Real Time Detection and Recognition of Vehicle License Plate in Bangla

Md. Mahmudul Hasan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

August 2011

Real Time Detection and Recognition of Vehicle License Plate in Bangla

A Thesis Submitted to the

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology

in Partial Fulfillment of the Requirement
for the Degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

by
Md. Mahmudul Hasan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
August 2011

The thesis entitled **"Real time detection and recognition of vehicle license plate in Bangla"** submitted by Md. Mahmudul Hasan, Student No.: 0409052042, Session: April, 2009 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING on August 7, 2011.

BOARD OF EXAMINERS

1. ____________________________________

(Dr. Md. Humayun Kabir) — **Chairman** (Supervisor)
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka - 1000, Bangladesh.

2. ____________________________________

(Dr. Md. Monirul Islam) — **Member** (Ex-officio)
Professor and Head
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka - 1000, Bangladesh.

3. ____________________________________

(Dr. Md. Monirul Islam) — **Member**
Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka - 1000, Bangladesh.

4. ____________________________________

(Dr. Md. Haider Ali) — **Member** (External)
Professor
Department of Computer Science and Engineering
Dhaka University
Dhaka - 1000, Bangladesh.

Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the candidate

(Md. Mahmudul Hasan)

Dedication

To my beloved parents.

Contents

Declaration **ii**

Dedication **iii**

Acknowledgement **xiii**

Abstract **xiv**

1 Introduction **1**

1.1 Automatic License Plate Recognition 1
1.2 Applications of ALPR 3
1.3 Bangladeshi License Plate 4
1.4 Real Time System 7
1.5 Proposed Scheme at a Glance 8
1.6 Organization of This Thesis 9

2 Supporting Algorithms and Techniques **10**

2.1 Introduction 10
2.2 Edge Detection 10
2.2.1 Horizontal and Vertical Edge 11
2.2.2 Canny Edge Detection Algorithm 12
2.3 Binarization 13
2.3.1 Sauvola Image Binarization Algorithm 14
2.3.2 Otsu Image Binarization Algorithm 14
2.4 Symmetry Detection 15
2.5 Horizontal and Vertical Projection 17
2.6 Moving Average Filter 18
2.7 Extrema Detection 20

2.8 Artificial Neural Networks 20
2.8.1 Activation Functions 23
2.9 Summary 24

3 Literature Review 25
3.1 Introduction 25
3.2 License Plate Detection Algorithms 25
3.2.1 Detection Algorithms Using Binary Image Processing . . . 25
3.2.2 Detection Algorithms Using Grey Scale Processing 27
3.2.3 Other License Plate Detection Algorithms 28
3.3 License Plate Segmentation Algorithms 30
3.3.1 Segmentation Algorithms Using Binary Image Processing . 30
3.3.2 Segmentation Algorithms Using Grey Scale Processing . . 31
3.3.3 Segmentation in Bangla OCR 32
3.4 License Plate Recognition Algorithms 33
3.4.1 Support Vector Machine 33
3.4.2 Neural Networks 33
3.4.3 Template Matching 33
3.5 Case Studies 34
3.5.1 An ALPR system for English LP 34
3.5.2 An ALPR system for Vietnamese LP 38
3.5.3 An ALPR system for Bangla LP 40
3.6 Summery 41

4 Proposed Scheme 43
4.1 Introduction 43
4.2 Detection of license plate region 43
4.3 Segmentation of characters 46
4.3.1 Separating the rows 48
4.3.2 Segmenting the characters and digits 50
4.4 Character Classification 53
4.4.1 Feature extraction 54
4.4.2 Network design 55
4.5 Summary 58

5 Experimental Studies **60**
5.1 Introduction 60
5.2 License Plate Detection 60
5.3 License Plate Segmentation 62
5.4 License Plate Recognition 63
5.5 English License Plate 66
5.6 Execution Time Comparison 70
5.7 Summary 70

6 Conclusion **71**
6.1 Summary 71
6.2 Suggestions for future work 71

References **73**

List of Figures

1.1 Stationary ALPR systems: (a) Orange county, USA, and (b) Use of CCTV camera in ALPR system. Mobile ALPR system: (c) Dubai, and (d) England. 2
1.2 Vehicle license plates from different countries: (a) USA, (b) Brazil, (c) England, and (d) India. 4
1.3 Some Bangla license plates with complex background. 5
1.4 Different types of Bangla license plate: (a) Private, (b) Public, (c) Government, and (d) Military. 6
1.5 Vehicle registration area in Bangladesh. 7
1.6 List of Bangla character that we recognize using neural network. . 7

2.1 (a) Original input images, (b) Vertical edge image, and (c) Horizontal edge image. 12
2.2 (a) Original input images, (b) Canny edge images (threshold = 0.5), and (c) Canny edge images (threshold = 0.3). 13
2.3 (a) Original input images, (b) Binarization using Otsu's algorithm, and (c) Binarization using Sauvola Algorithm. 16
2.4 (a) Strong reflection, (b) Uniform area, and (c) Background pattern are the cause of irregularities in vehicle symmetry. 17
2.5 Edge images are used to extract symmetry to deal with the problem of (a) Strong reflection, and (b) Uniform area. 17
2.6 (a) Original license plate image, (b) Horizontal projection, and (c) Vertical projection. 18
2.7 (a) Projections, and (b) Filtered projections. 19
2.8 Extrema detection. 21

2.9 Activation functions: (a)Hard limit, (b) Symmetric hard limit, (c) Pure linear, (d) Positive linear, (e) Saturating linear, (f) Symmetric saturating linear, (g) Log sigmoid, (h) Tan sigmoid, and (i) Radial basis. 23

3.1 Steps for license plate segmentation: (a) Original image, (b) Result of SCW segmentation technique after the application of the segmentation rule and thresholding, (c) Image after logical AND operation, and (d) License plate detection. 35
3.2 Flowchart indicating the license plate segmentation process [3]. . 36
3.3 Architecture of the PNN [3]. 37
3.4 The license plate detection stage [4]: (a) Candidate license plates detection using Hough transform and contours, and (b) Candidate license plate validation. 39
3.5 The license plate segmentation stage [4]: (a) Row segmentation using horizontal projection, and (b) Character segmentation using vertical projection. 39
3.6 (a)Initial image, (b) The result of applying empirical crop, (c) Image obtained after brute force scanning and cropping, (d) The lower half of the registration plate, and (e) Individual characters being extracted from the registration plate. 41

4.1 Sample outputs of license plate detection stage: (a) Input image containing vehicle, (b) Edge image with overlaying vertical axis of symmetry, and (c) Detected ROI. 48
4.2 Sample outputs of license plate segmentation stage: (a) Binary image of a simple license plate, (b) Horizontal projection of the pixels, and (c) Filtered projection. 50
4.3 Sample outputs of license plate segmentation stage: (a) Binary image of a complex license plate, (b) Horizontal projection of the pixels, and (c) Filtered projection. 50
4.4 Threshold initialization step: (a) Separated row containing digits, (b) Separated row containing characters, and (c) Calculating initial threshold. 52

4.5 Segmenting characters: (a) Binary image containing characters, (b) Vertical projection of the pixel, and (c) Filtered vertical projection. 54
4.6 Segmenting characters 55
4.7 Segmenting digits: (a) Binary image containing digits, (b) Vertical projection of the pixel, and (c) Filtered horizontal projection. . . 56
4.8 Some training samples for the character classifier neural network. 56
4.9 Some training samples for the digit classifier neural network. . . 57
4.10 Neural network for digit classification. 57
4.11 Neural network for character classification. 57

5.1 License plate detection: (a) Original input images, (b) Edge images, and (c) After the application of symmetry detection algorithm and ROID. 62
5.2 Character segmentation: (a) Original license plate, (b) Manual segmentation, (c) Segmentation with constant threshold [4], and (d) Segmentation with adaptive threshold (Algorithm SAT). . . . 64
5.3 License plate detection: (a) Original input images, (b) Edge images, and (c) After the application of Algorithm ROID. 67
5.4 License plate segmentation: (a) Original license plate, (b) Inverted binary images, (c) Vertical projections, and (d) Segmented characters and digits. 68
5.5 Execution time comparison among the algorithms. 70

List of Tables

5.1 Experiment set up for license plate detection stage. 61

5.2 Experimental results in license plate detection stage. 61

5.3 Experiment set up for license plate segmentation stage. 63

5.4 Experimental results in row segmentation stage. 64

5.5 Experimental results in character and digit segmentation stage. . 65

5.6 Experiment set up for digit and character recognition stage. . . . 65

5.7 Experimental results in digit and character classification stage using neural network. 66

5.8 Experimental results for the license plate written in English. . . . 69

5.9 Experimental results for the license plate written in English. . . . 69

List of Abbreviations

ROI	Region of Interest
LP	License Plate
ALPR	Automatic License Plate Recognition
PCP	Probable Crossing Point
IW	Inspection Window
OCR	Optical Character Recognition
ITS	Intelligent Transportation System
SMA	Simple Moving Average
CMA	Cumulative Moving Average
WMA	Weighted Moving Average
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
AR	Aspect Ratio
CCA	Connected Component Analysis
SCW	Sliding Concentric Windows
SVM	Support Vector Machine
HMM	Hidden Markov Model
SAT	Segmentation with Adaptive Threshold
LM	Levenberg-Marquardt back-propagation
GDA	Gradient Descent with Adaptive back-propagation
OSS	One Step Secant back-propagation
SCG	Scaled Conjugate Gradient back-propagation

List of Symbols

$D^h(i,j)$	Horizontal derivative
$D^v(i,j)$	Vertical derivative
$E(i,j)$	Edge image
$BI(i,j)$	Binary image
H	Height of an image
W	Width of an image
$T(x,y)$	The threshold at pixel (x,y) of Sauvola method
$J_{Otsu}(T)$	Otsu's objective function
$f_h(t)$	Horizontal projection
$f_{h1}(t)$	Filtered horizontal projection
$f_v(t)$	Vertical projection
$f_{v1}(t)$	Filtered vertical projection
$f_p(t)$	Horizontal projection of first row of LP
$f_{p1}(t)$	Filtered horizontal projection of first row of LP
$f_{p1}^1(t)$	Derivative of $f_{p1}(t)$
σ^2	Variance
m	Mean
net_j	Input to the neuron j
y_j	Output of neuron j
$g(.)$	Activation function
T_{int}	Initial threshold in SAT

Acknowledgement

I have passed an important hurdle of my life. The happiness from this will last forever in my mind. In this blissful moment, I feel obliged to show my gratitude to the persons accompanying me during this endeavour.

Above all, I would like to express my gratitude to the almighty Allah whose kindness has kept me alive and has given me a silent support.

I would like to express my sincere gratitude to my supervisor Dr. Md. Humayun Kabir for the continuous support of my MSc. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me all along this research as well as writing this thesis. He has always pointed me in the right direction when I was lost and supported me when I was on the right path. I could not have imagined having a better supervisor and mentor for my MSc. study.

Besides my supervisor, I would like to thank the rest of the exam committee: Prof. Dr. Md. Monirul Islam, Dr. Md Monirul Islam, Prof. Dr. Md. Haider Ali for their encouragement, insightful comments, and hard questions.

I am also thankful to Dr. Md. Ameer Ali for his generous assistance during this time.

Finally, I am forever indebted to my parents for their understanding, endless patience and encouragement when it was mostly required.

Abstract

Automatic license plate detection and recognition has numerous applications. A large number of schemes have already been proposed in order to make the detection and recognition process efficient. However, a very little work has been done on Bangla license plate recognition. Wide variation among the license plate patterns, complex background, and the difficulty in segmenting Bangla characters of Bangladeshi license plates make it inefficient to use the existing algorithms. In this thesis, we propose a solution for Bangla license plate detection and recognition. We use three stages of conventional license plate recognition system. However, we propose new algorithm in each stage, which are effective for Bangla license plate detection and recognition. We tested our algorithms for over 250 images taken from the road. We achieve over 95% success in Bangla license plate recognition.

Chapter 1

Introduction

1.1 Automatic License Plate Recognition

Automatic License Plate Recognition (ALPR) is a technology that uses optical character recognition (OCR) to automatically read license plate characters. It is one of the important module of Intelligent Transportation System (ITS). A fully functional ALPR system is comprised of three major stages such as,

1. Detection of license plate in the input image containing vehicles,
2. Segmentation of characters and digits in the license plate, and
3. Recognition and classification of these characters and digits.

There are two types of ALPR systems, such as stationary and mobile. In stationary ALPR systems, stationary cameras can be mounted on road signs, street lights, highway overpasses or buildings in a cost-effective way to monitor moving and parked vehicles. Camera software is made capable of identifying the pixel patterns that form the license plates and translating the letters and numbers on the plate into a digital format. The plate data is then compared in real-time to a list of plate numbers that belong to a set of vehicles of interest. If the system detects a match, it sends an alert to the dispatcher or other designated personnel. On the other hand, in the mobile ALPR systems, multiple cameras are mounted on the vehicle. As the vehicle moves, it takes the snaps of the license plates from other vehicle around it and transmits plate data to a database. In this thesis, we are only interested in stationary ALPR systems. Fig. 1.1 shows some existing stationary and mobile ALPR systems. A stationary ALPR system in Orange county, USA is shown in Fig. 1.1(a), where three cameras are mounted on three lanes of a road. CCTV cameras beside the road are seldom used in stationary ALPR system as shown in Fig. 1.1(b). Two mobile ALPR systems functioning in Dubai and London are shown in Fig. 1.1(c) and 1.1(d) respectively. Here, cameras are mounted on top of the cars.

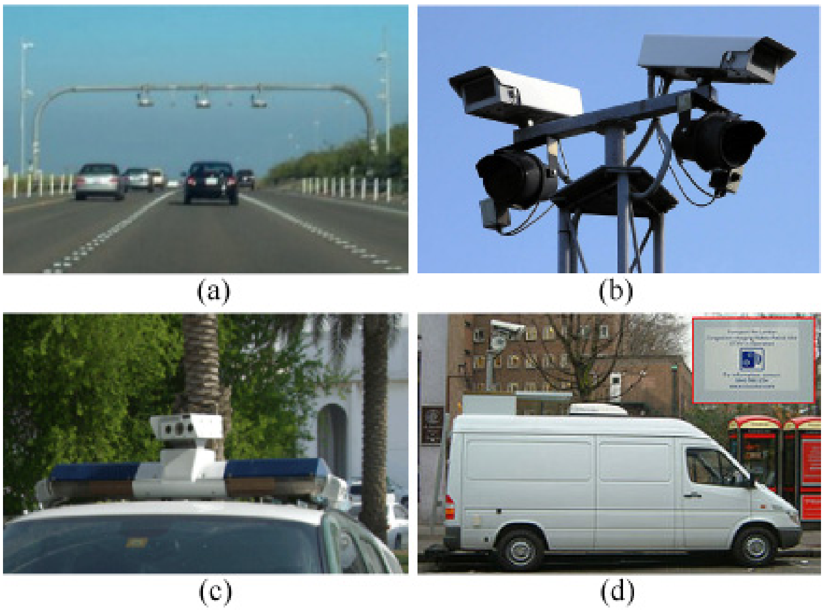


Fig. 1.1: Stationary ALPR systems: (a) Orange county, USA, and (b) Use of CCTV camera in ALPR system. Mobile ALPR system: (c) Dubai, and (d) England.

There are some difficulties that hinder the successful recognition of a license plate in the ALPR system [1].

1. Poor image resolution: Poor image resolution may result in if the target license plate is very far away from the camera. The use of a low-resolution camera may also cause this problem.

2. Blurry images: Snapped image might be blurred because of high vehicle speed. Camera with high shutter speed can be used to avoid this problem.

3. Poor lighting and low contrast: Sometime it might be very difficult to distinguish the license plate from the background due to overexposure, reflection or shadows. Image contrast can be adjusted by grey-scale histogram analysis to overcome this problem.

4. An object obscuring the plate such as, a tow bar, or dirt on the plate.
5. Some time different fonts are used in license plate. However, some countries do not allow such variations in the fonts of the plates, which eliminate the problem.
6. Different circumvention techniques in front of the vehicle near the license plate is a major difficulty in locating license plates.
7. The patterns of the license plates might be different from one country to another country.

1.2 Applications of ALPR

Research works [1,2] have listed many applications of ALPR, such as electronic payment systems, free-way and aerial management for traffic surveillance, recovering stolen cars, identifying cars with an open warrant for arrest, catching speeders by comparing the average time it takes to get from one stationary camera to another stationary camera, determining what cars do or do not belong to a parking garage, expediting parking by eliminating the need for human confirmation of parking passes etc.

As an essential part of Intelligent Transportation System (ITS), an ALPR system is capable of providing many benefits. It can be used to bring flexibility and automation in toll collection systems for highways, flyovers, and bridges. With the presence of an ALPR system tolls can be collected without manual intervention, that reduce time and traffic congestion. Traffic pattern analysis during peak and off peak hours can be done effectively with an ALPR system, which is an important component of urban planning. ALPR system can be used by metropolitan police department for effective law enforcement, effective enforcement of traffic rules, and enhanced vehicle theft prevention. It can also bring highest efficiency for border control systems. Other possible applications include building a comprehensive database of traffic movement, automation and simplicity of airport and harbour logistics, security monitoring of roads, checkpoints, etc.

Another important application of ALPR system is vehicle surveillance. It can be used to prevent non-payment at gas stations, drive-in restaurants, etc. After

Fig. 1.2: Vehicle license plates from different countries: (a) USA, (b) Brazil, (c) England, and (d) India.

integrating License Plate Recognition Software Engine into parking management systems, controlled and automatic vehicle entry and exit in car parks or secure zones becomes possible. Furthermore, the ability to recognise registration number is a significant added value for comprehensive parking solutions or inventory management. A parking lot equipped with ALPR system can provide many advantages, 1) Flexible and automatic vehicle entry to and exit from a car park, 2) Management information about car park usage. 3) Improved security for both car park operators and car park users, and 4) Improved traffic flow during peak periods.

Other possible applications include: 1) Vehicle recognition through date and time stamping as well as exact location, 2) Inventory management, and 3) Comprehensive database of traffic movement. State border control is one of the most important applications of automatic license plate recognition.

1.3 Bangladeshi License Plate

License plates from different countries in Fig. 1.2 clearly show that, they are very much different from one country to another country. For this reason, it is not effective to use an ALPR system developed in one country to another country without specific modifications. According to our study, no successful ALPR system has been developed so far for Bangladeshi license plates.

There are some unique characteristics of Bangladeshi license plates which are

Fig. 1.3: Some Bangla license plates with complex background.

written in Bangla language. Most of the other languages although have disjoint characters in a word, the characters in a Bangla word are often joined at the top by a horizontal line called Matra. Moreover, some Bangla characters are comprised of two or more separate regions. There might be some disjoint parts in a character either at the top or at the bottom. These special characteristics of Bangla characters make it very difficult to segment. Unlike the other countries, a lot of variations can also be seen among the license plate patterns in Bangladesh. This is shown in Fig. 1.3. In Bangladesh, government provides only the registration number to a vehicle and the vehicle owner individually makes the license plates without following a standard template. As a result, the pattern of license plate varies from one owner to another owner, which makes it increasingly difficult to locate license plate regions in an input image.

In addition, a large number of Bangladeshi license plates have two rows. The first row contains the registration area and its type and the second row contains the registration number. There might be an extra row at the top or at the bottom of the license plate containing some extra information. This extra information also makes it more challenging to extract the registration information from the license plates.

In Bangladesh, license plates can be divided into five categories, such as private, public, government, military, and foreign missions. Both private and public license plate have same format, which includes three major information, such as registration area, type, and number. Registration area is mainly a set of Bangla

Fig. 1.4: Different types of Bangla license plate: (a) Private, (b) Public, (c) Government, and (d) Military.

characters indicating the name of a district or city, where the vehicle is registered. Registration type is a single Bangla character expressing type of vehicles for example, small car, motor cycle, van, etc. Registration number is comprised of six Bangla digit, where first two are separated by a hyphen. The only difference between private and public license plate is that the foreground and the background of private license plates are white and black respectively, whereas in public license plate it is opposite. These license plate information are mainly written in two rows, but private or public license plate with only one row containing all the information can also be found. Government, military, and foreign mission license plates have only one row. The background of government owned vehicles are yellow and the foreground is black. It contains a Bangla character indicating registration type followed by a five digit registration number. License plates of military vehicles have a special arrow sign or the abbreviation of the name of the military force at the beginning, which is followed by a four digit registration number. License plate format of foreign missions are same as the government license plate format. Four different types of Bangla license plate are shown in Fig. 1.4.

Bangladesh Road Transport Authority (BRTA) is the only organization that issue vehicle registration number. Currently, there are twenty nine circle offices of BRTA are functioning around the country. These circle offices are listed in Fig. 1.5. Fig. 1.6 shows the list of Bangla characters that we recognize using neural network to find out the registration area.

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

Fig. 1.5: Vehicle registration area in Bangladesh.

ক	খ	গ	চ	জ
ঝ	ট	ড	ঢ	ম
প	ফ	ব	ল	ন
স	য	হ	ই	ঙ
ভ	ও	ঞ্জ	◌ে	◌া

Fig. 1.6: List of Bangla character that we recognize using neural network.

1.4 Real Time System

A real time system is one that must process information and produce a response within a specified time, else risk severe consequences, including failure [8]. In a real time digital signal or image processing, average processing time per sample must be less than the sampling period, which is the reciprocal of sampling rate. A system with a real time constraint can not be considered successful if it produce the correct action or the correct answer after a certain deadline. The correctness of a real time system is based on the correctness of the output and the timeliness. There are two types of real time system, such as hard and soft real time systems. In hard real time system deadline must be met with correct response or the system will lead to a total failure. In soft real time system, the usefulness of the responses degrade after deadline, which lead to the degradation of system's quality of service. Some examples of real time system are car engine control system, heart pacemaker, air traffic control system, industrial process controller, embedded system, etc.

1.5 Proposed Scheme at a Glance

It is obvious that a robust and effective ALPR system is required for Bangladeshi license plates. According to our study, no effective ALPR system has been developed yet for Bangladeshi license plates. In this thesis, we present an ALPR system for Bangladeshi license plates. Our system has three stages like all other ALPR systems. In the first stage, we locate the position of the vehicle in the input image by using the property of symmetry. We locate the license plate in this vehicle region by using our newly developed Region of Interest Detection (ROID) algorithm. In the second stage, we extract the registration information from the license plate. After separating the rows, we segment the characters and digits by using our newly developed Segmentation with Adaptive Threshold (SAT) algorithm. Finally, we recognize the individual characters and digits by using Artificial Neural Network (ANN). We use separate neural networks for the characters and digits. Horizontal and the vertical projections of the binary image of size 32×32 of characters and digits are used as the feature set to recognize them. We tested our algorithms using 250 images of vehicle containing the license plate written in Bangla. Experimental results prove the superiority of our proposed scheme over three existing algorithms on the detection and recognition of Bangladeshi license plates. We also test our algorithms for the license plates written in English. Experimental results prove that our algorithms are equally effective to detect, segment, and recognize license plate written in English.

1.6 Organization of This Thesis

The rest of the thesis is organized as follows:

In Chapter 2, we briefly describe several supporting technologies and algorithms that we use in our proposed scheme for the ALPR system in Bangladesh.

We present a survey of the existing algorithms and techniques that have been used in three stages of ALPR system in Chapter 3. We also describe three existing ALPR systems along with their shortcoming in recognizing the Bangla license plates.

We propose a set of algorithms to develop a comprehensive ALPR system for the Bangla license plates in Chapter 4.

Chapter 5 presents the experimental results of our proposed scheme. We also compare our results with several existing schemes.

Finally, some concluding remarks and suggestions for future works are provided in Chapter 7.

Chapter 2

Supporting Algorithms and Techniques

2.1 Introduction

We use a number of existing techniques and algorithms, such as edge detection, image binarization, symmetry detection, projections, moving average filters, extrema detection, and artificial neural networks in our ALPR system for Bangladeshi vehicles. In this chapter, we provide a brief description of these techniques and algorithms along with their mathematical explanations. We also present brief comparisons among the variants of these techniques and provide the reasons behind using the techniques in our ALPR system.

2.2 Edge Detection

Edge detection is a widely used method in computer vision and image processing. One of the important goals of edge detection is feature detection and extraction. It is used to identify sharp changes of brightness in a digital image. The fundamental goal of detecting sharp changes of brightness is to capture important objects, events, and changes in an image. Generally, the application of edge detector to an image may produce a set of connected curves indicating the boundaries of the objects and the discontinuities of the surface orientations. Therefore, application of edge detector will reduce the amount of data to be processed by a great deal, while preserving important structural properties of the objects in the image. Many algorithms exist in the literature to find the edges in an image. In our proposed scheme, we use three edge detection algorithm such as, i) Horizontal edge, ii) Vertical edge, and iii) Canny edge detection algorithm. These algorithms with their merits and demerits are illustrated in the following sections.

2.2.1 Horizontal and Vertical Edge

For a given grey-scale image $\{G(i,j); 1 \leq i \leq H, 1 \leq j \leq W\}$, its corresponding horizontal and vertical edge image can be computed by the following two steps [35]. The horizontal and the vertical derivatives of the image $G(i,j)$ are calculated by Equation 2.1 and 2.2 respectively. These derivative images are thresholded by Equation 2.3 to obtain the horizontal edge and the vertical edge images. Threshold T in Equation 2.3 is a algorithm parameter, which is chosen according the the nature and illumination of the input image.

$$D^h(i,j) = \tfrac{1}{3}\sum_{x=i-1}^{i+1} |G(x,j-1) - G(x,j+1)| \tag{2.1}$$

$$D^v(i,j) = \tfrac{1}{3}\sum_{y=j-1}^{j+1} |G(i-1,y) - G(i+1,y)| \tag{2.2}$$

$$E(i,j) = \begin{cases} 1, & \text{if } D(i,j) \geq T \\ 0, & \text{otherwise} \end{cases} \quad 1 \leq i \leq H, 1 \leq j \leq W \tag{2.3}$$

Horizontal and vertical edge images are very effective to detect rectilinear objects such as, license plate in an image. Yu et al. [35] presented an algorithm to locate license plate in an image that use vertical edge information. It is based on the fact that license plates are mostly comprised of a pair of horizontal edges and a pair of vertical edges. If any of the pairs can be detected correctly, the four corners of the license plate can be easily located. Edge image produced by other algorithms is very complex because it includes all kinds of edges. Horizontal and vertical edge images are comparatively simple that maintain important structural properties of an object. For illustration, the original grey-scale images and corresponding horizontal and vertical edge images are shown in Fig. 2.1(a), 2.1(b), and 2.1(c) respectively. These images are computed by Equations 2.1, 2.2, and 2.3 respectively.

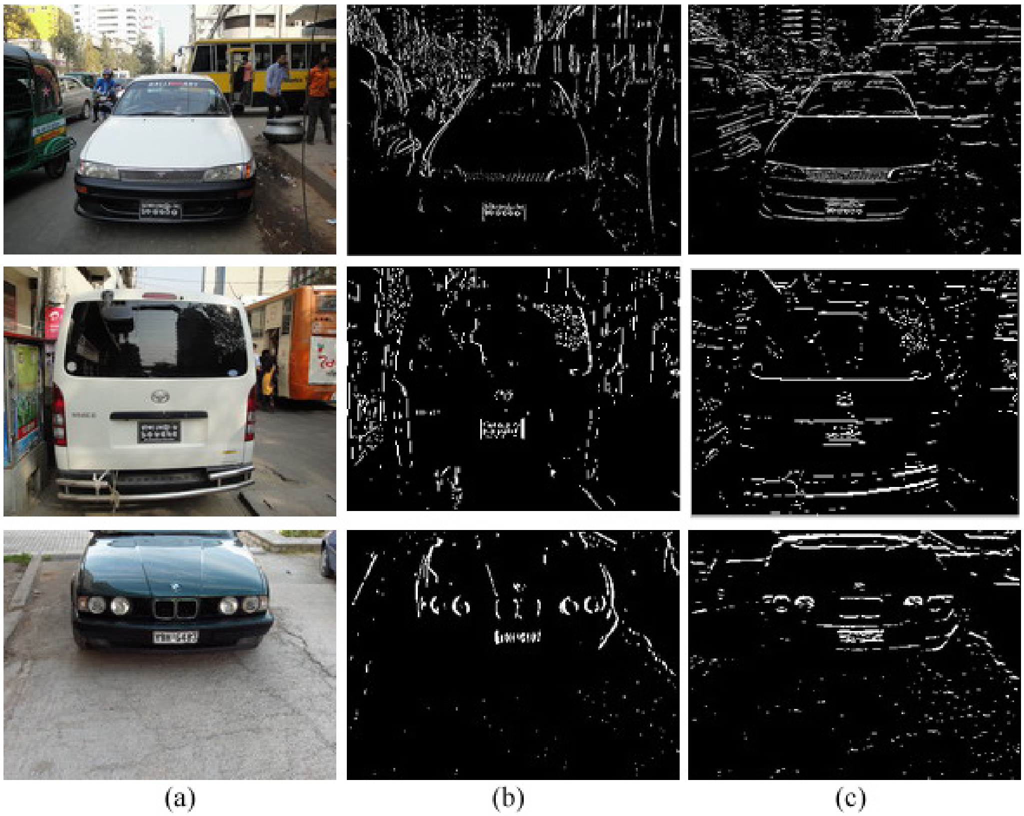
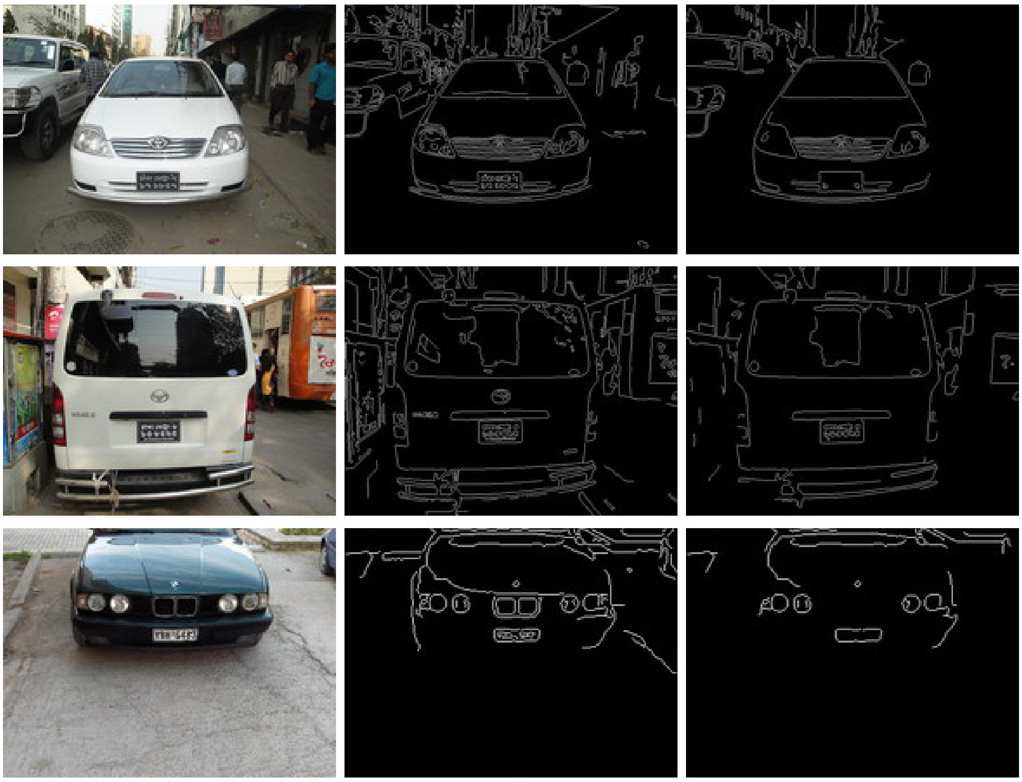


Fig. 2.1: (a) Original input images, (b) Vertical edge image, and (c) Horizontal edge image.

2.2.2 Canny Edge Detection Algorithm

Canny edge detection algorithm [31] was developed by John F. Canny in 1986. This is a multi-stage algorithm and is capable of detecting various kind of edges in an image. Canny developed this algorithm based on three important criteria. The first criteria is low error rate. It means an edge should not be missed and no false edge should be detected. Second criteria is localization. It means the distance between an actual edge and the detected edge should be minimum. Third criteria is, there should be only one response to a single edge. Canny edge detection algorithm proved its efficiency in many areas of image processing and computer vision. It is known to be the optimal edge detector. We use Canny edge detection algorithm in our proposed solution to locate the license plate in the image. License plate boundaries in an image shows sharp discontinuities of brightness. Canny edge detection algorithm can suppress many unwanted edges in the image of vehicles, while preserving the sharp edges in the license plate



(a) (b) (c)

Fig. 2.2: (a) Original input images, (b) Canny edge images (threshold = 0.5), and (c) Canny edge images (threshold = 0.3).

region. Original grey-scale images, and their corresponding Canny edge images with $T1 = 0.3$ and $T1 = 0.5$ are shown in Fig. 2.2(a), 2.2(b), and 2.2(c) respectively.

2.3 Binarization

Image binarization is the process of converting an image of up to 256 gray levels to a black and white image. Binarization is widely used as a pre-processing before Optical Character Reader (OCR). The simplest way of converting a grey scale image into a binary image is to choose a global threshold value, and classify all pixels with values above this threshold as white, and all other pixels as black. The problem of using global threshold is that the illumination of all images are not same. Some images are brighter and some others are darker. Therefore, a global threshold is not able to provide good results in all the cases. Finding one threshold compatible to the entire image set is very difficult and in many cases it is even impossible. Therefore, adaptive image binarization is needed where an optimal threshold is chosen for each image area.

2.3.1 Sauvola Image Binarization Algorithm

Sauvola's adaptive binarization technique [36] is characterized by the calculation of threshold for each pixel. The value of the threshold depends upon some local statistics like range, variance, and means or their logical combinations. It is typical of locally adaptive methods to have several adjustable parameters. The threshold $T(x, y)$ will be indicated as a function of the coordinates (x, y). This method adapts the threshold according to the local mean and standard deviation over a window size of $b \times b$. The threshold at pixel (x, y) is calculated as

$$T(x,y) = m(x,y) + \left[1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right] \tag{2.4}$$

where $m(i, j)$ and $\sigma(i, j)$ are the local sample mean and standard deviation respectively. Sauvola suggests the values of $k = 0.5, R = 128$, and $b = 10$. Thus, the contribution of the standard deviation makes the binarization adaptive. For example, in the case of badly illuminated areas, the threshold is lowered. Hence, image is binarized using

$$BI(x,y) \quad = \quad \begin{cases} 1, & \text{if } I(x,y) \geq T(x,y) \\ 0, & \text{if } I_(x,y) < T(x,y) \end{cases} \tag{2.5}$$

2.3.2 Otsu Image Binarization Algorithm

Otsu's binarization method [37] selects an optimal threshold Th^* for a given image by maximizing a discriminant criterion, i.e., the separability of the resultant classes in gray levels. It assumes that the grey level of the given image ranges in $\{0, 1, 2, \ldots, L-1\}$, where L is the total number of gray levels of the image. Otsu's method searches for an optimal threshold value Th^* in the range such that the objective function $J_{Otsu}(T)$, defined below, achieves its maximum:

$$J_{Otsu}(T) = \frac{P_1(T)P_2(T)\left[m_1(T) - m_2(T)\right]}{\sigma^2} \tag{2.6}$$

And thus

$$Th^* = \underset{0 \le T \le L-1}{\arg\max} J_{Otsu}(T) \tag{2.7}$$

Where $\sigma^2 = \sum_{l=0}^{L-1} [l - m(T)]^2 h(l)$ is the variance of the gray levels of the image; the $m(T) = \sum_{l=0}^{L-1} lh(l) / \sum_{l=0}^{L-1} h(l)$ is the mean of the total gray levels; $\{h(l), l = 0, 1, \ldots, L-1\}$ is the gray level histogram or distribution of a given image and $h(l)$ denotes the number of the pixels in the image with gray level of l; $P_1(T) = \sum_{l=0}^{T} h(l) / \sum_{l=0}^{L-1} h(l)$ denotes the prior probability of the foreground (the object class); $P_2(T) = \sum_{l=T+1}^{L-1} h(l) / \sum_{l=0}^{L-1} h(l)$ is the prior probability of the background (the non-object class); $m_1(T) = \sum_{l=0}^{T} lh(l) / \sum_{l=0}^{T} h(l)$ is the mean of the foreground and $m_2(T) = \sum_{l=T+1}^{L-1} lh(l) / \sum_{l=T+1}^{L-1} h(l)$ is the mean of the background. The foreground and the background contain pixels with the gray levels in $\{0, 1, \ldots, T\}$ and $\{T+1, \ldots, L-1\}$ respectively.

Some original grey-scale images, corresponding binary images using Otsu's and Sauvola's algorithms are shown in Fig. 2.3(a), 2.3(b), and 2.3(c) respectively. Though Sauvola's local adaptive binarization algorithm is widely used for document image binarization, its performance is not as good as the Otsu's algorithm in binarizing license plate with black background. Since the brightness of the license plates are different, using a constant threshold to binarize all the license plate image will lead to poor binaraization. We use Otsu's algorithm to binarize license plate in our scheme because it computes different binarization thresholds for each image based on the brightness of the image using histogram analysis.

2.4 Symmetry Detection

Symmetry is an important characteristic of man-made objects. An object is said to be symmetrical if there exist a vertical axis, which bisects the object equally to the left and to the right. Likewise, from the rear or from the front, most of

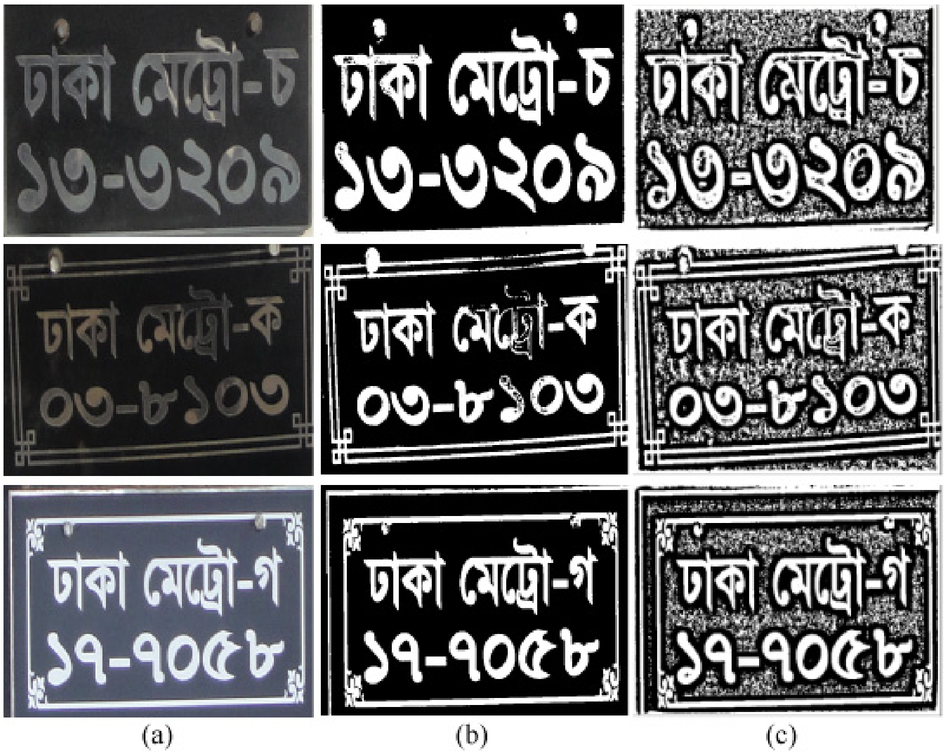
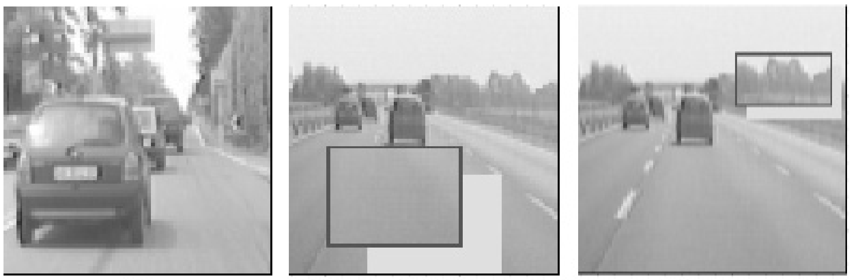


Fig. 2.3: (a) Original input images, (b) Binarization using Otsu's algorithm, and (c) Binarization using Sauvola Algorithm.

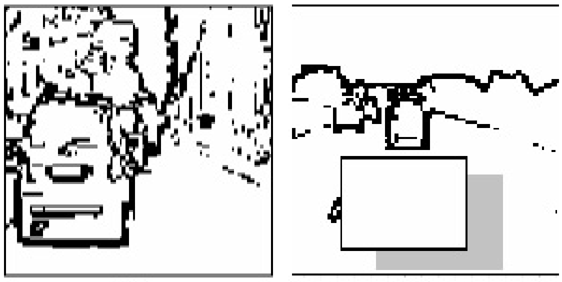
the vehicles are symmetrical along a vertical axis. We use this property to locate the vehicle in the image, which is an important part of our license plate detection process.

Gray level processing is not effective to find a symmetric pattern [32]. Because uniform areas and strong reflections could posses high level of symmetries. Fig. 2.4 shows that strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns present highly correlated symmetries. One solution to this problem is to compute symmetries in a binary image. At first, edges are extracted from the image using horizontal, vertical, or canny edge algorithms to get rid of uniform areas and strong reflections. Symmetries are computed in these edge images. Fig. 2.5 shows that although a strong reflection is present on the left side of the vehicle, edges are anyway visible and can be used to extract symmetries; moreover, in uniform areas no edges are extracted and therefore no symmetries can be detected.



(a) (b) (c)

Fig. 2.4: (a) Strong reflection, (b) Uniform area, and (c) Background pattern are the cause of irregularities in vehicle symmetry.



(a) (b)

Fig. 2.5: Edge images are used to extract symmetry to deal with the problem of (a) Strong reflection, and (b) Uniform area.

2.5 Horizontal and Vertical Projection

For a given binary image $\{BI(i,j); 1 \leq i \leq H, 1 \leq j \leq W\}$, its horizontal and vertical projections are computed by using Equations 2.8 and 2.9 respectively. An original binary image, and corresponding horizontal, and vertical projections are shown in Fig. 2.6(a), 2.6(b), and 2.6(c) respectively.

$$f_h(i) = \sum_{j=1}^{W} BI(i,j) \tag{2.8}$$

$$f_v(i) = \sum_{i=1}^{H} BI(i,j) \tag{2.9}$$

Horizontal and vertical projections are used in license plate segmentation stage in our proposed ALPR system. Horizontal projection of a license plate is evaluated and thresholded to separate the rows containing registration information, such as registration number, area, and type. Vertical projection of each row of the

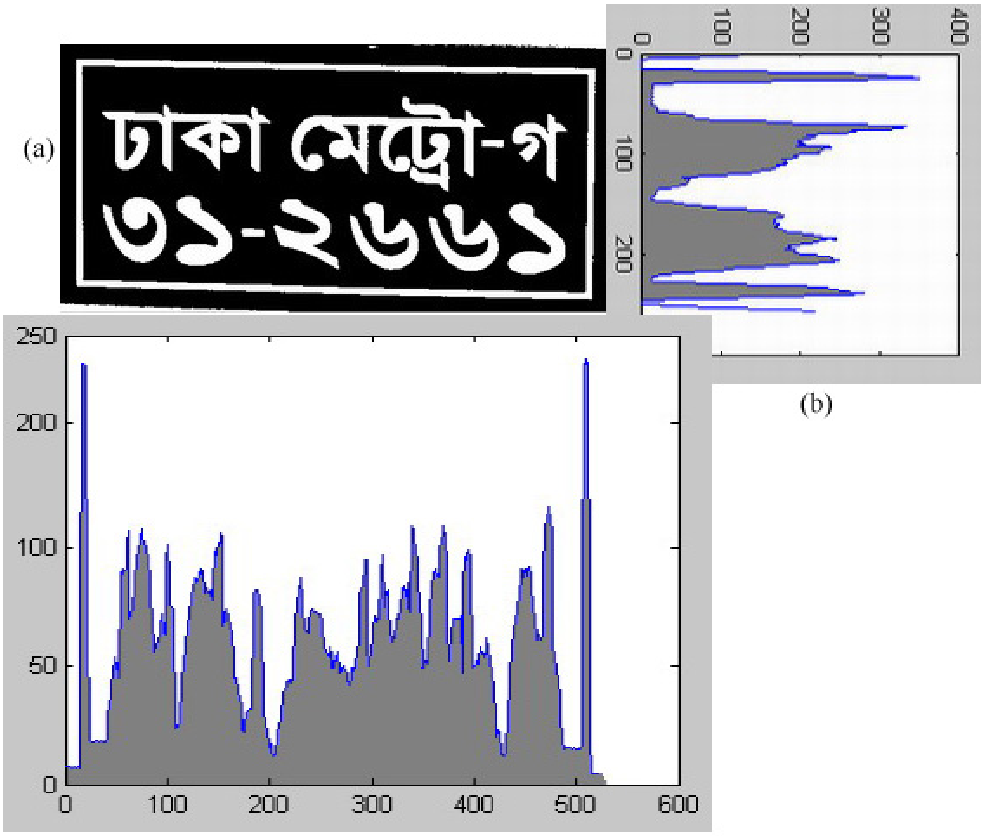


Fig. 2.6: (a) Original license plate image, (b) Horizontal projection, and (c) Vertical projection.

license plate is evaluated and thresholded to segment the characters and digits of the row. This threshold is adaptive because it changes based on the input image and the segmentation performance.

2.6 Moving Average Filter

A moving average filter is also known as running average filter. It is used to smooth a series of data by reducing the effect of random variations in the data. Its fundamental goal is to remove high frequency components and short-term fluctuations from a data set. In digital signal processing, moving average filter is normally used as the low pass filter. For a given data set and a fixed window size W, moving averages can be obtained by taking the average of the numbers belong to the first window. Then the window is shifted forward along the data set and a new average is calculated. This process is repeated over the entire data

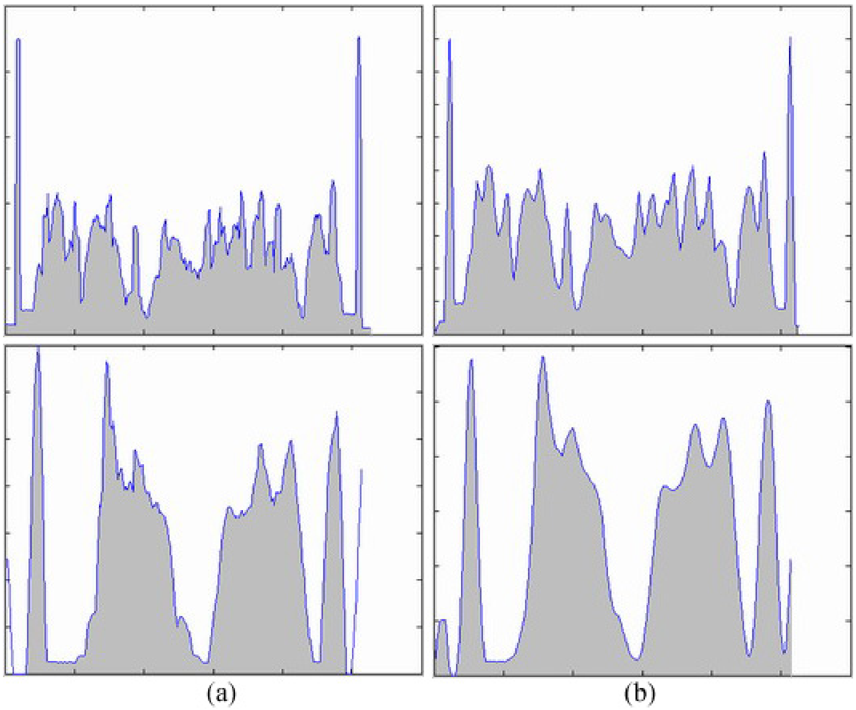


Fig. 2.7: (a) Projections, and (b) Filtered projections.

set. The line connecting all the averages is the moving average. The value of the parameter W is depend on the application of the filter. A higher value of W will result in long-term trend of data set. Short term fluctuations will be reduced by a great deal.

There are several types of moving average filters, such as simple moving average, cumulative moving average, and weighted moving average. In our proposed scheme we use simple moving average filter which is described below.

- Simple Moving Average

  A Simple Moving Average (SMA) of window size W is the unweighted mean of the previous W data points. Given a date set, $\{X : x_1, x_2, x_3, \cdots, x_{n-1}, x_n\}$, a simple moving average (SMA) is calculated by using the following formula

$$SMA(i) = \sum_{k=1}^{k=W} x_{i-k} \tag{2.10}$$

  where $i > W$. Sample projections and filtered projections using SMA are shown in Fig. 2.7(a) and 2.7(b) respectively.

In the license plate segmentation stage, the horizontal and the vertical projections are evaluated to separate the rows and to segment the characters and digits in a row respectively. Local maxima of the horizontal and the vertical projections represent the rows and characters or digits respectively. However, these projection are not smooth and contain many unwanted local maxima and minima. Simple Moving Average (SMA) filter is used to remove these unwanted local maxima and minima. SMA is used because of its simpleness and each member value in the projections has the same weight.

2.7 Extrema Detection

Given a function $f(x)$, extrema is the collective name of maxima and minima. There are two types of extrema, such as local extrema and global extrema. Local extrema are the largest or the smallest magnitude of $f(x)$ within a given neighbourhood. On the other hand, global extrema is the largest or the smallest magnitude of $f(x)$ within the entire domain of the function.

A real-valued function f defined on a real line is said to have a local (or relative) maximum point at the point $x*$, if there exists some $\varepsilon > 0$ such that $f(x*) \geq f(x)$ when $|x - x*| < \varepsilon$. The value of the function at this point is called maximum of the function. Similarly, a function has a local minimum point at $x*$, if $f(x*) \leq f(x)$ when $|x - x*| < \varepsilon$. The value of the function at this point is called minimum of the function. A function has a global (or absolute) maximum point at $x*$ if $f(x*) \geq f(x)$ for all x. Similarly, a function has a global (or absolute) minimum point at $x*$ if $f(x*) \leq f(x)$ for all x. Fig. 2.8 show the maxima and the minima of two functions.

Each arch of the horizontal projection and the vertical projection represents a row of the license plate and a characters in a row respectively. These arches are enumerated by counting the local maxima of the projections. Extrema detection technique is used to find the local maxima of the projections.

2.8 Artificial Neural Networks

An Artificial Neural Network (ANN) is usually known as Neural Network (NN). It is a mathematical model that is inspired by the structural and functional aspects

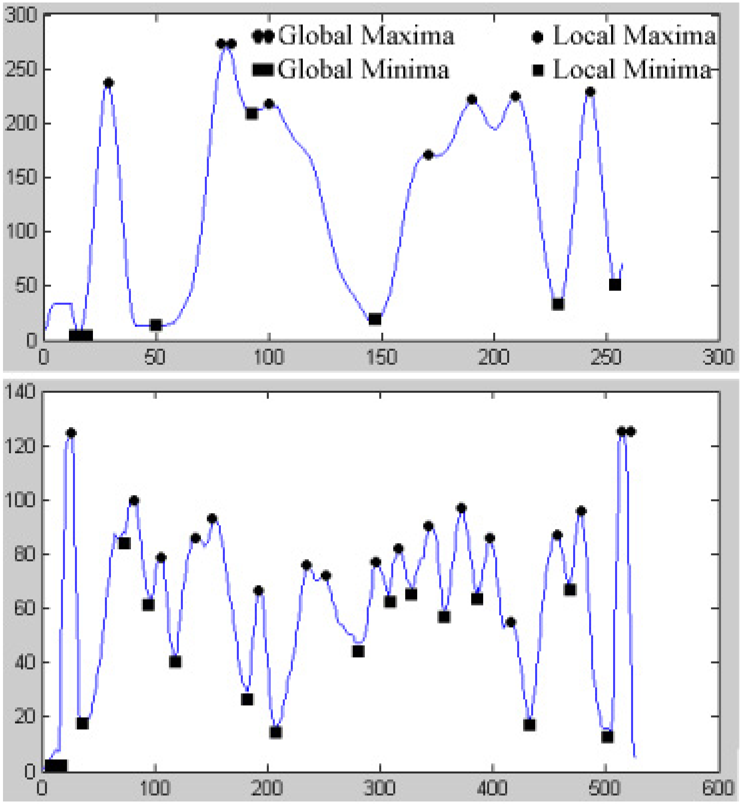


Fig. 2.8: Extrema detection.

of biological neural networks in our brain. A neural network consists of many interconnected processing units known as neurons. Neurons are connected in a structured way to accomplish a specific task. Neurons are arranged in layers. Each of the neurons in a layer have similar functionality. Generally, an ANN is an adaptive system that changes it structure and inputs and outputs relationship through a learning process. A fundamental goal of an ANN is to model complex relationship between inputs and outputs or to find patterns in data.

An Artificial Neural Network (ANN) has numerous applications, such as

- Regression analysis and modelling.
- Classification, including pattern and sequence recognition.
- Data processing, including filtering, clustering, etc.
- Robotics, including direction manipulators.

Based on the interconnection patterns between the layers of neurons, ANN can be divided into two categories, such as feed-forward network (acyclic) and recurrent or feedback network (cyclic). We use multilayer feed-forward neural network in order to recognize characters and digits because of its robustness and ease of use [49].

Feed-forward neural network is the simplest neural network, where connections between the units do not form any cycle. This network is called feed-forward because signals constantly move from one layer to the next layer(s). The properties of feed-forward networks are summarized as follows.

- The output only depend on the present input because there are no feedback connections.
- Neurons are arranged in layers. First layer receives input and the last layer produces outputs.
- There may be one or more layers between the input and output layers. These layers are called hidden layer because they have no interconnection with the external world.
- There is no connection among neurons in the same layer.

There are three types of feed-forward network: single layer feed-forward network, multi layer feed-forward network, and radial basis network. Each of the network has some merits and demerits.

Learning is an interactive and iterative process to find the optimal weights of the interactions between the neurons. The performance goal of the learning process is to minimize a cost function. A commonly used cost function is mean-squared error, which tries to minimize the average squared error between the network output and the target value. There are three major learning paradigms: supervised learning, unsupervised learning, and reinforcement learning. We use supervised learning scheme to train our neural networks because the target output for these training input images of characters and digits are known prior to the training.

In supervised learning, we are given a set of example pairs (X, Y), where $x_i \in X$ is the input and $y_i \in Y$ is the target output. Our goal is to find a function f so that $f(x_i) = y_i$.

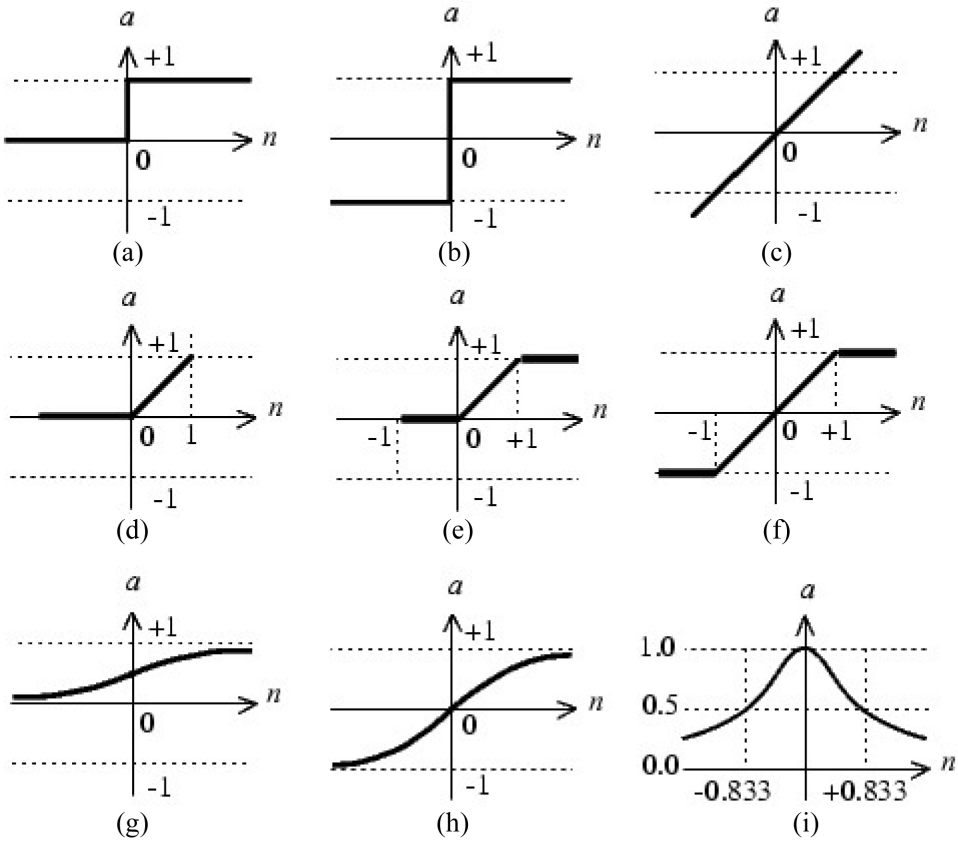


Fig. 2.9: Activation functions: (a)Hard limit, (b) Symmetric hard limit, (c) Pure linear, (d) Positive linear, (e) Saturating linear, (f) Symmetric saturating linear, (g) Log sigmoid, (h) Tan sigmoid, and (i) Radial basis.

2.8.1 Activation Functions

Behaviour of a processing unit or neuron is highly dependant on the activation functions. Activation functions can be divided into the following categories:

- **Linear:** The output of this kind of activation function is directly proportional to the total weighted input such as, pure linear, positive linear, saturating linear, and symmetric saturating linear activation functions. These functions are shown in Fig. 2.9(c), 2.9(d), 2.9(e), and 2.9(f) respectively.

- **Threshold:** The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value. Hard limit and symmetric hard limit are this kind of activation function. These functions are shown in Fig. 2.9(a), and 2.9(b) respectively.

- **Sigmoid:** The output varies continuously but not linearly as the input changes. Examples of this kind of activation function are Log sigmoid, tan sigmoid, and radial basis activation function. These functions are shown in Fig. 2.9(g), 2.9(h), and 2.9(i) respectively.

We use Log-sigmoid and pure linear activation functions in the hidden layers and the output layer respectively.

2.9 Summary

In this chapter, we have described several techniques and algorithms such as edge detection, binarization, symmetry detection, projections, moving average filters, extrema detection, and artificial neural network. We use horizontal, vertical, and Canny edge detection algorithm to detect edges of the input image. Symmetry detection algorithm is used to locate the vehicle in the edge image. Horizontal and vertical projections are used along with the simple moving average filters and extrema detection algorithm to segment the license plate. Finally, multilayer feed-forward network is used to recognize the characters and digits. Log-sigmoid and pure linear functions are used as the activation functions of the ANN.

Chapter 3

Literature Review

3.1 Introduction

Generally, license plate recognition algorithms of an ALPR system are comprised of three stages such as, detection, segmentation, and recognition of license plates. Due to the immense applications of license plate recognition, various techniques have been developed for these three stages. In this chapter, we present a survey of these techniques that are used in three stages of license plate recognition.

3.2 License Plate Detection Algorithms

In the first stage of an ALPR system, the location of the license plate is determined in the input image. This input image may be a video sequence or a still image which could be a binary, grey scale, or a color image. Therefore, binary, grey scale, or color image processing techniques are used in this stage. In this section, we discuss some research works that use these techniques.

3.2.1 Detection Algorithms Using Binary Image Processing

Among the binary image processing techniques, edge statistics and morphological analysis [6, 7, 11], connected component analysis [12], and spatial measurement [12] are the most widely used method.

Edge Statistics and Morphological Analysis

Some research works [6, 7, 11] used the combination of edge statistics and mathematical morphology. The change in brightness in the license plate region is more frequent than that of any other part in the image. This has been used as a distinguishing property to locate license plate. The gradient magnitude and the local variance of the edges of an image are computed. License plates are located based on the higher magnitude of the gradient and the variance. In a complex image, some other parts, such as complex background, radiator of car etc. can also show high variance of brightness. As a result, only the edge statistics such as local mean, and variance cannot produce a good result for these complex images. To resolve this problem, mathematical morphology such as erosion, dilation etc. are used along with the edge statistics to remove unwanted edges in the processed image. This makes the license plate detection rate higher and faster.

Connected Component Analysis

Connected component analysis (CCA) [12] is a very important technique in binary image processing. It scans a binary image and labels its pixels into components based on pixel connectivity (either 4-connected or 8-connected). Once all groups of pixels have been determined, each pixel is labelled with a value according to the component to which was assigned. Extracting and labelling of various disjoint and connected components in an image are used frequently to compute important measurements of binary objects, such as size, aspect ratio, orientation etc. In the license plate segmentation stage, we use CCA to remove all noises and unwanted objects except the characters and digits from the binary license plate image to make the segmentation process flawless.

Spatial Measurement

Spatial measurements, such as area, orientation, and aspect ratio (AR) are also incorporated with binary image processing algorithms in [12] to locate the license plate in an image. These criteria are used to filter candidate license plate regions, which are in fact binary objects. Binary objects whose area, orientation, and aspect ratio do not fall into the respective valid ranges are eliminated. We use aspect ratio in the license plate detection stage to locate the license plate.

3.2.2 Detection Algorithms Using Grey Scale Processing

Many research works [3,14,15,38,39] used grey scale image processing techniques to locate the license plate. Some grey scale image processing techniques are image transformation [14], region segmentation [3, 15], edge counting [39], and block based [38].

Image transformation

Image transformations are widely used for the license plate detection. They are very much effective to find the edges or the straight lines in the input image that comprise the boundary of the license plate. Gabor filters are used in [14] along with the spatial measurements, such as size and aspect ratio to locate the license plate. The Gabor Filters have received considerable attention because the frequency and the orientation representations of human visual system can be easily approximated by these filters. In addition, these filters have been shown to posses optimal localization properties in both spatial and frequency domain and thus are well suited for texture segmentation problems. Gabor filters have been used in many applications, such as texture segmentation, target detection, document analysis, edge detection, retina identification, image coding, image representation as well as for the license plate detection. This technique has the advantage of analysing texture in an unlimited number of directions and scales. However, the method is computationally expensive and slow for images with large analysis. Other than Gabor filters, Houge Transform (HT) is also used in literature [4] to detect the boundary of the license plates. The Hough transform is a feature extraction technique that is used to find imperfect instances of objects within a certain class of shapes by a voting procedure. The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identify the positions of arbitrary shapes, such as circles or ellipses.

Region Segmentation

An adaptive image segmentation technique named Sliding Concentric Windows (SCW) is proposed for license plate detection in [15]. The SCW method was developed based on the local characteristics such as mean and variance in the image. This method uses two concentric windows A and B of different sizes $X_1 \times Y_1$ and $X_2 \times Y_2$ respectively, where $X_1 > X_2$ and $Y_1 > Y_2$. This method is described in details in Section 3.5.1.

Edge Counting

A license plate localization technique based on edge counting is presented in [39]. In this method, the vehicle image is scanned with N-row distance, counting the edges. If the number of the edges is greater than a threshold value, the presence of a plate is assumed. If the plate is not found during the first scan, the algorithm is repeated with the reduced threshold value. The method features fast execution time as it scans only some rows of the image. However, this method is not capable of locating license plates in complex scenarios. Moreover, it is not independent of the size of the license plate and the distance of the vehicle from the camera.

Block Based

Block-based grey scale processing technique to locate the license plate was presented in [38]. In this method, the input image is divided into blocks. Blocks with a high edge magnitude or high edge variance are identified as the possible license plate regions. However, all the detected blocks are not license plate regions. Some geometrical criteria, such as area and aspect ratio are used to accept the detected block as the valid license plate. Since block processing does not depend on the edge of the license plate boundary, it can be applied to an image with an unclear license plate boundary and can be implemented easily. This technique is not capable of locating the license plate in an input image with complex background, where many regions other than the license plate region demonstrate high edge magnitude and variance.

3.2.3 Other License Plate Detection Algorithms

Other than binary and grey level processing many research works can be found that use color processing [9, 10] and fuzzy set theory [16, 17].

Colour Processing

Many algorithms [9, 10] based on colour processing have been proposed in the literature for the license plate detection. The fundamental idea of the detection of a license plate region using color processing is that, the color combination of a plate (background) and its characters (foreground) is unique, and this unique combination occurs mostly in a plate region and rarely in other regions. This colour combination is searched pixel-by-pixel in the image to locate the license plate. However, color processing techniques have the following disadvantages. Colour is not stable when the lighting conditions change. The same object with different illuminations may have considerably different colors. Consequently, license plate detection rate may vary in different illumination conditions. Moreover, colour based techniques are country specific because license plate colours vary from one country to another country.

Fuzzy Set Theory

Fuzzy logic has been applied to the problem of locating the license plates in [16], where the authors define the following intuitive rules based on human perception on the license plate object: 1) the license plate is a bright rectangular area within which there are some dark areas, such as digits and characters; 2) the license plate is located approximately in the middle or lower middle part of the image; 3) the border of the license plate is brighter than its surroundings; and 4) the approximate dimension of the license plate is 530×120 mm. The concepts of "brightness" or "darkness," which are present in rules 1) and 3), are described as a fuzzy set with membership functions on the interval [0, 255], where 0 represents black, and 255 represents the white color in grey scale. The method works as follows: At first, the input image is resized to 768×576 pixels and partitioned into sub images of size 75×25 pixels. After partitioning the image, a fitness value is computed for every partitioned sub image based on the four intuitive rules. A partitioned sub image, which has the maximum fitness value is accepted as the license plate. The algorithm successfully located the license plate in 97 images out of the 100 images presented by the authors. This algorithm may not perform well for a set of random input samples because rules 2) and 4) restrict this algorithm to identify license plates in a specific distance.

3.3 License Plate Segmentation Algorithms

In the second stage of ALPR system, the characters inside the license plates are segmented. Both binary and grey scale image processing techniques are used to segment the characters. According to our study, no color based technique exists in the literature to segment the license plate.

3.3.1 Segmentation Algorithms Using Binary Image Processing

In the binary image processing techniques, several methods such as horizontal and vertical projection [9,18,19], and mathematical morphology [19,20] are used to segment the license plate.

Projections

Computing the horizontal and vertical projections of the pixels is the most common and simplest method to segment characters [9, 18, 19]. Generally the projections are computed using a binary image of the license plate. Horizontal and vertical projections are computed using the Equations 2.8 and 2.9. The horizontal projection is used to separate the rows of the license plate and the vertical projection is used to segment the characters and digits of the rows. The simplest idea is to use the minimum values of the projections to separate the rows or to segment the characters. We use the both kind of projections with adaptive threshold in the license plate segmentation stage.

Mathematical Morphology

Research work [19] presented a character segmentation algorithm based on mathematical morphology. They described an adaptive approach for seriously degraded plate images. After converting the license plate into a binary image, they reduce the noises by using morphological operations, such as thickening and pruning. Thickening operation is used to detect boundaries between overlapping characters and pruning operation is used for cleaning up parasitic objects in the thickened image. After removing noises, they compute the vertical projection and search for the segmentation points in the projection based on the minimum values. As a result, some characters might be divided into multiple segments and two or more characters might be put into one segment. They decide whether the merging of segments is necessary or not by employing the prior knowledge of the maximum quantity of segments and the width of a character. Similarly, they separated overlapping or connected objects, if necessary.

3.3.2 Segmentation Algorithms Using Grey Scale Processing

In the grey scale processing, methods like local and adaptive threshold [21, 22], histogram processing [23,39], and classifiers [25,26] are the most commonly used techniques.

Local and Adaptive threshold

This method is based on converting a grey scale license plate image into a binary image. This conversion can be done by employing global and local thresholds. A global threshold based binarization technique namely Otsu's binarization algorithm has been described in Section 2.3.2. Pixels with value above this threshold are classified as white and all other pixels are classified as black. However, binarization with a global threshold does not always produce efficient result due to non-uniform illumination in different parts of the license plate image. Local threshold is used in binarization to deal with the problem of non-uniform illumination. There are two types of local threshold based binarization techniques. In the first type of local binarization method, an image is divided into $m \times n$ blocks, and thresholds are computed for each block. In the second type of local binarization method, thresholds are computed for each pixel. The pixel is classified as white, if its value is above the threshold. Otherwise it is classified as black. Research work [22] employs this type of local binarization method in license plate segmentation stage. A local threshold based binarization technique has been described in Section 2.3.1.

Classifiers

Hidden Markov chains are used to model a stochastic relation between an input image and the corresponding character segmentation. Franc and Hlavac [34] proposed a character segmentation method for noisy low-resolution license plate images, where the segmentation problem was expressed as the maximum a posteriori estimation from a set of admissible segmentations. The method was based on exploiting a priori knowledge such as the predetermined number of characters in the plate and their equal (but unknown) segmented width. The statistical model was created from a training set of ground-truth segmentation examples of Czech plates provided by a user.

3.3.3 Segmentation in Bangla OCR

Character segmentation is an important sub part of Bangla OCR, where segmentation is performed in three different levels, such as line segmentation, word segmentation, and character segmentation [40–44]. Line segmentation is performed by scanning the input image horizontally, where the frequency of black pixels in each row is summed up in order to construct the row histogram. A boundary between the lines is denoted by the zero value in the histogram. After line segmentation, each line is scanned vertically for word segmentation. Number of black pixels in each column is calculated to construct column histogram. The portion of the line with continuous black pixels is considered to be a word in that line. If no black pixel is found in some vertical scan that is considered as the spacing between words. In order to segment the characters in a word, the position of Matra is determined by constructing a row histogram. The row with the highest frequency value indicates the position of Matra. Matra is removed to isolate the characters. Linear and piecewise linear vertical scanning is used to segment the isolated characters.

However, simple horizontal scanning is not sufficient enough to segment the rows of the license plate. Because the pattern of the Bangla license plates vary from each other. Some license plates have boundary, some have a horizontal line between the rows, and some contain extra information other than registration area, type, and number. These variations make it impossible to separate rows using simple horizontal scanning. In OCR the position of the Matra is determined by using the maximam frequency of pixels along a row. However, a license plate is more subject to noise, where it is not efficient to use this simple method for finding the position of Matra. Moreover, deletion of Matra sometime segments a character without Matra into multiple segments. Success of this Matra deletion approach is highly depends on the skewness detection and correction algorithms.

3.4 License Plate Recognition Algorithms

The third and the last stage of an ALPR system is the character recognition stage. Different types of neural networks, such as Hidden Markov Model (HMM) [29], Support Vector Machine (SVM) [13], and Artificial Neural Network (ANN) [17, 30, 33, 39] are mainly used to classify the characters. Other than neural networks, pattern or template matching techniques [27, 28] are also used to recognize the characters.

3.4.1 Support Vector Machine

A support vector machine (SVM) is a type of supervised learning methods that analyse data and recognize patterns. The SVM is used for classification and regression analysis. Support Vector Machine (SVM) was used in [13] to recognize the characters in Korean license plates. They used four SVM base character recognizer in their system to recognize the characters and digits in four different parts of the license plate, such as upper character, upper numbers, lower character, and lower numbers.

3.4.2 Neural Networks

Multilayer feed-forward neural networks have been used in may research works [17, 30, 33, 39] to recognize the characters and digits of the license plate. The classical training method for feed-forward neural networks is back-propagation [48] training algorithm. The network has to be trained for many training cycles to achieve good recognition rate over unknown test data. Moreover, the number of hidden layers, the number of neurons in the hidden layers, and the activation functions have to be chosen by a trial-and-error procedure [49] to attain optimum result.

3.4.3 Template Matching

Template matching is a technique for finding small parts of an image which match a template image. This is a suitable technique for the recognition of single-font, non-rotated, and fixed-size characters. Template matching is successfully implemented in [28], where the whole recognition process is based on the computation of the root-mean-square error, for all the shifts of the template g over a sub-image f with size $m \times n$. In this method, the lowest root-mean-square score gives the estimate for the best position of template within the search image.

Among three recognition techniques discussed in this section, we use multi-layer feed-forward neural network to recognize Bangla characters and digits. We use error back-propagation learning algorithm to train the neural network. In English license plate recognition, there are thirty six different symbols need to be recognized and for Bangla there are thirty five symbols. Artificial neural networks are very much efficient in this kind of multi class classification. They are very much robust and easy to use. Neural networks take insignificant amount of time to recognize a symbol, if it is trained beforehand. On the other hand, template matching technique works well for single-font, non-rotated, and fixed-size characters. In Bangla license plate recognition, template matching is not an efficient technique because characters have different fonts and sizes. Moreover, it takes a lot of time during recognition and is not effective to use in real time. Support vector machines are excellent two class classifier. However, license plate recognition is a multi class problem, where support vector machines are not effective.

3.5 Case Studies

A lot of research works have been conducted to develop an efficient ALPR system because of its numerous applications. Most of the ALPR system have been developed based on the pattern of the license plates of a particular country or state. In this chapter, we describe three different ALPR systems [3–5] from three different regions in details. These ALPR systems were developed to recognize English, Vietnamese, and Bangla license plates respectively. We also discuss the shortcomings of these ALPR systems to recognize the Bangla license plates.

3.5.1 An ALPR system for English LP

In [3], Anagnostopoulos et al. presented a license plate recognition system for the license plate written in English. In the license plate detection stage, they develop an algorithm namely Sliding Concentric Windows (SCW), which employs the

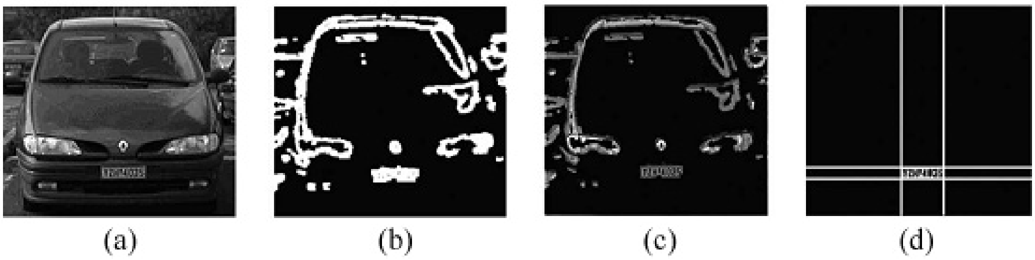


Fig. 3.1: Steps for license plate segmentation: (a) Original image, (b) Result of SCW segmentation technique after the application of the segmentation rule and thresholding, (c) Image after logical AND operation, and (d) License plate detection.

abrupt changes in the local characteristics of the image to determine the presence of a license plate in it. The algorithm uses the image statistics, such as standard deviation and mean value to describe the local irregularities in the image. This algorithm uses two concentric windows A and B of different sizes $X_1 \times Y_1$ and $X_2 \times Y_2$ respectively, where $X_2 > X_1$ and $Y_2 > Y_1$. At first, these windows are created centred at the first pixel of the image (upper left corner). The mean values or the standard deviations of the grey pixels of both the windows are computed. If the ratio of these mean values or standard deviations in the two windows exceeds a threshold T, which is set by the user, the pixel of the corresponding concentric windows is considered to belong to a license plate. They scan the image from left to right as well as from top to bottom using these two windows in order to determine which parts of the image belong to a license plate. The parameters X_1, X_2, Y_1, Y_2, and T are set by the user according to the specific application and aspect ratio of the license plate. The outcome of the SCW algorithm is a binary image containing candidate license plate regions. A logical AND operation is performed between this binary image and the original image. The resultant image is then binarize using the Sauvola locally adaptive binarization algorithm. Connected Component Analysis (CCA) is also used to identify connected regions in this binary image. A region is accepted as the valid license plate based on the size, the aspect ratio, the orientation, and the Euler number. The original image, and the corresponding image after the application of SCW algorithm, the image after logical AND operation, and the detected license plate region are shown in Fig. 3.1(a), 3.1(b), 3.1(c), and 3.1(d) respectively.

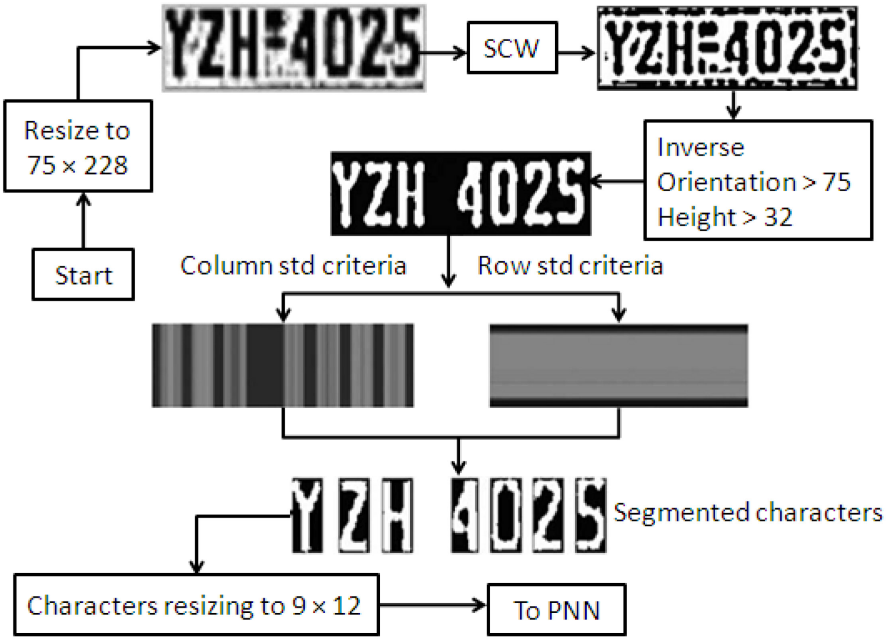


Fig. 3.2: Flowchart indicating the license plate segmentation process [3].

In the license plate segmentation stage, the cropped license plate from the input image is resized to 75×288 pixels using bicubic interpolation and then subjected to the SCW segmentation algorithm with the following parameters: $X1 = 2, Y1 = 5, X2 = 4, Y2 = 10$, and $T = 0.7$, where the standard deviation of the grey pixels is computed at each window. Above parameters are selected according to aspect ratio of the license plate and the threshold T is set after trial and error to optimize the result. After inverting the resultant image, binary objects are labelled using CCA. The heights and the orientations of these binary objects are measured. The objects whose measurement do not fulfil specific rules such as, orientation $> 75^o$ and height > 32 pixels are deleted. The remaining objects are segmented by calculating the standard deviations of the values in the rows and the columns of the license plate image. Segmentation points are selected based on the minimum standard deviations. The segmented characters and digits are then resized to 9×12 pixels. Steps of the license plate segmentation algorithm is shown in Fig. 3.2.

In the license plate recognition stage, they use a two layer Probabilistic Neural Network (PNN). The topology of this neural network is 108-180-36. The input layer consists of 108 nodes, which correspond to the 108-input vector (108 =

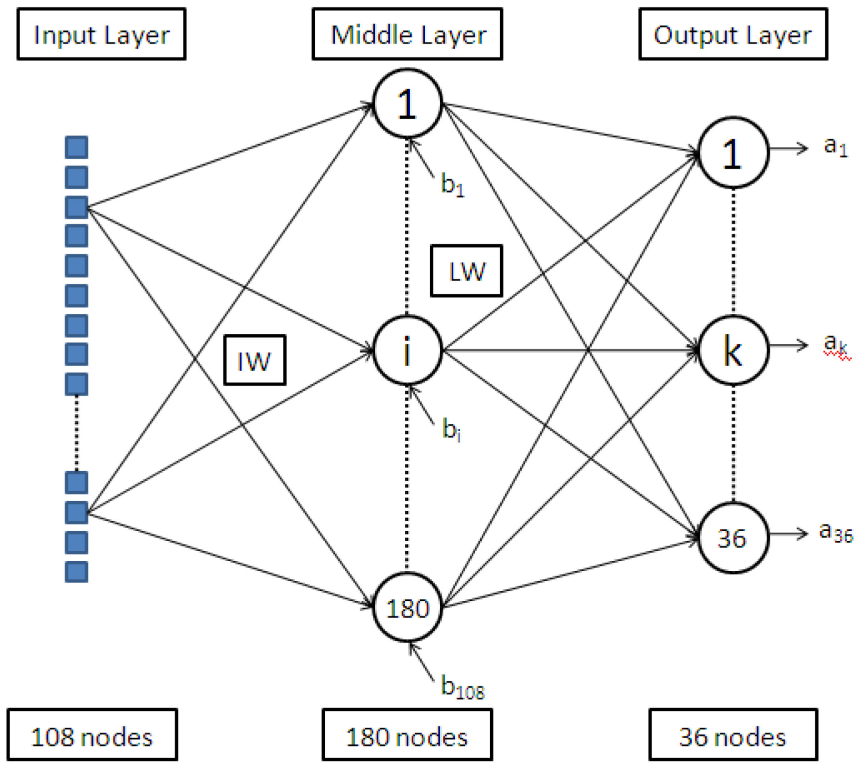


Fig. 3.3: Architecture of the PNN [3].

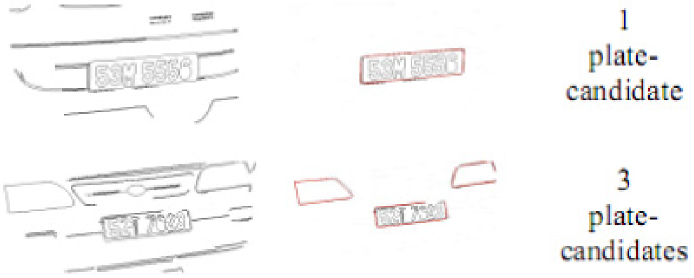
9×12). The middle layer has 180 nodes and the output layer has 36 nodes. Output layer corresponds to 36 characters and digits of English alphabet. A PNN uses a supervised training set to develop distribution functions within a pattern layer. These functions, in the recall mode, are used to estimate the likelihood of an input feature vector being part of a learned category or class. Architecture of the PNN is shown in Fig. 3.3.

The SCW algorithm sets the threshold T after trial and error to get an optimized result. This method might not perform well in a collection of diverse image samples, where the illuminations among the images are different. A threshold that performs well with a set of images may fail to produce good result with images having different illumination conditions. In addition, the SCW may falsely identify many license plate regions in an image having a complex background with many abrupt changes. In the license plate segmentation stage, they have used column standard deviation and row standard deviation to segment characters, which works effectively only on disjoint English characters. This method fails to segment Bangla characters, which are joint at the top by a horizontal line. They resize the segmented characters and digits to 12×9, which is too small to capture the distinguishing characteristics of the Bangla characters.

3.5.2 An ALPR system for Vietnamese LP

An automatic vehicle license plate recognition algorithm for Vietnamese license plate written in English is presented in [4]. In the license plate detection stage, the input image is converted into an edge image to find the contour lines of the objects in the image. Hough transform is applied to these contour lines to find two sets of parallel lines that comprise the boundary of the license plate. If these lines are not parallel to the horizontal plane, rotation transform is applied to adjust them to straight angle. Some candidate parallel lines are selected as the probable license plate regions that satisfy a pre-specified range of aspect ratios. Two horizontal crosscuts are used across the candidate regions and then the number of objects that are cut by these crosscuts are counted. A candidate is considered as a license plate if the number of cut objects is in the approximate range of the number of characters in a license plate. The license plate detection stage is shown in Fig. 3.4(a) and 3.4(b).

Like Bangla license plates, Vietnamese license plates may have two rows. They separate these two rows by using horizontal projection. The positions of the minimum values of horizontal projection are used as the start point or the end of a row in the license plate. Similarly, a simple vertical projection is used to segment the characters of a row. The segmentation process is re-evaluated by checking the number of characters. Vietnamese license plate contains 7 or 8 characters. In the license plate recognition stage, Hidden Markov Model (HMM) is used to recognize characters and digits. All character images are scaled into the size of 50×50 pixels. A window with the size of 9×9 is used to scan the image from left to right and top to bottom. These windows can overlap each other by two thirds of their size. For each window, the ratio of the foreground pixels and the window size are computed. By this way, a feature vector is computed which includes 196 values. Thus, the input layer of the HMM has 196 nodes. The output layer has 36 nodes, correspond to 26 characters and 10 digits. The license plate segmentation stage is shown in Fig. 3.5(a), and 3.5(b).



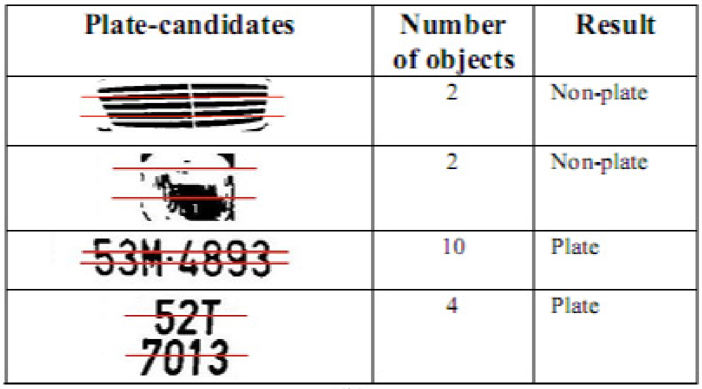


Plate-candidates	Number of objects	Result
	2	Non-plate
	2	Non-plate
53M-4893	10	Plate
52T 7013	4	Plate

(b)

Fig. 3.4: The license plate detection stage [4]: (a) Candidate license plates detection using Hough transform and contours, and (b) Candidate license plate validation.

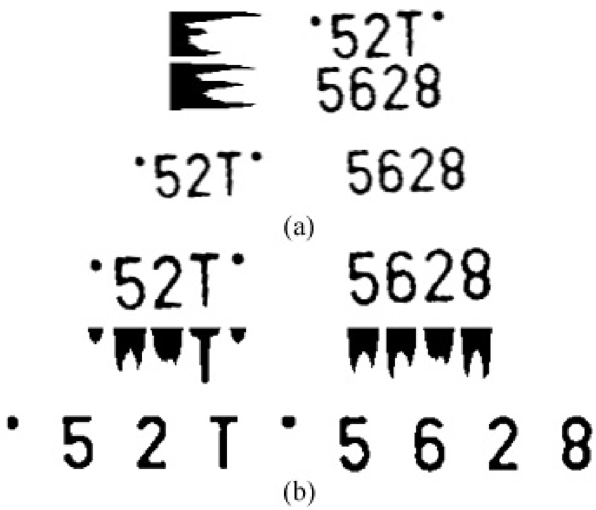


Fig. 3.5: The license plate segmentation stage [4]: (a) Row segmentation using horizontal projection, and (b) Character segmentation using vertical projection.

If the vehicle is too far away from the camera, characters inside the license plate will not be clearly visible. Hence, it will not be possible to find number of characters accurately by the horizontal crosscut. Although their simple vertical projection effectively segments characters in Vietnamese license plates, it fails with the Bangla characters in Bangladeshi license plates due to the presence of the Matra at the top of the most of the Bangla words in Bangladeshi license plates.

3.5.3 An ALPR system for Bangla LP

Automatic license plate detection and recognition system for Bangla license plate is presented in [5]. They also use detection, segmentation, and recognition stages in their system. Their input image samples were specially snapped by maintaining a certain distance and angle between the camera and the vehicle. In the detection stage, they crop the license plate region from the input image using some empirically measured criteria, such as the position of the license plate in the image and the size of the license plate. In the cropped license plate image, they perform a pixel-by-pixel search to locate the registration information.

In the segmentation stage, they use connected component analysis to locate disjoint regions in the cropped license plate image. A region is selected as a valid digit from these disjoint regions, if the area of this region falls into an assumed valid range of area. In the recognition stage, they recognize the digits using a four layer back-propagation neural network. The input layer and the hidden layers are comprised of 128 neurons. The output layer consists of 10 neurons. The license plate detection and segmentation stages are shown in Fig. 3.6(a), 3.6(b), 3.6(c), 3.6(d), and 3.6(e) respectively.

Their methods have some serious shortcomings. They completely ignored the first row of the license plate. The first row contains information on vehicle registration area and type. Registration area and type information cannot be ignored. Secondly, the position of the vehicle in the input image was used to locate the license plate. This method works well only for their set of input images but fails for arbitrary set of input images. Additionally, their brute force approach to search the registration information is time consuming. In the second stage, they segmented disjoint digits by connected component analysis which is not capable of segmenting joint Bangla characters.

Fig. 3.6: (a)Initial image, (b) The result of applying empirical crop, (c) Image obtained after brute force scanning and cropping, (d) The lower half of the registration plate, and (e) Individual characters being extracted from the registration plate.

3.6 Summery

In this chapter, a survey of the existing techniques in three stages of ALPR systems has been presented. Binary image processing, grey scale image processing, color image processing, and fuzzy set theory have been used in the license plate detection stage. Connected component analysis, mathematical morphology, Hough transform, Gabor filter, edge detector, etc. have been used along with the spatial measurements such as area, orientation, and aspect ratio to locate the license plates. From different binary and grey scale image processing techniques, projections and connected component analysis were mainly used to segment the license plates. Support vector machines, different types of neural networks, and template matching were used in the third stage to recognize the characters and digits of the license plates. In our proposed scheme for Bangla ALPR system, we use CCA, mathematical morphology, edge detector, spatial measurement, etc in different stages along with our newly developed techniques. We use multilayer feed-forward neural network with error back-propagation learning algorithm to recognize Bangla characters and digits because of its robustness and ease of use.

Three ALPR systems from three different regions have also been described in this chapter along with there shortcomings. The ALPR systems developed for English and Vietnamese license plates can efficiently detect, segment, and recognize their respective license plates but their algorithms are not capable of providing efficient result for the Bangla license plates in none of the three stages. Existing ALPR system for the Bangla license plate is not capable of locating license plate from an arbitrary input image. Its segmentation algorithm is not capable of segmenting all the characters and digits in the Bangla license plate.

Chapter 4

Proposed Scheme

4.1 Introduction

Three different ALPR systems have been described in Chapter 4. These ALPR systems were developed to recognize English, Vietnamese, and Bangla license plates. As discussed in Chapter 4, these ALPR systems work well under some constraints and are not capable of recognizing Bangla license plates efficiently for these constraints. It is obvious that a robust and effective ALPR system is required for the Bangla license plates. In this chapter, we propose a new ALPR system for the Bangla license plate. Like most of the ALPR systems, our proposed ALPR system is comprised of three stages, such as (i) detection, (ii) segmentation, and (iii) recognition. In each of these stages, we develop new algorithms, which we are going to present in the successive sections.

4.2 Detection of license plate region

We use the images of vehicles snapped by a digital camera mounted on the road. At first, we convert the color image into grey scale image because grey scale image is easy to process and our algorithm does not need color information. We produce an edge image from the grey scale image using Canny edge detection algorithm [31]. The Canny edge detection algorithm is a multi-stage algorithm to detect a wide range of edges in the images. It is known as the optimal edge detector with very low error rate. Other than Canny edge detector, we also use Horizontal and Vertical edge images. These edge images are processed further to locate the license plate regions.

The license plate region, which is our Region of Interest (ROI), could be any where in an image. We find the ROI in two steps. At the first step, we find the possible location of the vehicle. In the second step, the ROI is searched in the middle of the vehicle region along its vertical axis. The reason behind this action is that the probability of finding the license plate in the middle of a vehicle is very high. This two step search can locate the ROI efficiently in an image even in the presence of complex background and noise, which is very common in Bangladeshi road. Noises may exist in an image for several reasons. The boundary between the license plate and the vehicle body may not be distinctively sharp because of the distance between the camera and the vehicle. The license plates may be blur or may contain scratches in some old vehicles.

We use property of symmetry for locating the vehicle in an image. Property of symmetry was first used in [32] for the detection of on-road vehicles. Symmetry is a common characteristic in the nature and most of the vehicles that we see on the road posses this characteristic. If we see a vehicle from the front or from the rear, it will be symmetrical along a vertical axis. If an image contains more than one vehicle, then it will have multiple vertical axises of symmetry.

Algorithm 1 Symmetry detection algorithm

```
h ← Height of the image
w ← Width of the image
wh ← round((3/4)h); Height of the symmetry window
ww ← round((1/4)w); Width of the symmetry window
SM ← newMatrix(wh, ww); Symmetry Matrix
for c ← ww/2 to w − ww/2 do
  for r ← h/8 to 7h/8 do
    v ← 0;
    for k ← 1 to ww/2 do
      v ← v + abs(Image(r, c − k) + Image(r, c + k));
    end for
    SM(r − h/8 + 1, c − ww/2 + 1) ← v;
  end for
end for
```

We use Algorithm 1 to locate the vertical axis of symmetry in a binary edge image. We divide the whole image into overlapping windows. We detect symmetries in the windows rather than the whole image at once. The height of the windows are set to three-fourth of the input image's height. In all the 250 images we have taken from the road, the vehicles are located in the middle three-fourth portion of the image, which actually depends on the position and the operation of the camera. The width of the windows are set to one-fourth of the input image's width. The width of the region of interest or the license plate is less than to one-fourth of the input image's width in all the 250 input images. Hence, the windows of height three-fourth and width one-fourth of the input image's height and width respectively are sufficient to locate the vertical axis of symmetry. We start with the left most windows in the image. For each window, the center column is selected as its vertical axis. For each point on this vertical axis, a value is computed by summing up the absolute differences among the mirrored pixels equally separated by the axis. A value that is closer to zero represents a good symmetry. We put this value to a matrix, namely symmetry matrix. A column in this symmetry matrix represents the symmetry of the corresponding window. Similarly, we compute the symmetry of the next window, i.e., the next column of the symmetry matrix by shifting the symmetry window one pixel to the right. We repeat this process for all the windows in the image. We select a column as the vertical axis of symmetry whose sum of values is the minimum.

Most of the time, this vertical axis of symmetry intersects the license plate or the ROI. In Bangladesh, the aspect ratio of a rectangular region containing the license plate varies from 1 to 2. For this reason, we locate a rectangular region along the vertical axis having an aspect ratio between 1 and 2. This rectangular region is our ROI. We search ROI along the vertical axis in a binary image using Algorithm 2 which we named as ROID algorithm.

We start from the bottom of the vertical axis of symmetry. We search a white pixel along the vertical axis. This white pixel in the vertical axis could be a crossing point with the bottom edge of a ROI, which we named Probable Crossing Point (PCP). If the input image is not tilted, shape of the ROI will be rectangular. Therefore, we look for a rectangular shaped box along this PCP. We scan a square window named as Inspection Window (IW) of dimension IW pixels around the PCP to find more white pixels. The magnitude of IW is very important to deal with the problem of tilting image. If the chosen magnitude of IW is large, it will be able to find ROI in a seriously tilted image. If more white pixels are found in the current window, we continue this search in the next window in the right direction until a window is found without any white pixel. The last window with white pixel represents probable bottom-right corner of a ROI. If the first window along the PCP does not contain additional white pixel, we leave this PCP and search upwards along the vertical axis for a new PCP. If we successfully reach to a probable bottom-right corner of a ROI, we perform a search in the vicinity of this point to determine the beginning of a right edge of a ROI. We choose a position which has the maximum number of pixels above it. If no such position is found, we halt the current search and start from the next available PCP. We perform the above window search again to get the probable right-edge of a ROI. Similarly, we search for the top edge and the left edge of a ROI. If we fail to get any one edge, we start the search from the next available PCP. Finally, we calculate the aspect ratio and the size of the detected ROI. If the aspect ratio and the size are within their respective valid range, we accept the ROI as a valid ROI. Otherwise, we start new search from the next PCP along the vertical axis of symmetry. We continue this process until we get a valid ROI or we reach at the end of the vertical axis of symmetry.

Above license plate detection process is illustrated in Fig. 4.1. Original input images are shown in Fig. 4.1(a) containing only one vehicle with visible license plate. Images after applying the edge detection algorithm is shown in Fig. 4.1(b). After using the symmetry detection algorithm, we get the vertical axis of symmetry which is shown as an overlaid vertical line in Fig. 4.1(b). Fig. 4.1(c) shows the ROI found by the ROID algorithm.

4.3 Segmentation of characters

As mentioned before, Bangla characters in Bangladeshi license plates are very difficult to segment because of their various complexities as described in Section 1.3. To deal with these complexities, we develop a new algorithm that use horizontal and vertical projections with adaptive threshold to segment Bangla characters and digits efficiently. A lot of research works [9, 18, 19] used projections to segment license plates. Our novelty is that we use an adaptive threshold that changes based on the input image and the segmentation performance and capable of segmenting joint Bangla characters. We divide the segmentation process into two steps. In the first step, we separate the rows containing license

Algorithm 2 Algorithm ROID

```
input: Edge image (im), Vertical axis of symmetry
h, w ← Height and the width of the image.
vaxis ← Position of vertical axis of symmetry
mv ← Right; Moving direction
for r ← h to 1 do
  if im(r,cc) is a PCP then
    Set cr ← r
    Set cc ← vaxis
    while true do
      Wind ← Square window of size IW Centered at (cc, cr)
      if mv ← Right then
        if Wind contains any white pixel then
          increment cc by IW
        else
          Locate a transition point tp along the row cr
          Set cc ← tp, mv ← Up
          Save tp
        end if
      else if mv ← Up then
        if Wind contains any white pixel then
          decrement cr by IW
        else
          Locate a transition point tp along the column cc
          Set cr ← tp, Set mv ← Down
          Save tp
        end if
      else if mv ← Left then
        if Wind contains any white pixel then
          decrement cc by IW
        else
          Locate a transition point tp along the row cr
          Set cc ← tp, mv ← Left
          Save tp
        end if
      else if mv ← Down then
        if Wind contains any white pixel then
          increment cr by IW
        else
          Compute aspect ratio and size of the found ROI from tp's
          if aspect ratio and size are belong to an valid range then
            ROI found, exit loops.
          end if
        end if
      end if
    end while
  end if
end for
```

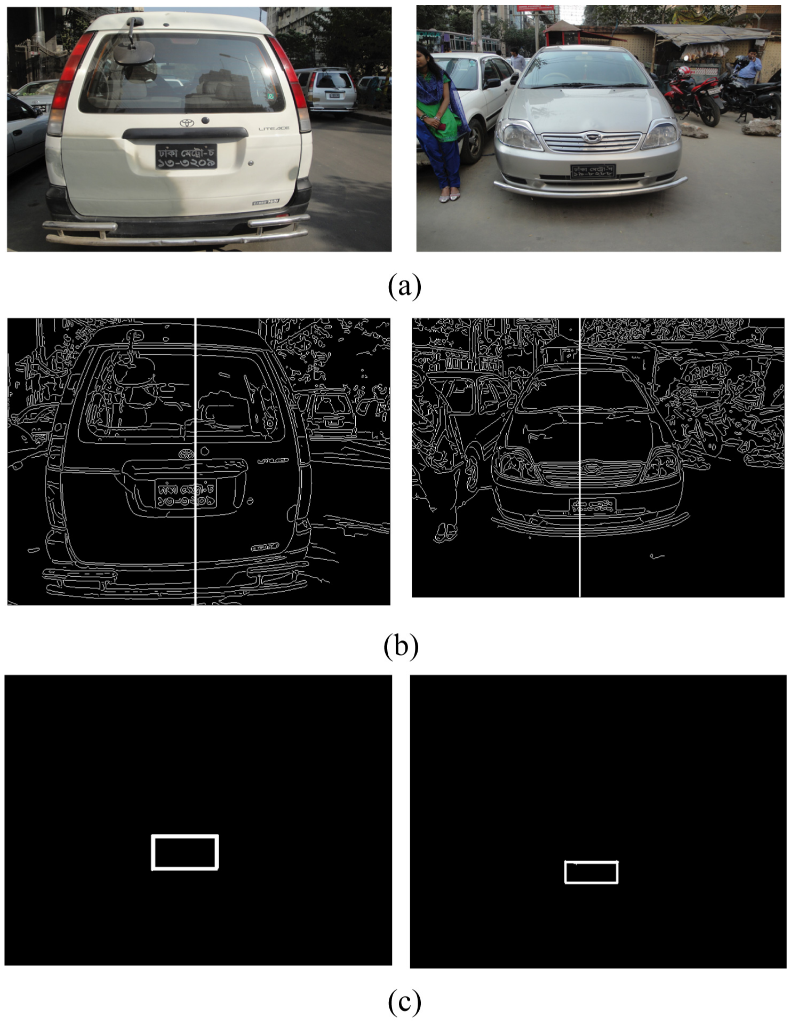


Fig. 4.1: Sample outputs of license plate detection stage: (a) Input image containing vehicle, (b) Edge image with overlaying vertical axis of symmetry, and (c) Detected ROI.

information. We segment the characters and digits in the second step.

4.3.1 Separating the rows

In this step, we use the cropped image of a license plate from an input image. We use the binary image containing vehicle registration information. The binary image might contain some noises. These noises are run-away pixels that should be removed to make segmentation process flawless. In order to remove the noise, we find the connected regions in the binary image. We measure the area of the connected regions and delete all the smaller regions having the area less than a threshold T_n.

After removing the noise from the binary image, we separate and crop the rows. We use Equation 4.1 to evaluate the horizontal projection $f_h(i)$ of the binary image. Resultant horizontal projection $f_h(i)$ is not smooth and may contain unwanted local minima and maxima. We filter this function to get rid of these unwanted local minima and maxima using a moving average filter depicted in Equation 4.2. Where W is the window size of the moving average filter. This filtering operation produces $f_{h1}(i)$ from $f_h(i)$. $f_{h1}(i)$ contains some convex arches. Each arch represents a row of the license plate. We get the arches from $f_{h1}(i)$ by enumerating the local maxima. If these arches are joint with each other, we separate them by shifting the x-axis upward by a parameter y_h. We calculate the value of y_h by using Equation 4.3. We select two largest arches based on the area covered by them. These two largest arches represent two rows in Bangla license plates that contain registration information.

Above rows Extraction process of the license plate is shown in Fig. 4.2 and 4.3. License plate in Fig. 4.2(a) is very simple containing only two rows. Resultant horizontal projection $f_h(i)$ is shown in Fig. 4.2(b). Fig. 4.2(c) shows the filtered horizontal projection $f_{h1}(i)$, which contains two arches corresponding to two rows. Calculated threshold is shown as the overlaid horizontal line in Fig. 4.2(c). The license plate shown in Fig. 4.3(a) is more complex than that of Fig. 4.2(a). It contains more than two rows. Resultant horizontal projection $f_h(i)$ is again shown in Fig. 4.3(b). Fig. 4.3(c) shows the filtered horizontal projection $f_{h1}(i)$, which contains five arches that correspond to five rows. Calculated threshold is again shown as the overlaid horizontal line in Fig. 4.3(c). We consider two largest arches based on the area covered by them. These two arches correspond to two rows having the registration information in the license plate.

$$f_h(i) = \sum_{j=1}^{col} LP(i,j) \tag{4.1}$$

$$f_{h1}(i) = \begin{cases} f_h(i), & \text{if } t \leq W \\ \frac{1}{W}\sum_{k=1}^{W} f_h(i-k), & \text{Otherwise} \end{cases} \tag{4.2}$$

$$y_h = Average(LocalMinima(f_{h1}(i))) \tag{4.3}$$

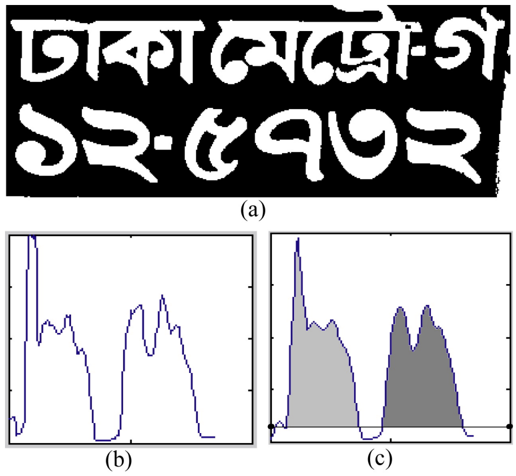


Fig. 4.2: Sample outputs of license plate segmentation stage: (a) Binary image of a simple license plate, (b) Horizontal projection of the pixels, and (c) Filtered projection.

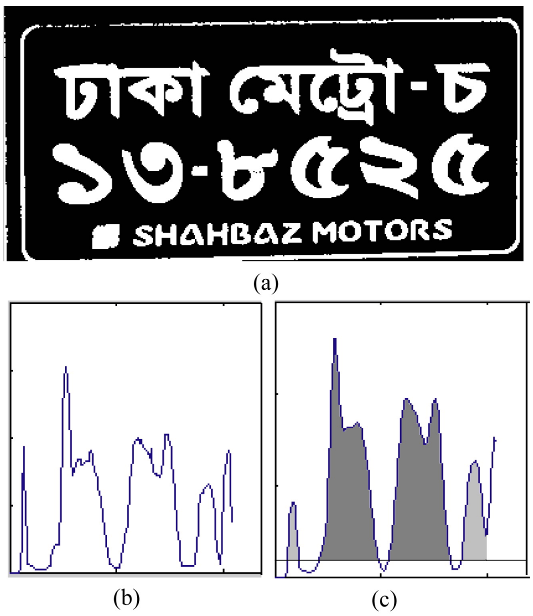


Fig. 4.3: Sample outputs of license plate segmentation stage: (a) Binary image of a complex license plate, (b) Horizontal projection of the pixels, and (c) Filtered projection.

4.3.2 Segmenting the characters and digits

Among two rows, segmenting the characters of the first row is the most tricky part because they are joint by Matra at the top. We develop a new algorithm named Segmentation with Adaptive Threshold (SAT) to segment Bangla characters in the first row of the license plate. In this algorithm we use an adaptive threshold. It adapts itself based on the input image and its performance in segmenting the characters. Value of this threshold is very critical. A low threshold will put two or more characters in one segment and a higher threshold will divide a character into multiple segments.

We set the initial value (T_{int}) of the threshold to the half of an approximate width of Matra. Bangla alphabet has eight characters without Matra. A higher value of the initial threshold would segment these characters into multiple segments and a lower value would slow down the segmentation process. In order to compute T_{int}, we evaluate the horizontal projection $f_p(i)$ of the first row, that contains Bangla characters and filter it to get the function $f_{p1}(i)$. $f_p(i)$ and $f_{p1}(i)$ are similar to $f_h(i)$ and $f_{h1}(i)$ of Equation 4.1 and 4.2 respectively. Since the number of pixels in the region of Matra is higher than that of its neighbourhood, a portion of the function $f_{p1}(i)$ corresponding to this region will be much steeper than its neighbourhood. We set T_{int} to the half of the extent of this steeper portion. We compute the extent of this steeper portion. We evaluate the derivative $f^1_{p1}(i)$ of the function $f_{p1}(i)$ using Equation 4.4. We list all the maxima and minima of $f_{p1}(i)$. We search for a sequence of numbers between a consecutive maximum and minimum in $f^1_{p1}(i)$. We select a sequence where the numbers are positive and one of the number is the maximum of $f^1_{p1}(i)$. Finally, we set the value of T_{int} to the length of this sequence. Fig. 4.4(a) and 4.4(b) show the first and second row of the license plate respectively. The extent of the steeper portion and T_{int} are shown by two overlaid vertical lines in Fig. 4.4(c).

$$f^1_{p1}(i) = \frac{1}{3}\sum_{x=1}^{3} |f_{p1}(i-x) - f_{p1}(i+x)| \tag{4.4}$$

After setting the value of T_{int}, we proceed to segment the characters. We evaluate the vertical projection $f_v(i)$ of the cropped binary image containing the first row using Equation 4.5 . This vertical projection $f_v(i)$ is not smooth either and may contain many unwanted local maxima and minima. We get rid of these unwanted local maxima and minima by filtering $f_v(i)$. We use a moving average filter as shown in Equation 4.6. This filtering operation produces a new smooth function $f_{v1}(i)$. Fig. 4.5(a), 4.5(b), and 4.5(c) show the first row of the license plate, vertical projection ($f_v(i)$), and filtered vertical projection ($f_{v1}(i)$)

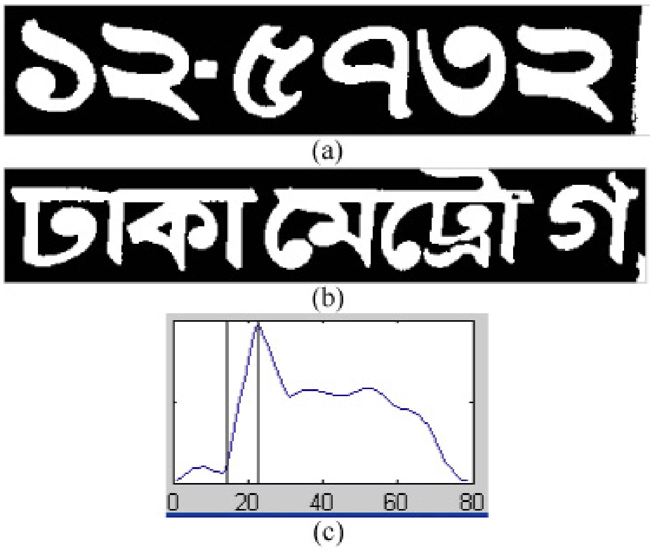


Fig. 4.4: Threshold initialization step: (a) Separated row containing digits, (b) Separated row containing characters, and (c) Calculating initial threshold.

respectively.

$f_{v1}(i)$ contains many convex arches equal to the number of characters present in the first row of the license plate. If these convex arches are joint with each other, we separate them by moving the x-axis upward to an amount equal to T_{int}. As a result some characters will be segmented and some might not be segmented depending on the value of T_{int}. We determine whether the segmentation is complete or not by counting the number of edges in a segmented part. Most of the Bangla characters have less than or equal to four edges if we project them horizontally across their middle. If a segmented part contains less than or equal to four edges then the segmentation can be assumed complete. On the other hand, if a segmented part contains more than four edges, segmentation cannot be assumed complete. In this case, we increase the value of T_{int} and repeat the process until the segmentation is complete.

In order to segment digits, we evaluate the vertical projection $f_d(i)$ of the cropped binary image containing digits and filter it to get the function $f_{d1}(i)$. $f_d(i)$ and $f_{d1}(i)$ are similar to $f_v(i)$ and $f_{v1}(i)$ of Equation 4.5 and 4.6 respectively. $f_{d1}(i)$ contains many convex arches equal to the number of digits present in the second row of the license plate. Since the digits in the license plate are not joined, these arches are separated by the x-axis. In a non ideal case, where the license plate is subject to various noises, we separate the arches by moving the x-axis upward to an amount equal to T_d. Magnitude of T_d depends on the size of the cropped license plate image.

Steps of the algorithm SAT are illustrated graphically in Fig. 4.6. After applying T_{int} to the function $f_{v1}(i)$, we get four separated parts of the function as shown in Fig. 4.6(a). Resultant four segments of the characters and an overlaid horizontal line are shown in Fig. 4.6(b). The numbers of edges in the segments are four, eight, twelve, and four respectively. We ignore the holes inside the characters while counting edges. The first and the fourth segments both have exactly four edges. The segmentations of these two characters can be considered complete. Since the second and the third segments have more than four edges, we need to segment them further. To complete these segmentations we increase the value of T_{int} as shown in Fig. 4.6(c). After applying this new threshold we get seven new segments from previous two segments as shown in Fig. 4.6(d). Among these seven segments five are complete because their edge count is less than or equal to four. We repeat this process until the last segment contains only one character.

$$f_v(i) = \sum_{i=1}^{row} LP(i,j) \tag{4.5}$$

$$f_{v1}(i) = \begin{cases} f_v(i), & \text{if } t \leq W \\ \frac{1}{W}\sum_{k=1}^{W} f_v(i-k), & \text{Otherwise} \end{cases} \tag{4.6}$$

Segmentation of digits is shown in Fig. 4.7(a). This is relatively easy task because unlike the characters digits are not joint by Matra. We take the vertical projection as shown in Fig. 4.7(b). The resultant arches are separated using a constant threshold. Fig. 4.7(c) illustrates the segmented arches corresponding to the digits.

4.4 Character Classification

In this section, we describe how the extracted characters and digits of the license plate are classified individually. We use Artificial Neural Network (ANN) for classifying characters and digits. From several types of ANN, we use a feed for-

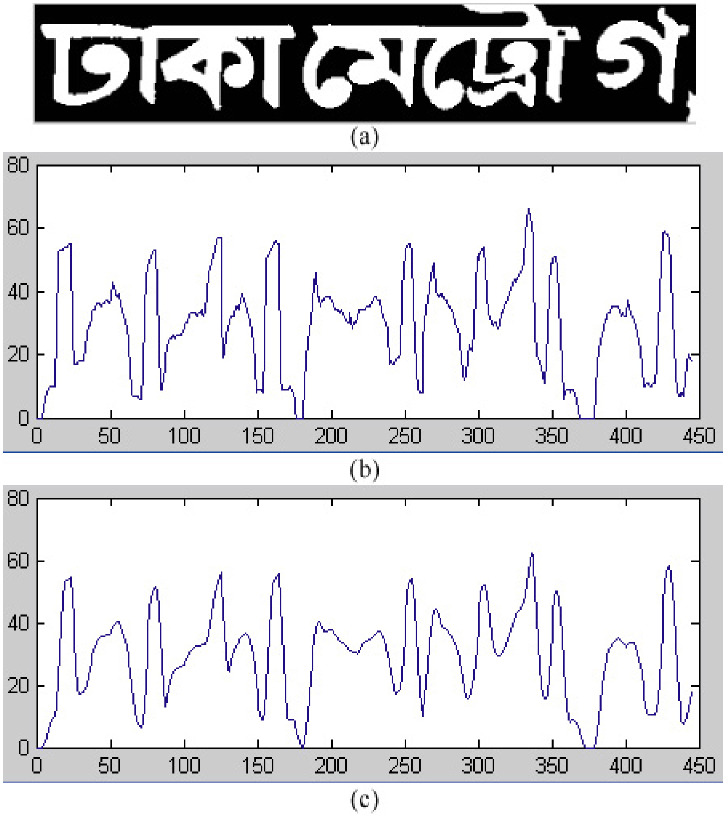


Fig. 4.5: Segmenting characters: (a) Binary image containing characters, (b) Vertical projection of the pixel, and (c) Filtered vertical projection.

ward back-propagation network, since it is robust and easy to use. Once they are trained, they can be used efficiently in real time [48]. The classification process includes resizing the image of individual characters and digits, feature extraction, normalizing features, network architecture design, training the network, and testing the network with unknown data. These steps are illustrated in the rest of this sub section.

4.4.1 Feature extraction

Image of the extracted characters and digits may have different sizes. We scale up or down the character and the digit images into 32×32 image before using them as the input to ANN. Bicubic interpolation is used to minimize the noise during resizing the images. We extract a feature set in order to train the ANN. At first, we take the vertical projection of the image. This produces a vector of size 32, where each element has a value between 0 to 32. This vector is normalized

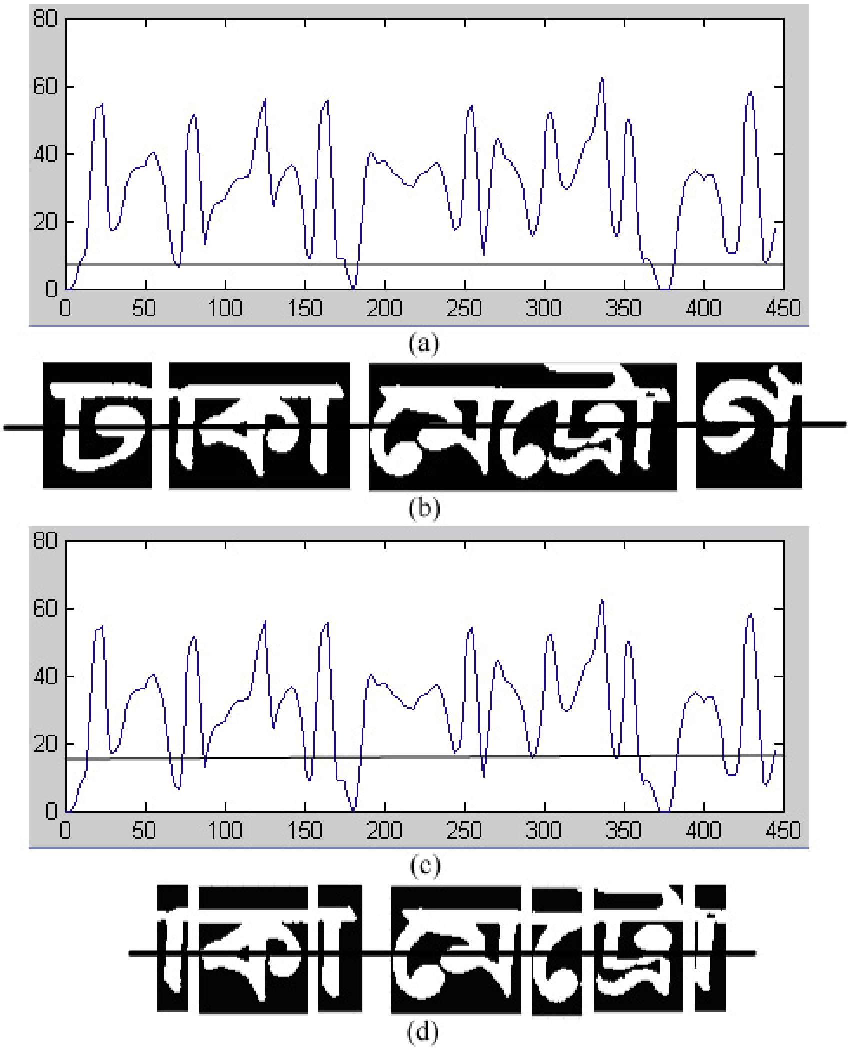


Fig. 4.6: Segmenting characters

by dividing each element by 32. As a result, the values of the elements in the vector become 0 to 1. Similarly, we get another vector of size 32 by taking the horizontal projection of the image. We merge these two vectors of size 32 and get a single vector of size 64. This vector is used as the input to the ANN. Some training samples of the character and digit classifications are shown in Fig. 4.8 and 4.9 respectively.

4.4.2 Network design

We use two separate ANN to classify the characters in the first row and the digits in the second row of the license plate. Since the size of the input vector is 64, both of the neural networks must have input layer of size 64. Neural network designed

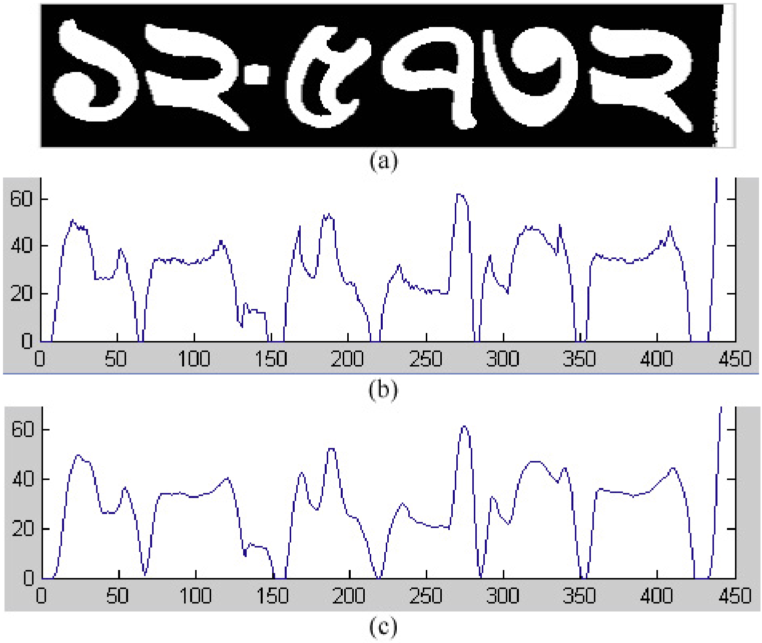


Fig. 4.7: Segmenting digits: (a) Binary image containing digits, (b) Vertical projection of the pixel, and (c) Filtered horizontal projection.

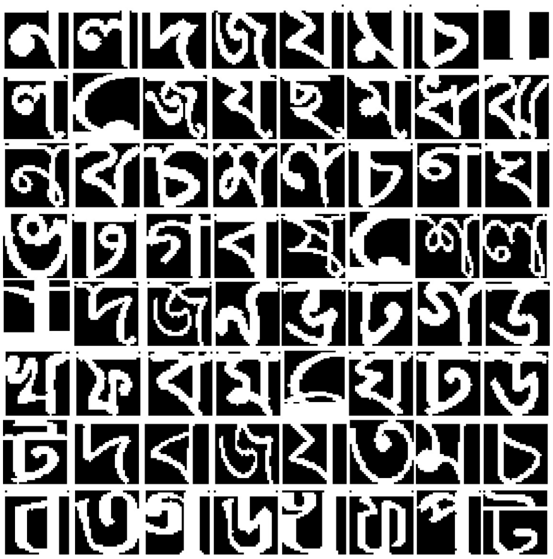


Fig. 4.8: Some training samples for the character classifier neural network.

for the digit classification must have 10 neurons in the output layer since we have ten different digits to recognize. The neural network for character classification must have 25 neurons in the output layer. Though Bangla Alphabet has 50 different characters, not all of them are used in the license plate. We observe that 25 different characters are used extensively. That's why 25 output neurons

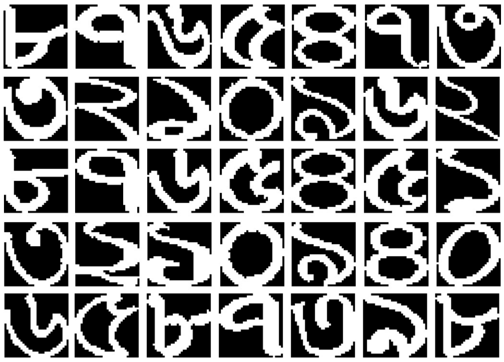


Fig. 4.9: Some training samples for the digit classifier neural network.

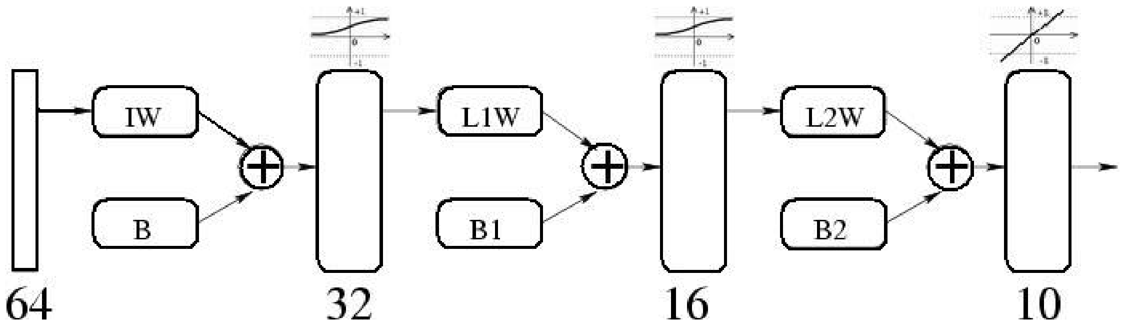


Fig. 4.10: Neural network for digit classification.

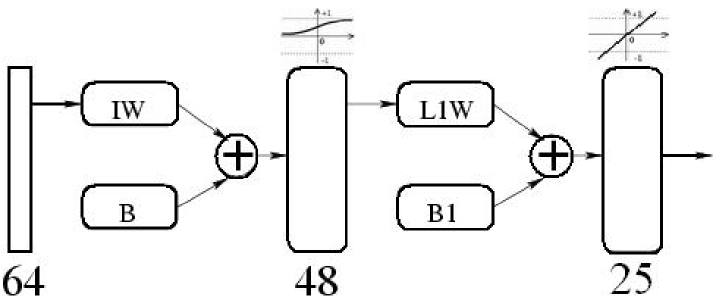


Fig. 4.11: Neural network for character classification.

are sufficient. We use pure linear transfer function as shown in Fig. 2.9(c) in the output neurons.

To devise a neural network suitable for recognizing Bangla digits of the license plate, we train neural network with different network architectures by varying the number of hidden layers, the number of neurons in the hidden layers, the transfer functions, and the training functions. Next, we simulate the network using test data and record the success rate in recognition. After some extensive experimentation, we observe that a three layer network with 32 and 16 neurons in the first and second hidden layers respectively can recognize the Bangla digits successfully. From our experiment, we observe that the performance of log-sigmoid transfer function (Fig. 2.9(g)) is much better than that of tan-sigmoid transfer function (Fig. 2.9(h)) over unknown test data. We also observe that the training convergence rate of Levenberg-Marquardt back-propagation (LM) [45] training function is higher than that of Gradient Descent with Adaptive back-propagation (GDA) [48], One Step Secant back-propagation (OSS) [47], and Scaled Conjugate Gradient back-propagation (SCG) [46] training functions. However, the performance of OSS back-propagation algorithm over unknown test data is much higher than the others. We use the OSS back-propagation algorithm as the training function in order to make our system robust. Fig. 4.10 shows the neural network for the character classification. Similarly, we design a neural network to recognize Bangla characters as shown in Fig. 4.11. This neural network has only one hidden layer with 48 neurons. Like the first neural network, we use log-sigmoid function as the transfer function and One Step Secant (OSS) back-propagation algorithm as the training function.

Since some Bangla characters are very similar to each other and have less distinguishing features, we use two different neural networks to recognize the characters and the digits to improve the success rate in recognition. As we discussed earlier, Bangladeshi license plates have two rows. In most of the cases, First row contains only the characters indicating the registration area and its type. Registration area is comprised of multiple characters and registration type is a single character. Second row contains six digits registration number. In this case, first and second rows are recognized by the character and digit classifiers respectively. However, in some cases, there is only one row containing all the registration information or second row contains both the registration type and number. In both of these cases, last six segmented symbols are digits. These symbols are recognized by the digit classifier and other symbols are recognized by the character classifier.

4.5 Summary

In this chapter, a new ALPR system for the Bangla license plates has been described. At first, the position of the vehicle is determined by using the property of symmetry. A boundary based algorithm has been proposed to locate the license plate in this vehicle region. License plate rows containing registration information have been separated by using horizontal projection with adaptive threshold. The characters and the digits of the rows have been segmented using vertical projection with adaptive threshold. Multilayer feed-forward back-propagation neural networks have been used to recognize the characters and the digits of the Bangla license plate.

Chapter 5

Experimental Studies

5.1 Introduction

We have proposed a new ALPR system to recognize the Bangla license plates in Chapter 5 with new algorithms for detection, segmentation, and recognition stages considering various features of the Bangla license plates. In this chapter, these algorithms are implemented using MATLAB 7.1 and applied on 250 grey scale images of Bangladeshi vehicles snapped under different illumination conditions, such as night, sunny, and cloudy. We also test the algorithms by using the vehicle images containing English license plates taken from an international license plate database. We also implement the ALPR systems presented in [3–5] using MATLAB 7.1. Experiment set up and parameter values of these algorithms in detection, segmentation, and recognition stages are shown in Table 5.1, 5.3, and 5.6 respectively.

5.2 License Plate Detection

Sample examples of license plate detection stages are shown in Fig. 5.1. Original input images, edge images, and the ROIs are shown in Fig. 5.1(a), 5.1(b), and 5.1(c) respectively. Experimental results of the vehicle and the license plate detection stages are shown in Table 5.2. In the presence of complex background and highly variable license plate patterns in the input images, the average detection rate of our algorithm is 92.8%. The successful license plate detection rates of the algorithms [3], [4], and [5] are 73.5%, 85%, and 0% respectively. It proves that the performance of our algorithm is better than that of the other algorithms in detecting Bangla license plate. The algorithm presented in [3] is based on

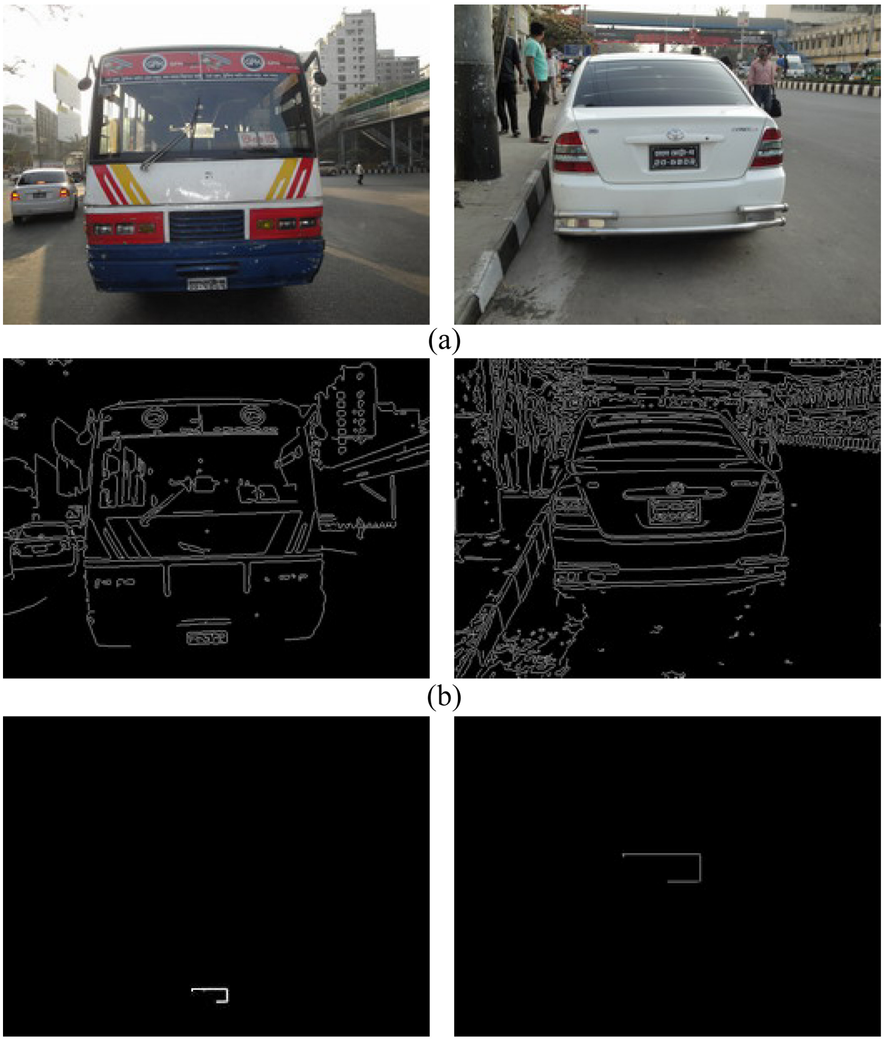
Algorithms	Parameters	Values
Proposed Scheme	Symmetry window: Height Symmetry window: Width Inspection Window (IW) Search area (SA)	$(3/4)^{th}$ of input image height $(1/4)^{th}$ of input image width 5×5 16 pixels
SCW [3]	Window: A Window: B Threshold: T Statistical measurement	$X_1 = 3, Y_1 = 6$ $X_2 = 6, Y_2 = 12$ 0.8 Standard deviation
Duan et al. [4]	1-row LP 2-row LP	$12 \leq Cut - objects \leq 20$ $5 \leq Cut - objects \leq 10$
Mashuk et al. [5]	-	-

Table 5.1: Experiment set up for license plate detection stage.

local abrupt changes and a threshold that needs to be set by trial and error, the algorithm presented in [4] depends on the distance between the camera and the vehicle, and the algorithm presented in [5] depends on the position of the license plate in the input image. Specific dependencies made the algorithms fail to locate Bangla license plate in a image, where the required dependencies are not present strongly.

Algorithms	Conditions	No. of images	Symmetry detection	Symmetry detection rate (%)	License plate detection	Success rate (%)
ROID	Sunny	145	130	89.6	136	93.8
	Cloudy	65	54	83.0	59	90.8
	Night	40	34	85.0	37	92.5
	Total	250	218	87.2	232	92.8
SCW [3]	Total	80	-	-	59	73.75
Duan et al. [4]	Total	80	-	-	68	85
Mashuk et al. [5]	Total	80	-	-	0	0

Table 5.2: Experimental results in license plate detection stage.



(a)

(b)

(c)

Fig. 5.1: License plate detection: (a) Original input images, (b) Edge images, and (c) After the application of symmetry detection algorithm and ROID.

5.3 License Plate Segmentation

The experimental results of row separation algorithm are shown in Table 5.4. The success rate of our algorithm is 98.4%. Out of the 250 license plates, it correctly separates the rows of 246. It fails to separate only 4 license plates due to the presence of noise in the input images. Algorithms presented in [3] and [5] do not include any method to separate the rows in the license plate. Success rate of algorithm [4] in separating the rows containing registration information is 93.75% only. Our row separation algorithm performs better than that of [4] because we use an adaptive threshold, which is more capable of separating rows

Algorithms	Parameters	Values
Proposed Scheme	SMA: Window size (W)	6
	T_n	$(1/100)^{th} of LParea$
	T_d	$(1/8)^{th} of rowheight$
SCW [3]	Window: A	$X_1 = 2, Y_1 = 5$
	Window: B	$X_2 = 4, Y_2 = 10$
	Threshold: T	0.7
	Statistical measurement	Standard deviation
Duan et al. [4]	-	-
Mashuk et al. [5]	-	-

Table 5.3: Experiment set up for license plate segmentation stage.

in the presence of noises.

Fig. 5.2 shows the segmentation performance of our SAT algorithm, manual segmentation, and the segmentation with constant threshold [4] over a license plate image. Fig. 5.2(a) shows the first row of a license plate containing characters. Fig. 5.2(b) shows the manual segmentation. Fig. 5.2(c) shows the segmentation with constant threshold [4], where some segments contain multiple characters and some characters are divided into two segments. Fig. 5.2(d) shows the segmentation performance of our SAT algorithm, which is very close to the performance of manual segmentation. Overall performance of our SAT algorithm is shown in Table 5.5. Its success rate in segmenting the characters and the digits in the license plates are 96.0% and 98.8% respectively. Algorithms presented in [3], [4], and [5] cannot be applied to segment Bangla characters in the license plate due to the presence of Matra. We implement these algorithms to segment the digits in the Bangla license plates. Success rates in digit segmentation of these algorithms are 97.5%, 95%, and 97.5% respectively. Our algorithm performs better than these algorithms since we deal with the noise presented in the image during digit segmentation by setting an appropriate threshold.

5.4 License Plate Recognition

Table 5.7 shows the experimental results of character and digit recognition stage. It shows that the correct recognition rates of our digit and character classifiers are 97.5% and 88.7% respectively. Correct recognition rate of the characters are little

Algorithm	No. of LP	Correct separation of rows	Success rate(%)
SAT	250	246	98.4
SCW [3]	80	-	-
Duan et al. [4]	80	75	93.75
Mashuk et al. [5]	80	-	-

Table 5.4: Experimental results in row segmentation stage.

Fig. 5.2: Character segmentation: (a) Original license plate, (b) Manual segmentation, (c) Segmentation with constant threshold [4], and (d) Segmentation with adaptive threshold (Algorithm SAT).

lower than that of the digits because the Bangla characters are joint to each other by Matra and the space between two adjacent characters is less compared to that of between two adjacent digits. The success rates in Bangla digit recognition of the classifiers in [3], [4], and [5] are 96.7%, 97.5%, and 96.7% respectively. The success rates in Bangla character recognition of these algorithms are 77.5%, 85%, and 70% respectively. Performances of these classifiers in recognizing Bangla characters are not as good as our neural network. Anagnostopoulos et al. [3] and Mashuk et al. [5] used input character images of size 12×9 pixels and 16×8 pixels respectively. These sizes are too small to capture the distinguishing features of Bangla characters. On the other hand, Duan et al. [4] use character images of size 50×50 pixels and the overlapping windows of size 9×9 pixels, and scan

Algorithm	No. of LP	Correctly segmented digits	Success rate (%)	Correctly segmented characters	Success rate(%)
SAT	250	247	98.8	240	96.0
SCW [3]	80	78	97.5	0	0
Duan et al. [4]	80	76	95	0	0
Mashuk et al. [5]	80	78	97.5	0	0

Table 5.5: Experimental results in character and digit segmentation stage.

Algorithms	Parameters	Values
Proposed Scheme	Number of training images	2100
	Performance function	Mean squared error (MSE)
	Training function	One Step Secant Back-prop.
	Learning rate	0.05
	Number of iteration	500
	Performance goal	10^{-5}
SCW [3]	Number of training images	1000
	Network type	Probabilistic
	Spread constant	0.1
	Number of iteration	500
	Performance goal	10^{-5}
Duan et al. [4]	Number of training images	1000
	Network type	Hidden Markov Model
	Training algorithm	Viterbi
	Tolerance	10^{-4}
	Number of iteration	100
Mashuk et al. [5]	Number of training images	2100
	Performance function	Mean squared error (MSE)
	Training function	One Step Secant Back-prop.
	Learning rate	0.05
	Number of iteration	500
	Performance goal	10^{-5}

Table 5.6: Experiment set up for digit and character recognition stage.

these windows in the image from left to right as well as from top to bottom, which helps them to achieve little better performance than that of the classifiers in [3] and [5]. However, the performance of the classifier in [4] is still worse than that of our classifier.

Methods	Neural network	No. of instances	Correctly classified	Success rate(%)
Proposed method	Digit classification	650	634	97.5
	Character classification	570	506	88.7
Anagnos. et al. [3]	Digit classification	120	116	96.7
	Character classification	120	93	77.5
Duan et al. [4]	Digit classification	120	117	97.5
	Character classification	120	102	85
Mashuk et al. [5]	Digit classification	120	116	96.7
	Character classification	120	84	70

Table 5.7: Experimental results in digit and character classification stage using neural network.

5.5 English License Plate

We evaluate our algorithms in terms of global usability by using the vehicle image samples from the database given at http://www.medialab.ntua.gr/research/LPRdatabase.html [1]. This database has many categories of images. We use small-sample-vehicle image category, which has sixty nine samples of car images. These samples include various types of images, taken under different illumination conditions.

Two sample experiments for the license plate written in English are presented in Fig. 5.3 and 5.4 for the license plate detection and segmentation stages respectively. In Fig. 5.3(a), the original input images are presented. After the application of edge detection algorithm we get the edge images which are illustrated in Fig. 5.3(b). Our ROID algorithm takes this edge image as the input and detects the ROI as illustrated in Fig. 5.3(c). English license plate segmentation stage is presented in Fig. 5.4. Original license plates are presented in Fig. 5.4(a). At first we convert this image to binary image. Since the background of English license plates are white unlike the Bangla license plates which are black, we invert the converted binary image. This image is illustrated

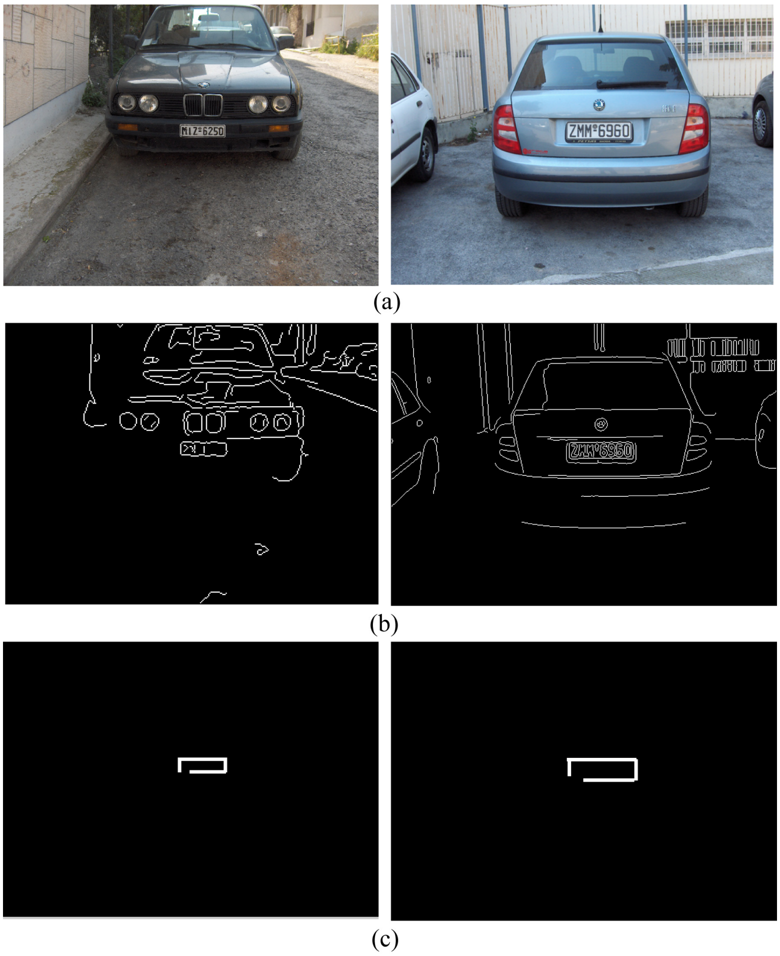


Fig. 5.3: License plate detection: (a) Original input images, (b) Edge images, and (c) After the application of Algorithm ROID.

in Fig. 5.4(b). Fig. 5.4(c) shows the vertical projection of this binary image. We apply our SAT algorithm to segment characters and digits. Segmented characters and digits are illustrated in Fig. 5.4(d).

The results of this experiment are presented in Table 5.9. The success rate of our, SCW [3], Duan et al. [4], and Mashuk et al. [5] schemes in detecting the license plates are 92.7%, 89.7%, 85.3%, and 0% respectively. The algorithm presented in [3] is based on local abrupt changes and a threshold need to be set in trial and error basis. In an image with complex background, many regions may show high magnitude of local abrupt changes other than the license plate region and setting an threshold appropriate for all the input images is impossible. The algorithm presented in [4] counts the number of characters to validate a candidate region as the license plate, which in not independent of the distance from the

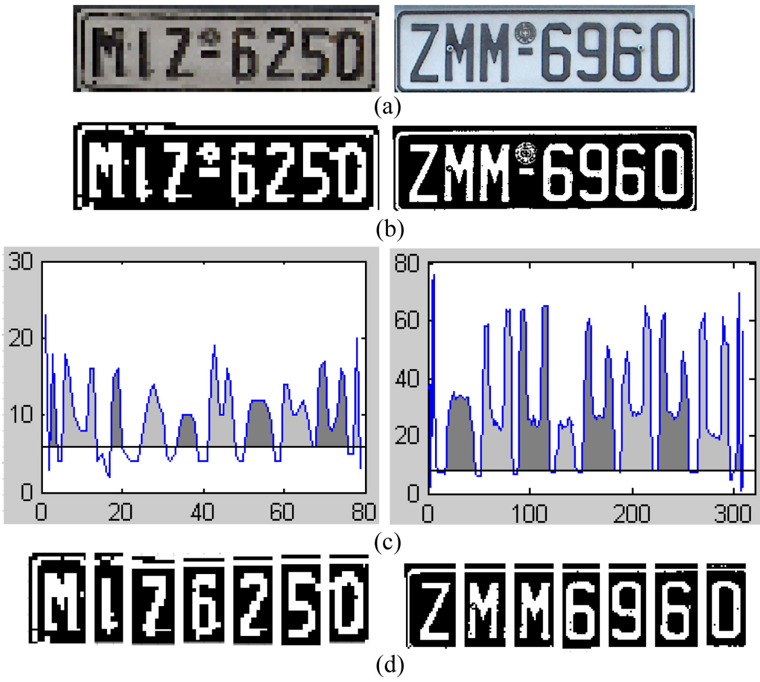


Fig. 5.4: License plate segmentation: (a) Original license plate, (b) Inverted binary images, (c) Vertical projections, and (d) Segmented characters and digits.

camera. The algorithm presented in [5] uses the empirical measurements of the position of the vehicle to locate the license plate. These constraints make these algorithms fail to locate the license plate in some cases, whereas our algorithm is capable of locating the license plate without being limited of these constrains. The success rate of our, SCW [3], Duan et al. [4], and Mashuk et al. [5] schemes in segmenting characters and digits are 100%, 94%, 95.4%, and 75.3% respectively. Our license plate segmentation algorithm performs better than these algorithms because it uses an adaptive threshold, which is capable of segmenting license plate in the presence of noise. The success rate our classifier and the classifiers presented in SCW [3], Duan et al. [4], and Mashuk et al. [5] in recognizing the characters and digits are 89%, 88.5%, 86.5%, and 82.5% respectively. Our classifier performs slightly better than these classifiers because we evaluate the horizontal and the vertical projections of 32×32 image as the feature, which is more capable of capturing distinguishing features of characters of the license plates. The classifiers presented in Anagnostopoulos et al. [3] and Mashuk et al. [5] used input character images of size 12×9 pixels and 16×8 pixels respectively. These sizes are too small to capture the distinguishing features. As a result their

Stages	Methods	No. of images	Correctly detected	Success rate(%)
Detection stage	Proposed scheme	69	64	92.7
	Anagnos. et al. [3]	69	66	89.7
	Duan et al. [4]	69	65	85.3
	Mashuk et al. [5]	69	0	0
Segmentation stage	Proposed scheme	64	64	100
	Anagnos. et al. [3]	66	62	94
	Duan et al. [4]	65	62	95.4
	Mashuk et al. [5]	69	52	75.3
Recognition stage	Proposed scheme	200	178	89
	Anagnos. et al. [3]	200	177	88.5
	Duan et al. [4]	200	172	86.5
	Mashuk et al. [5]	200	165	82.5

Table 5.8: Experimental results for the license plate written in English.

performance is poor compared to ours. Duan et al. [4] uses character images of size 50×50 pixels and capture the features using a window of size 9×9, whose performance is also poor than that of our classifier.

Stages	Methods	No. of images	Correctly detected	Success rate(%)
Detection stage	Proposed scheme	69	64	92.7
	Anagnos. et al. [3]	69	66	89.7
	Duan et al. [4]	69	65	85.3
	Mashuk et al. [5]	69	0	0
Segmentation stage	Proposed scheme	64	64	100
	Anagnos. et al. [3]	66	62	94
	Duan et al. [4]	65	62	95.4
	Mashuk et al. [5]	69	52	75.3
Recognition stage	Proposed scheme	200	178	89
	Anagnos. et al. [3]	200	177	88.5
	Duan et al. [4]	200	172	86.5
	Mashuk et al. [5]	200	165	82.5

Table 5.9: Experimental results for the license plate written in English.

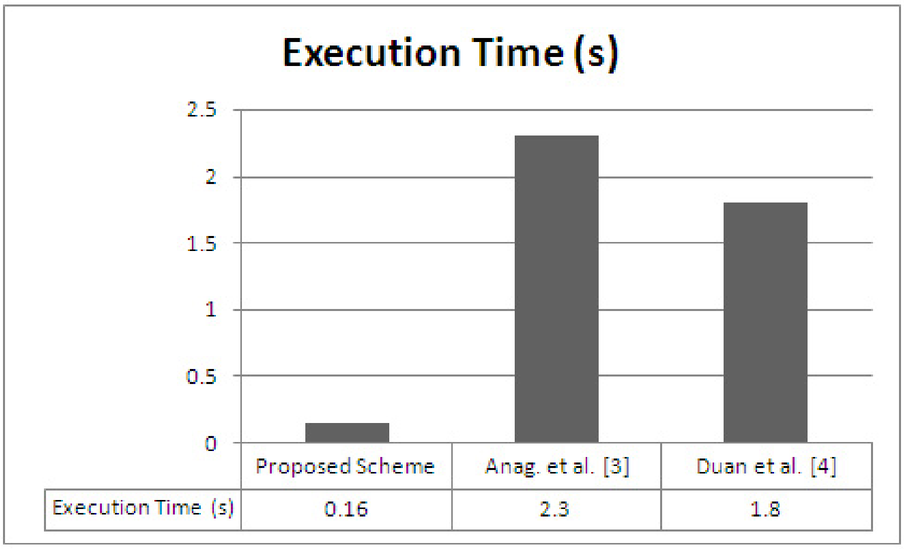


Fig. 5.5: Execution time comparison among the algorithms.

5.6 Execution Time Comparison

Suppose in a practical scenario, vehicles are moving at a speed of $60kmh^{-1} = 16.67ms^{-1}$ towards the camera and the distance between two vehicles is $5m$. Therefore, a real time ALPR system must process an input image in $\frac{5m}{16.67ms^{-1}} = 0.3s$. Our proposed ALPR system takes on an average of $0.16s$ to recognize a license plate in an input image, whereas the ALPR systems presented in [3, 4] take on an average of $2.3s$ and $1.8s$ respectively as shown in Fig. 5.5.

5.7 Summary

The results obtained from our proposed ALPR system are compared with those of the existing ALPR systems. We have compared the results for both Bangla license plates and English license plates. All the experimental results proved that our proposed ALPR system performs significantly better than that of the existing ALPR systems in recognizing Bangla license plates as well as English license plates.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we present a set of algorithms for the recognition of license plate number written in Bangla using three conventional stages of processing. In each stage we developed new techniques and algorithms suitable for recognizing Bangla license plates. At first, we detected the vehicle in the image by using the property of symmetry. We searched the license plate along this symmetry axis. Secondly, We separated the rows containing information in the license plate. We segmented the Bangla characters and digits by using our newly developed algorithm named SAT, which employs horizontal and vertical projections. Use of horizontal and vertical projections are not new in license plate segmentation. Our novelty is that we use an adaptive threshold, which is capable of segmenting joint Bangla characters. Finally, we recognized the extracted characters and digits by using Artificial Neural Network (ANN). We used two different neural networks to recognize the characters and the digits separately. We took 250 images containing vehicle under different conditions such as night, sunny day, and cloudy day. We test our algorithm for these inputs and achieve over 95% success rate. We also compared the experimental results against ALPR systems that were proposed in the literature. We found that our proposed scheme outperform existing algorithms in the recognition of Bangla license plate.

6.2 Suggestions for future work

License plate detection stage of our proposed ALPR system is based on the detection of rectangular boundary of the license plate. However, a license plate may not has a distinguishable boundary if the color of both the vehicle and the background of license plate are same. In this case, our proposed license plate detection algorithm will not be able to locate the license plate. In addition, Bangla language has two overlapping vowels. They are seldom used in the Bangla license plates. Our proposed segmentation algorithm SAT is not capable of segmenting these overlapping vowels. In future work, we intend to improve our license plate detection and recognition algorithms to solve these problems.

In addition, proposed ALPR system can be improved by introducing a number of strategies. Colour information and colour edge detectors can be used to increase the accuracy of locating license plates. Moreover, Decision tree can be employed in the character recognition stage to make early decision about the registration area and type, which will reduce execution time and improve accuracy. In order to apply our proposed scheme in real-time applications more efficiently, algorithms can be implemented in hard-wire and parallel devices, which require a lot of research in these fields.

Bibliography

[1] C. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License Plate Recognition From Still Images and Video Sequences: A Survey," IEEE Transactions On Intelligent Transportation Systems, vol. 9, no. 3, pp. 377-381, 2008.

[2] http://www.platerecognition.info/1106.htm

[3] C. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," IEEE Transactions On Intelligent Transportation Systems, vol. 7, no. 3, pp. 377-382, 2006.

[4] T. D. Duan, T. L. Hong, T. V. Phuoc, and N. V. Hoang, "Building an Automatic Vehicle License-Plate Recognition System," in the proceedings of International Conference on Computer Science: Research, Innovation & Vision of the Future, Can Tho, Vietnam, 2005.

[5] M. S. Mashuk, M. A. Majid, N. Basher, and T. R. Rahman, "Automatic Detection of Bangla Characters in Bangladeshi Car Registration Plates," in the proceedings of Second International Conference on Computational Intelligence, Modelling and Simulation, pp. 166-171, Bali, Indonesia, 2010.

[6] F. Martin, M. Garcia, and J. L. Alba, "New methods for automatic reading of VLPs (Vehicle License Plates)," in the proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, Greece, 2002.

[7] C. Wu, L. C. On, C. H. Weng, T. S. Kuan, and K. Ng, "A Macao license plate recognition system," in the proceedings of 4th International Conference

on Machine Learning and Cybernetics, pp. 4506-4510, Guangzhou, China, 2005.

[8] M. Ben-Ari, “Principles of Concurrent and Distributed Programming”, Prentice Hall, 1990. ISBN 0-13-711821-X. Ch. 16, pp. 164.

[9] X. Shi, W. Zhao, and Y. Shen, “Automatic License Plate Recognition System Based on Color Image Processing,” Springer-Verlag vol. 3483, pp. 1159-1168, New York, 2005.

[10] G. Cao, J. Chen, and J. Jiang, “An adaptive approach to vehicle license plate localization,” in the proceedings of 29th Industrial Electronics Conference, pp. 1786-1791, VA, USA, 2003.

[11] B. Hongliang and L. Changping, “A hybrid license plate extraction method based on edge statistics and morphology,” in the proceedings of International Conference on Pattern Recognition, pp. 831-834, Cambridge, UK, 2004.

[12] P. Wu, H. H. Chen, R. J. Wu, and D. F. Shen, “License plate extraction in low resolution video,” in the proceedings of 18th International Conference on Pattern Recognition, vol. 1, pp. 824-827, Hong Kong, 2006.

[13] R. Zunino and S. Rovetta, “Vector quantization for license-plate location and image coding,” IEEE Transaction on Industrial Electronics, vol. 47, no. 1, pp. 159-167, 2000.

[14] F. Kahraman, B. Kurt, and M. Gkmen, “License Plate Character Segmentation Based on the Gabor Transform and Vector Quantization,” Springer-Verlag, vol. 2869, pp. 381-388, New York, 2003.

[15] C. E. Anagnostopoulos, T. E. Alexandropoulos, S. Boutas, V. Loumos, and E. Kayafas, “A template-guided approach to vehicle surveillance and access control,” in the proceedings of IEEE Conference on Advanced Video Signal Based Surveillance, pp. 534-539, Italy, 2005.

[16] N. Zimic, J. Ficzko, M. Mraz, and J. Virant, “The fuzzy logic approach to the car number plate locating problem,” in the proceedings of Intelligent Information Systems, pp. 227-230, Bahamas, 1997.

[17] J. A. G. Nijhuis, M. H. ter Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema, and M. A.Westenberg, "Car license plate recognition with neural networks and fuzzy logic," in the proceedings of IEEE International Conference on Neural Network, vol. 5, pp. 2232-2236, Perth, Australia, 1995.

[18] T. H. Wang, F. C. Ni, K. T. Li, and Y. P. Chen, "Robust license plate recognition based on dynamic projection warping," in the proceedings of IEEE International Conference on Networking, Sensing, and Control, pp. 784-788, Taiwan, 2004.

[19] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," Pattern Recognition Letters, vol. 38, no. 11, pp. 1961-1975, 2005.

[20] S. Nomura, K. Yamanaka, O. Katai, and H. Kawakami, "A new method for degraded color image binarization based on adaptive lightning on grayscale versions," IEICE Transaction on Information System, vol. E87-D, no. 4, pp. 1012-1020, 2004.

[21] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," IEEE Transaction on Vehicular Technology, vol. 49, no. 6, pp. 2309-2319, 2000.

[22] C. Coetzee, C. Botha, and D.Weber, "PC based number plate recognition system," in the proceedings of IEEE International Symposium on Industrial Electronics, pp. 605-610, New Orleans, USA, 1998.

[23] S. Draghici, "A neural network based articial vision system for license plate recognition," International Journal on Neural Systems, vol. 8, no. 1, pp. 113-126, 1997.

[24] A. Broumandnia and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," in the proceedings of International Conference on Graphics, Vision and Image Processing, Cairo, Egypt, 2005.

[25] Y. Cui and Q. Huang, "Extracting characters of license plates from video sequences," Machine Vision and Application, vol. 10, no. 5/6, pp. 308-320, 1998.

[26] K. V. Mardia and T. J. Hainsworth, "A spatial thresholding method for image segmentation," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 919-927, 1988.

[27] P. Comelli, P. Ferragina,M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," IEEE Transaction on Vehicular Technology, vol. 44, no. 4, pp. 790-799, 1995

[28] Y. P. Huang, S. Y. Lai, and W. P. Chuang, "A template-based model for license plate recognition," in the proceedings of IEEE International Conference on Networking, Sensing and Control, pp. 737-742, Taiwan, 2004.

[29] D. Llorens, A. Marzal, V. Palazon, and J. M. Vilar, "Car License Plates Extraction and Recognition Based on Connected Components Analysis and HMM Decoding," Springer-Verlag, vol. 3522, pp. 571-578, NewYork, 2005.

[30] M. H. ter Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License plate recognition using DTCNNs," in the proceedings of IEEE International Workshop on Cellular NNs Applications, pp. 212-217, London, UK, 1998.

[31] J. Canny, "A Computational Approach to Edge Detection", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, 1986.

[32] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo Vision-based Vehicle Detection", in proceedings of IEEE Intelligent Vehicles Symposium, pp. 39-44, Dearbon, MI, USA, 2000.

[33] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach, for license plate recognition," in the proceedings of IEEE Signal Processing Society Workshop on NNs Signal Processing, vol. 2, pp. 614-623, NSW, Australia, 2000.

[34] V. Franc and V. Hlavac, License Plate Character Segmentation Using Hidden Markov Chains, Springer-Verlag, vol. 3663, pp. 385-392, Germany, 2005.

[35] M. Yu and Y. D. Kim, "An Approach to Korean License Plate Recognition Based on Vertical Edge Matching." IEEE international conference on Systems, Man, and Cybernetics, vol. 4, pp. 2975-2980, Chicago, USA, 2000.

[36] J. Sauvola and M. Pietikinen, "Adaptive document image binarization," Pattern Recognition Letters, vol. 33, no. 2, pp. 225-236, Feb. 2000

[37] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transaction on Systems, Man, and Cybernetics, vol. SMC-9, no. 1, pp. 62-66, 1979.

[38] H. J. Lee, S. Y. Chen, and S. Z. Wang, "Extraction and recognition of license plates of motorcycles and vehicles on highways," in the proceedings of International Conference on Pattern Recognition, pp. 356-359, Cambridge, UK, 2004.

[39] A. Broumandnia, and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," in the proceedings of International Conference on Graphics, Vision and Image Processing, Cairo, Egypt, 2005.

[40] M. M. Alam, and M. A. Kashem, "A Complete Bangla OCR System for Printed Chracters," Journal of Computer and Information Technology, vol. 1 no. 1 pp. 30-35, 2010.

[41] A. A. Chowdhury, E. Ahmed, S. Ahmed, S. Hossain and C.M. Rahman, "Optical Character Recognition of Bangla Characters using neural network: A better approach," in the proceedings of 2nd International Conference on Electrical Engineering (ICEE), Khulna, Bangladesh, 2002.

[42] B. B. Chaudhuri, and U. Pal, "A Complete Printed Bangla OCR System", Pattern Recognition, vol. 31, pp. 531-549, 1998.

[43] A. M. S. Shatil, and M. Khan, "Minimally Segmenting High Performance Bangla Optical Character Recognition Using Kohonen Network," in the proceedings of International Conference on Communications and Information Technology, Dhaka, Bangladesh, 2006.

[44] M. K. Shukla, T. Patnaik, S. Tiwari, and S. K. Singh, "Script Segmentation of Printed Devnagari and Bangla Languages Document Images OCR," International Journal of Computer Science and Technology, vol. 2, no. 2, pp. 367-370, 2011.

[45] Hagan, M. T., and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.

[46] Moller, M. F., "A scaled conjugate gradient algorithm for fast supervised learning," International Journal of Neural Networks, vol. 6, pp. 525-533, 1993.

[47] Battiti, R., "First and second order methods for learning: Between steepest descent and Newton's method," International Journal of Neural Computation, vol. 4, no. 2, pp. 141-166, 1992.

[48] Hagan, M. T., H. B. Demuth, and M. H. Beale, "Neural Network Design," PWS Publishing, Boston, 1996.

[49] S. Haykin, "Neural Networks: A Comprehensive Foundation," 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.