FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

INTELLIGENT PLATE NUMBER RECOGNITION SYSTEM USING SEGMENTIZED METHOD WITH ARTIFICIAL NEURAL NETWORKS

Auwal Salisu YUNUSA

Master's Thesis

DEPARTMENT OF MECHATRONICS ENGINEERING

Dr. Öğr. Üyesi Cafer BAL

SEPTEMBER 2020

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Mechatronics Engineering

Master's Thesis

INTELLIGENT PLATE NUMBER RECOGNITION SYSTEM USING SEGMENTIZED METHOD WITH ARTIFICIAL NEURAL NETWORKS

Author
Auwal Salisu YUNUSA

Supervisor
Dr. Öğr. Üyesi Cafer BAL

SEPTEMBER 2020
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Mechatronics Engineering

Master's Thesis

Title: INTELLIGENT PLATE NUMBER RECOGNITION SYSTEM USING SEGMENTIZED METHOD WITH ARTIFICIAL NEURAL NETWORKS

Author: Auwal Salisu YUNUSA

Submission Date: 29 August 2020

Defense Date: 14 September 2020

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor: Dr. Öğr. Üyesi Cafer BAL
Firat University

Chair: Dr. Öğr. Üyesı Deniz KORKMAZ
Malatya Turgut Özal University

Member: Dr. Öğr. Üyesi Gonca ÖZMEN KOCA
Firat University

This thesis was approved by the Administrative Board of the Graduate School on

.... / .... / 20 ....

Signature

Doç. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled " INTELLIGENT PLATE NUMBER RECOGNITION SYSTEM USING SEGMENTIZED METHOD WITH ARTIFICIAL NEURAL NETWORKS " in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

14 September 2020

Auwal Salisu YUNUSA

PREFACE

Although this research is a master's thesis, that includes many transformations in the field of image processing and artificial intelligence. There are a lot of methods used to perform some image operation to obtained an enhanced image. Recently, the technology in the field of image processing and mobile computers that have found places all in areas of our life, which has speedily developed these studies in this matter has increased day by day. However, we are going to research and develop an algorithm to be able to run on a mobile computer called Raspberry Pi. Therefore, based on this regard, this research contains important originality.

Primarily, I thank the almighty Allah for being able to complete this thesis project with success. I would also like to thank my supervisor Dr. Öğr. Üyesi Cafer BAL whose valuable support has been the ones that help me finished and make it a full-proof success. Also, to the entire staff of the Mechatronic Department of Technology faculty.

My completion of this research thesis could not have been accomplished without the support of my friends, especially Ahmad Muhammad, Abdullah Abba Abdullah, Salim Jibril Dambatta, and my little daughter Khadija Auwal Salis Yunusa (Janan). Thanks to my parents as well, for their financial support and prayer during my studies. Finally, to my loving, caring, and supportive wife, Ummi Abbas Sani; my deepest gratitude. Your encouragement when things got high is much appreciated and duly noted.

Auwal Salisu YUNUSA
ELAZIG, 2020

TABLE OF CONTENTS

ABSTRACT

INTELLIGENT PLATE NUMBER RECOGNITION SYSTEM USING SEGMENTIZED METHOD WITH ARTIFICIAL NEURAL NETWORKS

Auwal Salisu YUNUSA

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Mechatronics Engineering

September 2020, Page: ix + 63

This thesis research is concerned with the field of recognition of an image that used an automated plate number recognition system (APNR). In this thesis, the proposed system for the recognition of vehicle license plate numbers is designed with machine learning algorithms. The APNR system is an image processing technology that uses the license plate number to recognize/identify a vehicle. One of the objectives is to design an efficient, effective automatic vehicle identification system by using vehicle license plate number. One of the techniques that can be used to achieve this goal is the optical character recognition (OCR) system used for character recognition. The system was designed and simulated in MATLAB environment and its performance was tested on real plate images. It was observed that the system developed in experimental studies effectively detects and recognizes license plate images. The system captured the vehicle license plate images and send them to the computer where they are processed and identified. Different algorithms such as localization, segmentation, and OCR are used in the processing of the captured images.

Artificial neural networks are a powerful tool that can capture and characterize complex input / output relationships. The reason behind the development and use of artificial neural networks technology stems from the need to develop an artificial system that can perform intelligent tasks similar to those performed by the human brain. Feed-forward back propagation networks are created by generalizing the gradient descent learning rule to multilayer networks and nonlinear differentiable activation functions. Back propagation is a method for calculating the gradient for nonlinear multilayer networks. This method has many variations based on other standard optimization techniques such as Newton and conjugate gradient methods. Properly trained back propagation networks tend to give reasonable responses to input they have never seen. The variations of these back propagation networks are used for character recognition and comparative results on the performance of different algorithms are presented in this thesis.

Keywords: Vehicle plate, Character Recognition System, Artificial Neural Network

ÖZET

Yapay Sinir Ağları ile Segmentasyon Metodu Kullanılarak Akıllı Plaka Numarası Tanıma Sistemi

Auwal Salisu YUNUSA

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Mekatronik Mühendisliği Anabilim Dalı

Eylül 2020, Sayfa: ix + 63

Bu tez çalışması, otomatik plaka tanıma sistemi kullanan bir görüntü tanınma alanıyla ilgilidir. Bu tezde, araç plakalarının tanınması için önerilen sistem, makine öğrenmesi algoritmaları ile tasarlanmıştır. APNR sistemi, araç tanıma/tanımlama için plakayı kullanan bir görüntü işleme teknolojisidir. Amaçlardan biri, araç plaka numarasını kullanarak verimli, etkili bir otomatik araç tanımlama sistemi tasarlamaktır. Bu amacı gerçekleştirmek için kullanılabilecek tekniklerden biri karakter tanıma için kullanılan optik karakter tanımadır. Sistem MATLAB ortamında tasarlanmış ve benzetimi gerçekleştirilmiş ve performansı gerçek plaka görüntüleri üzerinde test edilmiştir. Deneysel çalışmalarda geliştirilen sistemin araç plaka görüntülerini etkin bir şekilde algılayıp tanıdığı gözlenmiştir. Sistem, araç plaka görüntülerini alarak işlenip tanılanacağı bilgisayara aktarmaktadır. Elde edilen görüntülerin işlenmesinde, localizasyon, bölütleme ve optik karakter tanıma gibi farklı algoritmalar kullanılmaktadır.

Yapay sinir ağları karmaşık girdi / çıktı ilişkilerini yakalayabilen ve karakterize edebilen güçlü bir araçtır. Yapay sinir ağları teknolojisinin geliştirilmesinin ve kullanılmasının ardındaki neden, insan beyninin gerçekleştirdiklerine benzer akıllı görevleri yerine getirebilecek yapay bir sistem geliştirme ihtiyacından kaynaklanmaktadır. İleri beslemeli geri yayılım ağları eğim düşme öğrenme kuralının çok katmanlı ağlara ve doğrusal olmayan türevlenebilir aktivasyon fonksiyonlarına genelleştirilmesiyle oluşturulmuştur. Geri yayılım, doğrusal olmayan çok katmanlı ağlar için eğimin hesaplandığı bir yöntemdir. Bu yöntem, Newton ve eşlenik eğim yöntemleri gibi diğer standart optimizasyon tekniklerine dayanan çeşitli varyasyonlara sahiptir. Düzgün bir şekilde eğitilmiş geri yayılım ağları, hiç görmedikleri girdilere karşı makul yanıtlar verme eğilimindedir. Bu geri yayılım ağlarının varyasyonları, karakter tanıma için kullanılmış ve bu tez kapsamında farklı algoritmaların performansı üzerine karşılaştırmalı sonuçlar sunuluştur.

Anahtar Kelimeler: Araç Plakası, Karakter Tanılama Sistemi, Yapay Sinir Ağı

LIST OF FIGURES

Page

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

Abbreviations

ANN : Artificial Neural Network
ANPR : Automatic Plate Number Recognition
CCD-camera : Charge-Coupled Device Camera
CPU : Central Processing Unit
CSI : Camera Serial Interface
DFT : Discrete Fourier Transform
DSI : Display Serial Interface
GPIO : General Purpose Input/output
GPU : Graphical Processing Unit
HD : High Definition
HDMI : High Definition Multimedia Interface
LCD : Liquid Crystal Display
LP : License Plate
LPR : License Plate Recognition
MLPNN : Multiple Layered Perceptron Neural Networks
OCR : Optical Character Recognition
OS : Operating System
PNR : Plate Number Recognition
RGB : Red Green Blue
SD : Secure Digital
SOC : System on Chip
USB : Universal Serial Bus

1. INTRODUCTION

Automation is one of the most frequently used terms in the area of intelligent transportation systems [1]. Nowadays, everything tends to moves towards automation. Unlike in the old days, where people used to deal with everything manually [2]. For instance, at the gate entrance, a person has to open a gate manually, access the vehicle visitor, before allowing the vehicle to pass through the gate. This process requires time consumption and manpower to open the gate, check the vehicle, authorize it to pass, and later close the gate. With the invention of remote-controlled garages, this lead to a great impact on consumers lives very easier by pressing the push button. However, with the recent advent of technology, the lives of our consumers become very easier, thus the system would automatically open and close the gate without the need of humans to press the push button.

This research work aimed to propose a design of an algorithm that detects a license plate number system using low-cost embedded system [3]. Considering that license plate recognition is a current research area in image processing and computer vision applications, also an essential stage in traffic intelligent transportation systems [4]. APNR also has been employed by many countries by various applications such as automatic speed control [5], traffic control [6], tracking stolen cars, [7], and toll collection systems for improving the traffic control system [8,9]. This process of recognizing the license plates results in delays by human involvement. To automate the processing from a mass surveillance camera is quite challenging, the system needs to deal with a different angle, illumination, distance, and resolution [10].

The system captures and analyzes the license plate image and recognizes the characters of the license plate. This research uses a built-in computer, often referred to as a Raspberry Pi processor. The Raspberry Pi is a bank card-like computer card developed by the Raspberry Pi Foundation in the UK to teach and learn basic computer science in schools. It has Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor (Firmware consists of several "Turbo" modes so user can try overclocking up to 1 GHz, warranty), first 256MB RAM and with Video Core IV GPU. It also uses an external SD card for booting and long-term storage, and the operating system uses a network time server or prompts the user for clock information during booting for access of to the time and date for file time and date stamp. This work presents the development of a prototype for characterization and extraction of license plates using a neural network placed on Raspberry-pi [9].

1.1. Problem Description

There are a lot of recognition systems used image processing methods and many mobile personal computers to be able to run the large size of algorithms on it. The license plate number recognition system uses for many practical purposes like security systems, automation systems, etc. It is important to make with less costly hardware as a portable: therefore, an algorithm to run on less costly hardware is an important requirement.

1.2. Motivation of The Study

An algorithm of the image processing is going to develop, and to recognize the license plate numbers by using intelligent methods. However, the method developed is going to be able to run on the lower performed hardware. To develop and simulate the algorithm that is going to used MATLAB program.

1.3. The Aim of Study

Recently the technology in the field of personal and mobile computers that have found places all in areas of our life has speedily developed and the studies, in this matter has increased day by day. Therefore, we are going to research and develop an algorithm, which could run on a mobile computer called Raspberry PI, etc. so that the algorithm is going to recognize the plate number using intelligent methods. This thesis proposes a license plate number recognition.

1.4. Related Works

This chapter is the analysis of related research on image processing using neural network embedded on to the Raspberry Pi. Moreover, some various studies published in this area of study analyzed, their outcomes as well as missing gaps are outlined. The last stage of this chapter talks about the originality of this study.

The use of intelligent systems has brought a tremendous effect on our daily lives. Thus, most of the companies are promoting computer vision research are now embedded into most of the intelligent system to capture physical images and process it.

This work aim is to design and implement an efficient plate number recognition system with an intelligent device which operates in real-time. A desired licensed number plate for recognition system used digital image processing techniques for detecting and recognizing characters of the license plate number and produced an output result as characters as a string. The study of the algorithm and its development focused on real-time image processing for plate number localization, segmentation, and character recognition.

The study in [11] proposed using of Raspberry pi to automatically recognize license plates. Using the Optical Character Recognition (OCR), the study extract license plate from the camera images. The captured image is processed by character segmentation, and authenticated by the Raspberry Pi. Although our algorithm is quite different, the study is similar to the approach used in this study. The experimental results showed 96% accuracy. Other studies that proposed using the Raspberry pi are reported in [11-13].

Raspberry pi in [11] is used as a low-cost non-instructive option in the project, which can be used for traffic data collection. In addition, Raspberry pi HD high-resolution camera module was in analyzing the moving object presence. However, after the detected object number, class and size are evaluated, the properties of the motion vector ware sent to the server node using the radio frequency transceiver. The use of video signal processing and analysis aiming on low power consumption and system consistency was proposed.

The work in [14] proposed real-time effective systems for detecting the presence of obstacles on vehicle tracks. The pi camera was used for a sensor in object recognition and image gaining in the project. The vision-based techniques for plate numbers was used for detection purposes in the study and because of its gain an interest in the area of intelligent system. The system has the ability for detecting the objects that are visible of the vehicle through activating the pi camera component. Moreover, the camera for the USB connects to the board of Raspberry pi, that can serve for the base of unit, and also the monitors of the web camera. This process is added to the picture in the panel. The Pi camera traces the picture and this process is achieved using library of OpenCV to find the edges of the detected picture image.

In [15] proposed an image processing method that uses a visual flow, whereby the technique can run faster by the use of the Raspberry pi processor. This technique show it can perform better by approximating the actual position and direction of the displayed object. Python OpenCV is used in set up the experiment, whereby the algorithm works on an embedded Linux Raspberry pi.

Abaya et al. proposed a project of an embedded security camera by the used of Raspberry pi. Has the ability to the captured picture at night. This system first detects human and smoke then send the detected image to the electronic mail through Wi-Fi network. It behaves like an intruder when detects the problem [16].

The author in [17] Presented a rapid intelligent method based on its connection with (OCR), with multiple layered perceptron of artificial neural networks (MLPNN) models, in which Pakistan vehicles license plate image recognized in conjunction with color region. The algorithm is based on real time feature for the different font style and colored license plate localization as well as recognition. Image can be enhancing by detecting its white region by applying an optimal threshold. The performance analysis of the system was conFigured by localization method and recognition method by using the proposed algorithms. For OCR based system, the overall performance time was 2.44 and for MLPNN based system, it was 2.74.

In [18], the neural network-based method perceptron is trained by setting a few intelligent rules with a sample set. One of the problems with neural networks is the training a perceptron which is quite difficult and involves massive sample sets to train the network. If the system is not trained in a suitable method, scale and orientation invariance may not be addressed.

In [19] clustering methods using fuzzy C-mean and K-mean were presented. The two methods were for both automating Grabcut and improving segmentation accuracy. Besides, a method called "SOFM clustering" was employed for the automation of the Grabcut which is the additional to the user interface for the requirement by the old algorithm. This Grabcut is a global optimal result that performed at the boundary region between the object and background in order to make Grabcut global optimal solution.

In [20], an efficient plate number extraction algorithm based on geometric properties and multi-level threshold is proposed. This type of algorithm can be used to identify, segment the vehicle license plate while capturing the vehicle image. Segmentation algorithm for extraction has 75% localization accuracy. There were some difficulties with license plate detection, with poor image quality.

The author in [21] approaches the problem of license plate number recognition by using a group of three multilayer perceptron based on a mixture of expert's architecture for each segmented character. Edge detection was used to detect the license plate, while the character segmentation was obtained through vertical projection. The accuracy for localization is 95.39% and the total accuracy of the system is 92.45%.

The author in [22] proposed an algorithm called the low-complexity histogram modification technique. Usually it does not require any splitting operation, instead it deals with histogram increments to realize black and white extensions and adjusts the level of enhancement adaptively to the system. Thus, the dynamic range of the system is better utilized while meeting the requirement for noise visibility and natural appearance.

Several factors can affect the license plate number recognition accuracy, such as the size of the characters in number plate, light and weather conditions, resolution of the camera, and so on. In analyzing the vehicle plate image, MATLAB is chosen as one of the very powerful software, and preferable by many researchers to be used, in the designing and development of algorithms for the image processing.

1.5. Overview of The Study

The thesis discusses in detail the introduction, related research, design methodologies, and implementation details of said research project (plate number recognition with intelligent device). It is aimed at providing the complete technical details needed to fully re-implement the stated research, and lastly conclusions and recommendations.

The thesis is organized as follows:

- **Chapter one:** Covers the introduction of image processing and Raspberry Pi systems, the problem description, motivation of the study, aim of the study as well as the overall overview of the entire study.
- **Chapter two:** A review of previous works that contribute to this research. A brief literature review of the techniques used in the license plate number recognition.
- **Chapter three:** A description of the basic methods which are used in this thesis research. License plate number recognition using a neural network with Raspberry-pi.
- **Chapter four:** Experimental analysis and the results of the techniques used in this thesis research are discussed.
- **Chapter five:** Conclusion of the study, and recommendation, suggestions of our work future work are presented.

2. CONCEPT OF DIGITAL IMAGE PROCESSING

In this chapter are given informs about some of the image processing methods, describing a digital image, the methods of image enhancement, spatial filtering, image segmentation, and morphological image processing.

2.1. Digital Image

The concepts of digital image processing can be defined as the method of processing digital images using a digital computer. Digital image can also be represented as a two-dimensional function $f(x, y)$, where x and y are the coordinates of spatial (plane). The amplitude of f at any point of the image is known as grey level or intensity level of the image. If the values x, y, and amplitude are finite and discrete, this type of image obtained is called a digital image. At certain point of the image, the position and intensity value at that point of image is called pixel [11].

2.1.1. RGB (Color) Images

An R, G, B image, also known as true color image is the mixture three standard of red, blue and green color components of pixel image. The red, green and blue has a dimension of $m * n * 3$ data array that describe the color components of each pixel in an image. Images can be of class, unit8, uint16, single or double are arrays of real color. If the color image is of class single or double, then it is color components value is in the range of 0 and 1. However, if a pixel color component is at position 0, it is displayed as black, and if it is a pixel with color component is 1, can be displayed as white as illustrated in Figure 2.1.

Figure 2.1. Color Image of Class Double. [12]

2.1.2. Grayscale (intensity) images

Grayscale image is a type of image that has shades of gray colors. This type of image has three color space of *R, G,* and *B* equal to the intensity red, green and blue components. Therefore, it only required to identify single value intensity of each pixel, which is not as same case as *R, G*, *B* color. Grayscale intensity needs an 8-bit integer to store values of each pixel. Figure 2.2. is the gray scale image of class Uint8.

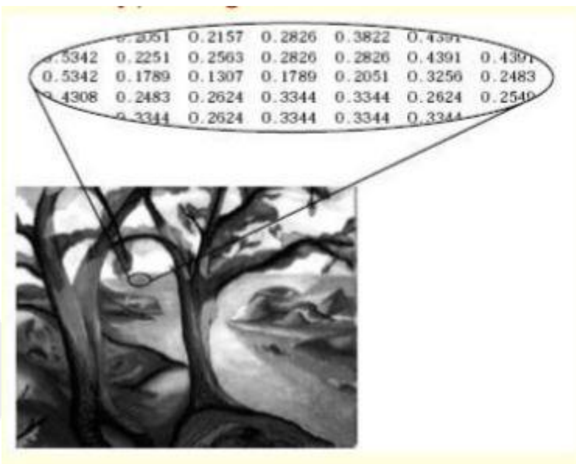

Figure 2.2. Grayscale Image of Class Uint8 [24]

2.1.3. Black and white (Binary) images

Each pixel has one of two grey levels either black (0) or white (1). It requires 8-bit to store a greyscale value as shown in Figure 2.3.

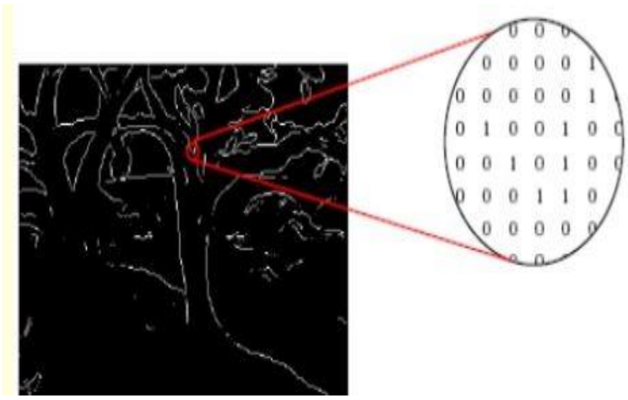

Figure 2.3. Binary Image of Class 8 [24]

2.2. Sampling and Quantization

A digital image may be nonstop around its amplitude and x, y, coordinate. Converting the image into digital form is achieved by sampling the function image in both amplitude and coordinates. To digitizing the coordinate and amplitude of a digital image are called sampling and quantization [11].

The results of sampling and quantization give a matrix of real numbers. If the sampled image $f(x, y)$, then the subsequent digital image would have has M rows and N columns. These coordinates (x, y), are called discrete quantities.

Figure 2.4. shows how a digital image is presented for image coordinate convection a

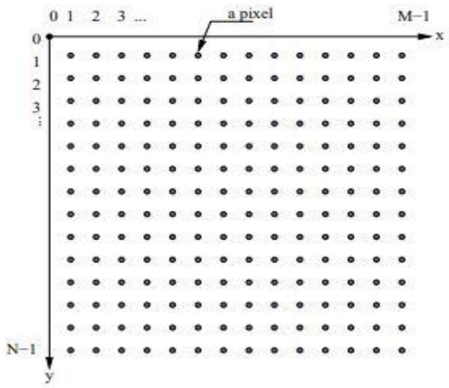

Figure 2.4. Coordinate Convention of Digital Image [25].

The complete $M * N$ digital image is presented in the compact matrix form as shown in Figure 2.5.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figure 2.5. Compact Matrix of Digital Image [13].

The process of digitization requires decisions about some values, M, N, and L which is the number of discrete grey levels that allowed for each number of pixels. These values of M and N must be positive numbers. Moreover, based on capacity, performances and hardware sampling consideration, the total of grey levels is the integer power of 2 as expressed in equation 2.1.

$$L = 2^k \tag{2.1}$$

where the discrete levels are assumed to be equally space, and have an integer in the interval of *[0, L-1]*.

2.3. Neighbourhood

A pixel p is at coordinate x, y, has four horizontal and vertical neighbors at the following coordinate $(x-1, y)$, $(x+1, y)$, $(x, y+1)$, and $(x, y-1)$. These sets of pixels are called 4-neighbors of pixel p, it can be denoted as $N4(p)$. Moreover, if diagonal pixels i.e., $(x-1, y+1), (x-1, y-1), (x+1, y+1)$, and $(x+1, y-1)$, are all included in this set, then the whole sets are called 8-neighbors of p, and is denoted by $N8(p)$ [11].

2.4. Image Enhancement

Is a technique used in digital image processing, this image enhancement is one of the most challenging issues used in image processing. The idea of image enhancement is to develop a precise image so that the its output would be more appropriate for a certain application than that of the original image. These techniques emphasize certain features of image which includes borders, contrast and edges, thus making the graphic representation more useful for further analysis. This procedure does not rise the intrinsic data content of the image, rather the dynamic rage of the selected features is increase so that can be easily detected. The techniques of image enhancement are of two types, frequency and spatial domains [14].

2.4.1. Spatial Domain Enhancement Methods

Spatial field methods are applied to the image plane itself, and these operations rely on direct manipulation of the pixels of an image. These processes are formulated as shown in equation 2.2.

$$g(x, y) = Tf(x, y) \tag{2.2}$$

The value of g is the output image, and the value of f is the input image, and the value of T is an operation on f that describe over some neighborhood of (x, y).

2.4.2. Frequency Domain Enhancement Methods

Such methods are used to enhance an image $f(x, y)$ by wrapping the image with a linear and position invariant operator. This 2D image convolution is performed in frequency domain by discrete Fourier transform, DFT.

2.5. Enhancement by point processing

These processing techniques are based only on the intensity of single pixels of an image. These techniques are based on either intensity transformation or histogram processing.

2.5.1. Image Negatives

This is one of the simplest methods in image transformation [15]. It is useful in many applications such as negatives of digital images, viewing medical images, and recognition of the license plate. A gray level image is a negative image in the range of range *[0, L-1]* is obtained by negative transformation as shown in equation 2.3.

$$g(x, y) = L - f(x, y) \tag{2.3}$$

The value of L is the maximum image intensity value. Whereas, a negative image is an inversion of the original image, whereby a light area may appear dark and dark area appear light. Figure 2.6 illustrates the example of the negation of an image. The purpose of this techniques is used in such a way that, each value of the input image can be subtracted from $L - 1$ and link to the output image.

Figure 2.6. Original Image [27]

In an 8-bit environment, the actual number of grey levels in the image is 256, when applied to the above equation it becomes $S = 256 - r$. Therefore, each value is subtracted by 255 and resulting in lighter pixels becoming dark and the darker image becomes light. Hence the image negation can be illustrated in Figure 2.7.

Figure 2.7. Image Negative [27]

2.6. Spatial Filtering

Spatial filtering is another method in image enhancement, in which sub-image with values (coefficient) is referred to as kernel, filter, mask, or window, which are convolved over the entire image. Moreover, the size of the kernels is the one to determine the number of neighboring pixels that influence the coefficient at (x, y). This sliding process is ran across the entire digital image, pixel by pixel. Then, at each point (x, y), the output response of the filter is weighted using the predefined relationship. Figure 2.8 illustrates the simple 3 * 3 kernel with coefficient w.

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

Figure 2.8. 3*3 Kernel with Coefficient w [23, 28]

However, linear filtering, the output response of R, when using 3*3 kernel is based on given equation $R = w(-1,1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots\ldots\ldots\ldots + w(0,0)f(x,y) + w(1,0)\, f(x+1,y) + w(1,-1)f(x+1,y-1)$, where, $w(s,t)$ is the kernel. Moreover, when the coefficient $w(0,0)$ matches with $f(x,y)$ specifying the kernel is positioned at point (x,y), when the calculating of the sum-product is taking place. However, when using a kernel of size $m * n$, an assumption that $m = 2a + 1$ and $n = 2b + 1$ is made, where a and b are non-negative integer [11].

In general, the response of any linear filtering using a kernel of size $m * n$ is given by equation 2.4

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x+s,y+t) \tag{2.4}$$

$$a = \frac{m-1}{2} \tag{2.5}$$

$$b = \frac{n-1}{2} \tag{2.6}$$

where a and b are the coefficient of equation 2.4. This equation has to be applied to completely filter an image, for all values of $x = 0,1,2 \ldots M-1$ and $y = 0,1,2 \ldots N-1$.

2.6.1. Histogram

The histogram of an image is used to largely show the distribution of gray levels in the image. Also, an image's histogram provides a general description of their appearance. Also, the shape of the histogram of any image will provide useful information about the likelihood of contrast stretching. Figure 2.9 is a gray scale image of an 8-bit and 15 pixels' distributions at certain intensity values that range from 0 to 250. However, it can be seen from the graphic in Figure 2.10, that most of the pixels in the image are in the following range. 115 - 182 [16].

Figure 2.9. Grayscale Image [30]

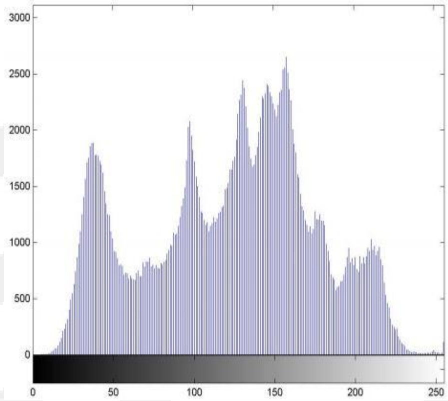

Figure 2.10. Histogram of the Grayscale Image [30].

2.6.2. Histogram Equalization

Histogram is a method that can be used for altering image intensities in order to improve the contrast of the image. The algorithm of histogram equalization is illustrated below.

Let $r_k, k = 1, 2, \dots, m$ to be the intensity of the image, and $p(r_k)$ be its normalized histogram of the function. Therefore, the intensity transformation function for histogram equalization are illustrated in equation 2.7 as

$$T(r_k) = \sum_{j=1}^{k} P(r_k) \tag{2.7}$$

Therefore, we can add the values of the normalized histogram function from *1* to *k* value, so that we can find where the intensity r_k will be mapped. The range of equalized images should be in the interval of [0, 1]. Figure 2.11 illustrated the grayscale image and its histogram equalization.

Figure 2.11. Contrast of an Image [30].

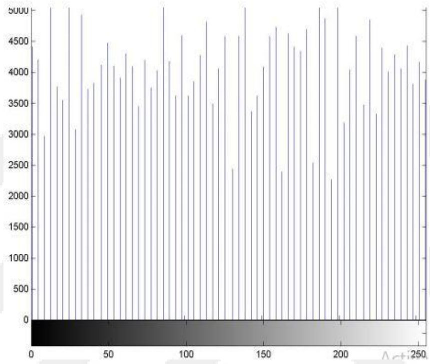

Figure 2.12. Histogram Equalization of Image [30]

2.7. Image Segmentation

This is the process of partition and image or dividing an image into segments that are different in certain aspects of some color, features. Each region of the image is a collection of pixels. Image segmentation can be applicable in so many ways, in which some of the methods used similarity detection, dissimilarity detection, or histogram-based method [17].

2.7.1. Edge Detection

Edge detection is important steps in artificial intelligent and computer vision practice, it is necessary to specify the actual edges to get the best results from the matching process because it is very important to select the edge detectors that are suitable for the best application. There are many types of edge detection algorithms such as Canny, Robert, Sobel, and Prewitt edge detection algorithm. All these algorithms, canny and Sobel edge detection algorithm has higher efficiency compared to others. This research used Sobel edge detection algorithms [17]. The Sobel operator usually approximate the gradient by using the kernel. This process approximates the first derivative in each pixel direction. The Sobel operator is used to detect edges in both vertical and horizontal directions and later combine the information to form a single metric. The Sobel filter kernel of *x* and *y*-direction is illustrated in Figure 2.13 and Figure 2.14.

-1	0	+1
-2	0	+2
-1	0	+1

Figure 2.13. X- directional Sobel Filter Mask [32]

+1	+2	+1
0	0	0
-1	-2	-1

Figure 2.14. Y- directional Sobel Filter Mask [32]

The masks in Figure 2.13 and Figure 2.14 are used to achieve spatial filtering. A mask generates a set of $S = \{r_1,\ r_2\}$ representing the response of the filter in each pixel of the rendered image. The magnitude $|\bar{r}|$ of the edges is computed using equation 2.8

$$|\bar{r}| = \sqrt{r_1 + r_2} \tag{2.8}$$

in case to find the direction of the edge, then

$$d = arctan\frac{r_1}{r_2} \tag{2.9}$$

Figure 2.15. Grayscale Image [33]

Figure 2.16. Image after applying Sobel operator [33]

2.8. Thresholding

Thresholdings are effective methods used for image segmentation. It is method in which each value of pixels would be compare to threshold value T. If the intensity value of the pixel equal to or less than the threshold value. This value can be changed from 1 or 0 based on the application as reported in [18]. If the threshold is less than the pixel value, they assigned the other value. Moreover, this process is an image analysis that isolates an object by converting grayscale images to binary image, as illustrated in the equation of $T\ (x, y)$:

$$T(x,y) = \begin{cases} 1: & f(x,y)\ > T \\ 0: & f(x,y)\ \leq T \end{cases} \tag{2.10}$$

where $f(x, y)$ is the pixel intensity of image at a given point $f(x, y)$. the operation is carried out based on two processes. Global and Local thresholding, if the operation depends on the whole image, is called global thresholding, while if the operation depends on other properties of pixels' neighborhood, its local threshold. An example is the average intensity values in this neighborhood. Moreover, if the local threshold is selected individually for each pixel or group of some pixels, then the operation is called adaptive thresholding [19]. In this research, the method used to determine the threshold value of the license plate number is global thresholding which is one of the simplest among other thresholding.

The method performance success entirely depends on how good the histogram can be partitioned. Since segmenting using some interval values say T results in undesirable output result. A better method should be approached first in selecting the correct threshold value estimate that would be used to segment the image. This method splits the pixels in two groups of modules.

$D1 = \{0, 1, 2, \dots t\}$ and $D2 = \{t, t+1, t+2, \dots L-1\}$, where t is the threshold value. Moreover, these classes correspond to objects of interest, or foreground object, and background in an image. Their average gray level values μ_1 and μ_2 for the pixels are then computed, which is the new threshold value can be calculated using equation 2.11.

$$T = \frac{1}{2}(\mu_1 + \mu_2) \tag{2.11}$$

Repeat the steps described above, till the difference in equation 2.11 is continuous iteration lesser than the predefined parameter $T_{D.}$

Figure 2.17. Grayscale Image [35] and [36]

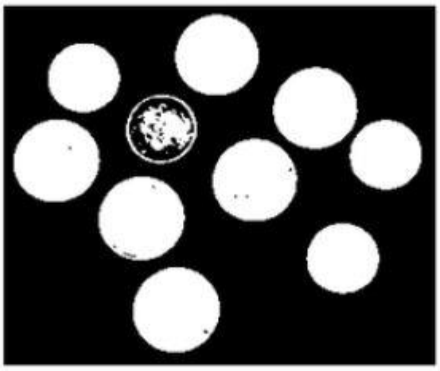

Figure 2.18. Result after applying global thresholding [36]

2.9. Morphological Image Processing

This process is based on mathematical morphology, which takes its name from the biological morphology concepts that deal with the forms and structure of plants and animals. In image processing, morphological image processing is used for extracting the image components related to features such as skeletons, borders and convex body for representation and explanation of image shapes. Mathematical morphology uses set theory, which is a very powerful method for image processing problems [20].

The set representing the objects of an image in mathematical morphology, all black pixel of an image sets system can be completed morphological definition. Moreover, in binary image sets there are members of the 2D integer space of Z^2, where each element consists of the coordinates of the black pixels of an image [21]. In this research, we consider some concept of morphological image processing such as dilation, erosion, opening, and closing, with *A* and *B*, are sets of Z^2, and B is the structuring elements.

2.9.1. Dilation

Dilation operation is one of the basic fundamental morphological image processing. However, the expansion of *A* by *B* is the set of all displacements such that *A* and *B* has overlap by at least one element. Some of the extension applications are used to repair intrusions enlarge objects. Dilation can also be defined based on the equation 2.12

$$A \oplus B = \{z | [(\hat{B})z \cap A] \subseteq A\} \tag{2.12}$$

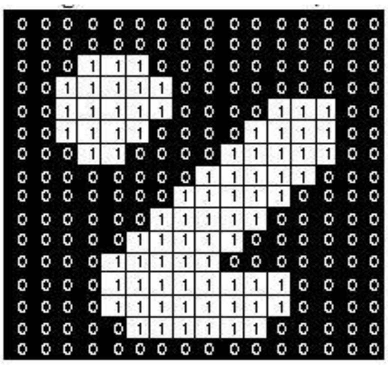

Figure 2.19. Original Image [37]

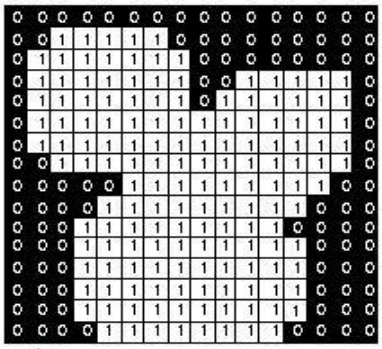

Figure 2.20. Dilated image using square 3*3 structural element [37]

2.9.2. Erosion

Erosion operation is another basic fundamental morphological image processing, the basic operator effect on binary image is to remove pixels on the boundary image. Therefore, erosion of A by B is the set of z points such that B surrounded by z is located at A. some of the applications of erosion are for enlarging holes, also can be used to split apart joined objects. Erosion can be defined based on the equation 2.13

$$A \ominus B = \{z \mid (\hat{B})z \subseteq A\} \qquad (2.13)$$

These two operations are the basic to morphological processing, although most of morphological algorithms are based on these two simple processes.

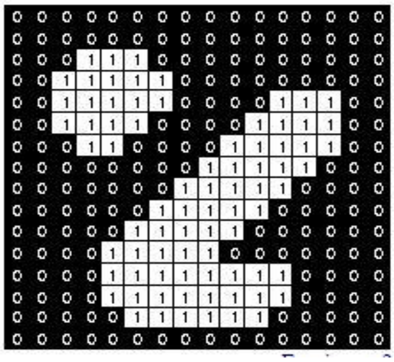

Figure 2.21. Original Image [38]

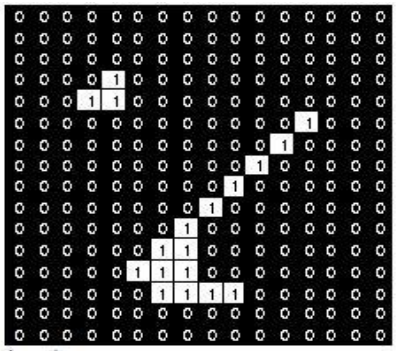

Figure 2.22. Eroded image using square 3*3 structural element [38]

2.9.3. Opening

Opening is another important morphological operation, like dilation and erosion, opening involve or consists of performing dilation after erosion. This operator is used to smooth the contours of an object, as well as to eliminate thin protrusions. Thus, the span can be shown in equation 2.14.

$$A \circ B = (A \ominus B) \oplus B \tag{2.14}$$

Hence, the opening of A by B is the erosion of A by B, and follow by the expansion of B as an output result.

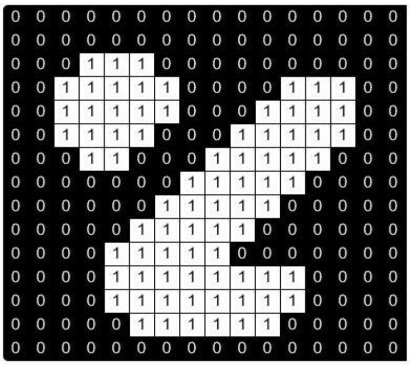

Figure 2.23. Original Image [38]

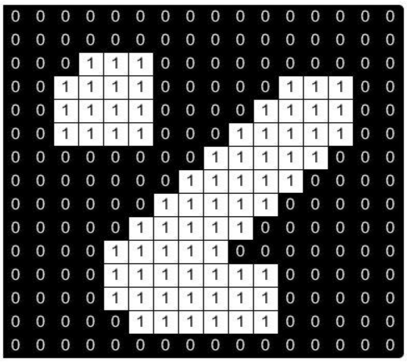

Figure 2.24. Opening image using square 3*3 structural element [38]

2.9.4. Closing

Closing like opening, is another important morphological operation, closing consist of performing erosion after dilation. This operator used to fill some gaps in contour, also used to removed small holes. Thus, closing can be illustrated as in equation 2.15.

$$A \bullet B = (A \oplus B) \ominus B \tag{2.15}$$

Similarly, A closes with B, A expands with B, and then the output result is the erosion of B.

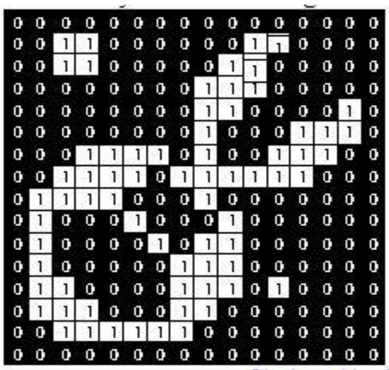

Figure 2.25. Original binary image [38]

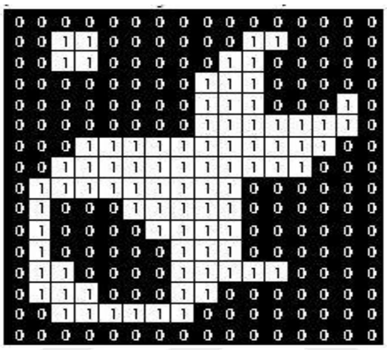

Figure 2.26. Closing with 3*3 structuring element [38]

3. ARTIFICIAL NEURAL NETWORK

A lot of research working on artificial neural networks (ANN) gone far in recent years [22], defined a neural network as an immensely distributed processor, in which its made of simple processing units, and have an expected learning for putting empirical knowledge and formulating. Also [23], describes the neural networks as an imitation of the human brain. A human brain has a lot of ability to adapt and learn new things. A human brain is one of the most amazing abilities in analyzing unclear and incomplete, also fuzzy information, and make its final decision out of it.

3.1. Brain Neuron

The human brain consists of cells called neurons, these neurons (cells) are interconnected and form a network. The human brain has about 1011 neurons and 10,000 connections to each other. An artificial neural network is a copy of a natural neural network that the artificial neurons are linked in a similar pattern as the human brain network. These connections of the network and the strengths of the individual synaptic establish a function called network. Figure 3.1 illustrates the human brain.

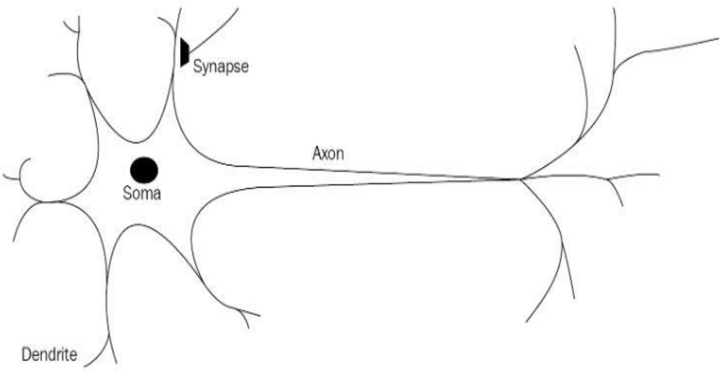

Figure 3.1. A biological Neuron [40]

Based on Figure 3.1, a biological neuron consists of a cell body, dendrites, and axon. A dendrite is used in receiving a signal from the neurons, while an axon is capable of conveying the signals generated of the same neuron to other connected neurons [23].

Artificial neural is similar to the biological neuron, which made up of the following:

The input layer used to accept input features, by providing information from the outside to the network. They only need to pass the information to the other layer, no computation is taken place at this node. While the output layer, used to carries the information learned by the network to the output. The hidden layer performs all calculations on the structure and properties are provided into the input layer and thereby transmits the result to the output layer.

An activation function is used to make a decision on which neuron should be activated by calculating its weighted sum and also adding bias with it. Thus, its main function is to add a nonlinearity to the output of neurons as shown in Figure 3.2.

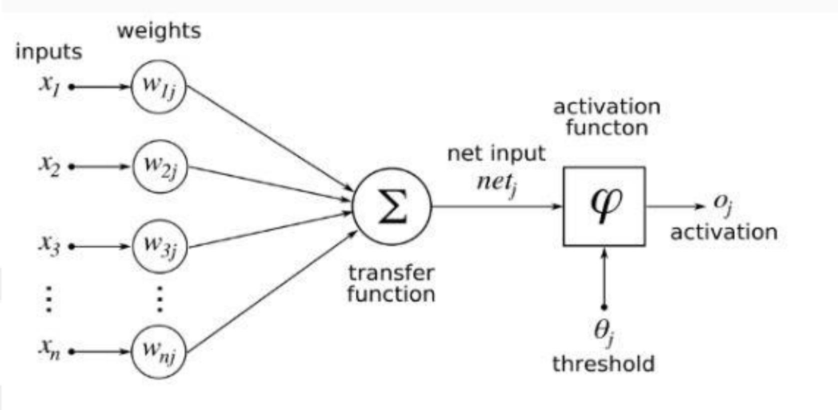

Figure 3.2. Non-linear Model of Neural Network [40]

The function of activation is often referred to as the crush function as the limits the allowable amplitude range output signal of the neuron.

Mathematically, a neuron can be express by the equation 3.1 – 3.3

$$u_k = \sum_{j=0}^{n} w_{ij}\, x_n \tag{3.1}$$

$$v_k = u_k + b_k \tag{3.2}$$

$$y_k = \varphi(u_k + b_k) \tag{3.3}$$

where $x_1, x_2, \dots x_n$ are the inputs signals, $w_{1j}, w_{2j}, .. w_{nj}$ are the weight of the neuron, $\varphi(v)$ is the activation function, u_k is the output summation, and v_k is also called the adder output that combines with bias bk. As in Figure 4.1.2, the deviation sum of the new input signal with a stable value and the value of weight would be equal to b_k. There are various function used for activation used in artificial neural network, but the most general activation function is called sigmoid, in which is strictly based on an increasing function with S-shape graph, as illustrated in equation 3.4

$$\varphi(x) = \frac{1}{1+e^{-x}} \tag{3.4}$$

Another sigmoid function that can be used is the hyperbolic tangent as illustrated in equation 3.5.

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3.5}$$

Figure 3.3, shows the activation function graph of sigmoid and hyperbolic tangent. The basic difference between the logistic and hyperbolic tangent is hyperbolic tangent limit the output neuron signal in the range of [-1, 1], while log sigmoid it limits in the range of [0,1].

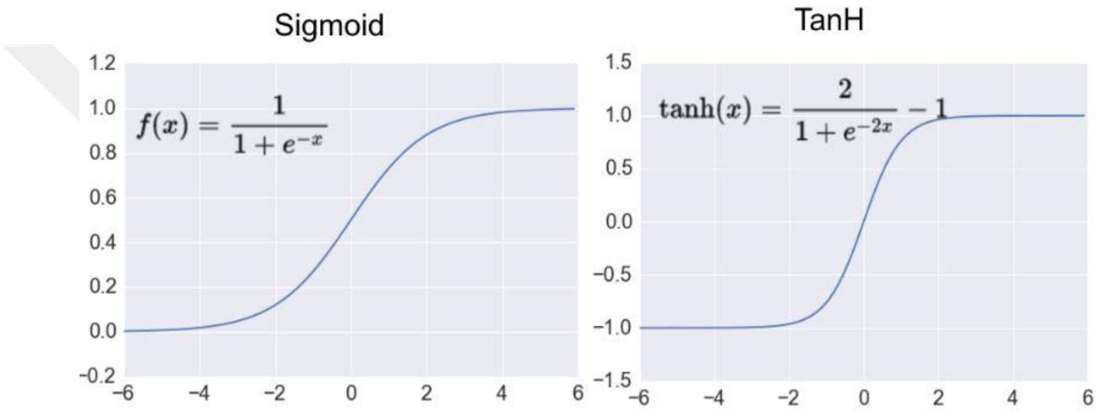

Figure 3.3. The Sigmoid and Hyperbolic Tangent Function [41]

Basic advantages of using these two type of activation function is that their derivative can be achieved very easily compared to other activation function. The derivative of the hyperbolic tangent function is shown in equation 3.5, while for a logistic function it is shown in equation 3.6 - 3.7.

The derivative of logistic is

$$\varphi'_j\big(v_j(n)\big) = \frac{e^{vj(n)}}{\left(1+e^{vj(n)}\right)^2} = \varphi_j(v_j(n))(1 - \varphi_j(v_j(n)) \tag{3.6}$$

then, the derivative of hyperbolic is given as

$$\varphi'_j\big(v_j(n)\big) = 1 - \frac{(e^{x} - e^{-x})^2}{(e^{x} + e^{-x})^2} = 1 - (\varphi_j(v_j(n)))^2 \tag{3.7}$$

3.2. Structure of Feedforward Neural Network

In a simple NN, there is only one input and output layer. This structure is referred to as a single layer perceptron. In this type of network, the function of the input layer is to "hold" the values entered into the network. Figure 3.4 shows a single layer sensor.

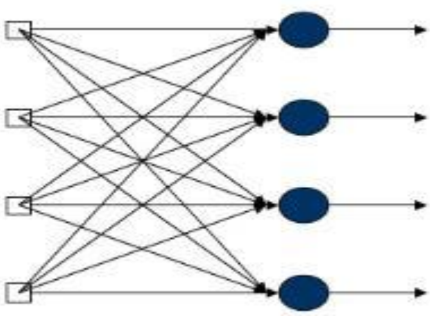

Figure 3.4. Structure of Single-Layer Perceptron [41]

However, the feed forward network consists of a hierarchy of units organized into two or more groups of mutually exclusive layers. The function of the input layer is to keep the input to the network, while the function of the output layer is to carry the information entered by the network to the output. There are hidden layers between the input and output layers where the weights connect each unit in one layer to the next higher layer [24]. However, these connections implied directionality, at the output of a layer, connecting weight is scaled by a value, then It is fed forward to provide some of the activations for the next higher layer. Figure 3.5 shows a two-layer sensor with one hidden layer [25].

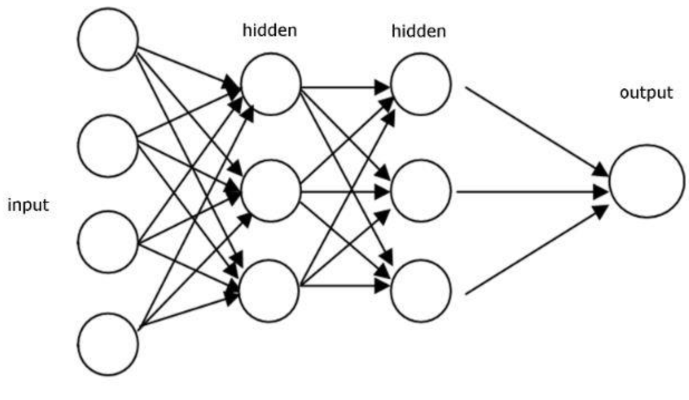

Figure 3.5. Structure of Two-Layer Perceptron [41]

This research thesis has designed various neural networks that changing in their structure and principle functioning. Also, this research emphasis on MLPFF NN trained with back propagation algorithm.

3.3. Backpropagation Algorithm

The error back propagation algorithm is a procedure that is often used to train the multi-layer detector in a controlled method. This process has two transistors in between the two layers, i.e., input layer and output layer. In forward propagation, input vector would be applied to neurons of the network, and the values of weight are unchanged. However, on the back propagation, the synaptic weights will be adjusted based on algorithm error learning rule. [24–26]. Therefore, this indicated that actual response of the NN is going to be subtracted from the desired response of NN, that result in an error signal. The value of error signal is going to propagate the backward across the entire NN that goes against the direction of the weight. This error can be called back propagation algorithm. Equation 3.8 shows the error signal e_j (n) in the output of neuron j in n iterations;

$$e_j(n) = d_j(n) - y_j(n) \tag{3.9}$$

From equation 3.9. The value of $y_j(n)$ is the actual response of the network and the value of $d_j(n)$ is the desire response for the network. therefore, an error energy is defined as $\frac{1}{2}e_j^2$, then the error value $\varepsilon(n)$ of total error given in equation 3.10 is obtained by adding the $\frac{1}{2}e_j^2$ based on all the neurons in the output layer.

$$\varepsilon(n) = \frac{1}{2}\sum_{j\in C} e_j^2 \tag{3.10}$$

The value of C indicates all the set of output layer neurons in the NN, and also the N indicates the total number of element in the training set of the NN. Therefore, average square error can be calculated by adding $\varepsilon(n)$ over all the values of n as it seem in equation 3.11, and then the result would be normalized to the size of N [26].

$$\varepsilon(n)_{av} = \frac{1}{N}\sum_{n-1}^{N} \varepsilon(n) \tag{3.11}$$

The aim of learning algorithm is to reduce or minimize the mean square error. In this research, the error can be minimized using a simple method by training the updated synaptic weight according to the corresponding error signals from each model represented by the network. [26].

This induced field $v_j(n)$ which is at the input of the activation function of the network is associated with neuron j is giving in the equation

$$v_j(n) = \sum_{i=0}^{m} w_{ij}(n) y_i(n) \tag{3.12}$$

The total number of inputs that applied to the neurons j is the value m the output signal of the neuron can now be

$$y_j(n) = \varphi_j(v_j(n) \tag{3.13}$$

The backpropagation algorithm would apply a correction $\Delta w_{ji}(n)$ to the weights of the network Δw_{ji}, that is proportional to the partial derivative of $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$. Therefore, the correction of $\Delta w_{ji}(n)$ that applied to the synaptic weight is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \tag{3.14}$$

where η is the value of learning rate. This method is called delta rule. The negative sign indicates gradient descent in weight field. These process shows steps taken in $\varepsilon(n)$ to minimize the weight space are done in the opposite direction of the gradient [26].

Sometimes, the correction can be written as

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{3.15}$$

where $\delta_j(n)$ is called the local gradient, it can be represented as

$$\delta_j(n) = -\frac{\partial \varepsilon(n)}{\partial v_j(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) {\varphi'}_j\left(v_j(n)\right) \tag{3.16}$$

Based on the equation above, the gradient of the output layer can be obtained as the product of error signal $e_j(n)$ and its derivative function of ${\varphi'}_j\left(v_j(n)\right)$ with the activation function.

Therefore, the NN has desired output response for the output layer, that make it easier to calculate its gradient [26]. Moreover, for the hidden layer neurons, no desired response would make it complicated. Therefore, the gradient for a hidden layer neuron can be illustrated in equation 2.31.

$$\delta_j(n) = \varphi'_j\left(v_j(n)\right)\sum_k \delta_k(n)\, w_{kj}(n) \tag{3.17}$$

where the j neuron is a hidden neuron. This equation says that the gradient of the neuron which is the product of the derivative function of the activation function of that neuron, which would be the same neuron as sum product of the next layer neuron gradient and the weight that is in between the j and k neurons. Equation 3.18 is the summary of the discussion above.

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) \tag{3.18}$$

The $\Delta w_{ji}(n)$ correction is a product of the $\delta_j(n)$ gradient and parameters of the learning rate with input of the neuron. Therefore, to generate the gradient of the NN it depends if the connected hidden neurons. However, if the neuron is at departure neuron $\delta_j(n)$, therefore would be the product of the derivative of the activation function $\varphi'_j\left(v_j(n)\right)$with the error signal $e_j(n)$. However, if the neuron is hidden, the derivative of $\delta_j(n)$ is the product of layers such as $\varphi'_j\left(v_j(n)\right)$and the two-layer perceptron, the output layers δ_j associated synaptic weight between j and k neurons [25,26].

3.3.1. Learning Rate and Momentum

The purpose of learning rate η is the NN is used to regulate the sizes changes made by the weight from one epoch of the backpropagation to the next epoch of the next epoch. If small value of η is obtained, then the smoother would be in the weight space. Also, if the value of η increases to speed of the learning rate, the weight of the network is going to be change which result is the NN unstable [25].

To obtained the best learning rate is going to be difficult, the simple method that can be used to avoid this instability is by modifying the delta rule as in equation 3.19.

$$\Delta w_{ij}(n) = \alpha w_{ij}(n-1) + \eta\delta_j(n)y_i(n) \tag{3.19}$$

where α positive number is called momentum constant. The idea for applying the momentum constant to the delta rule is that the local gradient is a weighted sum of successive iterations of the backpropagation algorithm. Thus, it filters out the large gradient oscillations that make the network more stable as before. [25].

3.3.2. Initial Weight

To start the initial weight with good values is an important is successful neural network design, and thereby the performance of the network would be effectively. If we assign the initial weight of to be small quantity, the performance of the algorithm would be operating in lower region or area that is close to the error surface. If the initial weight is too big, then the algorithm would be operating at the saturation region. To avoid this problems, the gradient back propagation algorithm takes some value that would slower the learning algorithm.

For selecting initial values [26] suggested a series of methods that can help to achieved good result. Firstly, if the average of the inputs over the entire training system set is close to zero the training has to be optimized, since its convergence is closer. Therefore, any change in the average input is away from zero alter the weight, by updating this particular direction would result in slow down the learning methods. Secondly convergence can be enhanced by scaling the input nearly the same as covariance. The value of covariance would match the sigmoid used.

This process is useful to all the layers in NN, and the output neuron we want to make it so close to zero. To achieved this process by selecting the proper activation function. The sigmoid origin is preferably symmetric, as they produce an output that is very close to zero and thus converges the network [27,28]. It suggests using the modified hyperbolic tangent. The constant value (hyperbolic tangent) of the activation function is chosen so that the variance of the outputs is close to 1 when used with normalized inputs. and the standard deviation given by this equation. $\delta_w = m - \ 1/2$, where *m* is the number of entries to neurons. [25, 29].

3.3.3. Teaching the Neural Network

The supervised learning approach involves having the desired output available for a particular set of input signals [25]. Therefore, the input and output signals consist the training examples. Also, a table with input and output data will be required to represent the process and its behavior. Based on this information, neural network structures will express "theory" about the learned system. However, this supervised learning practice may only depend on the availability of the table. The weights and threshold of the network will be adjusted repeatedly through the application of comparative actions influenced by the learning algorithm itself. This tends to control the difference between produced output and desired output. Based on this difference, the network is considered "educated". If this difference is within the satisfactory range value, then the supervised learning system is pure inductive inference, where the free variables of the network will be set by knowing the one before the desired output for the system under consideration [23]. To teach the neural network the following steps must be follow:

❖ To set an arbitrary initial values according to weights of the network

❖The training data is feed to the network till error signal nearly stops decreasing.

❖The procedures in step 1 and 2 are repeated, that lead in finding the error in the surface which would be easier to find the minima.

❖The network performance has never seen the training set before testing.

Therefore, in case there is difference in the training and testing set of data, then the NN performance is too big. This training set does not represent the population. Also, there is a possibility that there are too many neurons in the latent layer. He proposes this idea to find out how many hidden layers of neurons are needed [25]. If there are m neurons in the output layer and there are n inputs into the neural network, there must be √ (m * n) hidden layer neurons. In the neural network process, the method will not work for a training set of this size. Over time, the size of the hidden layer became a situation through trial and error.

3.3.4. Training Modes

In training modes, the learning outcomes of multiple performances of a training set are determined based on the neural network. If the one performance of the entire training set during the learning process is called the age. This procedure is usually held till the error signal reaches a predetermined minimum value. It is good practice to rearrange the performance order of the training set between consecutive terms. Back-propagation learning takes one of the following procedures forgiven for a training set:

1. The stochastic mode: the weight of the network in this mode is updated after training each example.

2. The batch mode: the weight of this network in this mode is updated completed after all the set of examples in the training set have been taken to the network.

This research work on the first mode, because one sequential mode requires less local storage for each connection and resources that are rare on a mobile device. Moreover, when training data is redundant, as in this case, this mode will be able to take advantage of this redundancy as this sample can be obtained in one go. However, sequential mode has several drawbacks compared to batch mode. Sequential mode which is better performance may be used in this research because it is easy to implement and often provides effective solutions to large difficult problems.

3.3.5. Criteria for Stopping

There are no defined criteria to stop the back propagation algorithm from working because it cannot show convergence. However, some logical criteria have some useful benefits. This type of criteria can be used to end weight changes. Although there is simple method used to predetermined the least mean square error, and the algorithm terminate when the minimum error reached. This method can lead to early expiry of the learning process. Secondly this method involves of testing network performance after each epoch using set of tests as described above. The learning process is stopped if the performance of the NN stretch or reaches a sufficient level.

The methods described above can be called as early stop method. When performance goal set as good, then the training would be very hard to understand when would it stopped. However, if training does not stay in the correct location, the network may over fit training data. However, over-compliance undermines the simplification performance of the network and is therefore undesirable to use [28,30], suggesting that stopping the learning process early is to treat symptoms but not virus. They proposed a technique that uses some few of neurons in which initially initiated their network performances were trained after it was tested with a lot of set of data that can be called the validation. This validation set consists of inputs which cannot be seen by the NN during the training process, thereby the neuron would be add to the process which would be repeated from the beginning. The process continues till the set error of the set data is reach an acceptable point.

4. METHODOLOGY AND IMPLEMENTATION

In this thesis research, the license plate number recognition methods consist of the following processing steps:

i. Detection of license plate.
ii. Segmentation of an image.
iii. Recognition of Character.

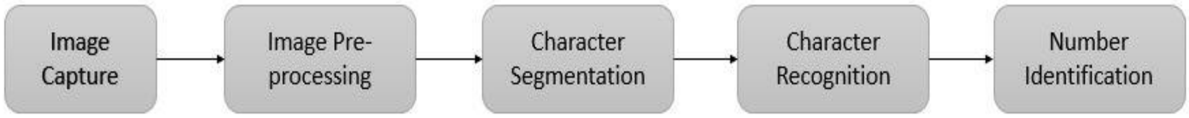

Figure 4.1. A Building Block of the License Plate Recognition System (LPR) [4]

Figure 4.1 illustrated a building block of the proposed design system. This process can be addressed as follows:

4.1. License Plate Detection

Image acquisition is the first step in plate number recognition. It is a very essential part of the LPR system. The image acquisition system helps the algorithm to run in a dynamic environment, and support the system of license plate localization and character segmentation. There are many ways to acquire images, such as digital cameras and CCD cameras. The use of a Raspberry pi camera can also be used to capture the vehicle's license plate number. To enhance the accuracy of the detection algorithm, it is obvious to know the exact size, color, and location of the license plate.

4.1.1. Pre-processing Image

The main goal of image preprocessing is to improve the quality of an image that would be suitable for easy recognition. The stages of preprocessing are so important and are commonly obtained in any computer vision application. In this thesis research, to processing an image required two methods:

i. Adjustment size - One of the basic stages of image preprocessing is by reducing its size after taken from the camera into an appropriate aspect ratio.
ii. Color transformation ratio – Most of the image captured are encoded in software standards. Therefore, these captured image are of RGB type which has 3 channel as red green and blue. These channel values in the captured image indicates the amount

of color information that image has. The image is in RGB form, by converting it to gray scale which offers less information for each pixel of an image.

A flowchart (see Figure 4.2) described the procedure of the whole system. This procedure described a detail operation of the plate number recognition. Firstly, to initial the system by turn on the touch screen and also the raspberry pi camera. Where by the raspberry pi find out is there any available image in the camera. If it obtained the image, then the MATLAB would start the process of extraction and recognition of characters. When this process is complete finished, the output result would be displayed on the screen. The whole process can be summarized as shown in Figure 4.2. This process is completely efficient and fast as long as the images are captured properly. Figures 4.3 and 4.4 show the captured image and the pre-processed result of the image.

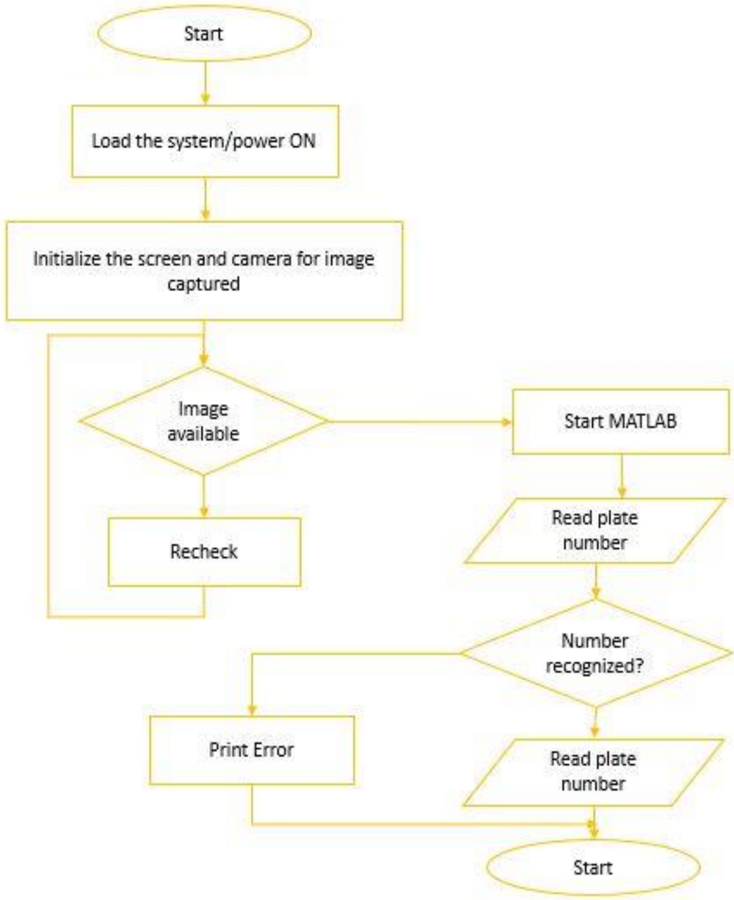

Figure 4.2. Flowchart of the Plate Number Recognition System [4]

Figure 4.3. Capture (RGB) Image [25]

Figure 4.4. Grayscale Image [25]

4.1.2. Filtering of Image

During the process of acquisition, there is a tendency to have a high number of noises which comes with the original image (like Gaussian noise, salt, and pepper noise, etc.). Because every image has a piece of useful and non-useful information. The license plate is our target that the useful information we want, but the rest are not useful information, and are called noise. A median filter will be used to eliminate the noises from the image. This filter is non-linear digital filtering techniques used for data processing such as edge detection of an image, also this filter not only to eliminate the noise but make the frequency of image high in concentration. This median filter is implemented for removing such purpose.

4.1.3. Edge Detection

Another important and interesting step in license plate recognition is edge detection. Edge detection can be defined as the technique of detecting an edge of the image. This technique is quite useful and important in understanding the features and information of the images. It usually affects the image by decreasing the less significant information, like the sizes of the images, and thereby filters which keep the structural properties of an image [36].

This edge detection technique can also help to reduce the size of images in which computers do not require too much memory. If images are classified based on the number of objects contain, this technique is greatly used in the process of segmenting the images, because edges are located in the image, in which boundaries of an object can be seen. There are various methods in edge detection, such as Sobel, canny, and Prewitt operators, in this research we used the canny edge detection method because it is among the easiest and popular for detection an edge of the license plate number. Figure 4.5 illustrated the result of a canny edge detection algorithm for the license plate number.

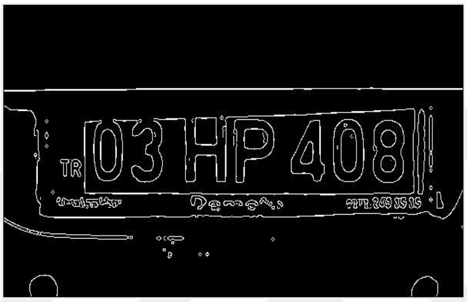

Figure 4.5. Result of Canny Edge Detection [34]

4.2. Character Segmentation

Segmentation of character is an important stage of (OCR). This process can be achieved by portioning an image into smaller parts. Moreover, segmentation of character is by looking at the special characters of each contour on the license plate. This contour can be anything that has a close surface, after obtaining the result, the license plate number would also have this contour feature.

To filter the LPR from the result obtained, we loop all the results and cross-check among the result which one has a rectangular shape of contour with a four-sided and close Figure, the LPR found has the right contour feature. If the segmentation process is successful the recognition would also be successful but if it fails, it hinders recognition accuracy [37]. Figure 4.6 illustrated the segmentation process using the bounding box technique.

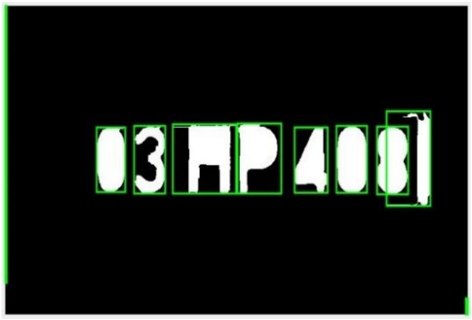

Figure 4.6. Segmentation of License Plate [35]

4.3. Recognition of Character

This is the last step in this process, is to read the plate number information from the segmented image. ANN is going to be used in recognizing the characters on the license plate image. Neural network is a model that mimic the human brain [30]. Moreover, it is capable of modeling linear and nonlinear systems based on expectations indirectly as in the case of statistical methods. This research used a MATLAB neural network toolbox that would recognize more characters.

However, the weights of the system can be adjusted by the training algorithm. A feed-forward backpropagation algorithm is used in the training of the network, which is a systematic process for training MLPNN [25]. This type of network makes use of an input layer, hidden layer, and output layer that is used in forecast the output characters. This network must memorize and train 26 English alphabets from (A-Z) and 10-digit numbers from (0–9). To validate the network, the software is going to be developed which reads the sequence of characters, and recognizes each character, and recognizes the identified character.

4.4. Back-propagation Neural Network

Back-propagation algorithm is a technique that trains an MLNN using a scientific basis. This technique is an advanced gradient descent method. Also, this technique used a delta learning rule which is in conjunction with the supervised machine learning process [38]. This technique uses a nonlinear thresholding function. The goal of back-propagation is to train the NN to adjust the input pattern used in the process and also to the input patterns.

4.5. Proposed Stages of OCR System

The proposed (OCR) system in this thesis research are based on two stages. Firstly, the education stage, and secondly which is the last stage is the recognition stage. These stages are illustrated in Figure 4.7.

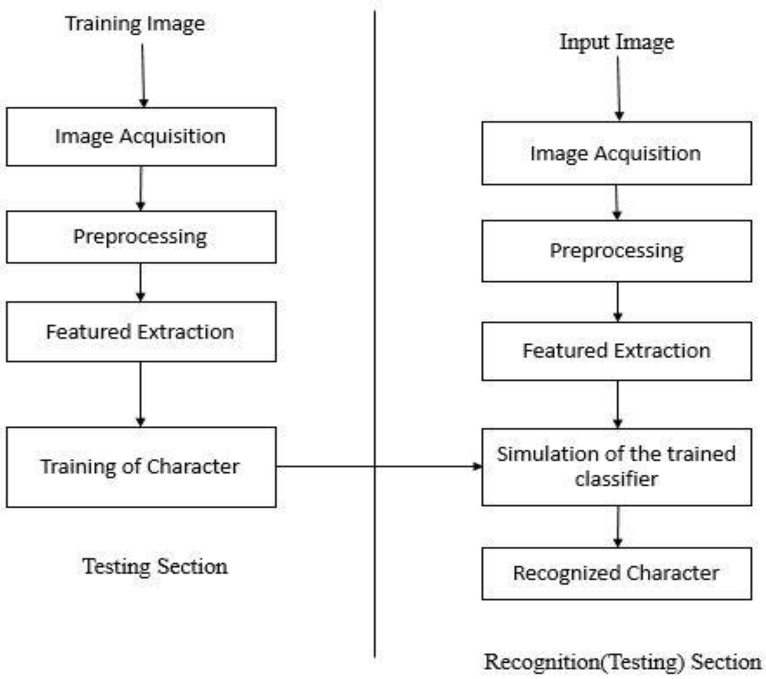

Figure 4.7. Stages of Optical Character Recognition (OCR) [39]

Figure 4.7. shown both the training and recognition schemes of (OCR). However, in each scheme have the acquisition of the image, image pre-processing, and extraction of image features. Also, the training scheme involves training classifiers and their model of the trained classifier.

4.5.1. Image Acquisition

In this stage, the proposed OCR would start with the image acquisition method, by captured the input image using a camera. Most of the formats such as JPEG, PNG, are used as the input image. The image that has been capture is to be used in testing the NN as shown in Figure 4.7.

4.5.2. Pre-processing of Image

This is the starting stage of the preprocessing of image whereby the actual data set are exposed that make it functional in the analysis of characters. The basic idea behind this method is to generate the actual raw data very easily for the systems to work correctly. These preprocessing steps are as follows:

a. Gray image conversion

At this stage, the Pixel values of a grayscale image have a value from 0 to 255. Black pixel density represents the lowest value; white pixel density represents the highest value. Therefore, RGB images can be converted to a grayscale image using equation 4.1 below.

$$g(x,y) = 0.299xR(x,y) + 0.587xG + 0.114xB(x,y) \tag{4.1}$$

where $R(x,y)$, $R(x,y)$, and $R(x,y)$ are the red, green and blue components of the color image respectively and $g(x,y)$ are grayscale images.

b. Noise Reduction

To improve the quality of the image is by reducing its noise. Median filtering is one of the techniques that can be applied is this process. The process is used to take the input image and a mask with dimensions 3 * 3, 5 * 5, or 7 * 7 and find/calculate the median given to the kernel of each pixel of the input image. And lastly, by replacing pixels with a median value that equivalent to the center of the kernel.

Usually, noise reduction is one of key factors to improve the quality of the image. Median filter would be used to eliminate the noises from the image. This filter is non-linear digital filtering techniques used for data processing such as edge detection of an image, also this filter not only to eliminate the noise but make the frequency of image high in concentration. This median filter is implemented for removing such purpose.

c. Thresholding

Thresholding is another technique in image processing. The basic idea of thresholding an image is by setting background pixels' values of the image below a threshold value, and the foreground value assigns to another value [40]. Global thresholding and local thresholding are two types of this technique. Equation 4.2 is used to arrive at a thresholding technique.

$$f(x,y) = \begin{cases} 255, & if\ f(x,y) \ > T \\ 0, & otherwise \end{cases} \tag{4.2}$$

where *T* is the threshold value.

d. Binarization

Binarization is like thresholding, in which grayscale images are converted into a binary image. However, this type of transforming image is defining by the levels, and these levels are called binary images.

e. Line Extraction

Segmentation of character is an important stage of optical character recognition (OCR). This technique used in portioning an image into smaller parts. Moreover, this technique can be achieved by looking at the special characters of each contour on the license plate. This contour can be anything that has a close surface, after obtaining the result, the license plate number would also have this contour feature. There are many ways to find boundaries of characters, a character string is one of them, whereby a string of characters is parsed into a potion of the single characters.

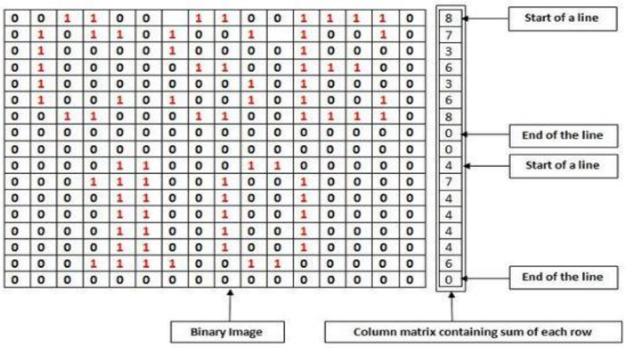

Figure 4. 8. Line Extraction Approach [38].

f. Extraction of Character

To extract the characters of an image is to scan the context or text region starting from left to right by obtaining its pixels mean. When their sum value is zero then a character is subtracted. This technique can be illustrated in Figure 4.9.

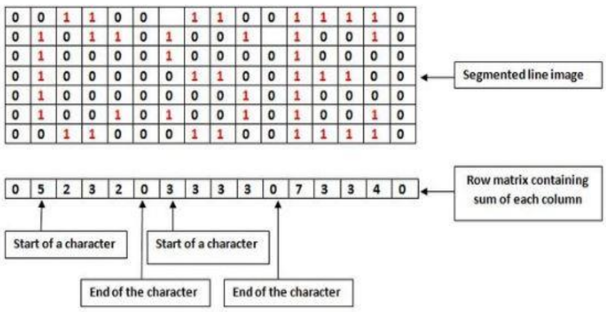

Figure 4.9. Character Extraction Approach [38].

g. Normalization

To normalization the characters is by synchronizing all the extracted binary image, in which the character variation are resizes. Besides, normalization is provided so that the extracted characters are equal to the matrix size. The size of the normalization matrix is *28x28*.

4.6. Feature Extraction

This process is used to extract and distinguish image features from the matrices of the output characters. Also, in this technique, each character would be itemized as an attribute vector whereby it identity would be known. Moreover, this technique its aiming extract some set of image features that would be used in maximizing the rate of recognition with a minimum number of items [38]. The character matrix is normalized to *28x28* in its size and has *0* and *1* value that is converted into column matrix. The output result served as the feature vector which used as the input of the backpropagation NN.

4.7. Training of Classifier

Training of classifier are the decision making of the recognition process. The use of feedback and backpropagation neural network in the classification and recognition of English alphabet characters. In the segmentation process, the derived pixels which were normalized can form an input to the classifier as illustrated in table 4.1.

Table 4.1. Classes of Recognition problem [38]

0	1	2	3	4	5	6	7
8	**9**	**A**	**B**	**C**	**D**	**E**	**F**
G	**H**	**I**	**J**	**K**	**L**	**M**	**N**
O	**P**	**Q**	**R**	**S**	**T**	**U**	**V**
W	**X**	**Y**	**X**				

Table 4.1 illustrated the class of character that can be recognized. These characters are trained with a neural network which has 864 patterns of 36 characters, and these characters have 24 instances, and are written in 24 fonts size.

4.8. Simulation of Trained Classifier

After the classifier has been trained with patterns, the extracted features of the test image are going to be simulated with training the NN. Lastly, neural network would have recognized the tested characters as verified from the neural network architecture.

4.9. Implementation

To implement all the processes in MATLAB, which is algorithms are implemented on it. Line and characters of the extraction algorithm has been prepared. Firstly, a neural network model was designed using trial and error which used a backpropagation algorithm. The network has two parts, the first one a scanned copy of image is going to be used in both the training and testing part. After obtaining the result, it can be converted to grayscale to remove the noises attached to the image. After that, the characters that are extracted are going to be normalized to make the image size fixed thereby subtracting the extracted features from the normalized images. However, in the training unit, the extracted features are one to be used in creating the input data set for training the network. At the recognition part, a test data set would be generated to stimulate the extracted features for the training network. The whole system can be illustrated in the flow chart diagram of Figure 4.10

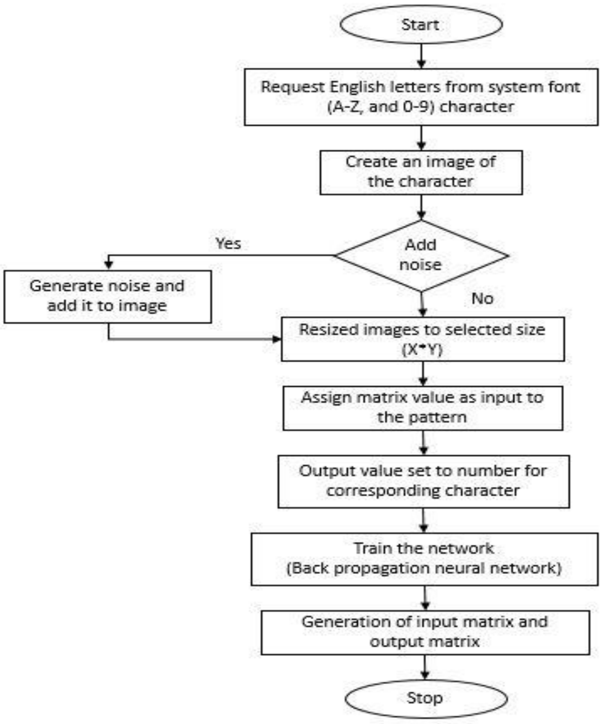

Figure 4.10. Input/output Matrix Data Scheme [41]

5. RESULT AND DISCUSSIONS

This is where we obtained our results, firstly some techniques used in obtaining the license plate number recognition are explained in details, which includes image acquisition, grayscale conversion, filtering, dilation, edge, and character segmentation methods.

The initial stage of intelligent plate number recognition is by obtain images of vehicles. Some electronic devices such as optical (digital/video) camera, and web cameras etc., are used to captured the acquired image which is in RGB format as shown in figure 5.1.

Figure 5.1. Input RGB image

Figure 5.1 is the captured RGB image, now is converted to a gray scale image format for easy analysis in which it contained only two color properties, since gray scale contains less information which we needed in providing for each pixel.

Figure 5.2. Gray scale image

Figure 5.2 Is the gray scale image. The next is to apply a median filtering, which is a preprocessing stage, in order to improve the visibility and quality of the captured image as in figure 5.2.

Figure 5.3. Filtered Image

Also, in figure 5.3. Is used to enhance the quality of the captured image, filtering is necessary to improve contrast enhancement, noisy and blurry issues and data reduction. This captured image is now cropped which is a recognition process which extract the rectangle that contains the edges of the license plate number. Disc of radius 1 is used as structural element for morphological operation, because using disc approximation run much faster when the structural approximation element uses approximation.

After applying the structural element, then dilating the gray scale image with disc structural element. This method called Imdilate, it performs gray scale dilation for all images except images of data type logical. It main objectives of the operator on binary images which gradually enlarge the boundaries of regions of foreground pixels. The next stage is edge detection method, which is an independent process in image processing as shown in figure 5.3.

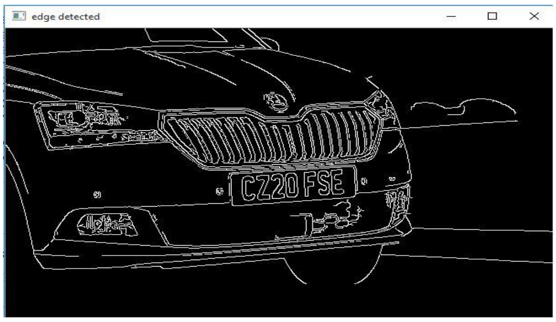

Figure 5.4. Edges of captured image

It usually finds the region boundaries as shown in figure 5.4. Firstly, before proceed to segmentation of an object from the captured image. The next stage is segmentation method which segment the character on the chosen candidate region i.e., the license plate, such that each characters can be send to the optical character recognition unit individually for recognition purposes.

Figure 5.5. Cropped Image

The captured image is now cropped as shown in figure 5.5, Which serve as a recognition process that extract the rectangle that contains the edges of the license plate number.

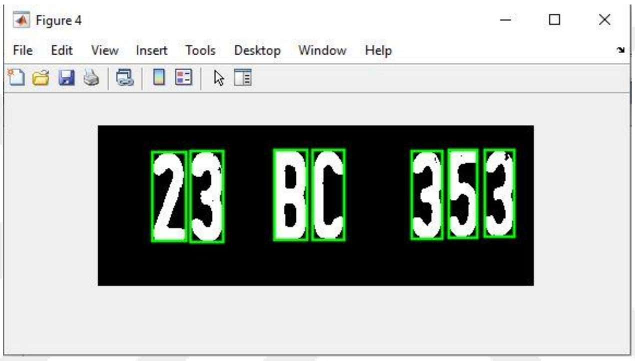

Figure 5.6. Segmented character of license plate

The next stage is segmentation method as shown in figure 5.6. This process segments the character on the chosen candidate region i.e., the license plate, such that each characters can be send to the optical character recognition unit individually for recognition purposes.

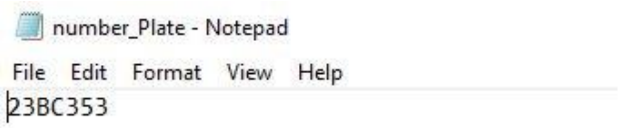

Figure 5.7. Corrected character recognized

Figure 5.7. Shows how the correct character of the license plate that fully recognized. Some other license plate number that fully recognized can be illustrated in figure 5.7.

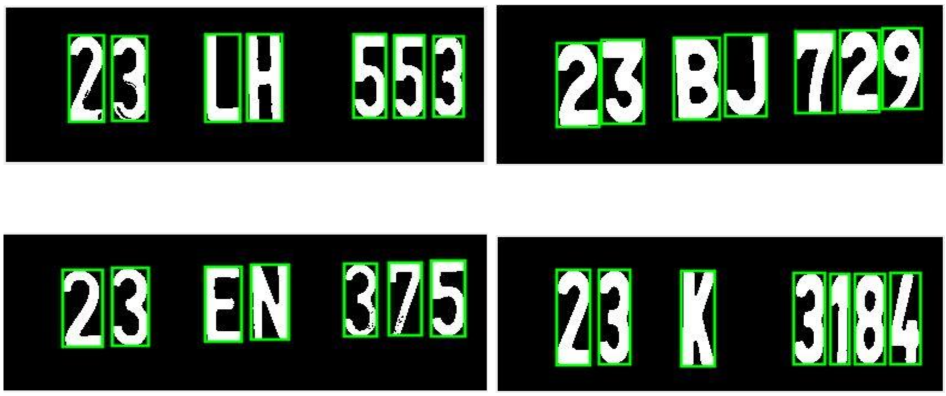

Figure 5.8. Some of the segmented license plate images.

Figure 5.8. Illustrated some of the character image of the license plate, which were fully segmented using image processing techniques, and later were recognized using optical character recognition.

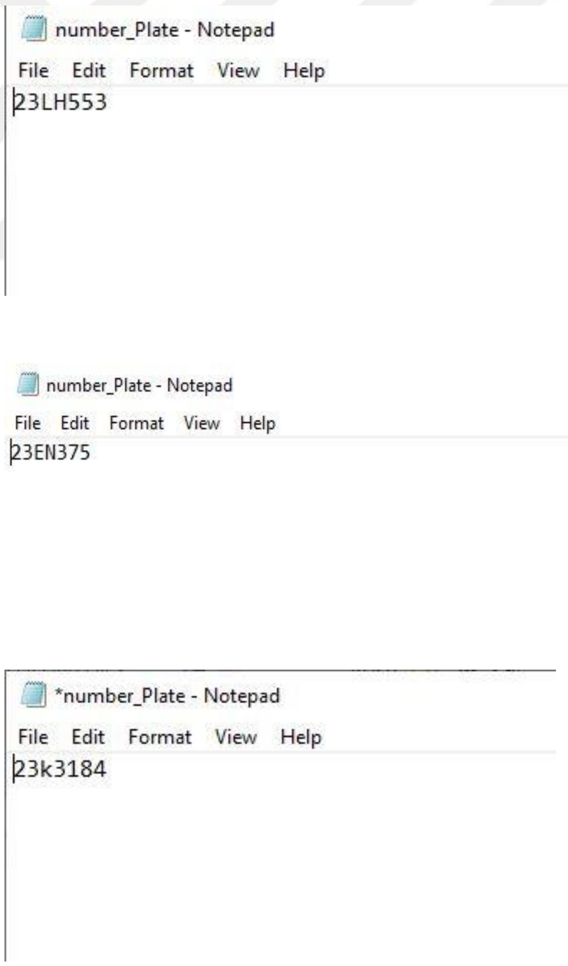

Figure 5.9. (a, b, c). Some of the recognized characters of license plate

Some of the recognized license plate number is shown in figure 5.9. These characters of license plate were initially acquired using image processing algorithms to their last stage which were recognized using optical character recognition figure 5.9.

5.2. EXPERIMENTAL RESULT

MATLAB software is used for writing simulation programs related to training and testing for perceptron and PatternNet models of ANN. The performances of the two models can be evaluated. Firstly, train a model, by trying to solve its optimization problems. By optimizing the model, the weights of the neural networks are constantly updating to reach their optimal values. To know how these weights are optimized depends on the optimization algorithm. The most widely used optimization algorithm is stochastic gradient descent (SGD). The objective of this algorithm is to minimize the loss for a given function as close to zero as possible. There are several loss functions which can be used in minimizing errors, such as (MSE) means square error, (MAE) mean absolute error, etc. we used (MAE) in calculate the performance of the models by supply a model with the data and labels to that data. We created 2 ANN models (pattern network and perceptron).

A pattern net is also an MLPNN classifier that uses softmax as an activation function, also used cross entropy for determining the error function. This softmax function has the power to calculate small probabilities of the model. Contract with linear activation function, a softmax is a probability function that does not give information about the absolute value but rather the class in which a particular data can be contained. In this research, one hidden layer pattern network is used as one of the classifiers. This layer uses Tanh as an activation function.

A perceptron can be called one input layer and one output layer, which is a machine learning algorithm for supervised learning of a binary classifier. It involves one input layer neuron that would adjust the synaptic weights, thereby classify the pattern into two classes. The hyperbolic tangent used as an activation function of this model.

The following result was obtained using the training of the Perceptron ANN model seems in Figure 5.10.

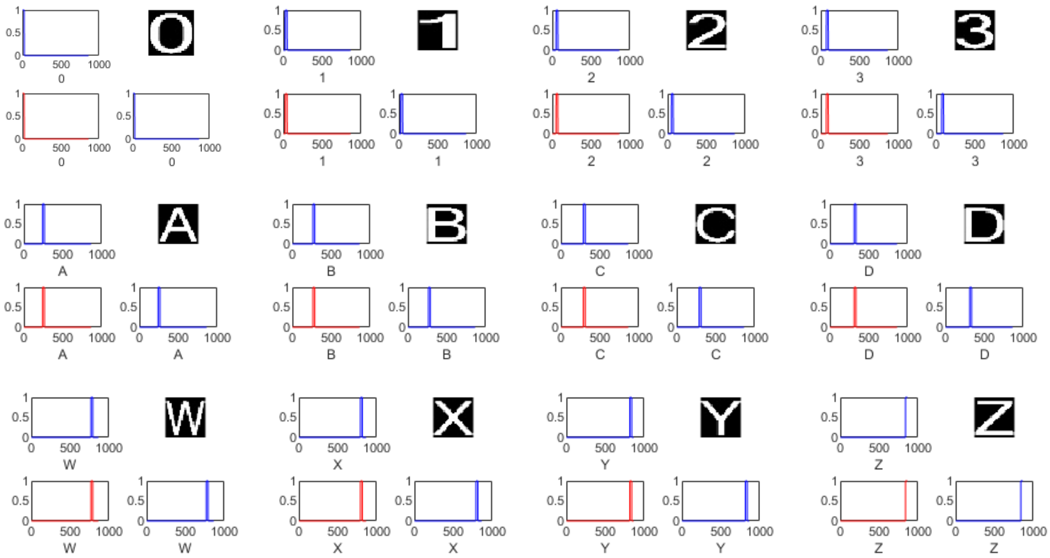

Figure 5.10. Illustrated some images of characters and numbers used at training perceptron neural networks, target vectors, and outputs vectors of networks.

Figure 5.10. Illustrated some images of characters and number used in training the perceptron model, which indicated the actual output and the desire output of the perceptron network, or by saying the target vectors and the actual vectors of neural network.

Table 5.1. Results of the characters for perceptron model

Characters	Number of Trail	Correct Recognition	False Recognition	% of Recognition
A	33	30	3	90.90
B	31	29	2	93.55
C	26	22	4	84.62
D	34	30	4	88.23
E	27	25	2	92.59
F	29	23	6	79.31
G	30	19	11	63.33
H	28	27	1	93.10
I				
J	30	27	3	90
K	23	21	2	91.3
L	26	25	1	96.1
M	21	18	3	85.71
N	30	27	3	90
O				
P	17	15	2	88.23
Q	24	20	4	83.33
R	25	23	2	92
S	21	19	2	90.47
T	28	17	9	60.71
U	29	21	8	72.41
V	31	19	12	61.29
W	32	24	8	75
X	18	8	10	44.44
Y	27	17	10	62.96
Z	28	22	6	78.57

Table 5.1. Shows the total number of characters used each characters are subjected to certain number of trail. Also, false recognition of character obtained by subtracting the number of correct character from the number of trail. Their percentage of recognition was also calculated.

Numbers	Number of Trail	Correct Recognition	False Recognition	% of Recognition
0	35	31	4	88.57
1	29	28	1	96.55
2	33	30	3	90.9
3	28	22	6	78
4	31	28	3	90.3
5	27	24	3	88.89
6	30	25	5	83.33
7	29	21	8	72.4
8	22	20	2	90.9
9	25	22	3	88

Table 5.2. Results of the Numbers for perceptron model

Table 5.2. Shows the total number of alpha numeric used each alpha numeric are subjected to certain number of trail. Also, false recognition of alpha numeric obtained by subtracting the number of correct alpha numeric from the number of trail. Their percentage of recognition was also calculated.

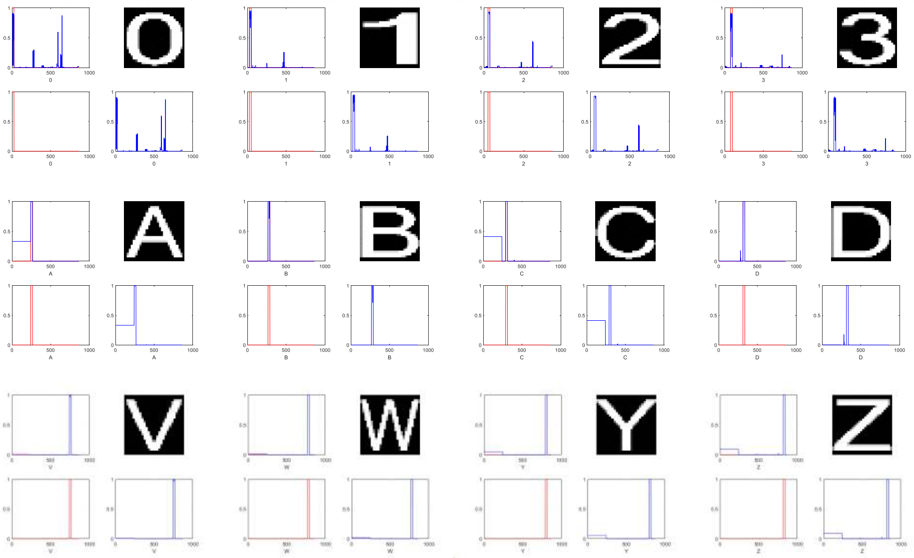

Figure 5.11. Illustrated some images of characters and numbers used at training Pattern Net neural networks, target vectors, and outputs vectors of networks.

Figure 5.11. Illustrated some images of characters and number used in training the perceptron model, which indicated the actual output and the desire output of the perceptron network, or by saying the target vectors and the actual vectors of neural network.

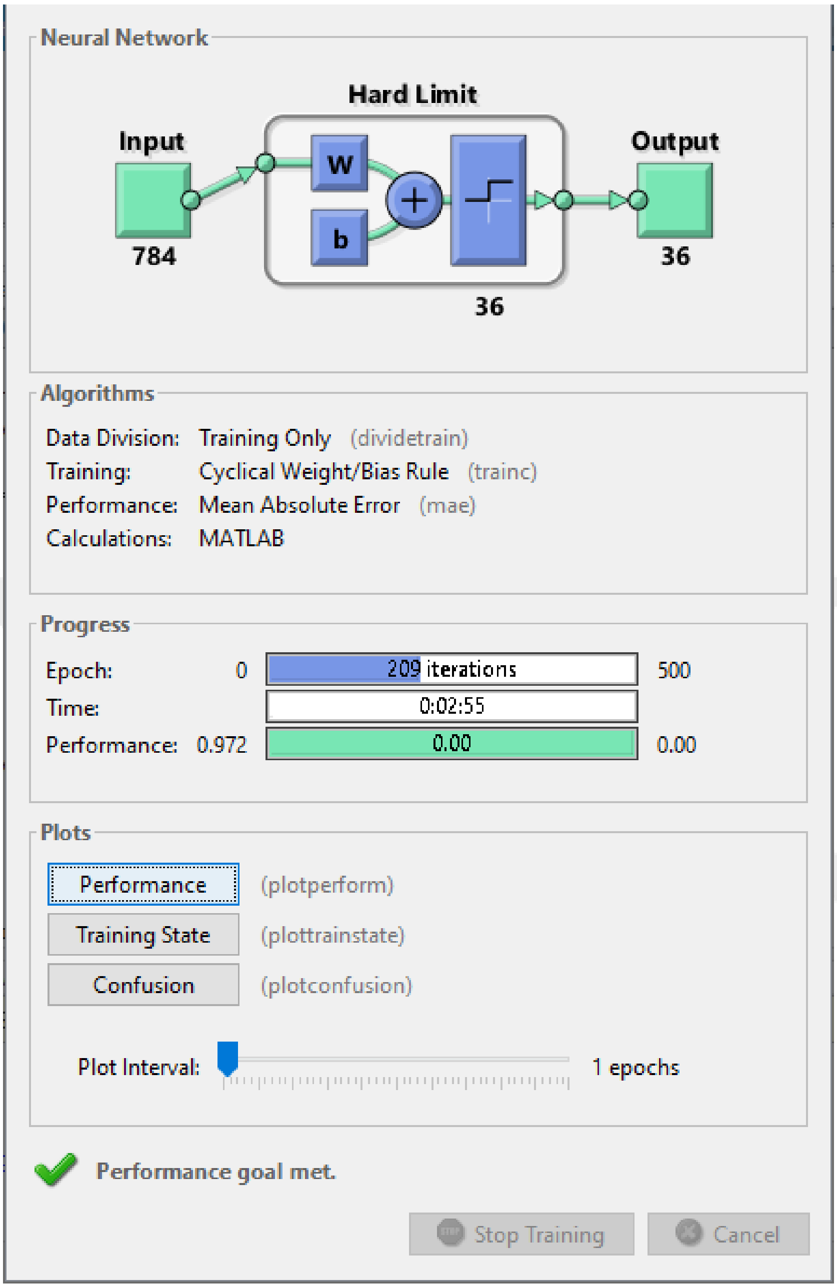

Figure 5.12. Illustrated the architecture of the Perceptron for training ANN model.

Figure 5.12. Is a perceptron ANN model was created. By choosing the perceptron ANN model, its architecture displayed 784 input vector, and one output layer of 36 number of neurons respectively. A hyperbolic Tangent activation function is used for the model. This activation function is an improved version of the sigmoid, it typically helps in creating the scores that to be considered in the next layer and which one can be ignored. By training the ANN model, we set all the initial settings and start the training the network as can be illustrated in Figure 5.12.

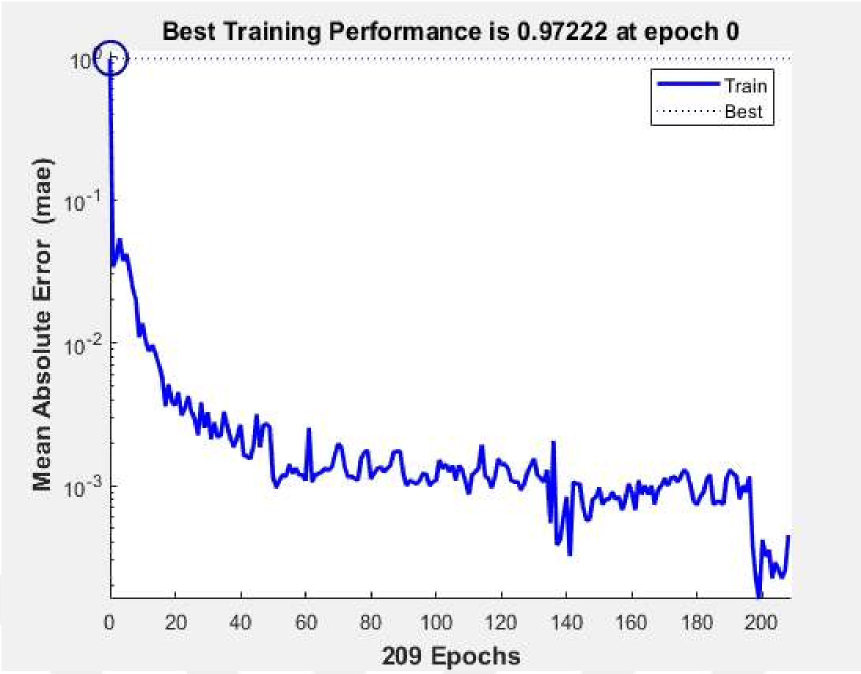

Figure 5.13. Best training performance of a Perceptron ANN model.

We train the perceptron ANN model, by setting all the initial and start the training the network as shown in Figure 5.13. By default, 500 number of the epoch is to be set. This number may be changed by increasing the number of iterations, and also the training parameters. Based on Figure 5.13. It can be determined the model trained with 97 number of epoch of the training algorithm (performance 0.9722) which can be shown in the performance graph. Based on the performance curve, by increasing the number of epoch the mean absolute error is decreasing.

Table 5.3. Specification of neural network model

Input Layer	1
Hidden Layer	1
Output Layer	1
Number Neurons in input layer	784
Number Neurons in Hidden Layer	39
Number Neurons in Output Layer	36

Table 5.3. Illusrated the specification of the neural network model of pattern network, which has one input layer with 784 number of neurons, and one hidden layer with 39 number of hidden neorons and lastly with one output layer of 36 number of neurons.

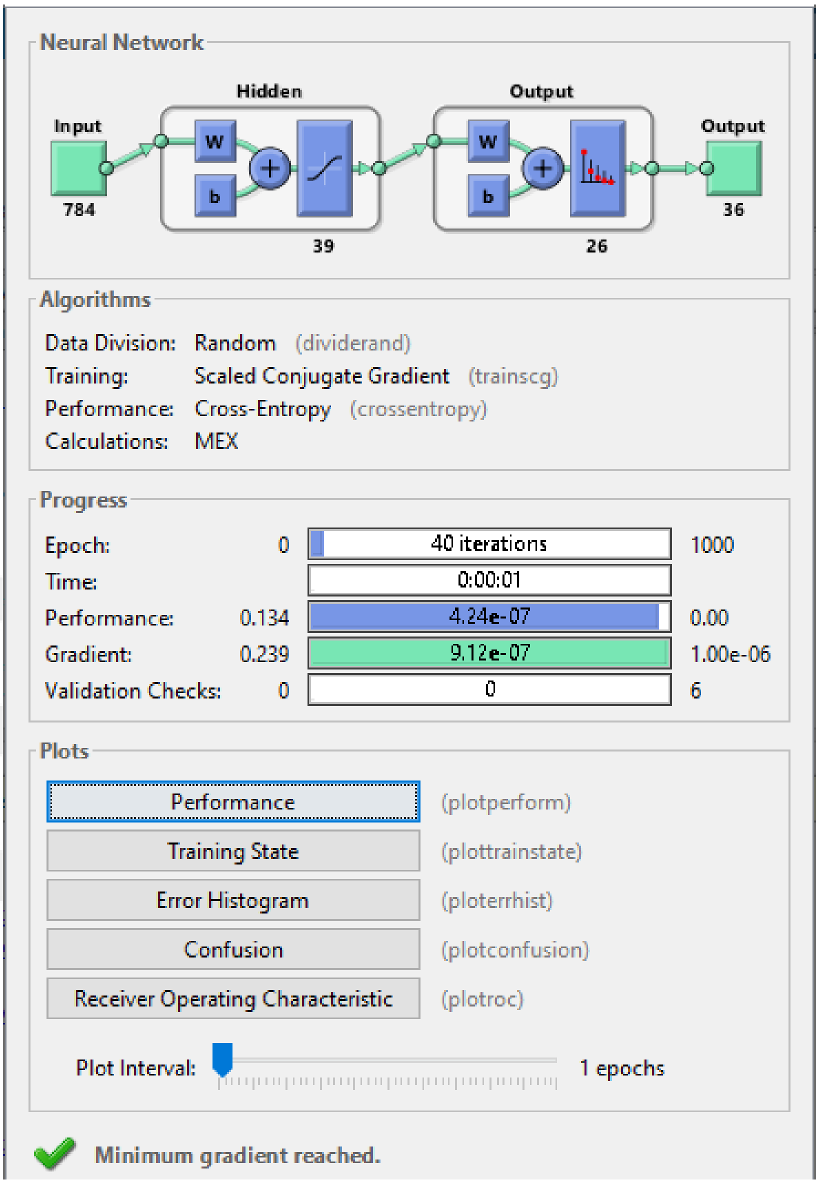

Figure 5.14. Illustrated the architecture of the pattern Net testing ANN model.

Performance and the error percentage is shown in Figure 5.14. Therefore, at the hidden layer, the network is going to train with 39 number of neurons. The train function (train Function) and performance function (performance Function) is used for the default values. The type of activation function used at the hidden layer is Trainscg which signified as scale conjugate gradient descent algorithm. However, we train the network with the MNIST training data set and thereby obtained the percentage error with the training data and the percentage error was recorded. Also, the ANN pattern net would test MNIST dataset to obtained the accuracies of the network. These results would be saved into the variable percentage error data set. The ANN model can be trained and observed its status.

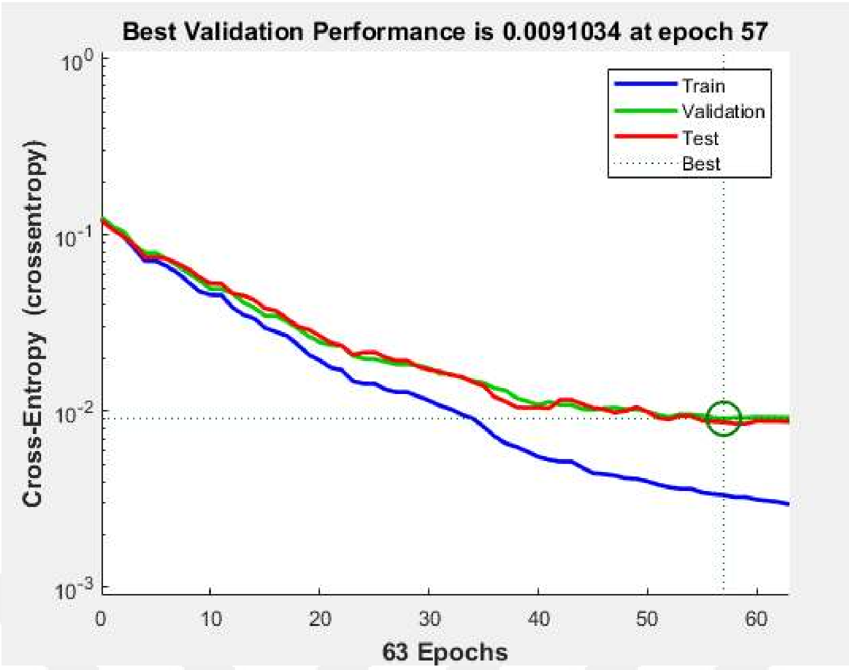

Figure 5.15. Illustrate the performance of the testing data set using the pattern Net ANN model.

We train the pattern Net ANN model, by setting all the initial and start testing the network as shown in Figure 5.15. By default, 1000 number of the epoch is to be set. This number may be changed by increasing the number of iterations, and also the training parameters. Also based on Figure 5.6. It can be determined that the neural network has trained with of 57 number of epoch of the training algorithm which can be shown in the performance graph. Based on the performance curve, as the number of epoch increased in the training data set, the mean absolute error is decreasing. However, all the three data set were used. Based on the performance curve, as the number of epoch increased the cross-entropy of the testing, and validation data set is decreasing, it would reach a point where the ANN model would stop. At this point the ANN model comes to learn beyond the 63 epoch, the model results in overfitting.

Table 5.4. Recognition Model Parameters

Number of Iteration	**57**
Performance Goal	0.01
Mean Square Error	0.0128

Table 5.4. Illustrated the recognition model parameters of the neural network model. Show that the model has 57 number of epoch and mean square error of 0.0128. The percentage of recognition of the network can be obtained by using the formula.

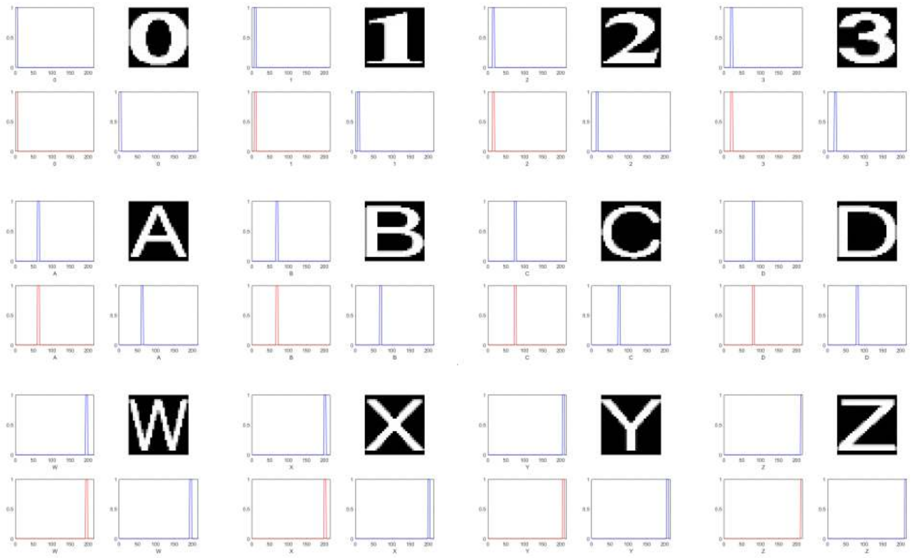

Figure 5.16. Illustrated some images of characters and numbers used at testing neural networks, target vectors, and outputs vectors of networks for Pattern Net ANN model.

Figure 5.16. Illustrated some images of characters and number used in testing the perceptron model, which indicated the actual output and the desire output of the perceptron network, or by saying the target vectors and the actual vectors of neural network. It calculates percentage of recognition as equation 5.1 to find the performance of the neural network.

$$Percentage\ of\ recognition \frac{(total\ number\ of\ test\ character - number\ of\ misclassified\ character)}{total\ number\ of\ test\ character}\ 100\% \tag{5.1}$$

Table 5.5. Pattern Recognition Model Evaluation

Classes	Number of test characters	Number of correct classification	Number of misclassification	% of Recognition
0-9	60	57	3	95%
A-Z	156	145	11	92.9%
0-9, & A-Z	216	202	14	96.8%

Table 5.5. Illustrated the evaluation of network model, from the table, it can be observed the total actual number of characters used is 216 which has the % of recognition of about 97%. While for the alphabet and numbers has about 93% and 95% recognition rate respectively.

In the testing process of Perceptron, the following result was obtained as shown in Figure 5.8

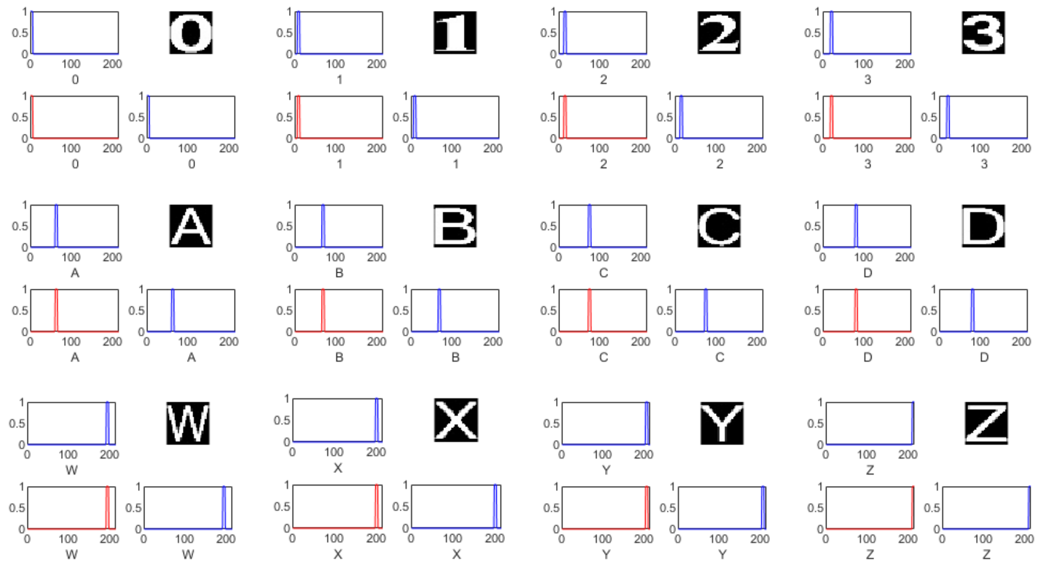

Figure 5.17. Illustrated some images of characters and numbers used at testing neural networks, target vectors, and outputs vectors of networks for the Perceptron ANN model.

Figure 5.18. Illustrated some images of characters and number used in testing the pattern net model, which indicated the actual output and the desire output of the perceptron network, or by saying the target vectors and the actual vectors of neural network.

We used three different type of neural networks in this research, (pattern net, perceptron, and multilayer neural network) to observe their performances. Also, the result of validation of the neural networks for different type of plate number using each neural networks are illustrated in table 5.6.

For pattern net, four different types of architectures of neural network are used to observe the performances of the network, such as Net7 PTRN 123_36 tansig_softmax, Net21 PTRN 63_36 tansig_softmax, Net7 PTRN 123_36 tansig_softmax, and Netp1 PTRN 63_ tansig_softmax. Among them, Netp1 PTRN 63_ tansig_softmax has best performance with zero false recognition, and the rest has only one false recognition of license plate, which shows that pattern net has a very good performance in recognizing the license plate images compared to other two types of network. Moreover, pattern net has an advantage of less training time compared to other types of neural network.

For MLP neural networks, four different types of architectures ware used to observe the performances of networks. Some of the network includes: Net1 MLP 123_36 tansig_tansig, Net11 MLP 785_36 tansig_tansig, Net23 MLP 101_101_36 tansig_tansig_softmax, Net3 MLP 36_36 tansig_tansig, Network1 MLP 55_36 tansig_tansig. Among the four networks, Net3 MLP 36_36 tansig_tansig, has the best recognition performance which recognize all the license plate images.

For perceptron, two different type of network were used to observe their performance, these are Net2 PCRN(tansig)_36, Net5 PCRN(hardlim) 36. Based on the observation, none of them able to recognize one license plate recognition, this show that perceptron has poor recognition performance when compared with MLP and pattern net.

Table 5.6. Validation of 7 License Plate Images

Type of Architecture	Type of license plate	True/False
Net1 MLP 123_36	23HC353	True
tansig_tansig	238J721	False
	238S611	False
	23EN375	True
	73ES352	False
	23K318S	True
	23LH553	True
Net11 MLP 785_36	2SBEE5H	False
tansig_tansig	2JBJ7Z9	False
	22BE611	True
	2HFNH75	False
	7HFGR52	False
	2JKJ184	False
	ZHLH55H	False
Net2 PCRN(tansig)_36	7RHCH6H	False
	27RJ72V	False
	22RE611	False
	7HHNH7U	False
	7HFRH62	False
	2JKJ1H6	False
	7HLH66H	False
Net21 PTRN 63_36	23BC353	True
tansig_softmax	23BJ729	True
	23BS611	True
	23EN375	True
	23FS352	False
	23K318G	True
	23LH553	True

Net23 MLP 101_101_36	73BC353	False
tansig_tansig_softmax	23BJ729	True
	238S611	False
	23FN375	False
	73FS357	False
	23K3184	True
	73LH553	False
Net3 MLP 36_36	23BC353	True
tansig_tansig	23BJ729	True
	23BS611	True
	23EN375	True
	23ES352	True
	23K3184	True
	23LH553	True
Net5 PCRN(hardlim) 36	Z0RL000	False
	ZZ0JTZ0	False
	Z00SSTT	False
	7ZRNJ75	False
	700S007	False
	Z0K0T0G	False
	70LR550	False
Net7 PTRN 123_36	23BC353	True
tansig_softmax	23BJ729	True
	Z3BS6TJ	False
	23EN375	True
	23ES35Z	False
	23K3184	True
	Z3LH553	False
Netp1 PTRN 63_	23BC353	True
tansig_softmax	2JBJ729	True
	23BS6T1	False
	23EN375	True
	23ES352	True
	23K318S	True
	23LH553	True

Netp2 PTRN 785_36	23BC353	True
tansig_softmax	Z3BJ729	False
	23BS611	True
	23EN375	True
	23ES352	True
	23K318A	True
	23LH553	True
Network1 MLP 55_36	23BC353	True
tansig_tansig	Z3BJ729	False
	238S611	False
	23BNH75	False
	23HS3S2	False
	23K3185	True
	23LH553	True

From table 5.6, we also observed that some of the networks fully recognized the license plate correctly, like pattern net5, net21, and net7, netp1 and netp2. Some of the network that couldn't recognized license plate correctly are perceptron network net1, net5, and net23. This is due to so many reasons, like improper captured license plate image, preprocessing techniques and etc.

5.3. DISCUSSION

In this study, an experiment was conducted for the evaluation of the performance of recognition of characters. Two model were used to compared the performances of the neural network system, based on time, speed, and accuracy. The two models are pattern net and perceptron. In the training mode, perceptron which is a simple network takes longer time before it converges unlike than the pattern net which has less time. Also, pattern net give much better accuracy compared to perceptron as shown in Figure 5.6. Moreover, pattern net model has a percentage error with less number of epoch when compared with perceptron model which is not the same with the pattern net model.

Also Figure 5.10 and Figure 5.11 show the output result of testing patterns using perceptron and pattern net respectively. Table 5.1 and table 5.2 provides a summary of the specification of the network architecture and performance parameters based on its error performance, timing and accuracy. Also, the lost function and the accuracy are also obtained in Figure 5.10 and 5.11 respectively.

6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

The use of Back-propagation (BP) in optical character recognition (OCR) would give good output results. However, these OCR can be affected by many factors like brightness, noise, and color background. Therefore, pre-processing is essential to be used in documented images as an initial step for character recognition, to remove the effects of such factors. There are different pre-processing techniques needs for each image, which depend on the factors that may affect the image quality.

Also, from Figure 5.15, the pattern net model shows that its performance is better when compared with perceptron model in terms of its accuracy and error percentage. Moreover, it has less number of epoch as compared with perceptron model.

6.2. Future Work

In future work, a lot of project may be extended to other various classification problems such hand writing zip code recognition, text recognition and alphabet character recognition using any type of machine learning algorithm, like perceptron or pattern net.

REFERENCES

[1] J. Wang, B. Bacic, W.Q. Yan, An effective method for plate number recognition, Multimed. Tools Appl. 77 (2018) 1679–1692. https://doi.org/10.1007/s11042-017-4356-z.

[2] A.O. Agbeyangi, O.A. Alashiri, A.E. Otunuga, Automatic Identification of Vehicle Plate Number using Raspberry Pi, in: 2020: pp. 1–4. https://doi.org/10.1109/icmcecs47690.2020.246983.

[3] and K.H.G. N. Sulaiman, A. F. Razali, M. S. Jadin, A. Abdul Hadi, M. Mustafa, Development of Online Vehicle Plate Recognition System, J. Electr. Electron. Control Instrumentations. 1 (2016).

[4] A. D G, A. O T, F. A S, Nigerian Vehicle License Plate Recognition System using Artificial Neural Network, Ijarcce. 4 (2015) 1–5. https://doi.org/10.17148/ijarcce.2015.41101.

[5] K. Khorgade, Security for Campus Vehicle by Automatically Recognition of Number Plate with Notification, Int. J. Res. Appl. Sci. Eng. Technol. 7 (2019) 3983–3985. https://doi.org/10.22214/ijraset.2019.5657.

[6] K.W. Maglad, A vehicle license plate detection and recognition system, J. Comput. Sci. 8 (2012) 310–315. https://doi.org/10.3844/jcssp.2012.310.315.

[7] K. Vallapreddy, S. Sunkari, A New method of License plate recognition system using Raspberry Pi processor, (n.d.) 1–5.

[8] S.J. Danbatta, N.D. Mohammed, Nigerian License Plate Using Bounding, 2 (2016) 239–244.

[9] J. Jagtap, S. Holambe, Multi-Style License Plate Recognition using Artificial Neural Network for Indian Vehicles, 2018 Int. Conf. Information, Commun. Eng. Technol. ICICET 2018. (2018) 1–4. https://doi.org/10.1109/ICICET.2018.8533707.

[10] D. Zang, Z. Chai, J. Zhang, D. Zhang, J. Cheng, Vehicle license plate recognition using visual attention model and deep learning, J. Electron. Imaging. 24 (2015) 033001. https://doi.org/10.1117/1.jei.24.3.033001.

[11] Digital_image_processing_by_Rafael_C._Go.pdf, n.d.

[12] W.M. Programs, With MATLAB Programs & Examples, (2017).

[13] S. Shawal, M. Shoyab, S. Begum, Fundamentals of Digital Image Processing and Basic Concept of Classification, Int. J. Chem. Process Eng. Res. 1 (2014) 98–108. https://doi.org/10.18488/journal.65/2014.1.6/65.6.98.108.

[14] A. Haris, A.A. Almisreb, Southeast Europe Journal of Soft Computing Available online : http://scjournal.ius.edu.ba A Review on Image Enhancement Techniques, 8 (2019). https://doi.org/10.21533/scjournal.v8i1.175.

[15] E. Baidoo, Implementation of Gray Level Image Transformation Techniques, Int. J. Mod. Educ. Comput. Sci. 10 (2018) 44–53. https://doi.org/10.5815/ijmecs.2018.05.06.

[16] R. Chandwadkar, S. Dhole, V. Gadewar, D. Raut, S. Tiwaskar, Comparison Of Edge Detection Techniques, 6th Annu. Conf. IRAJ. (2013) 133–136. https://doi.org/10.13140/RG.2.1.5036.7123.

[17] S. Bassel, Z. Nizar, Recognition of Vehicle Numberplate Using Matlab, (2016). https://doi.org/10.13140/RG.2.1.1459.8640.

[18] N.M. Basheer, A.M. Aaref, D.J. Ayyed, International Journal of Advanced Research in Digital Image

Sobel Edge Detection Using FPGA, (2018).

[19] S. Norul, K. Omar, K. Omar, A. Salimi, M. Petrou, M. Khalid, Determining adaptive thresholds for image segmentation for a license plate recognition system, Int. J. Adv. Comput. Sci. Appl. 7 (2016). https://doi.org/10.14569/ijacsa.2016.070667.

[20] A. G, H.N. T, J. Kumari, S. M, Analysis of Digital Images Using Morphlogical Operations, Int. J. Comput. Sci. Inf. Technol. 5 (2013) 145–159. https://doi.org/10.5121/ijcsit.2013.5112.

[21] N. MOHANAN, A. AHMAD, S.S. AL-BUSAIDI, L. KHIRIJI, A. ABDULGHANI, M. AL NADABI, Use of Mathematical Morphology in Vehicle Plate Detection, Orient. J. Comput. Sci. Technol. 11 (2018) 195–200. https://doi.org/10.13005/ojcst11.04.04.

[22] A.A. Universit, MATLAB Code of Artificial Neural Networks Estimation MATLAB Code of Artificial Neural Networks Estimation, (2016).

[23] H. Kukreja, An introduction to artificial neural networks, Hardw. Archit. Deep Learn. (2020) 3–26. https://doi.org/10.1049/pbcs055e_ch1.

[24] M. Thakur, I. Raj, P. Ganesan, The cooperative approach of genetic algorithm and neural network for the identification of vehicle License Plate number, ICIIECS 2015 - 2015 IEEE Int. Conf. Innov. Information, Embed. Commun. Syst. (2015) 1–6. https://doi.org/10.1109/ICIIECS.2015.7193090.

[25] S. Samarasinghe, Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition, (2007) 582. http://www.amazon.com/Neural-Networks-Applied-Sciences-Engineering/dp/084933375X/ref=pd_sim_b_3%0Ahttp://www.amazon.com/Neural-Networks-Applied-Sciences-Engineering/dp/084933375X/ref=pd%7B_%7Dsim%7B_%7Db%7B_%7D3.

[26] S.S. Cross, R.F. Harrison, R.L. Kennedy, Introduction to neural networks, 1995. https://doi.org/10.1016/S0140-6736(95)91746-2.

[27] P. Surekha, P. Gurudath, R. Prithvi, V.G. Ritesh Ananth, Automatic License Plate Recognition Using Image Processing and Neural Network, ICTACT J. Image Video Process. 8 (2018) 1786–1792. https://doi.org/10.21917/ijivp.2018.0251.

[28] I. Portugal, P. Alencar, D. Cowan, The use of machine learning algorithms in recommender systems: A systematic review, Expert Syst. Appl. 97 (2018) 205–227. https://doi.org/10.1016/j.eswa.2017.12.020.

[29] X. Zhao, C. District, Z. Jiang, C. District, A Tangent Distance Preserving Dimensionality Reduction Algorithm, (n.d.).

[30] A. Dey, Machine Learning Algorithms: A Review, Int. J. Comput. Sci. Inf. Technol. 7 (2016) 1174–1179. www.ijcsit.com.

[31] W.F. Abaya, J. Basa, M. Sy, A.C. Abad, E.P. Dadios, Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV, 2014 Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2014 - 7th HNICEM 2014 Jt. with 6th Int. Symp. Comput. Intell. Intell. (2014). https://doi.org/10.1109/HNICEM.2014.7016253.

[32] V. Vujović, M. Maksimović, Raspberry Pi as a Sensor Web node for home automation, Comput. Electr. Eng. 44 (2015) 153–171. https://doi.org/10.1016/j.compeleceng.2015.01.019.

[33] C. Audio, C.E. Storage, C. Power, L. Basics, I. Debian, U. External, S. Devices, Raspberry Pi ® User

Guide, n.d.

[34] A.N. Ansari, M. Sedky, N. Sharma, A. Tyagi, An Internet of things approach for motion detection using Raspberry Pi, Proc. 2015 Int. Conf. Intell. Comput. Internet Things, ICIT 2015. (2015) 131–134. https://doi.org/10.1109/ICAIOT.2015.7111554.

[35] Programming the Raspberry Pi, Second Edition: Getting Started with Python, (2015) 208.

[36] P. Ganesan, G. Sajiv, A comprehensive study of edge detection for image processing applications, Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017. 2018-Janua (2018) 1–6. https://doi.org/10.1109/ICIIECS.2017.8275968.

[37] X. Liu, Z. Deng, Y. Yang, Recent progress in semantic image segmentation, Artif. Intell. Rev. 52 (2019) 1089–1106. https://doi.org/10.1007/s10462-018-9641-3.

[38] I. Journal, O.F. Engineering, P. Evaluation, O.F. Variances, I.N. Backpropagation, N. Network, U. For, H. Character, *1 *1 & 2, 6 (2017) 372–378.

[39] J. Wiley, 09/11/2017, (2017).

[40] J. Shrivastav Ravindra Kumar Gupta Shailendra Singh, A Modified Back propagation Algorithm for Optical Character Recognition, Compusoft. 2 (2013) 180–184.

[41] C.E.S.E. Al O Matan, R.K. Kiang, Handwritten Character Recognition Using Neural Network Architectures, 4th USPS Adv. Technol. Conf. 2 (1990) 1003–1011.

CURRICULUM VITAE

Auwal Salisu YUNUSA

PERSONAL INFORMATIONS

Birth of Place : Kano
Birth of Date : 05-05-1986
Nationality : Nigeria
Address : No 345 Alkali street, Sabuwar Gandu Quarter, Kano. Nigeria.
E-mail : auwalsalis1984@gmail.com
Languages : Turkish (C1), English (B2)

RESEARCHER INFORMATION

Student Orcid ID : 0000-0001-7654-3907
Advisor Orcid ID : 0000-0002-1199-2637

EDUCATION

Master Degree : "Intellıgent Plate Number Recognition System using Segmentized Method with Artificial Neural Network"
Firat University, Graduate School, Department of Mechatronic, Year of 2020
Supervisor: Dr. Öğr. Üyesi Cafer BAL

Bachelor : Bayero University Kano, Faculty Technology, Department of Electrical Engineering, 2009

High School : Dawakin Kudu Science College, Kano City, 2001

RESEARCH EXPERIENCES

- ✓ Computer Programming languages available (C-C++, MATLAB, PYTHON, vb.)
- ✓ Computer programs you can use (CAT/CAM, PROTEUS, CALCULUS, vb.)

WORK EXPERIENCE

2015 – Date: Lecturer
2010 – 2015: Lecturer

ACADEMIC ACTIVITIES