pp. 469-477, 2017. Retrieved July 31, 2020 from https://doi.org/10.3844/ajassp.2017.469.477

[102] Ibrahim, N. K., Kasmuri, E., Jalil, N. A., Norasikin, M. A., Salam, S., and Nawawi, M. R. M., “License Plate Recognition (LPR): A Review with Experiments for Malaysia Case Study.” *The International Journal of Soft Computing and Software Engineering*, vol. 3, no. 3, 2013. Retrieved July 31, 2020 from https://arxiv.org/abs/1401.5559

[103] Simin, N. and Mei, F. C. C., “Automatic Car-Plate Detection and Recognition System.” *EURECA*, pp. 113-114, 2013. Retrieved July 31, 2020 from https://pdfs.semanticscholar.org/aeba/9d78448ad3f7abff42223b3556f76d80b364.pdf

[104] Manisha, R., and Kumari, S., “Tracking Number Plate from Vehicle using MATLAB.” *International Journal in Foundations of Computer Science and Technology*, vol. 4, no. 3, 2014. Retrieved July 31, 2020 from https://wireilla.com/papers/ijfcst/V4N3/4314ijfcst04.pdf

[105] Kaur, H., and Garg, N. K., “Number Plate Recognition using Neural Network Classifier and KMEAN.” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 8, 2014. Retrieved July 31, 2020 from https://scholar.google.com/scholar?cluster=6349262307990714589

[106] Chibuisi, I., and Madhubuezi, O., “MATLAB Based Vehicle Number Plate Recognition.” *International Journal for Research in Applied Science and Engineering Technology*, vol. 3, no. 4, 2015. Retrieved July 31, 2020 from https://www.ripublication.com/ijcir17/ijcirv13n9_10.pdf

[107] Amusan, D. G., Arulogun, O. T., and Falohun, A. S., “Nigerian Vehicle License Plate Recognition System using Artificial Neural Network.” *International Journal of Advanced Research in Computer and Communication Engineering*,

vol. 4, no. 11, 2015. Retrieved July 31, 2020 from https://ijarcce.com/wp-content/uploads/2015/12/IJARCCE-1.pdf

[108] Tiwari, B., Sharma, A., Singh, M. G., and Rathi, B., "Automatic Vehicle Number Plate Recognition System using MATLAB." *IOSR Journal of Electronics and Communication Engineering*, vol. 11, no. 4, pp. 10-16, 2016. Retrieved July 31, 2020 from http://www.iosrjournals.org/iosr-jece/papers/Vol.%2011%20Issue%204/Version-2/B1104021016.pdf

[109] Chhikara, K., and Tomar. P., "A Smart Technique for Accurate Identification of Vehicle Number Plate using MATLAB and Raspberry Pi 2." *International Journal of Engineering and Technology, Management and Applied Science*, vol. 4, no. 5, ISSN: 2349-4476, 2016. Retrieved July 31, 2020 from https://www.researchgate.net/publication/309426937_A_Smart_Technique_for_Accurate_Identification_of_Vehicle_Number_Plate_Using_MATLAB_and_Raspberry_Pi_2

[110] Stefanovic, H, Veselinovic, R., and Milic, D., "Some optimization of MATLAB-based number plate recognition algorithm," in *10th International Scientific Conference, "Science and Higher Education in Function of Sustainable Development"*, pp. 29-36, 2017. Retrieved July 31, 2020 from http://www.vpts.edu.rs/sed17/CD%20Proceedings%202017/proceedings/2-11.pdf

[111] Omran, S. S., and Jarallah. J. A., "Automatic Iraqi Cars Number Plates Extraction." *Iraqi Journal of Computers and Informatics*, vol. 44, no. 1, 2018. Retrieved July 31, 2020 from https://www.iasj.net/iasj?func=article&aId=148374

[112] Azam, S., *Automatic License Plate Detection in Hazardous Condition*, M.Sc. Thesis, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2011. Retrieved August 3, 2020 from http://lib.buet.ac.bd:8080/xmlui/bitstream/handle/123456789/3348/Full%20Thesis.pdf

[113] Alam, M. M. and Kashem, M. A., "A Complete Bangla OCR System for Printed Characters". *The Journal of Cases on Information Technology (JCIT)*, vol. 1, no. 1, ISSN 2218-5224, 2010. Retrieved July 31, 2020 from http://www.uap-bd.edu/jcit_papers/vol-1_no-1/JCIT-100707.pdf

[114] Alom, M. Z., Sidike, P., Taha, T. M., and Asari, V. K., "Handwritten Bangla Digit Recognition using Deep Learning." *Cornell University*, 2017. Retrieved July 31, 2020 from https://arxiv.org/abs/1705.02680

[115] Knight, W. (2017) *The Dark Secret at the Heart of AI.* MIT Technology Review. Retrieved July 31, 2020 from https://www.technologyreview.com/2017/04/11/5113/the-dark-secret-at-the-heart-of-ai/

[116] Shawon, A., Rahman, J., Mahmud, F., and Zaman, M. M. A., “Bangla handwritten digit recognition using deep learning CNN for large and unbiased dataset,” in *International Conference on Bangla Speech and Language Processing*, 2018. Retrieved July 31, 2020 from https://ieeexplore.ieee.org/document/8554900

[117] Hoq, M. N., Islam, H. M., Nipa, N. A., and Akbar, M. M., “A comparative overview of classification algorithm for Bangla handwritten digit recognition,” in *International Joint Conference on Computational Intelligence, Algorithms for Intelligent Systems*, Springer, Singapore, 2019. Retrieved July 31, 2020 from https://doi.org/10.1007/978-981-13-7564-4_24

[118] Shahed, M. T., Udoy, M. R. I., Saha, B., Khan, A. I., and Subrina, S., “Automatic Bengali number plate reader,” in *IEEE Region 10 Conference (TENCON)*, Malaysia, 2017. Retrieved July 31, 2020 from https://ieeexplore.ieee.org/document/8228070

[119] Ahmed, A. J. M. M. U., Hasan, M. T., Alam, M. R., and Hoque, M. S., “Characteristics of fundamental diagrams due to shockwave by non-lane based heterogeneous traffic,” in *Resilient Structures and Sustainable Constructions, International Structural Engineers Conference-9*, Valencia, Spain, vol. 4, no. 1, 2017. Retrieved August 5, 2020 from https://www.isec-society.org/ISEC_PRESS/ISEC_09/html/O-8.xml

[120] Pai, A. (2020) *CNN vs. RNN vs. ANN – analyzing 3 types of neural networks in deep learning*. Analytics Vidya. Retrieved August 4, 2020 from https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning

[121] *What is deep learning? 3 things you need to know*. MathWorks, Accelerating the Pace of Engineering and Science (2020). Retrieved August 4, 2020 from https://www.mathworks.com/discovery/deep-learning.html

[122] Specht, D. F., “Probabilistic Neural Network.” *Neural Networks*, vol. 3, no. 1, pp. 198-118, 1990. Retrieved August 5, 2020 from https://doi.org/10.1016/0893-6080(90)90049-Q

[123] Satapathy, S. K., Dehuri, S., Jagadev, A. K., and Mishra, S., *EEG Brain Signal Classification for Epileptic Seizure Disorder Detection*. Academic Press, Elsevier, 2019. Retrieved August 5, 2020 from https://doi.org/10.1016/B978-0-12-817426-5.00001-6

[124] Malik, D. M. G. *Development of Passenger Train Service Quality Model for Special Occasion through Neural Networks and Fuzzy Inference System*, M.Sc. Engg. Thesis, Department of Civil Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2017. Retrieved July 28, 2020 from http://lib.buet.ac.bd:8080/xmlui/handle/123456789/4878

[125] *Millions of engineers and scientists trust MATLAB*. MathWorks, Accelerating the Pace of Engineering and Science (2020). Retrieved August 5, 2020 from https://www.mathworks.com/products/matlab.html

[126] *PYPL PopularitY of Programming Language: PYPL Index*. PYPL (2020). Retrieved August 5, 2020 from http://pypl.github.io/PYPL.html

[127] Bhatt, S. (2021) *Pros and cons of using python for web development*. Data Science Central. Retrieved August 5, 2020 from https://www.datasciencecentral.com/profiles/blogs/pros-and-cons-of-using-python-for-web-development

[128] Dataflair Team (2019) *Pros and cons of R programming language – unveil the essential aspects!* Data Flair. Retrieved August 5, 2020 from https://data-flair.training/blogs/pros-and-cons-of-r-programming-language

[129] *Anfis: Tune Sugeno-type fuzzy inference system using training data.* MathWorks. Retrieved August 5, 2020 from https://www.mathworks.com/help/fuzzy/anfis.html

[130] *Location of the Motijheel Shapla circle*. Google Maps (2020). Retrieved August 6, 2020 from https://www.google.com/maps/place/Shapla+Chattar/@23.7260166,90.4189483,16z

[131] *Tallest buildings in Dhaka: city center*. Emporis (2020). Retrieved August 6, 2020 from https://www.emporis.com/buildings/1208095/city-centre-dhaka-bangladesh

[132] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979. Retrieved August 5, 2020 from https://ieeexplore.ieee.org/document/4310076

[133] Canny, John, "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, 1986. Retrieved August 5, 2020 from https://ieeexplore.ieee.org/document/4767851

[134] Sarangi, P. P., Sahu, A. and Panda, M., "A Hybrid Differential Evolution and Back-Propagation Algorithm for Feedforward Neural Network Training." *International Journal of Computer Applications*, vol. 114, no. 14(4). Retrieved August 5, 2020 from https://www.ijcaonline.org/archives/volume35/number5/4398-6106

[135] *Confusion matrix – explained: How to take advantage of a confusion matrix.* Towards Data Science (2020). Retrieved August 19, 2020 from https://towardsdatascience.com/confusion-matrix-explained-34e4be19b3ec

APPENDIX-A

separatedcharacter.m file in MATLAB

```
function [rt1] = separatedcharacter(I)
% this function returns avg. no. of separated characters
% along width throughout the height of the BW image 'I'
% note that, I image has white pixels for characters in
% black background

J = I;
% a vector created with '0's inintially
M = zeros(size(J,1),1);
% a variable for counting
Msz = 0;

for d = 1:size(J,1)
    if J(d,1) == 1
        count = 1;
    else count = 0;
    end
    for x = 2: size(J,2) % along total width
        if J(d,x) == J(d,x-1)
        elseif J(d,x) == 1
            count = count+1;
        elseif J(d,x) == 0
        end
    end
    Msz = Msz+1;
    M(Msz,1) = count;
end

% unnecessary rows eliminated first
M(Msz+1:end,:) = [];
% mean of M is determined as funtion return
rt1 = mean(M);
end
```

APPENDIX-B

FeatureTraining.m file in MATLAB

```
clear all;
close all;
clc;
% clearing all variables and closing all other tasks in
% matlab and cleaning command window

% ask user to give multiple files as input
P = uigetfile({'*.jpg;*.tif;*.png;*.gif','All Image Files'},...
    'Please Select an Image','MultiSelect','on');

if isequal('double', class(P))
    display(['No image was selected, please select at' ...
        'least two images'])
    for n = 1:5
        beep
        pause(1)
    end
    clear all
    clc
elseif isequal('char', class(P))
    display(['Only one image was selected, please ' ...
        'select at least two images'])
    for n = 1:5
        beep
        pause(1)
    end
    clear all
    clc
else

    % This function returns three matrices containin geometric
    % features of the images, A containing 1 and 0 status based
    % on limit, L containing actual value of features and D
    % containing features inputs for ANFIS in last two columns

    L = cell(size(P,2),30); %Contains Questions

    D = {size(P,2),6}; %ANFIS Data

    A = cell(size(P,2),30); %contains answers

    Compare = cell(size(P,2),2); % contains comparative images

    for h = 1: size(P,2)
        L{h,1} = P{1,h};

        I = imread(P{1,h});


        if size(I,3)==3 % RGB image
            I=rgb2gray(I);
        end
        if size(I,3) < 3 && any(I(:) > 1)
            threshold = graythresh(I);
            I = ~im2bw(I,threshold);
        end

        %remove all disconnected objects fewer
        % than size(I,1) pixels
        I = bwareaopen(I,size(I,1));

        I = I(any(I ~= 0,2),:); %removes all blank rows
        I = I'; % rotates the image for removing blank columns
        I = I(any(I ~= 0,2),:); % removes all blank columns
        % (original image) and rows(for rotated image)
        I = I'; % gives original orientation back

        %     I = imresize(I,[256 512]);

        %separatedcharacter gives average disconnected no.
        % of vertical lines
        if separatedcharacter(I) >= 2.25
             % Umo and Borgiyo Jo might create problem
             % which is not covered in this research
            I = imresize(I,[256 512]);
            % elseif separatedcharacter(I) < 2.25
            cmp = 2;
        else
            I = imresize(I,[256 256]);
            cmp = 1;
        end




        % might be unnecessary
        width = size(I,2);
        height = size(I,1);

        % two limits set, the character nnz value must be
        % within this range
        ulim = 0.40;
        llim = 0.30;
        while ~(nnz(I)/(size(I,1)*size(I,2)) > llim && nnz(I)/...
                (size(I,1)*size(I,2)) < ulim)
            if nnz(I)/(size(I,1)*size(I,2)) <= llim
                I = bwmorph(I,'thick',1.1);
            elseif nnz(I)/(size(I,1)*size(I,2)) >= ulim
                I = bwmorph(I,'thin',1.2);
            end
        end


        L{h,2} = I;

        % middle of zero
        T1 = I(floor(0.3*height):floor(0.7*height),...
            floor(0.3*width):floor(0.7*width));

        Q1 = nnz(T1)/(size(T1,1)*size(T1,2));
        if Q1 < 0.12
            A1 = 1;
        elseif Q1 >= 0.12
            A1 = 0;
        end



        %left middle of one
        T19 = I(floor(0.3*height):floor(0.7*height),...
            1:floor(0.3*width));

        Q2 = nnz(T19)/(size(T19,1)*size(T19,2));

        if Q2 < 0.12
            A2 = 1;
        elseif Q2 >= 0.12
            A2 = 0;
        end



        %right top side of one
        T3 = I(1:floor(0.5*height),floor(0.5*width):end);

        % triangular area taken
        trin1 = zeros(size(T3,1),size(T3,2));
        for f = 1:size(T3,1)
            for g = f:size(T3,2)
                trin1(f,g) = T3(f,g);
            end
        end

        Q3 = nnz(trin1)/(size(trin1,1)*size(trin1,2)/2);

        if Q3 < 0.12
            A3 = 1;
        elseif Q3 >= 0.12
            A3 = 0;
        end




        %bottom left side of two
        T4 = I(floor(0.7*height):end,1:floor(0.8*width));
        trin2 = zeros(size(T4,1),size(T4,2));

        tn2y = size(T4,1);
        tn2x = size(T4,2);

        for f = 1:tn2y
            for g = 1:floor(tn2x*f/tn2y)
                trin2(f,g) = T4(f,g);
            end
        end


        Q4 = nnz(trin2)/(0.5*tn2y*tn2x);

        if Q4 < 0.12
```

```
    A4 = 1;
elseif Q4 >= 0.12
    A4 = 0;
end




%top left side of three
T5 = I(1:floor(0.15*height),1:floor(0.35*width));

Q5 = nnz(T5)/(size(T5,1)*size(T5,2));


if Q5 < 0.08
    A5 = 1;
elseif Q5 >= 0.08
    A5 = 0;
end



%top left side downward of three
T6 = I(1:floor(0.3*height),...
    floor(0.15*width):floor(0.30*width));

Q6 = nnz(T6)/(size(T6,1)*size(T6,2));

if Q6 < 0.15
    A6 = 1;
elseif Q6 >= 0.15
    A6 = 0;
end



% middle of three
T7 = I(floor(0.45*height):floor(0.7*height),...
    floor(0.35*width):floor(0.65*width));

Q7 = nnz(T7)/(size(T7,1)*size(T7,2));


if Q7 < 0.15
    A7 = 1;
elseif Q7 >= 0.15
    A7 = 0;
end



%bottom middle of four
T8 = I(floor(0.65*height):floor(0.75*height),...
    floor(0.45*width):floor(0.55*width));

Q8 = nnz(T8)/(size(T8,1)*size(T8,2));
```

```
if Q8 < 0.15
    A8 = 1;
elseif Q8 >= 0.15
    A8 = 0;
end




% middle part of four
T9 = I(floor(0.45*height):floor(0.55*height),...
    floor(0.25*width):floor(0.75*width));

Q9 = nnz(T9)/(size(T9,1)*size(T9,2));

if Q9 >= 0.80
    A9 = 1;
elseif Q9 < 0.80
    A9 = 0;
end





% right portion of five
fT10 = flipdim(I,2); %horizontal flip
T10 = fT10(floor(0.35*height):end,1:floor(0.25*width));
trin10 = zeros(size(T10,1),size(T10,2));

tx10 = size(trin10,2);
ty10 = size(trin10,1)/2;

for f = 1:size(trin10,1)
    if f < 0.5*size(trin10,1)
        for g = 1:round(f*tx10/ty10)
            trin10(f,g) = T10(f,g);
        end
    elseif f == 0.5*size(trin10,1)
        for g = 1:size(trin10,2)
            trin10(f,g) = T10(f,g);
        end

    elseif f> 0.5*size(trin10,1)
        for g = 1:((2*ty10-f)*tx10/ty10)
            trin10(f,g) = T10(f,g);
        end
    end
end

% triangular area
Q10 = nnz(trin10)/(0.5*size(trin10,1)*size(trin10,2));

if Q10 < 0.09
    A10 = 1;
```

```
        elseif Q10 >= 0.09
            A10 = 0;
        end




        % right top of six
        T11 = I(1:floor(0.25*height),floor(0.75*width):end);

        Q11 = nnz(T11)/(size(T11,1)*size(T11,2));

        if Q11 < 0.05
            A11 = 1;
        elseif Q11 >= 0.05
            A11 = 0;
        end




        % bottom left of seven
        T12 = I(floor(0.70*height):end,1:floor(0.75*width));

        Q12 = nnz(T12)/(size(T12,1)*size(T12,2));


        if Q12 < 0.12
            A12 = 1;
        elseif Q12 >= 0.12
            A12 = 0;
        end





        % top right of eight
        T13 = I(1:floor(0.30*height),floor(0.25*width):end);

        Q13 = nnz(T13)/(size(T13,1)*size(T13,2));


        if Q13 < 0.12
            A13 = 1;
        elseif Q13 >= 0.12
            A13 = 0;
        end





        % bottom right of eight
        T14 = I(floor(0.70*height):end,floor(0.80*width):end);

        Q14 = nnz(T14)/(size(T14,1)*size(T14,2));


        if Q14 < 0.08
            A14 = 1;
        elseif Q14 >= 0.08
            A14 = 0;
        end




        % difference of nine with one, Q2, Q3 also
        % properties of nine
        T15 = I(floor(0.75*height):end,1:end);

        Q15 = separatedcharacter(T15);


        if Q15 >= 1.72
            A15 = 1;
        elseif Q15 <= 1.72
            A15 = 0;
        end





        % bottom left of Ga
        T16 = I(floor(0.80*height):end,1:floor(0.45*width));

        Q16 = nnz(T16)/(size(T16,1)*size(T16,2));

        if Q16 < 0.12
            A16 = 1;
        elseif Q16 >= 0.12
            A16 = 0;
        end




        % Difference between Kha and Ga
        T17 = I(floor(0.65*height):floor(0.85*height),...
            floor(0.45*width):floor(0.65*width));

        Q17 = nnz(T17)/(size(T17,1)*size(T17,2));


        if Q17 < 0.25
            A17 = 1;
        elseif Q17 >= 0.25
            A17 = 0;
        end





        %% this portion might be unncessary for the research

        %mirror properties of image
        %both vertical and horizontal flip
```

```
sym = flipdim(flipdim(I,2),1);


sym = sym-I;

Qsym = nnz(sym)/(size(I,1)*size(I,2));


%Symmetry about axis
if Qsym < 0.22
    Asym = 1;
elseif Qsym >= 0.22
    Asym = 0;
end






% no. holes within character e.g., 1 for zero,
% 2 for four

Thole = zeros(size(I,1)+2, size(I,2)+2);
for f = 2:size(Thole,1)-1
    for g = 2:size(Thole,2)-1
        Thole(f,g) = I(f-1,g-1);
    end
end

Thole = ~Thole;
%if Q26 = 2, means one hole, 3 means 2 holes and so on
[~, Qhole] = bwlabel(Thole);

Ahole = Qhole-1;





% nos. of avg vertical lines in character
Qsep = separatedcharacter(I);
%see above, also condition given in top about image size

if Qsep >= 2.25
    Asep = 2;
else
    Asep = 1;
end




%% Necessary portion stars again


%top middle portion of sylhet and rajshahi
T18 = I(1:floor(0.15*height),...
    floor(0.35*width):floor(0.65*width));
```

```
Q18 = nnz(T18)/(size(T18,1)*size(T18,2));

if Q18 <= 0.05    %indicates sylhet and rajshahi
    A18 = 1;
elseif Q18 > 0.05
    A18 = 0;
end




%top left of metro and chotto, also rajshahi

T19 = I(1:floor(0.15*height),1:floor(0.45*width));

Q19 = nnz(T19)/(size(T19,1)*size(T19,2));

if Q19 <= 0.1
    A19 = 1;
elseif Q19 > 0.1
    A19 = 0;
end




%top right blank of barisal

T20 = I(1:floor(0.15*height),...
    floor(0.45*width):floor(0.85*width));

Q20 = nnz(T20)/(size(T20,1)*size(T20,2));

if Q20 <= 0.1     %% i.e., barisal
    A20 = 1;
elseif Q20 > 0.1
    A20 = 0;
end




% separatedcharacter gives difference between
% metro and chotto

T21 = I(floor(0.15*height):end,1:end);
Q21 = separatedcharacter(T21);

if Q21 < 6.15 && Q21 > 3.90
    A21 = 1;  % as well as A31 indicates metro
elseif Q21 > 2.4 && Q21 < 3
    A21 = 2;  %as well as A31 indicates chotto
elseif Q21 > 6.5
    %as well as A32 indicates barisal & Rajshahi
    A21 = 3;
else A21 = 0;
end
```

```
% A33 and A34 both > 0.2 indicates either
% Dhaka or Khulna
% bottom right portion free indicates Khulna

T22 = I(floor(0.92*height):end,floor(0.35*width):end);
Q22 = nnz(T22)/(size(T22,1)*size(T22,2));

if Q22 <= 0.05
    % indicates Khulna, remember condition
    % A33 and A34 as well
    A22 = 1;

elseif Q22 > 0.05
    % indicates Dhaka, remember condition
    % A33 and A34 as well
    A22 = 0;
end




% Condition for Ha, matra at top and character
% similar of two

T23 = I(1:floor(0.02*height),1:end);
Q23 = nnz(T23)/(size(T23,1)*size(T23,2));

if Q23 >= 0.4
    % matra, remember to match with A6 as well...
    A23 = 1;
elseif Q23 < 0.4
    A23 = 0;
end





%difference between Ga and Seven
T24 = I(1:floor(0.02*height),1:floor(0.6*width));
Q24 = nnz(T24)/(size(T24,1)*size(T24,2));

if Q24 <= 0.02
    A24 = 1;      % Ga
elseif Q24 > 0.02
    A24 = 0;      %Seven
end




%top left of Dhaka and Khulna, also rajshahi

T25 = I(floor(0.05*height):floor(0.25*height),1:end);

Q25 = nnz(T25)/(size(T25,1)*size(T25,2));
```

```
        if Q25 >= 0.15
            A25 = 1;
        elseif Q25 < 0.15
            A25 = 0;
        end







        %%%%%%%%% Qeustion Answer Session

        %%% Questions

        L{h,3} = Q1;
        L{h,4} = Q2;
        L{h,5} = Q3;
        L{h,6} = Q4;
        L{h,7} = Q5;
        L{h,8} = Q6;
        L{h,9} = Q7;
        L{h,10} = Q8;
        L{h,11} = Q9;
        L{h,12} = Q10;
        L{h,13} = Q11;
        L{h,14} = Q12;
        L{h,15} = Q13;
        L{h,16} = Q14;
        L{h,17} = Q15;
        L{h,18} = Q16;
        L{h,19} = Q17;
        L{h,20} = Q18;
        L{h,21} = Q19;
        L{h,22} = Q20;
        L{h,23} = Q21;
        L{h,24} = Q22;
        L{h,25} = Q23;
        L{h,26} = Q24;
        L{h,27} = Q25;



        %%% Answers
        A{h,1} = L{h,1};
        A{h,2} = L{h,2};
        A{h,3} = A1;
        A{h,4} = A2;
        A{h,5} = A3;
        A{h,6} = A4;
        A{h,7} = A5;
        A{h,8} = A6;
        A{h,9} = A7;
        A{h,10} = A8;
        A{h,11} = A9;
        A{h,12} = A10;
        A{h,13} = A11;
        A{h,14} = A12;
        A{h,15} = A13;
        A{h,16} = A14;
        A{h,17} = A15;
        A{h,18} = A16;
        A{h,19} = A17;
        A{h,20} = A18;
        A{h,21} = A19;
        A{h,22} = A20;
        A{h,23} = A21;
        A{h,24} = A22;
        A{h,25} = A23;
        A{h,26} = A24;
        A{h,27} = A25;

        %%% Trial value variable is out
        % out1 denotes single character, then out2 = 0
        % out2 denotes multicharacter, then out1 = 0


        if Qsep < 2.25
            out2 = 0; % denotes single character

            if A1 == 1
                out1 = 1; % denotes 0
            elseif A2 == 1 && A3 == 1
                out1 = 2; % denotes 1
            elseif isequal([1 1 0 0],[A3 A4 A23 A17])
                out1 = 3; % denotes 2
            elseif isequal([1 1 1 0],[A5 A6 A7 A16])
                out1 = 4; % denotes 3
            elseif isequal([1 1 0 0 0],[A8 A9 A12 A13 A14])
                out1 = 5; % denotes 4
            elseif isequal([1 0 0 0],[A10 A23 A13 A16])
                out1 = 6; % denotes 5
            elseif isequal([0 1 0 0 0 0],[A10 A11 A13 A15 ...
                    A21 A12])
                out1 = 7; % denotes 6
            elseif isequal([1 0 0],[A12 A23 A24])
                out1 = 8; % denotes 7
            elseif isequal([1 1],[A13 A14])
                out1 = 9; % denotes 8
            elseif isequal([1 1 0],[A3 A15 A21])
                out1 = 10; % denotes 9
            elseif isequal([1 1 0 1],[A16 A17 A21 A24])
                out1 = 11; % denotes Ga
            elseif isequal([1 0 0 0],[A16 A17 A21 A23])
                out1 = 12; % denotes Kha
            elseif isequal([1 1],[A4 A23])
                out1 = 13; % denotes Ha

            else out1 = Inf; % undefiend

            end



        elseif Qsep >= 2.25
            out1 = 0; % denotes multi character


            if isequal([1 0],[A18 A19])
```

```
                out2 = 1; % denotes Sylhet
            elseif isequal([1 1 3],[A18 A19 A21])
                out2 = 2; % denotes Rajshahi
            elseif isequal([1 3],[A20 A21])
                out2 = 3; % denotes Barisal
            elseif isequal([1 2],[A19 A21])
                out2 = 4; % denotes Chittagong
            elseif isequal([1 1 0],[A19 A21 A20])
                out2 = 5; % denotes Metro
            elseif isequal(1,A25) && A22 == 1
                out2 = 6; % denotes Khulna
            elseif isequal(1,A25) && A22 == 0
                out2 = 7; % denotes Dhaka


            else out2 = Inf; % undefiend

            end

        end




        %% data for ANFIS modelling
        D{h,1} = A{h,1};
        D{h,2} = A{h,2};
        D{h,3} = Asym;  % Symmetry
        D{h,4} = Ahole;  % Nos. of holes within character
        D{h,5} = out1;  % Single or multi text based on nos
        % of vertical lines in average
        D{h,6} = out2; % Desired non-numeric value

    end
end

clearvars -except D
% clears all variables generated using this program except
% the variable named 'D'
```

APPENDIX-C

RecognizeLicensePlate.m file in MATLAB

```
%%% Code for Recognizing Model of ALPR BD

% closing all additional windows, clear everything and
% cleaning command window
close all;
clear all;
clc;



% Read an RGB Image
[FileName,PathName] = uigetfile({'*.jpg;*.tif;*.png;*.gif',...
    'All Image Files'},'Please Select an Image');

if FileName == 0 % if no image is selected
    display('No image was selected')
    pause(3)
    clear all
    clc
else

    % read the RGB image from the directory
    Inputimage = imread([PathName FileName]);

    % send the image to newrefine1 function which fially
    % generates modified license plate region BW image
    % and shows the image
    Inputimage = newrefine1(Inputimage);

    % show the input image for the user to define LP boundary
    imshow(Inputimage, []);
    % Axis on to collect boundary coordinates
    axis on;
    % define a font size for the image
    fontSize = 16;
    % give a title for the images using that font
    title('Original Grayscale Image', 'FontSize', fontSize);
    % Maximize figure.
    set(gcf, 'Position', get(0,'Screensize'));
    hold on

    % for multiple line for same character vector [] is used
    answer = ...
        questdlg(['Would you like to define License Plate ' ...
        'Area Manually?'], ...
        'Manual LP Area Define', ...
        'Yes','No thank you','No thank you');

    % Switch answer to follow desired directions
    switch answer
        % if user wants to manually define LP area
        case 'Yes'
            message = sprintf(['Simply click on 4 corners '...
                'of the numberplate']);
            uiwait(msgbox(message));
            coord = zeros(4,2);  % Pairs of x-y coordinates

            % take four coordinates one by one by clicking
            for n = 1:4
                [x,y] = ginput(1);
                coord(n,1:2) = [x,y];
                % show the point with '-o' label
                plot(coord(1:n,1:1),coord(1:n,2:2),'-o')
            end

            % if user denies to manually define LP area
        case 'No thank you'
            % take whole image as LP area from four corners
            coord = [1 1;1 size(Inputimage,1);...
                size(Inputimage,2) size(Inputimage,1);...
                size(Inputimage,2) 1];
            plot(coord(1:end,1:1),coord(1:end,2:2),...
                '-o','Linewidth', 5)

            % if user closes the dialogue box
        case ''
            % still assume as 'No thank you' option as before
            coord = [1 1;1 size(Inputimage,1);...
                size(Inputimage,2) size(Inputimage,1);...
                size(Inputimage,2) 1];
            plot(coord(1:end,1:1),coord(1:end,2:2),...
                '-o','Linewidth', 5)
    end

    % masking of the image done according to the defined
    % coordinates of LP area before
    bw = ~poly2mask(coord(1:end,1:1), coord(1:end,2:2),...
        size(Inputimage, 1), size(Inputimage, 2));  % 2D mask

    % Replicate the mask to make it 3D
    bw = repmat(bw, [1 1 size(Inputimage, 3)]);
    masking = Inputimage;

    % convert remaining portion into pure white pixels
    masking(bw) = 255;

    % Inputimage redefined as the masking image
    Inputimage = masking;

    % show the Inputimage
    imshow(Inputimage);
    % Axis on to give title using font size defined earlier
    axis on;
    % Title given
    title('Transparent License Plate Number', ...
        'FontSize', fontSize);

    % Remove all objects containing fewer than 30 pixels
    Inputimage = bwareaopen(Inputimage,30);
    % Wait 1 second
    pause(1);

    % Invert the BW image so black pixels converted to white
    % and vice versa for individual character detection as
```

```
% labeling done for connected white pixels
Inputimage = ~Inputimage;

% Pseudo Regionprops from the Inputimage as order is not
% modified
% L matrix of labeled objects and Ne returns no. of
% connected objects i.e., possible characters in LP
[L, Ne]=bwlabel(Inputimage);

% Creating bounding boxes around the connected objects
% note that this labels are unordered for BD LP format
propied = regionprops(L,'BoundingBox');



% Again invert the image to get black colored character
% on white background and showing that image
imshow(~Inputimage);
hold on

% creating a variable npropied which will contain ordered
% connected objects later on
npropied = cell(size(propied,1)*size(propied,1),1);
% area of boxes created as a measure of checking character
% zero vectors created to reduce execution time of program
% later on extra elements contianin zero will be eliminated
area_of_box = zeros(size(propied,1),1);

% inappropriate objects will be deleted of which matrix
% has been generated
connected_objects_to_be_deleted = zeros(size(propied,1)...
    *size(propied,1),1);

% Area of every bounding box made from Inputimage
% as 3rd and 4th element contains width height or vice versa
for n = 1:size(propied,1)
    area_of_box(n,1) = propied(n).BoundingBox(3)...
        *propied(n).BoundingBox(4);
end

% variable for taking counts, remaining elements will be
% eliminated based on these
NPR = 0;
COD = 0;

% for treating overlapping boundingbox of letters as one
for n = 1:size(propied,1)
    % trial area to compare the dominated area and overlap
    trial_area = zeros(size(propied,1),1);
    % variable for taking counts
    Y = 0;
    % all boundingbox characters will be checked each time
    for m = 1:size(propied,1)
        % if the boxes are not the same and areas of two
        % boxes are not almost equal is area/size
        if m ~= n && overlapratio(propied(n).BoundingBox,...
                propied(m).BoundingBox) > 0.4
            % count increases
            Y = Y + 1;
            % area of the box taken
            trial_area(Y,1) = area_of_box(m);
        end
    end
    % extra elements are elimniated
    trial_area(Y+1:end,:) = [];

    % if no trial area is found from the condition of the
    % loop above
    if isempty(trial_area)
        % count increases
        NPR = NPR+1;
        % npropied of NPRth element replaces propied of n
        npropied{NPR,1} = propied(n);

        % otherwise if area of n is greater than the maximum
        % of the trial area
    elseif area_of_box(n) > max(trial_area)
        % count increases
        NPR = NPR+1;
        % npropied of NPRth element replaces propied of n
        npropied{NPR,1} = propied(n);
        % if non of above means the concerned area is smaller
        % which indicates overlapping and this element should
        % not be treated as a new character
    else
        % count increases
        COD = COD + 1;
        % this bounding box no. is stored for elimination
        connected_objects_to_be_deleted(COD,1) = n;
    end
end

% unused 0 elements of this vector are eliminiated
npropied(NPR+1:end,:) = [];

% unused 0 elements of this vector are eliminiated
connected_objects_to_be_deleted (COD+1:end,:) = [];

% if no npropied could be generated then replace it with
% propied as before
if isempty(npropied)
    npropied = propied;
end

% create a variable fal_order of letter equal to
% propied size first indicating false order of letter
fal_order_of_letter = (1:size(propied,1))';
% for taking x coordinates
x_of_letter = (1:size(propied,1))';
% for taking y coordinates
y_of_letter = (1:size(propied,1))';

% if any secondary boundingbox then delete those numbers
% from fal_order_of_letter
for g = 1:size(fal_order_of_letter)
    x_of_letter(g) = propied(g).BoundingBox(1);
    y_of_letter(g) = propied(g).BoundingBox(2);
    if any(connected_objects_to_be_deleted(:) == g)
        fal_order_of_letter(g) = 0;
        x_of_letter(g) = 0;
```

```
            y_of_letter(g) = 0;
        end
    end

    % delete unused rows having 0 values from all three
    fal_order_of_letter(fal_order_of_letter(:,1) == 0, :) = [];
    x_of_letter(x_of_letter(:,1) == 0, :) = [];
    y_of_letter(y_of_letter(:,1) == 0, :) = [];



    % new false order variable order matrices defined as changing
    % elements of fal_order_of_letter could hamper logical basis
    % also x and y variables for those
    % fal_order1 for first row of the LP i.e. 'Dhaka Metro Ga'
    % fal_order2 for second row of the LP i.e., '11-2200'
    fal_order_of_letter1 = zeros(size(fal_order_of_letter,1),1);
    x_of_letter1 = zeros(size(fal_order_of_letter,1),1);
    y_of_letter1 = zeros(size(fal_order_of_letter,1),1);
    fal_order_of_letter2 = zeros(size(fal_order_of_letter,1),1);
    x_of_letter2 = zeros(size(fal_order_of_letter,1),1);
    y_of_letter2 = zeros(size(fal_order_of_letter,1),1);

    % variable for taking counts
    FOL1 = 0;
    FOL2 = 0;

    for b = 1: size(fal_order_of_letter,1)
        % taking a line towards middle(apprx) of the max and
        % min y coord of image portion selection by user
        if y_of_letter(b) <= (max(coord(1:end,2)) + ...
                min(coord(1:end,2)))/2

            % count increases
            FOL1 = FOL1+1;
            % first row, ordered characters added accordingly
            fal_order_of_letter1(FOL1,1) = fal_order_of_letter(b);
            x_of_letter1(FOL1,1) =  x_of_letter(b);
            y_of_letter1(FOL1,1) = y_of_letter(b);

            % else it goes to the second row characters
        else
            % count increases
            FOL2 = FOL2 + 1;
            % second row, ordered characters added accordingly
            fal_order_of_letter2(FOL2,1) = fal_order_of_letter(b);
            x_of_letter2(FOL2,1) = x_of_letter(b);
            y_of_letter2(FOL2,1) = y_of_letter(b);
        end
    end

    % Extra elements on the vectors/matrices are eliminated
    fal_order_of_letter1(FOL1+1:end,:) = [];
    x_of_letter1(FOL1+1:end,:) =  [];
    y_of_letter1(FOL1+1:end,:) = [];

    fal_order_of_letter2(FOL2+1:end,:) = [];
    x_of_letter2(FOL2+1:end,:) = [];
    y_of_letter2(FOL2+1:end,:) = [];
    % A new vector created to contain all of the characters
    % of both first and second row of the LP therefore, size
    % is equal to sum of those two vectors
    order_of_letter = zeros(size(fal_order_of_letter1, 1)+...
        size(fal_order_of_letter2, 1),1);

    % for taking counts (in case needed)
    OL = 0;

    % original order of letter is created by the entry of
    % ordered first row characters then ordered second row
    % characters
    for p = 1: size(fal_order_of_letter1, 1)
        % take the value and index of the letter having
        % minimum x value i.e., closer to left border
        [v,ind] = min(x_of_letter1);
        % increase counts
        OL = OL + 1;
        % reorder the vector according to that index number
        order_of_letter(OL,1) = fal_order_of_letter1(ind);
        x_of_letter1(ind) = inf;
    end


    % after top row completed, second row same operations starts
    for p = 1: size(fal_order_of_letter2, 1)
        [v,ind] = min(x_of_letter2);
        % increase counts
        OL = OL + 1;
        % reorder the vector according to that index number
        order_of_letter(OL,1) = fal_order_of_letter2(ind);
        x_of_letter2(ind) = inf;
    end
    % Now we have already got the order of letter...

    % eliminate unnecessary rows if exist
    order_of_letter(OL+1:end,:) = [];


    % show all identified chracters in LP bounded by rectangular
    % boxes on the LP image
    for n=1:size(npropied,1)
        rectangle('Position',npropied{n,1}.BoundingBox,...
            'EdgeColor','g','LineWidth',2)
    end
    hold off
    figure



    % Geometric Feature Extraction

    % create a matrix of Extracted_objects which will contain
    % geometric features of different characters
    Extracted_objects = cell(size(order_of_letter,1),3);

    % variable for counts
    EO = 0;
```

```
for n=1:size(order_of_letter,1)
    % ignoring the connected objects to be deleted made
    % before for overlapping characters
    if ~ismember(order_of_letter(n),...
            connected_objects_to_be_deleted)
        % take only the connected objects searching the
        % coordinates of label made on the LP image
        [r,c] = find(L==order_of_letter(n));
        % Collect extent of this object in the image
        n1 = Inputimage(min(r):max(r),min(c):max(c));
        % count increases
        EO = EO+1;
        % include in first column
        Extracted_objects{EO,1} = n1;
    end
end

% eliminate unnecessary row of the matrix, now size fixed
Extracted_objects(EO+1:end,:) = [];

% Replace the first column with cropped image
for n = 1: size(Extracted_objects,1)
    I = Extracted_objects{n,1};
    I = I(any(I ~= 0,2),:);
    I = I';
    I = I(any(I ~= 0,2),:);
    I = I';
    Extracted_objects{n,1} = I;
end


% avg black dots along total width & total height for
% each distinguished character
for n = 1:size(Extracted_objects,1)
    II = Extracted_objects{n,1};
    % vector defined with zeros initially
    M = zeros(size(II,1),1);
    % variable for counts (if needed later on)
    Msz = 0;

    for d = 1:size(II,1)
        if II(d,1) == 1 % dots
            % count is the variable for calculating dots
            count = 1;
        else count = 0;
        end
        for x = 2: size(II,2)
            if II(d,x) == II(d,x-1)
            elseif II(d,x) == 1
                count = count+1;
            elseif II(d,x) == 0
            end
        end
        Msz = Msz+1;
        M(d,1)=count;
    end
    II = II';
    N = zeros(size(II,1),1);
    Nsz = 0;
    for d = 1:size(II,1)
        if II(d,1) == 1
            count = 1;
        else count = 0;
        end
        for x = 2: size(II,2)
            if II(d,x) == II(d,x-1)
            elseif II(d,x) == 1
                count = count+1;
            elseif II(d,x) == 0
            end
        end
        Nsz = Nsz + 1;
        N(d,1)=count;
    end
    II = II';

    % Eliminate unnecessary rows
    M(Msz+1:end,:) = [];
    N(Nsz+1:end,:) = [];

    % Now determine average dots along width and height
    Extracted_objects{n,2} = mean(M);
    Extracted_objects{n,3} = mean(N);
end


% new vector containing status of unnecessary dots or hyphens
KK = cell(size(Extracted_objects,1),1);

% if average dots along either width or height is found 1,
% it means that character is hyphen like rectangular object

for n = 1:size(KK,1)
    % if either along width or height avg. dot is found 1
    if Extracted_objects{n,2} == 1 || ...
            Extracted_objects{n,3} == 1
        % status in KK is '1'
        KK{n,1} = 1;
        % otherwise status '0'
    else KK{n,1} = 0;
    end
end

% for deleting objects having pure black
% pixels throughout e.g., hypens
% if status of KK is 1, then delete those objects from
% the list of the characters
Extracted_objects = Extracted_objects(~any(cellfun(@(x)...
    isequal(x,1), KK,'un',1),2),:);



% if still any irrelavant area exists, those will be deleted
% logic for identifying those characters is to see the average
% of areas of all characters remaining after
% deleting from KK status
```

```
    % new variable created with '0' initially
    area_avg_sum = 0;
    for n = 1:size(Extracted_objects,1)
        % for each character, the area is added every time
        area_avg_sum = area_avg_sum + ...
            size(Extracted_objects{n,1},1)...
            *size(Extracted_objects{n,1},2);
    end

    % Finally average of the area calculated dividing sum
    % by the no. of characters
    area_avg_sum = area_avg_sum/size(Extracted_objects,1);

    % New matrix created to collect actual characters' features
    Extraction2 = cell(size(Extracted_objects,1),3);
    % variable for counting
    EX2 = 0;



    % code for refining existing irrelevant objects assuming
    % they contain less then or equal 30% of the average height
    % of all objects after removing pure  black dots

    for n = 1:size(Extracted_objects,1)
        if size(Extracted_objects{n,1},1)...
                *size(Extracted_objects{n,1},2) > 0.3*area_avg_sum
            I = Extracted_objects{n,1};

            % Single or multi character resize algorithm
            if separatedcharacter(I) >= 2.25
                % Umo and Borgiyo Jo might create problem
                % which is not covered in this research
                I = imresize(I,[256 512]);
                % elseif separatedcharacter(I) < 2.25
            else
                I = imresize(I,[256 256]);
            end
            % count increases
            EX2 = EX2+1;
            % First column contains image matrix uint8
            Extraction2{EX2,1} = I;
            % second and thrid column average dots
            % it might not be necessary though
            Extraction2{EX2,2} = Extracted_objects{n,2};
            Extraction2{EX2,3} = Extracted_objects{n,3};

        end
    end

    % delete all unnecessary rows
    Extraction2(EX2+1:end,:) = [];



    % a variable P created which will be used to send into
    % the function extractfeature built keeping similarties
    % with FeatureTraining.m
    P = cell(1,size(Extraction2,1));

    % P will contain all images of finalized characters
    for n = 1:size(Extraction2,1)
        P{1,n} = Extraction2{n,1};
    end



    % if we provide blank image initially for ALPR then
    % this logic will work
    if isempty(P)
        display('No character was found')
        close all
        for n = 1:5
            beep
            pause(1)
        end
        clear all
        clc

        % otherwise it will work perfectly accepting else
    else

        %calling extractfeature function
        [D,A,S] = extractfeature(P);

        % D has 6 columns see extractfeature function for details
        for n = 1:size(D,1)
            % Subploting recognized characters' images
            subplot(1,size(D,1),n);imshow(D{n,2});
        end




        % a file named Trained_FIS.fis will have to be saved
        % with trained data, that file will be called

        % read the ANFIS calibrated file
        fis = readfis('Training_FIS');

        % a variable created which will contain 2 inputs for
        % ANFIS and later on adds output of ANFIS to another
        % additonal column
        u = zeros(size(D,1),2);


        for n = 1:size(D,1)
            % 5th element of D is out1, i.e., 1st input for ANFIS
            u(n,1) = D{n,5};
            % 6th element of D is out2, i.e., 2nd input for ANFIS
            u(n,2) = D{n,6};
        end

        % third column of u will contain the output of ANFIS
        u(:,3) = evalfis(u,fis);

        % a vector named RES will contain recognized
        % characters in text format, therefore size equal to
        % no. of characters has been chosen
        RES = cell(size(u,1),1);
        % for each character in u
        for n = 1:size(u,1)
            % output variable is equal to the output of ANFIS
            % for that character
            output = u(n,3);

            % output rounded to nearby with limited extents
            % total 20 characters are numbered here
            if output >= -0.9 && output< 0.1
                result = 0;
            elseif output >= 0.9 && output< 1.1
                result = 1;
            elseif output >= 1.9 && output< 2.1
                result = 2;
            elseif output >= 2.9 && output< 3.1
                result = 3;
            elseif output >= 3.9 && output< 4.1
                result = 4;
            elseif output >= 4.9 && output< 5.1
                result = 5;
            elseif output >= 5.9 && output< 6.1
                result = 6;
            elseif output >= 6.9 && output< 7.1
                result = 7;
            elseif output >= 7.9 && output< 8.1
                result = 8;
            elseif output >= 8.9 && output< 9.1
                result = 9;
            elseif output >= 9.9 && output< 10.1
                result = 10;
            elseif output >= 10.9 && output< 11.1
                result = 11;
            elseif output >= 11.9 && output< 12.1
                result = 12;
            elseif output >= 12.9 && output< 13.1
                result = 13;
            elseif output >= 13.9 && output< 14.1
                result = 14;
            elseif output >= 14.9 && output< 15.1
                result = 15;
            elseif output >= 15.9 && output< 16.1
                result = 16;
            elseif output >= 16.9 && output< 17.1
                result = 17;
            elseif output >= 17.9 && output< 18.1
                result = 18;
            elseif output >= 18.9 && output< 19.1
                result = 19;

                % if no similar character value found then
            else result = inf;
            end

            % another coulmn is created where output of ANFIS
            % are rounded as above are kept
            u(n,4) = result;

            % according to the 'result', 'RES' vector is filled
            % up with recognized character texts
            switch result
                case 0
                    RES{n,1} = '0';
                case 1
                    RES{n,1} = '1';
                case 2
                    RES{n,1} = '2';
                case 3
                    RES{n,1} = '3';
                case 4
                    RES{n,1} = '4';
                case 5
                    RES{n,1} = '5';
                case 6
                    RES{n,1} = '6';
                case 7
                    RES{n,1} = '7';
                case 8
                    RES{n,1} = '8';
                case 9
                    RES{n,1} = '9';
                case 10
                    RES{n,1} = 'Ga ';
                case 11
                    RES{n,1} = 'Kha ';
                case 12
                    RES{n,1} = 'Ha ';
                case 13
                    RES{n,1} = 'Sylhet ';
                case 14
                    RES{n,1} = 'Rajshahi ';
                case 15
                    RES{n,1} = 'Barisal ';
                case 16
                    RES{n,1} = 'Chittagong ';
                case 17
                    RES{n,1} = 'Metro ';
                case 18
                    RES{n,1} = 'Khulna ';
                case 19
                    RES{n,1} = 'Dhaka ';
                case Inf
                    RES{n,1} = '!!';
                otherwise
                    RES{n,1} = '!!';
            end
        end

        %delete empty cells
        RES = RES(~any(cellfun('isempty', RES), 2), :);

        % a variable 'out' which will contain the total
        % recognized license plate number later on initially
        % started with empty character variable
        out = '';

        % each character text is added sequentially within
        % the variable 'out'
        for n = 1: size(RES,1)
            if n > 1
                % blank space after letters and words not numerals
```

```
                if ~ismember(RES{n-1,1},['0','1','2','3',...
                        '4','5','6','7','8','9'])
                    out = char(strcat(out,{' '},RES{n,1}));
                else out = strcat(out,RES{n,1});
                end
            else
                out = strcat(out,RES{n,1});
            end
        end
        % print the result in command window to display
        fprintf('The license plate number is:  %s\n', out);

    end
end

clearvars -except out
% it clears all variables generated using this program except
% the variable named 'out' containing license plate number
```

APPENDIX-D

newrefine1.m file in MATLAB

```
function Ref_image = newrefine1(J)
% Take RGB image as input

%threshold to convert into BW image
thresh = graythresh(J);

BW = im2bw(J,thresh);

%detect all boundaries within the image
[B,~] = bwboundaries(BW);

% A vector is created to receive areas of all boundaries
CC = zeros(size(B,1),1);
for n = 1:size(B,1)
    CC(n,1) = polygonarea(B{n,1})/(size(BW,1)*size(BW,2));
end

% A vector is created to keep intensity values from loop
average = zeros(size(B,1),1);

for n = 1:size(B,1)
    %Each bounrady element denoted as a variable in loop
    boundary = B{n};
    %calling boundary property function for shape and area
    [Shape,Area,~] = boundary_property(boundary,BW);
    %some conditions for possible LP area
    if CC(n,1) > 0.05 && (Shape > 0.25 && Shape < 0.90) &&...
            (Area/polygonarea(B{n,1}) > 0.85 && ...
            Area/polygonarea(B{n,1}) < 1.15)

        % Background color of that boundary will be checked
        bw = ~poly2mask(boundary(1:end,2:2), ...
            boundary(1:end,1:1), size(J, 1), size(J, 2));
        % 2D masking done

        % Replicate the mask to make it 3D
        bw = repmat(bw, [1 1 size(J, 3)]);
        masking = J;
        % remaining portion is converted to black
        masking(bw) = 0;
        %keeping other regions black only LP region in image
        h = masking;

        % converted to grayscale
        nn = rgb2gray(h);
        % as image is uint8 format, it is converted to double
        nn = im2double(nn);
        % unique values in the matrix used to create new one
        a = unique(nn);
        %histogram count is used
        out = [a,histc(nn(:),a)];
        summ = 0;
        needed = 0.01*sum(out(2:end,2:2));
        sz = size(out,1);
        for z = 1:sz-1
            if summ <= needed
                summ = summ + out(sz-z+1,2);
                s = sz-z;
            end
        end
        % average background color of possible LP regions
        average(n,1) = mean(out(s:end,1));
    end
end



% most intesity contianed LP region chosen as LP
r = 1;
avg = average(1);
for n = 1:size(average)
    if average(n) > avg
        avg = average(n);
        r = n;
    end
end

% four corner coordinates of the LP region taken
[~,~,POLY] = boundary_property(B{r,1},BW);

% new matrix fboundary containing all coordinates of actual
% LP boundary detected above

fboundary = B{r};

%Masking of the possible LP area of actual LP boundary
%detected above
bw = ~poly2mask(fboundary(1:end,2:2), fboundary(1:end,1:1), ...
    size(J, 1), size(J, 2));  % 2D mask

% Replicate the mask to make it 3D
bw = repmat(bw, [1 1 size(J, 3)]);
masking = J;
masking(bw) = 0;
h = masking;

% Show the masked image
imshow(h)
hold on;
% Plot a curve using the boundary points of the actual LP region
plot(fboundary(:,2), fboundary(:,1), 'b', 'LineWidth', 2);
hold off;

%Take the frame of the plotted image over the LP image
F = getframe;
% Take current directory of the program
L = pwd;
% Give a file name of refined image to save in the directory
fname = char('\Refined_image.jpg');
% Add full directory path with the image name
L = strcat(L,fname);
% Save the image in that directory
imwrite(F.cdata,L)
% read the refined image
```

```
F = imread('Refined_image.jpg');
close all;

% Take top left and top right coordinates to get alignment
% of LP region

dsy2 = POLY(2,1);
dsy1 = POLY(1,1);
dsx2 = POLY(2,2);
dsx1 = POLY(1,2);

% rotation angle measured
rotation = atand((dsy2 - dsy1)/(dsx2 - dsx1));

% image rotated using rotation angle
Rotated_image = imrotate(F,rotation);

%show the rotated image
imshow(Rotated_image);

%Take the frame of the rotated image over the LP image
F = getframe;

% Take current directory of the program
L = pwd;

% Give a file name of refined image to save in the directory
% As same name, it overwrites the current image
fname = char('\Refined_image.jpg');
% add the image name for creating full path
L = strcat(L,fname);
% saving the image as file in the directory
imwrite(F.cdata,L)
close all

% Rotated image is sent to the new function newrefine2
Ref_image = newrefine2(Rotated_image);

end
```

APPENDIX-E

Boundary_property.m file in MATLAB

```matlab
function [shape,area,POLY] = boundary_property(X,BW)
% This function returns two scalers 'shape' and 'area'
% and one matrix POLY from a polygonal region's coordinate
% matrix named 'X' within the BW image named 'BW'

% four outer corners of the the polygon assuming parallelogram
% are defined as 'a', 'b', 'c' and 'd' measurement is done
% by taking lowest distance from four corners of BW image

% 'a' is the left top so distance from (0,0) taken
a = [Inf Inf]; %first one Y and second one X
for n = 1:size(X,1)
    if sqrt(X(n,1)^2 + X(n,2)^2) < sqrt(a(1,1)^2 + a(1,2)^2)
    a = [X(n,1) X(n,2)];
    end
end

% 'b' is the right top corner of polygon so distance from
% right top of BW image taken
y = 1;
x = size(BW,2);
b = [Inf Inf]; %first one Y and second one X
for n = 1:size(X,1)
    if sqrt((X(n,1)-y)^2 + (X(n,2)-x)^2) < sqrt((b(1,1)-y)^2 ...
            + (b(1,2)-x)^2)
    b = [X(n,1) X(n,2)];
    end
end


% 'c' is the bottom left corner of polygon so distance from
% bottom left of BW image taken
y = size(BW,1);
x = 1;
c = [Inf Inf]; %first one Y and second one X
for n = 1:size(X,1)
    if sqrt((X(n,1)-y)^2 + (X(n,2)-x)^2) < sqrt((c(1,1)-y)^2 ...
            + (c(1,2)-x)^2)
    c = [X(n,1) X(n,2)];
    end
end


% 'd' is the bottom right corner of polygon so distance from
% bottom right of BW image taken
y = size(BW,1);
x = size(BW,2);
d = [Inf Inf]; %first one Y and second one X
for n = 1:size(X,1)
    if sqrt((X(n,1)-y)^2 + (X(n,2)-x)^2) < sqrt((d(1,1)-y)^2 ...
            + (d(1,2)-x)^2)
    d = [X(n,1) X(n,2)];
    end
end
```

```
% four corners of the parallelogram assumed polygon
ax = a(1,2);
ay = a(1,1);
bx = b(1,2);
by = b(1,1);
cx = c(1,2);
cy = c(1,1);
dx = d(1,2);
dy = d(1,1);
% therefore POLY matrix is found
POLY = [ay,ax;by,bx;dy,dx;cy,cx];

% average of 'ab' and 'cd' length taken as width
x_shape = mean([sqrt((bx - ax)^2 + (by - ay)^2) ...
    sqrt((dx - cx)^2 + (dy - cy)^2)]);

% average of 'ac' and 'bd' length taken as height
y_shape = mean([sqrt((ax - cx)^2 + (ay - cy)^2) ...
    sqrt((bx - dx)^2 + (by - dy)^2)]);

% height to width ratio of the rectangle assumed polygon
% is 'shape'
shape = y_shape/x_shape;

% area of the paralellopgram assumed region is calculated
% using customized function 'polygonarea'
area = polygonarea(POLY);

end
```

APPENDIX-F

polygonarea.m file in MATLAB

```
function I = polygonarea(B)

% B is a matrix having 2 columns and n rows having
% 'n' number of x,y coordinates each representing coordinates
% of a polygonal region

% two variables defined with '0' initial value
add = 0;
subtract = 0;

% note that area of polygon having coordinates
%[(x1,y1(,(x2,y2),...(xn,yn)] is
% = | [(x1y2+x2y3+...+xny1)-(y1x2+y2x3+...ynx1)]/2 |
% = | [add - subtract]/2 |



% based on above formula the program is created below
for n = 1:size(B,1)
    if n < size(B,1)
        add = add+ B(n,1)*B(n+1,2);
        subtract = subtract + B(n,2)*B(n+1,1);
    elseif n == size(B,1)
        add = add+ B(n,1)*B(1,2);
        subtract = subtract + B(n,2)*B(1,1);
    end
end

% finally I gives the area of the polygon
I = abs((add - subtract)/2);
end
```

APPENDIX-G

nmax.m file in MATLAB

```
function [K,Index] = nmax(X, i)
% this function calculates ith maximum value within X vector
% as K, and also finds first index of that value (if multiple
% ith maximum value exists) within X vector

% if highest maximum value is expected
if i == 1
    [K,Index] = max(X);

% if any other maximum value except highest is expected
else
    for n = 2:i
        [~,Index] = max(X);
        X(Index) = 0;
        [K,Index] = max(X);
    end
end
end
```

APPENDIX-H

newrefine2.m file in MATLAB

```
function I = newrefine2(i)
% this function assumes a properly oriented (rotated to
% horizontal) RGB image of license plate as i and modifies
% the region to a refined BW image of LP named I where
% characters are in black color within white background


% convert to gray scale from rgb format
i1 = rgb2gray(i);

% mapping intensity values, it saturates the bottom 1% and
% the top 1% of, all pixel values default
i11 = imadjust(i1);


% find all possible edges in the intensity image, it also
% coverts image to binary image
i2 = edge(i11,'canny',0.4);


% creating structural element of square shape to a dilated
% image using strel
se = strel('square',2);

% dilating i.e., thickening of image using
% structural element 'se' created before
i3 = imdilate(i2,se);


% filling holes of the image which converts the license
% plate area to white
i4 = imfill(i3,'holes');


% still after filling hole if some areas open are found then
% those are eliminated using bwareaopen where pixel limit
% is fixed as 'sz'

sz = floor(0.1*size(i4,1)*size(i4,2));

% inintially refined image 'e'
e = bwareaopen(i4, sz);


% evaluating threshold value for converting original image
% to binary image IBW later on
threshold = graythresh(i);
IBW = im2bw(i,threshold);


% referring to 'e', blank and filled spaces are
% marked in IBW referring same coordinates
```

```
for f = 1:size(IBW,1)
    for g = 1:size(IBW,2)
        if ~e(f,g) == 1
            IBW(f,g) = 1;
        end
    end
end

% IBW image is inverted i.e., black pixels to white pixels
% and vice versa
I = ~IBW;




% we will be removing layers from the original rgb image,
% that is why we are individually removing the same portion
% from r, g and b ... eventually combine those to get original
% rgb image having only licenseplate afterwards

redChannel = i(:,:,1);
greenChannel = i(:,:,2);
blueChannel = i(:,:,3);

    redChannel = redChannel(any(I ~= 0,2),:);
    greenChannel = greenChannel(any(I ~= 0,2),:);
    blueChannel = blueChannel(any(I ~= 0,2),:);
    I = I';
    redChannel = redChannel';
    greenChannel = greenChannel';
    blueChannel = blueChannel';
    redChannel = redChannel(any(I ~= 0,2),:);
    greenChannel = greenChannel(any(I ~= 0,2),:);
    blueChannel = blueChannel(any(I ~= 0,2),:);
%     I = I';
    redChannel = redChannel';
    greenChannel = greenChannel';
    blueChannel = blueChannel';

    % concatenating three channels to create one RGB image
    I =  cat(3, redChannel, greenChannel, blueChannel);

% converting the image to RGB
I = rgb2gray(I);

% taking threshold value for converting grayscale to BW image
threshold = graythresh(I);
I = im2bw(I,threshold);
% clearing unnecessary border pixels from I
I = imclearborder(~I);
I = ~I;
% showing figure of I as well
figure,imshow(I)
end
```

APPENDIX-I

overlapratio.m file in MATLAB

```
function overlappingratio = overlapratio(first,second)
% this functin returns overlapratio of two boundingboxes
% named 'first' and 'second'
% in BoundingBox elements are [xleft,ytop,width,height]
% therefore, y is measured from top

XA1 = first(1); % taking xleft value
YA1 = first(2); % taking ytop value
XA2 = first(1)+first(3); % taking xright value
YA2 = first(2)+first(4); % taking ybottom value

XB1 = second(1); % taking xleft value
YB1 = second(2); % taking ytop value
XB2 = second(1)+second(3); % taking xright value
YB2 = second(2)+second(4); % taking ybottom value

% SI is the overlapped portion's area (Intersecting area)

SI = max(0,min(XA2,XB2)-max(XA1,XB1))...
    *max(0,min(YA2,YB2)-max(YA1,YB1));

% area of first boundingbox
SA = first(3)*first(4);
% area of second boundingbox
SB = second(3)*second(4);

% overlappingratio in this function is defined as the ratio
% of intersecting area to the smaller boundingbox area
overlappingratio = SI/min(SA,SB);
end
```

APPENDIX-J

extractfeature.m file in MATLAB

```
function [D,A,L] = extractfeature(P)
% This function returns three matrices containin geometric
% features of the images in P, A containing 1 and 0 status
% based on limit, L containing actual value of features and D
% containing features inputs for ANFIS in last two columns


L = cell(size(P,2),30); %Contains Questions

D = {size(P,2),6}; %ANFIS Data

A = cell(size(P,2),30); %contains answers


% for each character images in P this loop runs
for h = 1: size(P,2)
    % Giving name according to the character order
    L{h,1} = strcat('Featured Character no. ',num2str(h));

    % first element is the image
    I = P{1,h};


    if size(I,3)==3 % RGB image
        I=rgb2gray(I);
    end
    % if grayscale image is given as input
    if size(I,3) < 3 && any(I(:) > 1)
        threshold = graythresh(I);
        I = ~im2bw(I,threshold);
    end

    % removal of unnecessary dots
    I = bwareaopen(I,size(I,1));

    % get cropped image for actual extent of characters
    I = I(any(I ~= 0,2),:);
    I = I';
    I = I(any(I ~= 0,2),:);
    I = I';

    % initiall resize might be skipped also
    I = imresize(I,[256 512]);

    % resizing based on single and multi character types
    if separatedcharacter(I) >= 2.25  % multi characters
        % Umo and Borgiyo Jo might create problem
        % though those are not covered in the research
        I = imresize(I,[256 512]);
    else % single characters
        I = imresize(I,[256 256]);
    end
    % might be unnecessary
    width = size(I,2);
    height = size(I,1);

    % two limits set, the character nnz value must be within
    % this range
    ulim = 0.40;
    llim = 0.30;
    while ~(nnz(I)/(size(I,1)*size(I,2)) > llim && nnz(I)/...
            (size(I,1)*size(I,2)) < ulim)
        if nnz(I)/(size(I,1)*size(I,2)) <= llim
            I = bwmorph(I,'thick',1.1);
        elseif nnz(I)/(size(I,1)*size(I,2)) >= ulim
            I = bwmorph(I,'thin',1.2);
        end
    end


    L{h,2} = I;

    % middle of zero
    T1 = I(floor(0.3*height):floor(0.7*height),...
        floor(0.3*width):floor(0.7*width));

    Q1 = nnz(T1)/(size(T1,1)*size(T1,2));
    if Q1 < 0.12
        A1 = 1;
    elseif Q1 >= 0.12
        A1 = 0;
    end



    %left middle of one
    T19 = I(floor(0.3*height):floor(0.7*height),...
        1:floor(0.3*width));

    Q2 = nnz(T19)/(size(T19,1)*size(T19,2));

    if Q2 < 0.12
        A2 = 1;
    elseif Q2 >= 0.12
        A2 = 0;
    end



    %right top side of one
    T3 = I(1:floor(0.5*height),floor(0.5*width):end);

    % triangular area taken
    trin1 = zeros(size(T3,1),size(T3,2));
    for f = 1:size(T3,1)
        for g = f:size(T3,2)
            trin1(f,g) = T3(f,g);
        end
    end
    Q3 = nnz(trin1)/(size(trin1,1)*size(trin1,2)/2);

    if Q3 < 0.12
        A3 = 1;
    elseif Q3 >= 0.12
        A3 = 0;
    end





    %bottom left side of two
    T4 = I(floor(0.7*height):end,1:floor(0.8*width));
    trin2 = zeros(size(T4,1),size(T4,2));

    tn2y = size(T4,1);
    tn2x = size(T4,2);

    for f = 1:tn2y
        for g = 1:floor(tn2x*f/tn2y)
            trin2(f,g) = T4(f,g);
        end
    end


    Q4 = nnz(trin2)/(0.5*tn2y*tn2x);

    if Q4 < 0.12
        A4 = 1;
    elseif Q4 >= 0.12
        A4 = 0;
    end





    %top left side of three
    T5 = I(1:floor(0.15*height),1:floor(0.35*width));

    Q5 = nnz(T5)/(size(T5,1)*size(T5,2));


    if Q5 < 0.08
        A5 = 1;
    elseif Q5 >= 0.08
        A5 = 0;
    end




    %top left side downward of three
    T6 = I(1:floor(0.3*height),...
        floor(0.15*width):floor(0.30*width));

    Q6 = nnz(T6)/(size(T6,1)*size(T6,2));

    if Q6 < 0.15
```

```
        A6 = 1;
    elseif Q6 >= 0.15
        A6 = 0;
    end




    % middle of three
    T7 = I(floor(0.45*height):floor(0.7*height),...
        floor(0.35*width):floor(0.65*width));

    Q7 = nnz(T7)/(size(T7,1)*size(T7,2));


    if Q7 < 0.15
        A7 = 1;
    elseif Q7 >= 0.15
        A7 = 0;
    end




    %bottom middle of four
    T8 = I(floor(0.65*height):floor(0.75*height),...
        floor(0.45*width):floor(0.55*width));

    Q8 = nnz(T8)/(size(T8,1)*size(T8,2));


    if Q8 < 0.15
        A8 = 1;
    elseif Q8 >= 0.15
        A8 = 0;
    end





    % middle part of four
    T9 = I(floor(0.45*height):floor(0.55*height),...
        floor(0.25*width):floor(0.75*width));

    Q9 = nnz(T9)/(size(T9,1)*size(T9,2));

    if Q9 >= 0.80
        A9 = 1;
    elseif Q9 < 0.80
        A9 = 0;
    end






    % right portion of five
    fT10 = flipdim(I,2); %horizontal flip
    T10 = fT10(floor(0.35*height):end,1:floor(0.25*width));
    trin10 = zeros(size(T10,1),size(T10,2));

    tx10 = size(trin10,2);
    ty10 = size(trin10,1)/2;

    for f = 1:size(trin10,1)
        if f < 0.5*size(trin10,1)
            for g = 1:round(f*tx10/ty10)
                trin10(f,g) = T10(f,g);
            end
        elseif f == 0.5*size(trin10,1)
            for g = 1:size(trin10,2)
                trin10(f,g) = T10(f,g);
            end

        elseif f> 0.5*size(trin10,1)
            for g = 1:((2*ty10-f)*tx10/ty10)
                trin10(f,g) = T10(f,g);
            end
        end
    end

    % triangular area
    Q10 = nnz(trin10)/(0.5*size(trin10,1)*size(trin10,2));

    if Q10 < 0.09
        A10 = 1;
    elseif Q10 >= 0.09
        A10 = 0;
    end



    % right top of six
    T11 = I(1:floor(0.25*height),floor(0.75*width):end);

    Q11 = nnz(T11)/(size(T11,1)*size(T11,2));

    if Q11 < 0.05
        A11 = 1;
    elseif Q11 >= 0.05
        A11 = 0;
    end



    % bottom left of seven
    T12 = I(floor(0.70*height):end,1:floor(0.75*width));

    Q12 = nnz(T12)/(size(T12,1)*size(T12,2));


    if Q12 < 0.12
        A12 = 1;
    elseif Q12 >= 0.12
        A12 = 0;
    end
```

```
% top right of eight
T13 = I(1:floor(0.30*height),floor(0.25*width):end);

Q13 = nnz(T13)/(size(T13,1)*size(T13,2));


if Q13 < 0.12
    A13 = 1;
elseif Q13 >= 0.12
    A13 = 0;
end




% bottom right of eight
T14 = I(floor(0.70*height):end,floor(0.80*width):end);

Q14 = nnz(T14)/(size(T14,1)*size(T14,2));


if Q14 < 0.08
    A14 = 1;
elseif Q14 >= 0.08
    A14 = 0;
end



% difference of nine with one, Q2, Q3 also
% properties of nine
T15 = I(floor(0.75*height):end,1:end);

Q15 = separatedcharacter(T15);


if Q15 >= 1.72
    A15 = 1;
elseif Q15 <= 1.72
    A15 = 0;
end




% bottom left of Ga
T16 = I(floor(0.80*height):end,1:floor(0.45*width));

Q16 = nnz(T16)/(size(T16,1)*size(T16,2));

if Q16 < 0.12
    A16 = 1;
elseif Q16 >= 0.12
    A16 = 0;
end




% Difference between Kha and Ga
T17 = I(floor(0.65*height):floor(0.85*height),...
    floor(0.45*width):floor(0.65*width));

Q17 = nnz(T17)/(size(T17,1)*size(T17,2));


if Q17 < 0.25
    A17 = 1;
elseif Q17 >= 0.25
    A17 = 0;
end





%% this portion might be unncessary for the research

%mirror properties of image
%both vertical and horizontal flip
sym = flipdim(flipdim(I,2),1);


sym = sym-I;

Qsym = nnz(sym)/(size(I,1)*size(I,2));


%Symmetry about axis
if Qsym < 0.22
    Asym = 1;
elseif Qsym >= 0.22
    Asym = 0;
end






% no. holes within character e.g., 1 for zero, 2 for four

Thole = zeros(size(I,1)+2, size(I,2)+2);
for f = 2:size(Thole,1)-1
    for g = 2:size(Thole,2)-1
        Thole(f,g) = I(f-1,g-1);
    end
end

Thole = ~Thole;
%if Q26 = 2, means one hole, 3 means 2 holes and so on
[~, Qhole] = bwlabel(Thole);

Ahole = Qhole-1;




% nos. of avg vertical lines in character
Qsep = separatedcharacter(I);
%see above, also condition given in top about image size

if Qsep >= 2.25   %Umo and Borgiyo Jo might create problem
    Asep = 2;
else
    Asep = 1;
end



%% Necessary portion stars again


%top middle portion of sylhet and rajshahi
T18 = I(1:floor(0.15*height),...
    floor(0.35*width):floor(0.65*width));

Q18 = nnz(T18)/(size(T18,1)*size(T18,2));

if Q18 <= 0.05    %indicates sylhet and rajshahi
    A18 = 1;
elseif Q18 > 0.05
    A18 = 0;
end




%top left of metro and chotto, also rajshahi

T19 = I(1:floor(0.15*height),1:floor(0.45*width));

Q19 = nnz(T19)/(size(T19,1)*size(T19,2));

if Q19 <= 0.1
    A19 = 1;
elseif Q19 > 0.1
    A19 = 0;
end




%top right blank of barisal

T20 = I(1:floor(0.15*height),...
    floor(0.45*width):floor(0.85*width));
```

```
Q20 = nnz(T20)/(size(T20,1)*size(T20,2));

if Q20 <= 0.1     %% i.e., barisal
    A20 = 1;
elseif Q20 > 0.1
    A20 = 0;
end



% separatedcharacter gives difference between
% metro and chotto

T21 = I(floor(0.15*height):end,1:end);
Q21 = separatedcharacter(T21);

if Q21 < 6.15 && Q21 > 3.90
    A21 = 1;  % as well as A31 indicates metro
elseif Q21 > 2.4 && Q21 < 3
    A21 = 2;  %as well as A31 indicates chotto
elseif Q21 > 6.5
    A21 = 3;  %as well as A32 indicates barisal & Rajshahi
else A21 = 0;
end



% A33 and A34 both > 0.2 indicates either Dhaka or Khulna
% bottom right portion free indicates Khulna

T22 = I(floor(0.92*height):end,floor(0.35*width):end);
Q22 = nnz(T22)/(size(T22,1)*size(T22,2));

if Q22 <= 0.05
% indicates Khulna, remember condition A33 and A34 as well
    A22 = 1;

elseif Q22 > 0.05
% indicates Dhaka, remember condition A33 and A34 as well
    A22 = 0;
end



% Condition for Ha, matra at top and character
% similar of two

T23 = I(1:floor(0.02*height),1:end);
Q23 = nnz(T23)/(size(T23,1)*size(T23,2));

if Q23 >= 0.4
% matra, remember to match with A6 as well...
    A23 = 1;
elseif Q23 < 0.4
    A23 = 0;
end
```

```
%difference between Ga and Seven
T24 = I(1:floor(0.02*height),1:floor(0.6*width));
Q24 = nnz(T24)/(size(T24,1)*size(T24,2));

if Q24 <= 0.02
    A24 = 1;      % Ga
elseif Q24 > 0.02
    A24 = 0;      %Seven
end



%top left of Dhaka and Khulna, also rajshahi

T25 = I(floor(0.05*height):floor(0.25*height),1:end);

Q25 = nnz(T25)/(size(T25,1)*size(T25,2));

if Q25 >= 0.15
    A25 = 1;
elseif Q25 < 0.15
    A25 = 0;
end






%%%%%%%%% Qeustion Answer Session

%%% Questions

L{h,3} = Q1;
L{h,4} = Q2;
L{h,5} = Q3;
L{h,6} = Q4;
L{h,7} = Q5;
L{h,8} = Q6;
L{h,9} = Q7;
L{h,10} = Q8;
L{h,11} = Q9;
L{h,12} = Q10;
L{h,13} = Q11;
L{h,14} = Q12;
L{h,15} = Q13;
L{h,16} = Q14;
L{h,17} = Q15;
L{h,18} = Q16;
L{h,19} = Q17;
L{h,20} = Q18;
L{h,21} = Q19;
L{h,22} = Q20;
```

```
    L{h,23} = Q21;
    L{h,24} = Q22;
    L{h,25} = Q23;
    L{h,26} = Q24;
    L{h,27} = Q25;



    %%% Answers
    A{h,1} = L{h,1};
    A{h,2} = L{h,2};
    A{h,3} = A1;
    A{h,4} = A2;
    A{h,5} = A3;
    A{h,6} = A4;
    A{h,7} = A5;
    A{h,8} = A6;
    A{h,9} = A7;
    A{h,10} = A8;
    A{h,11} = A9;
    A{h,12} = A10;
    A{h,13} = A11;
    A{h,14} = A12;
    A{h,15} = A13;
    A{h,16} = A14;
    A{h,17} = A15;
    A{h,18} = A16;
    A{h,19} = A17;
    A{h,20} = A18;
    A{h,21} = A19;
    A{h,22} = A20;
    A{h,23} = A21;
    A{h,24} = A22;
    A{h,25} = A23;
    A{h,26} = A24;
    A{h,27} = A25;

    %%% Trial value variable is out
    % out1 denotes single character, then out2 = 0
    % out2 denotes multicharacter, then out1 = 0



    if Qsep < 2.25
        out2 = 0; % denotes single character

        if A1 == 1
            out1 = 1; % denotes 0
        elseif A2 == 1 && A3 == 1
            out1 = 2; % denotes 1
        elseif isequal([1 1 0 0],[A3 A4 A23 A17])
            out1 = 3; % denotes 2
        elseif isequal([1 1 1 0],[A5 A6 A7 A16])
            out1 = 4; % denotes 3
        elseif isequal([1 1 0 0 0],[A8 A9 A12 A13 A14])
            out1 = 5; % denotes 4
        elseif isequal([1 0 0 0],[A10 A23 A13 A16])
            out1 = 6; % denotes 5
        elseif isequal([0 1 0 0 0 0],[A10 A11 A13 A15 A21 A12])
            out1 = 7; % denotes 6
        elseif isequal([1 0 0],[A12 A23 A24])
            out1 = 8; % denotes 7
        elseif isequal([1 1],[A13 A14])
            out1 = 9; % denotes 8
        elseif isequal([1 1 0],[A3 A15 A21])
            out1 = 10; % denotes 9
        elseif isequal([1 1 0 1],[A16 A17 A21 A24])
            out1 = 11; % denotes Ga
        elseif isequal([1 0 0 0],[A16 A17 A21 A23])
            out1 = 12; % denotes Kha
        elseif isequal([1 1],[A4 A23])
            out1 = 13; % denotes Ha

        else out1 = Inf; % undefiend

        end



    elseif Qsep >= 2.25
        out1 = 0; % denotes multi character


        if isequal([1 0],[A18 A19])
            out2 = 1; % denotes Sylhet
        elseif isequal([1 1 3],[A18 A19 A21])
            out2 = 2; % denotes Rajshahi
        elseif isequal([1 3],[A20 A21])
            out2 = 3; % denotes Barisal
        elseif isequal([1 2],[A19 A21])
            out2 = 4; % denotes Chittagong
        elseif isequal([1 1 0],[A19 A21 A20])
            out2 = 5; % denotes Metro
        elseif isequal(1,A25) && A22 == 1
            out2 = 6; % denotes Khulna
        elseif isequal(1,A25) && A22 == 0
            out2 = 7; % denotes Dhaka


        else out2 = Inf; % undefiend

        end

    end


    %% data for ANFIS modelling
    D{h,1} = A{h,1};
    D{h,2} = A{h,2};
    D{h,3} = Asym;  % Symmetry
    D{h,4} = Ahole;  % Nos. of holes within character
    D{h,5} = out1;  % Single or multi text based on nos
    % of vertical lines in average
    D{h,6} = out2; % Desired non-numeric value
end

end
```

APPENDIX-K

confusion_matrix.m function file in MATLAB

```
function [] = confusion_matrix(T,Y)
% this function returns a confusion matrix plot of two sets
% of data T and Y, where T indicates vector of true or actual
% results and Y indicates vector contianing the predicted
% results

% calculates unique results i.e., number of classes

% if the class size is greater than 2, i.e., multi class
% confusion matrix will have to be generated
if size(unique(T),2) > 2

M = size(unique(T),2);
N = size(T,2);
targets = zeros(M,N);
outputs = zeros(M,N);
% as multi-dimensional matrix, sub2ind is used for indices
targetsIdx = sub2ind(size(targets), T, 1:N);
outputsIdx = sub2ind(size(outputs), Y, 1:N);
targets(targetsIdx) = 1;
outputs(outputsIdx) = 1;
% Plot the confusion matrix
plotconfusion(targets,outputs)

% else if it is binary class confusion matrix then
else
    % simply plot it
    plotconfusion(T,Y)
end
end
```

APPENDIX-L

Chart containing Original License Plate State and Model Predicted License Plate State

	Original License Plate State	Predicted License Plate State
Observation no.	(exists indicated by 1 and does not exist indicated by 0)	
1	1	1
2	1	1
3	1	1
4	1	1
5	0	0
6	0	0
7	0	0
8	0	0
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	1	1
22	1	1
23	1	0
24	1	1
25	1	0
26	1	1
27	1	1
28	1	0
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1
36	1	1
37	1	1
38	1	1
39	1	1
40	1	1
41	0	0

	Original License Plate State	Predicted License Plate State
Observation no.	(exists indicated by 1 and does not exist indicated by 0)	
42	0	0
43	0	0
44	0	0
45	0	0
46	0	0
47	0	0
48	0	0
49	0	0
50	0	0
51	0	0
52	0	0
53	0	0
54	0	0
55	1	1
56	0	0
57	1	1
58	0	0
59	1	1
60	0	0
61	1	1
62	1	1
63	1	1
64	1	1
65	1	1
66	0	0
67	0	0
68	1	1
69	1	1
70	1	1
71	0	0
72	1	1
73	0	0
74	0	0
75	0	0
76	1	1
77	1	1
78	1	1
79	1	1
80	1	1
81	1	1
82	1	1
83	1	1
84	1	1
85	0	0
86	0	0

	Original License Plate State	Predicted License Plate State
Observation no.	(exists indicated by 1 and does not exist indicated by 0)	
87	0	0
88	0	0
89	0	0
90	0	0
91	0	0
92	1	1
93	1	1
94	1	1
95	1	1
96	1	1
97	1	1
98	1	1
99	1	1
100	1	1
101	1	1
102	1	1
103	1	1
104	1	0
105	1	1
106	1	1
107	1	1
108	1	1
109	1	1
110	1	1
111	1	1
112	1	1
113	1	0
114	1	1
115	1	1
116	1	1
117	1	1
118	1	0
119	1	1
120	1	1
121	1	0
122	1	1
123	1	1
124	1	1
125	1	1
126	1	1
127	1	1
128	1	1
129	1	1
130	1	0
131	1	1

	Original License Plate State	Predicted License Plate State
Observation no.	(exists indicated by 1 and does not exist indicated by 0)	
132	1	1
133	1	1
134	1	1
135	0	0
136	0	0
137	0	0
138	0	0
139	0	0
140	0	0
141	1	1
142	1	1
143	1	1
144	1	1
145	1	1
146	1	1
147	1	1
148	1	1
149	1	0
150	1	1
151	1	0
152	1	1
153	1	1
154	1	1
155	1	1
156	1	1
157	1	1
158	1	1
159	0	0
160	0	0
161	0	0
162	0	0
163	0	0
164	0	0
165	0	0
166	0	0
167	0	0
168	0	0
169	0	0
170	0	0
171	0	0
172	0	0
173	0	0
174	0	0
175	0	0
176	0	0

	Original License Plate State	Predicted License Plate State
Observation no.	(exists indicated by 1 and does not exist indicated by 0)	
177	0	0
178	0	0
179	0	0
180	1	1
181	1	1
182	1	1
183	1	1
184	1	1
185	1	1
186	1	1
187	1	1
188	1	1
189	1	0
190	1	1
191	1	1
192	1	1
193	1	0
194	1	1
195	1	1
196	1	1
197	0	0
198	1	1
199	0	0
200	0	0

Actual Characters and Model Predicted Characters for single-character types

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
1	14	14
2	14	3
3	14	4
4	14	6
5	14	14
6	14	2
7	14	5
8	14	14
9	14	14
10	14	8
11	14	9
12	14	14
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	2	2
27	2	2
28	2	2
29	2	2
30	2	2
31	2	2
32	2	2
33	2	2
34	2	2
35	2	2
36	2	2
37	2	2
38	2	2
39	2	2
40	2	2
41	2	2
42	3	3
43	3	3

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
44	3	3
45	3	3
46	3	3
47	3	3
48	3	2
49	3	3
50	3	2
51	3	3
52	3	3
53	3	3
54	4	4
55	4	4
56	4	4
57	4	4
58	4	4
59	4	4
60	4	4
61	4	4
62	4	4
63	4	4
64	4	4
65	5	5
66	5	5
67	5	5
68	5	5
69	5	5
70	5	5
71	5	5
72	5	5
73	5	5
74	5	5
75	5	5
76	5	5
77	5	5
78	5	5
79	5	5
80	6	6
81	6	6
82	6	6
83	6	6
84	6	6
85	6	6
86	6	6
87	6	6
88	6	6

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
89	6	6
90	6	6
91	6	5
92	6	6
93	6	6
94	6	6
95	6	6
96	7	7
97	7	7
98	7	7
99	7	7
100	7	7
101	7	7
102	7	7
103	7	7
104	7	7
105	7	7
106	7	7
107	7	7
108	7	7
109	7	7
110	7	7
111	8	8
112	8	8
113	8	8
114	8	8
115	8	8
116	8	8
117	8	7
118	8	8
119	8	8
120	8	8
121	8	8
122	8	8
123	8	8
124	8	8
125	8	8
126	8	8
127	8	8
128	8	8
129	8	8
130	8	8
131	8	8
132	8	8
133	8	8

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
134	8	8
135	9	9
136	9	9
137	9	9
138	9	9
139	9	9
140	9	2
141	9	9
142	9	9
143	9	9
144	9	9
145	9	9
146	9	9
147	9	2
148	9	2
149	9	9
150	9	9
151	9	9
152	9	9
153	9	9
154	9	9
155	9	9
156	10	10
157	10	10
158	10	10
159	10	10
160	10	10
161	10	10
162	10	10
163	10	10
164	10	11
165	10	10
166	10	10
167	10	10
168	10	11
169	10	10
170	10	10
171	10	10
172	10	10
173	10	10
174	10	10
175	10	10
176	10	10
177	10	10
178	11	11

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
179	11	11
180	11	11
181	11	11
182	11	11
183	11	11
184	11	11
185	11	11
186	11	11
187	11	11
188	11	11
189	11	11
190	11	11
191	12	12
192	12	12
193	12	12
194	12	12
195	12	12
196	12	12
197	12	12
198	12	12
199	12	12
200	12	12
201	12	11
202	12	12
203	12	12
204	12	11
205	13	13
206	13	13
207	13	13
208	13	13
209	13	3
210	13	13
211	13	13
212	13	13
213	13	13
214	13	3
215	13	13
216	13	13
217	13	13
218	13	13
219	13	13
220	13	13
221	13	13
222	13	13
223	13	13

	Original character value	Predicted character value
Observation no.	character values according to the values of $out1$ except 14 for the unknown characters which are out of research extent	
224	13	13
225	13	13

Actual Characters and Model Predicted Characters for multi-character types

	Original character value	Predicted character value
Observation no.	character values according to the values of $out2$ except 8 for the unknown characters which are out of research extent	
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	2	2
15	2	2
16	2	2
17	2	2
18	2	2
19	2	2
20	2	2
21	2	2
22	2	2
23	2	2
24	2	2
25	2	2
26	2	2
27	2	2
28	2	2
29	2	2
30	3	3
31	3	3
32	3	3
33	3	3
34	3	3
35	3	3
36	3	3
37	3	3
38	3	3
39	3	3
40	3	3
41	3	3
42	4	4
43	4	4

	Original character value	Predicted character value
Observation no.	character values according to the values of $out2$ except 8 for the unknown characters which are out of research extent	
44	4	4
45	4	4
46	4	4
47	4	4
48	4	4
49	4	4
50	4	4
51	4	4
52	4	4
53	5	5
54	5	5
55	5	5
56	5	5
57	5	5
58	5	5
59	5	5
60	5	5
61	5	5
62	5	5
63	5	5
64	5	5
65	5	5
66	5	5
67	5	5
68	6	6
69	6	5
70	6	6
71	6	6
72	6	6
73	6	6
74	6	6
75	6	6
76	6	5
77	6	6
78	6	6
79	6	6
80	6	6
81	6	6
82	6	6
83	6	6
84	7	7
85	7	7
86	7	7
87	7	7
88	7	7

	Original character value	Predicted character value
Observation no.	character values according to the values of $out2$ except 8 for the unknown characters which are out of research extent	
89	7	6
90	7	6
91	7	7
92	7	7
93	7	7
94	7	7
95	7	7
96	7	7
97	7	7
98	7	7
99	8	8
100	8	8
101	8	8
102	8	5
103	8	8
104	8	5
105	8	8
106	8	8
107	8	8
108	8	8
109	8	6
110	8	8
111	8	8
112	8	8
113	8	6
114	8	8
115	8	8
116	8	8
117	8	5
118	8	8
119	8	8
120	8	8
121	8	8
122	8	8
123	8	8
124	8	8
125	8	8
126	8	8