IRIS RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

ZHUANG YUAN

FACULTY OF ENGINEERING

UNIVERSITY OF MALAYA

KUALA LUMPER

2018

IRIS RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

ZHUANG YUAN

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE MASTER'S DEGREE OF ENGINEERING

[INDUSTRIAL ELECTRONICS AND CONTROL]

FACULTY OF ENGINEERIG

UNIVERSITY OF MALAYA

KUALA LUMPUR

2018

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: ZHUANG YUAN
Registration/Matric No: KQC160005
Name of Degree: MASTER OF ENGINEERING (INDUSTRIAL ELECTRONICS AND CONTROL)
Title of project Paper / Research Report: IRIS RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK
Field of Study: IMAGE PROCESSING

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature: Date:

Subscribed and solemnly declared before,

Witness's Signature: Date:

Name:

Designation:

Abstract

This research aims at designing a practical system to recognize and classify human's iris. Iris recognition is an advanced technology that is being actively researched for many decades and is gaining wider popularity. The rising of artificial intelligence provides a great opportunity to further popularize iris recognition. Convolutional neural network is a practical algorithm that is highly suitable for image processing and pattern recognition, its effectiveness and flexibility allows it to be used in many fields. This study focuses on building a system of iris recognition based on convolutional neural network, that is able to recognize human's eyes precisely and efficiently.

Abstrak

Kajian ini bertujuan untuk merekabentuk sistem praktikal untuk mengenali dan mengklasifikasikan iris manusia. Pengiktirafan Iris adalah teknologi canggih yang sedang dikaji secara aktif selama beberapa dekad dan semakin popular. Peningkatan kecerdasan buatan menyediakan peluang yang baik untuk mempopularkan lagi pengiktirafan iris. Rangkaian saraf konvolusi adalah algoritma praktikal yang sangat sesuai untuk pemprosesan imej dan pengiktirafan corak, keberkesanan dan kelenturannya membolehkan ia digunakan dalam banyak bidang. Kajian ini memberi tumpuan kepada membina sistem pengiktirafan iris berdasarkan rangkaian neural convolutional, yang dapat mengenali mata manusia secara tepat dan efisien.

Acknowledgement

Frist of all, I would like to express my special gratitude to my supervisor, Ir. Dr. Chuah Joon Huang who has given this opportunity on researching on pattern recognition on iris images in the VIP Research Lab. It is my great honor to have the valuable supervision and patient guidance from him. He has kindly helped me immensely in solving all critical situations and challenging issues throughout my project, it is indeed an unforgettable experience to study as one of his students.

Secondly, I would also like to extend my sincere thanks to my parents, for their love and support. They have encouraged me and helped me gaining my confidence to overcome all the challenges I faced.

A special gratitude to all staff of the following organization for allowing me to have their resources as the important database for my study: The National Laboratory of Pattern Recognition of Chinese Academy of Science.

Last but not least, sincere appreciation to those who have offered great help to me during my time in the University of Malaya.

TABLE OF CONTACT

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CNN: convolutional neural network

DNN: deep neural network

RNN: recurrent neural network

DBN: deep belief neural network

RBM: restricted Boltzmann machine

DTC: discrete cosine transform

HVC: hierarchical visual classification

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

Iris recognition refers to the automated process of recognizing individuals based on their iris patterns. Iris recognition algorithms have demonstrated very low false match rates and very high matching efficiency in large databases [1]. It is due to the peculiar property and different patterns of iris of every single person. Identifying individuals by their iris is not novel but it is truly being widely applied. Iris recognition is not a new technology but the theory of using iris to identify persons was proposed in the 1930s. In the 1980s, iris recognition entered commercial domains, and was gradually applied in others in 1990s.

A human eye structure includes three layers: fibrous tunic, vascular tunic and retina [2]. Vascular tunic consists of the choroid, ciliary body, pigmented epithelium and iris. Iris is able to control the diameter and size of pupil and define the color of human eye. In additional, one of the special points is that the tiny things on iris like spots and filaments make up its particular features and uniqueness, it is much more complex than structure of fingerprint. Meanwhile, a study conducted by Chinese Academy of Science shows that iris recognition is ten thousands more accurate than finger print recognition.[2] Therefore, false match of iris and even forging iris are nearly impossible.

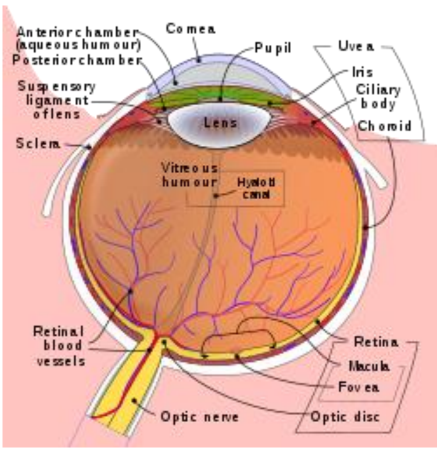

Figure 1.1: Structure of human's eye

Biometrics	Universality	Uniqueness	Collectability	Performance	Acceptability
Face	High	Low	High	Low	High
Fingerprint	Medium	High	Medium	High	Medium
Vascular	Medium	Medium	Medium	Medium	Medium
Iris	High	High	Medium	High	Low
Voice	Medium	Low	Medium	Low	High
DNA	High	High	Low	High	Low

Figure 1.2: Capability of iris recognition compares with others

Nowadays iris recognition has been widely used in many domains such as business, national defense as well as personal use. The obvious example showing that iris recognition becomes more common is Galaxy Note7 (a smart phone produced by Samsung Electrics), everyone can own a phone with iris recognition by just spending RM3000. However, there are some problems awaiting to be solved. For example, unlocking a Galaxy note7 at outdoor environment is much harder than indoor because of the effect of hard light. Therefore, developing methods to further improve flexibility safety as well as reliability in harsh condition of iris recognition are still very challenges.

1.2 Significance

It has been proven that iris recognition is the safest and most precise recognition technology, compared to others. Iris is believed to have the potential to protect the security of the next generation mobile devices and it may replace face and fingerprint in the near future. However, conventional iris recognition require a particular equipment and complex algorithm. That may cause a high and unnecessary cost and complex algorithm will also result in a low processing speed. In the last decade, the reinvigoration of research of machine learning brings new possibilities to improve performance as well as decreases costs for applying iris recognition technique. It has thus made iris recognition a trend of research recently.

In machine learning, a convolutional neural network is a class of depth, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery [3]. It is also a simple but highly efficient deep learning algorithm. The combination of both iris recognition and convolution neural network could better enhance efficiency and accuracy for recognizing irises, hence it will be a great advancement for the research of iris recognition as well.

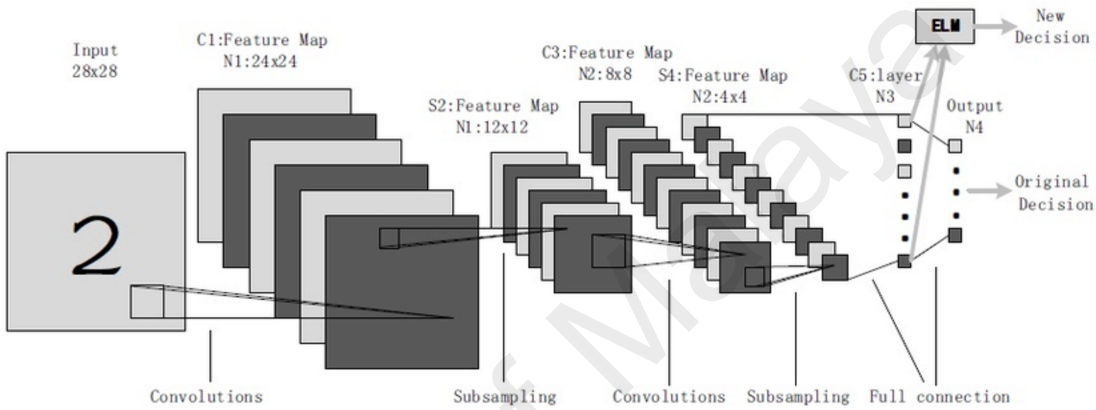

Figure 1.3: A program of a convolutional neural network

1.3 Challenges and problem statements

(1) Pupil size will change with variation of light intensity. Change of pupil size may have effects on recognition.

(2) Due to different races of human, color of eyes are different. Brown and black iris are much more difficult to recognize.

(3) High-intensity light will make a huge effect on recognition. How to enhance accuracy of recognition at outdoor area is still a great challenge.

1.4 Scope

This research project aims at designing a system to recognize and identify iris from different individuals. To improve the accuracy and reliability, the recognition system is built based on convolutional neural network. Meanwhile to seek a best method for reducing cost and solving the mentioned problem above, the system is built by using

MATLAB, and the database used in the system is sourced from National Laboratory of Pattern Recognition of Institution of Automation of Chinese Academy of Science. At the end of study, different iris from different individuals are able to be identified correctly with a minimum error.

1.5 Thesis organization

This thesis includes five chapters. Chapter 1 mainly introduces the advantages and importance of iris recognition, besides stating the problems of researching iris recognition and explaining the benefits using CNN technique to enhance performance of iris recognition.

Chapter 2 is based on previous research work of both artificial intelligence and iris recognition, describing their relationship and further discussing about more applications and characteristics using on deep learning or others, which are related to this project.

Chapter 3 is the description of the methodology used to design and develop the system that is applied at iris recognition

Chapter 4 is an analysis of all the results obtained from this system. It further discusses on weakness and strengths found during the execution of the system.

Chapter 5 concludes the research project.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter explains on previous study and research of artificial intelligence and iris recognition. It discusses about the theoretical background of understanding of development and applications of artificial intelligence and iris recognition. It is important to understand the developmental history of both technologies. Furthermore, by understanding the different working principle of different applications introduced in this chapter, readers are able to know the feature of each applications.

2.2 Deep learning

2.21 Developmental History

Deep learning is rapidly developing in the last decade. Recently it attracts a lot of public attention, i.e. AlphaGO of Google. It shows a new trend of research in deep learning with no doubt. Meanwhile, history of deep learning is still important as necessary part of understanding it. The study of deep learning has started very long ago. In 1943, W.S.McCilloch and W.Pitts introduced MCP in their thesis <A logical calculus of the ideal immanent in nervous activity>[4]. MCP is built as a simplified model according to the structure and operating principle of biological neuron, It is the first model that aims to behave like biological neuron.

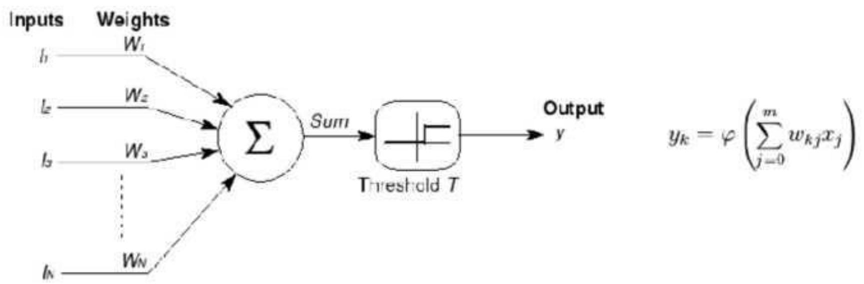

Figure 2.1: MCP model

In 1958, Rosenblatt introduced a double-layer neural network, which was named Perceptrons. It is an algorithm for supervise learning of binary classifiers and use MCP for the first time. Its first implementation in custom hardware was one of the first artificial neural networks. Based on that, Alexey Lvakhnenko and Lapa published the first generation of deep learning algorithms for supervised and multilayer perceptrons in 1965.

During the 1970's the first AI winter kicked in, the result of promises that could not be kept. The impact of this lack of funding limited both DL and AI research. Fortunately, there were individuals who carried on the research without funding[5].

In 1986, Geoffrey Hinton showed a new method called Backpropagation, this method can generate useful internal representations of incoming data in hidden layers of neural networks. This algorithm is commonly used to train deep neural networks, a term used to explain neural with more hidden layers[6]. It represented the second generation of deep learning algorithm, and this method successfully aroused the second tide of researching deep learning.

After the second AI winter, the next significant evolutionary step of deep learning started in 1999 since computers were becoming faster and more powerful. However, around 2000, the vanishing gradient problem appeared, it became a large roadblock of deep learning at that time. A few years later to 2006, Geoffrey Hinton and his student Ruslan Salakhutdinow introduced a method to solve the problem that vanishing gradient in training deep neural network. Deep learning was able to attract the interest of researchers again.

In 2012, applications of convolution neural network started to catch public attention. AlexNet, a system based on convolution neural network consecutively won the champion of several international competitions without a hitch. Thereafter, many researchers and companies started researching at convolutional neural networks.

Currently, the processing of big data and evolution of artificial intelligence are both dependent on deep learning. Deep learning is still evolving and in need of creative ideas[7].

2.22 Models of deep learning

Backpropagation neural network

The backpropagation neural network was presented by David Rumelhart and Geoffrey Hinton in their thesis<learning representation by backpropagation error> in 1986, BP uses the gradient descent and the aim is to minimise the square error between neural network's output and the expected output. A BP neural network diagram shows below.

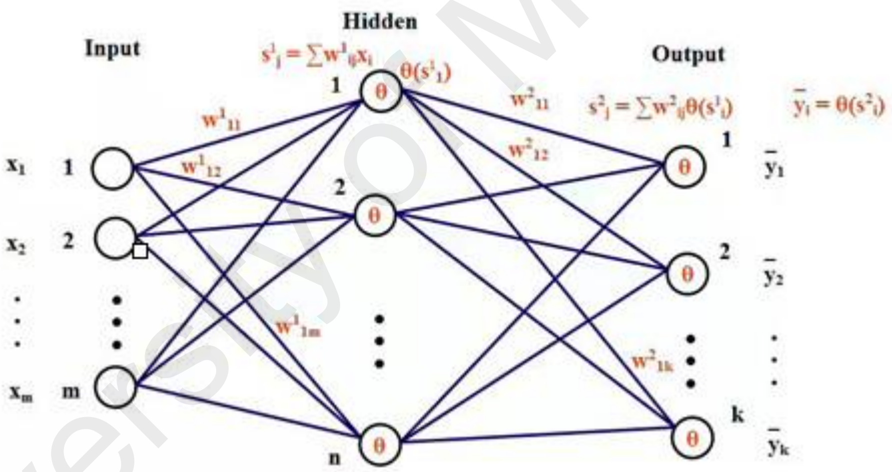

Figure 2.2: BP neural network

A simple BP neural network must include input layer, hidden layers and output layer, meanwhile hidden layers can be multi-layers. Its operating principle is that when a gourps of inputs (such as X1 to Xm) are given to input layer, them will produce a group of data(S1 to Sm) with connected weight and this a group of data will be inputs of the hidden layer. After nodes of hidden layer activate function θ, this a group of date can be expressed to $\theta(S_j)$, while S_j is the output of node j of hidden layers. These outputs of hidden layers will produce outputs of the output layer with the connected weights of hidden layers and the output layer. Thereafter, output y_j will be produced at the output layer, j represents output of node j.

The Output of this network is

$$E(\vec{w}) = \frac{1}{2}\sum_{d \in D} \sum_{k \in outputs} (t_{kd} - o_{kd})^2 \tag{2.1}$$

Here o_{kd} is the target output and t_{kd} is the output of training sample D in output layer K. Error E is a function to connect weights of each layer, it is determined by all weights.

Deep neural network

The word "Deep neural network" was created by Geoff Hinton's research group in 2006, it was a new concept different with Multi-layer Perceptron. As its name implies, the feature of DNN is its "depth", a DNN may include up to hundreds of layers. In 2014, Google created a DNN with 22 layers and won the champion of ImageNet, in 2015, ResNet improved the number of layers to 152, but just one year later to 2016, the number of layers amazingly was increased to 1207 by SenseTime. Nowadays, the number of layers are still going up.

A DNN is a typical feedforward network, which means data flows from its input layer to its output layers and never loops back, a simple structure of DNN is shown below.

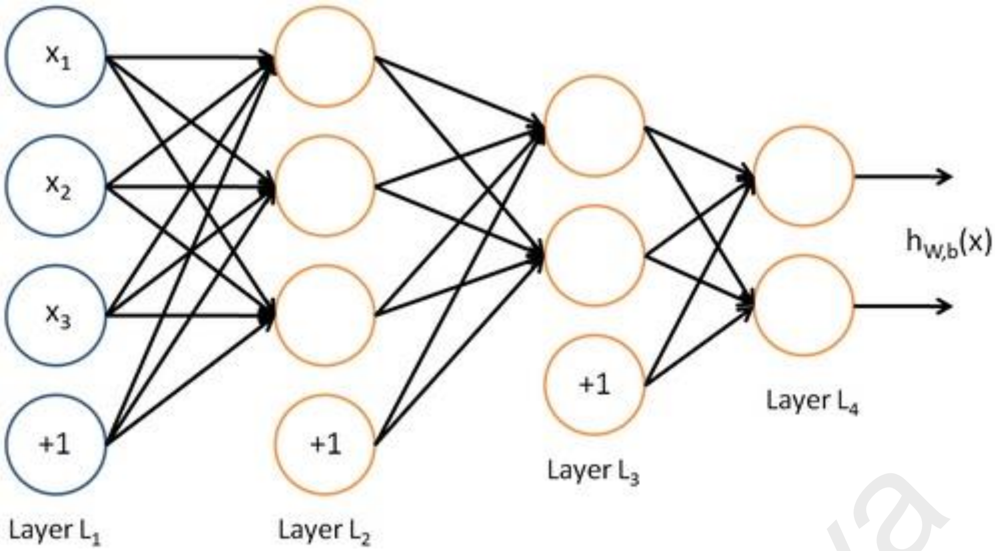

Figure 2.3: Structure of a deep neural network

DNNs can build complex non-linear relationships. Architectures of DNNs generate compositional models, these models are consisted by several composition of primitives. Lower layers allow the composition of feature to be activated by the extra layers, complex data can be modelling with less units compared to a shallow network with the same performance [8]. Meanwhile, many variants of some basic approaches are involved in deep architecture and each architecture is successfully used in specific fields.

Recurrent neural network

RNN was first created by John Hopfield in 1982. It is the class of artificial neural network that includes recurrent neural network and recursive neural network. Recursive neural networks can describe dynamic time behavior since it is different with feedforward neural networks, which only accept inputs of specific structures. The states of RNN are transferred on their own networks, so it can accept extensive inputs from time-series structures. Handwriting recognition is the earliest successful application that uses RNN, however now, speech recognition is the most relative one with RNN. The structure of RNN is shown below.

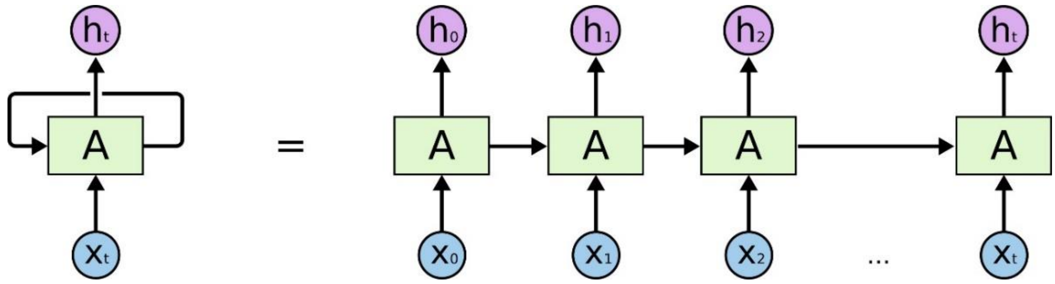

Figure 2.4: Structure of RNN

Deep belief neural network

DBN can be explained as a model generated by Bayesian probability, consisted of multilayers random variables. The two layers on the top share an undirected symmetric connection, other layers receive a directed symmetric connection from the layer above it. DBN is constructed by several cells stacking together, these cells are normally restricted Boltzmann machine (RBM) as shown in figure 2.3. The numbers of visible neurons of each RBM cell in the stack are equal to the numbers of hidden layer neurons of last RBM cell. According to the mechanism of deep learning, it is using input sample to train the first layer of RBM cell, its output can be used to train the second layer of RBM cell. In training process, after inputting encoding DBN to the top RBM cell, decoding state of top to the cell of bottom can reconstructed inputs.

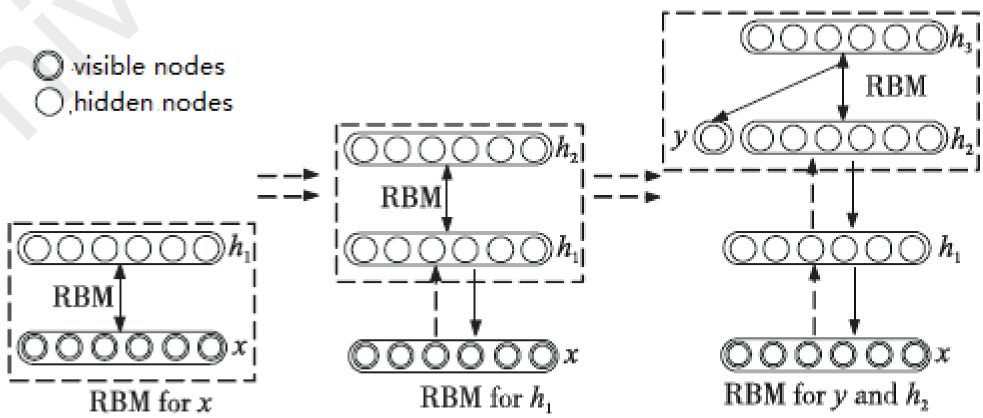

Figure 2.5: The generative process of a RBM

Diagram of RBM is shown below in figure 2.6, RBM share parameters with DBM of every layers in it is the structural unit of DBM. RBM is a special kind of Boltzmann machines, its variables are connected in limited form, only the visible node and hidden nodes are connected with weights, but there is no weight in the connection between two visible nodes or two hidden nodes.

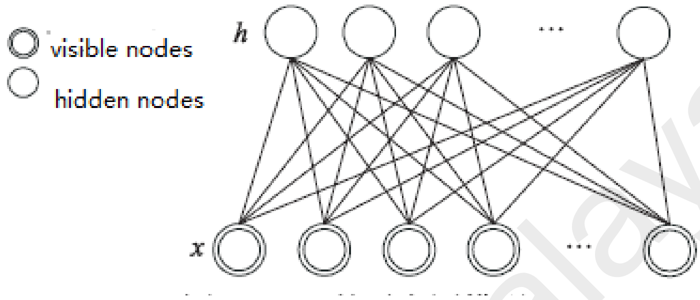

Figure 2.6: Undirected RBM model

Stacked auto-encoder network

Structure of stacked auto-encoder network is similar with DBN, both of them are constructed by structural cells stacking, but the difference is that the structural cells of stacked auto-encoder are auto-encoder instead of RBM.

Auto-encoder is a double-layer neural network, including encoding layer and decoding layer. Diagram of auto-encoder shows below, the purpose of training this model is using c(·) to make input x become function c(x), then using decoder g(x) to decode c(x) and input function $r(x) = g(c(x))$ to it. Therefore, the output of an auto-encoder is its input.

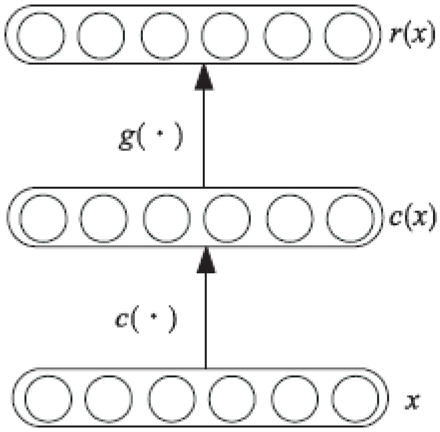

Figure 2.7: Structure of an auto-encoding model [9]

The training algorithm for an auto-encoder can be summarized to: there is a feed-forward pass made by system to compute the activations in all hidden layers for each input x, then output x' can be obtain at output layer. Meanwhile, the deviation of x' will be measured, errors will also be propagated backward the net, performing as the weight update. [9]

2.3 Iris recognition

The first significant plan proposing iris to identify individuals was presented by two ophthalmologists Leonard and Aran Safir in 1987. During the year 1993, Dr.John G.Daugman created an algorithm that is able to recognize human's iris, it is the pioneer at the field of iris recognition. Nowadays, applications of iris recognition are various, their recognizing methods are also different, but their identifying processes are almost same, a diagram of iris recognition process shows below.

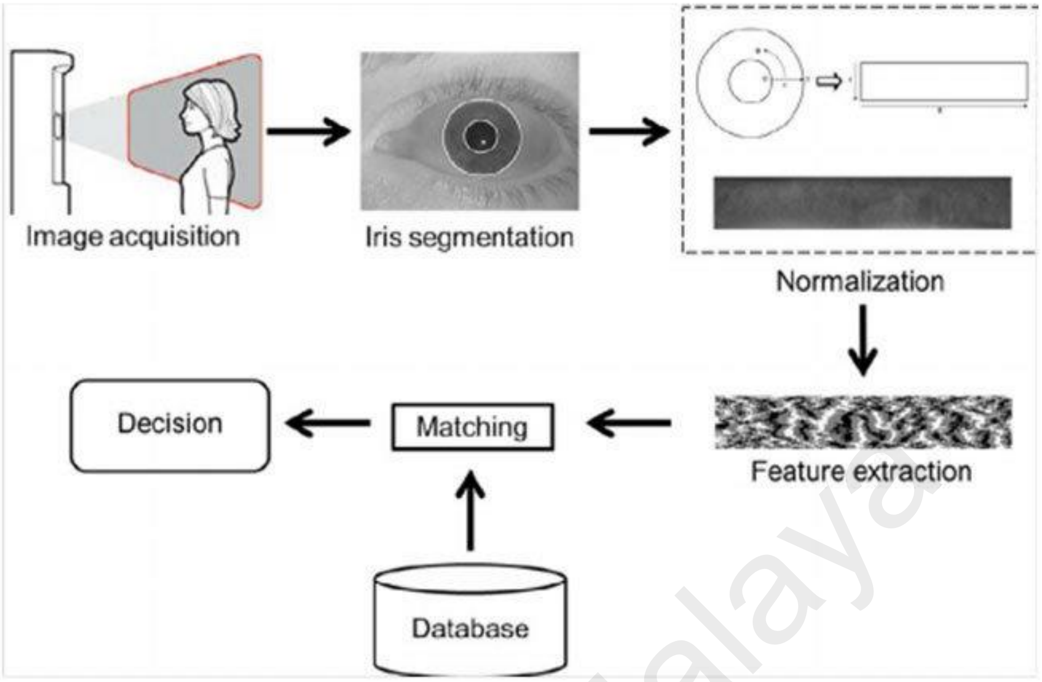

Figure 2.8: A process of iris recognition

Identifying iris is comparing the iris feature collected with iris feature in iris library. If the iris collected can be matched with one of irises in the library, the program ends and the match successes. Inversely, when the match fails, the program ends. As the frame shown in figure 2.9, it assumes that the number of databases in the library is M, setting the searching times as N. Once N is equal to M, that means there is no database in the library can be matched with the iris collected and the program ends.

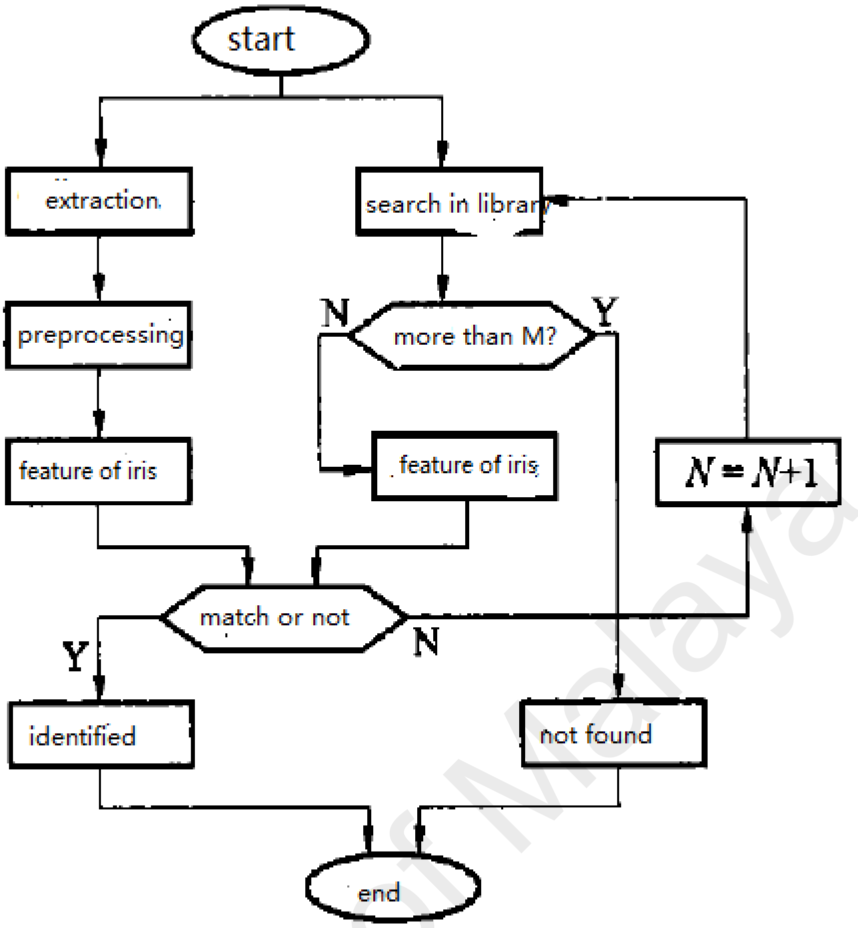

Figure 2.9: Flow diagram of identifying iris

2.31 Image processing of iris recognition

Image processing is highly necessary for computers to recognize iris, it is not only making the feature of iris image easier to be collected, but also provides a standard procedure for iris recognition.

Preprocessing and Iris Localization

The method preprocessed all collected images of iris pattern by using red channel, since wavelengths of red light are the longest in human's visible spectra. Then it uses classic Daugman's Intergro-differential operator for extracting the iris region, based on Gaussian smoothing function. After the processing of operator, the images of iris has been processed to localize.

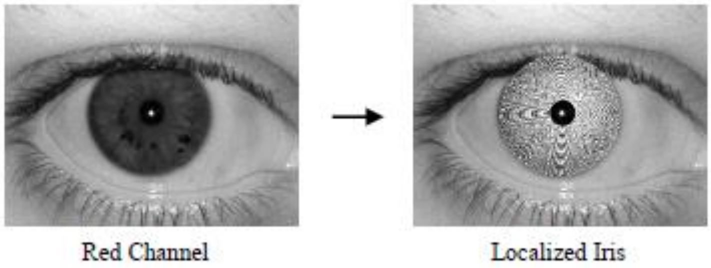

Figure 2.10: Localizing iris [10]

Eyelid Suppression

To reduce the effect made by eyelashes and enhance the accuracy when recognizing iris, the process of suppressing a part of iris which is not exactly circular and covered by the eyelids is in need.

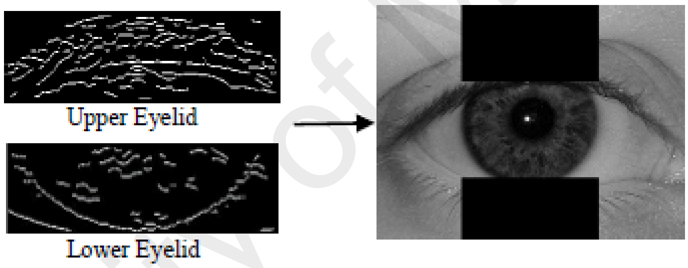

Figure 2.11: Eyelid suppression [10]

Normalization

For successfully segment the iris part and suppressed eyelids, the Daugman's rubber sheet model is necessary.

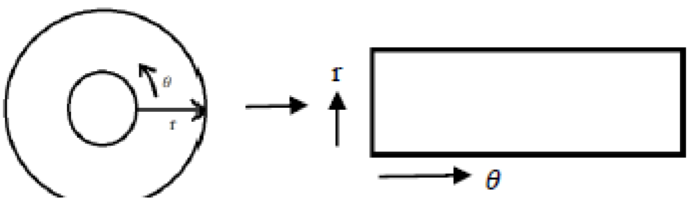

Figure 2.12: Segmenting circular ring by Daugman's rubber sheet model

Remapping model shown below

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$$

(2.2)

and

$$x(r,\theta) = (1-r)xp(\theta) + rxi(\theta)$$

$$y(r,\theta) = (1-r)yp(\theta) + ryi(\theta)$$

(2.3)

Where $(x,)$ is the region of iris, $(y,)$ are original Cartesian coordinates and $(r,)$ are corresponding normalized polar coordinates, $(xp,)$ and $(xi,)$ are the central coordinates of pupil and iris boundary along the θ direction[10].

Finally, iris image has been processed to the diagram shown in figure 2.13.

Figure 2.13: A segmented iris pattern

Once all databases are segmented and normalized and noises are removed, the part of extracting the relevant texture and intensity information to train a classifier can be operated.

These flows of image processing are used by nearly all project of iris recognition. Results show that histogram equalization improving the recognition and training accuracy are finally increased over 2%.

2.32 Deployed applications of iris recognition

The United Arab Emirates is the first nation that employs iris recognition technology for border control, in 2003, the tracking system starts to be operated. Today, it has been used at the entire land of UAE, including air and sea ports of entry and requiring all foreign nationals. All visitors will be checked through a camera with iris recognition. The system apprehended more than 343,000 persons who was trying to re-enter the UAE with either another nationality or name, or even fraudulent travel documents.

In India, an iris recognition system named Aadhaar start operation at 2011, more than one billion Indian citizens gain the benefits bring by Aadhaar, Indian government is still making their efforts for slogan which is “to give poor an identity.” now, this project can enroll around 100000 persons per day, across over 36000 stations conducted by 83 agencies. Based on a report of Indian sociologist, it really has a positive effect on enhancing social inclusion.

To reduce the crime rate and manage floating population, Police forces across America plan to start using a Mobile offender recognition and information system created by BI2 Technology in 2012, this system includes iris recognition, simply takes 3 seconds to identify a person. New York City police department is the first that installs this system.

The Samsung Galaxy Note 7 is the first smartphone that carries the function of iris recognition, it was unveiled at 3rd August 2016, released on 19th August 2016 offically. Samsung has been conducting research on Iris scanning of phone for many years but now this is the first time Samsung actually applies the technique in a major phone. Marketing supervisor of Samsung says it is more secure than fingerprint scanning, since it is almost impossible to spoof, it only works with living tissue and can not be tricked by a high-resolution photos of your iris. Thus, unless someone knocks you out and then holds your unconscious head up to the Note 7's iris scanner, it is relatively safe[11].

2.4 Feature extraction

Techniques of iris recognition can be mainly linked to Discrete Cosine Transform (DTC), Wavelet, Zernike, Dual tree complex Wavelet and Hierarchical Visual classification.

2.41 DCT technique

DCT is a Fourier-related transform which is usually used in signal and image processing especially DCT-II due to its strong “energy compaction” property.

A specify DCT-II can be expressed as

$$X_k = \sum_{n=0}^{N-1} x_n cos[\frac{\pi}{N}(n+\frac{1}{2})k] \tag{2.4}$$

A thesis <*A Iris Recognition Algorithm Based on DCT*>[12]published in 2007 introduces DCT technique used on iris recognition. After preprocessing and segmented, authors used DCT algorithm to collect the features of iris pattern. DCT transform can be modelled as: set $f(x)(x = 0,1,2,3 \ldots N-1)$ is a discrete signal in time domain, the one-dimensional DCT is

$$F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad u = 0,1,2 \cdots, N-1$$

$$F(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \tag{2.5}$$

$F(u)$ is the DCT coefficient, it is clear to understand from equations above, their DC component of the discrete signal in frequency domain is same with Fourier transform, it is the average value of time domain signal.

In the process of feature extraction of iris. Authors further transformed the coefficient transformed by DCT to a characteristic matric, texture features of iris were mainly distributed closed to pupil, thus features further from pupil were less.

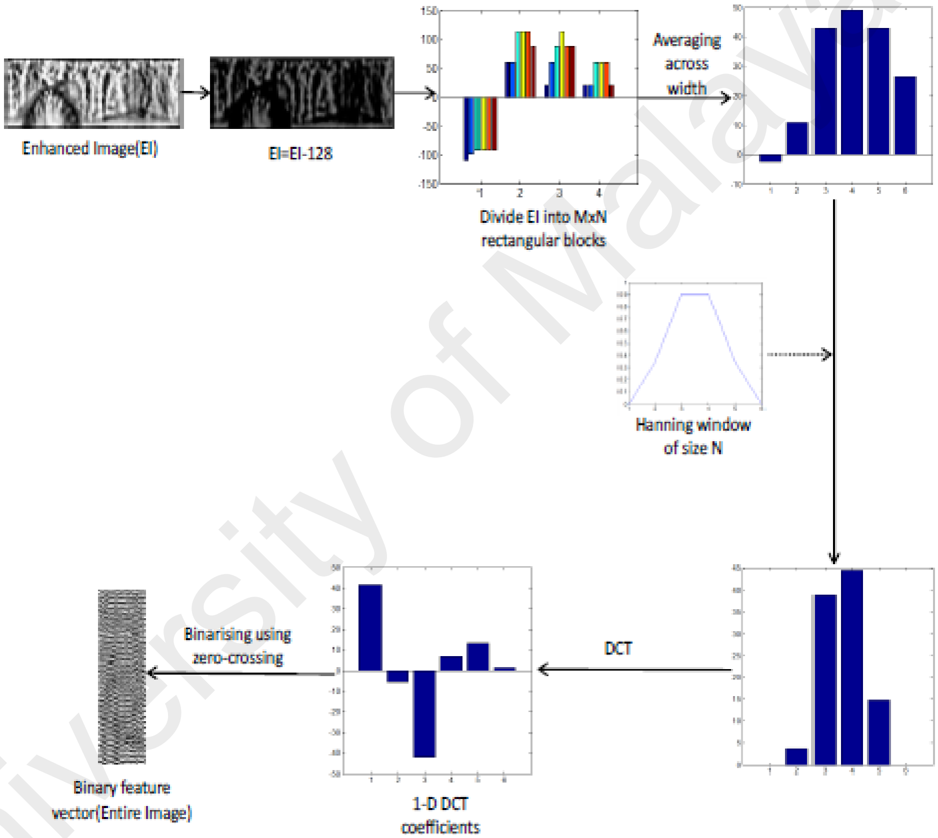

Figure 2.14 Process of feature extraction by DCT[12]

After training 63 times of identify, they gained the result shown the accuracy is 92% and processing time is 7.831s. Comparing to the result obtained by wavelet transform, DCT has a faster processing speed and higher accuracy.

2.42 Zernike polynomials

Zernike polynomials are a sequence of polynomials that are orthogonal on the unit disk. They are introduced by Zernike in 1934 [24], named as Zernike polynomials in 1954. Recent years, the Zernike moments have been successfully applied to image analysis and processing, and being widely used at many application of iris recognition since it can greatly increase accuracy of recognizing iris. Zernike polynomials are shown below

Even ones are

$$Z_n^m(\rho,\varphi) = R_n^m(\rho)cos(m\varphi) \tag{2.6}$$

Odd ones are

$$Z_n^{-m}(\rho,\varphi) = R_n^m(\rho)sin(m\varphi) \tag{2.7}$$

When m and n are positive and n $\geq$ m, φ is the azimuthal angle and ρ is limited at $0 \leq \rho \leq 1$.

Radial polynomials are

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} \frac{-1^k(n-k)!}{k!\left(\frac{n+m}{2}-k\right)!\left(\frac{n-m}{2}-k\right)!}\rho^{n-2k} \tag{2.8}$$

In the thesis <*Zernike's feature descriptors for Iris Recognition with SVM*>[13], authors useed Zernike's feature descriptors based on Zernike moments for feature extraction. After preprocessing and normalizing, databases were processed to an enhanced image shown below

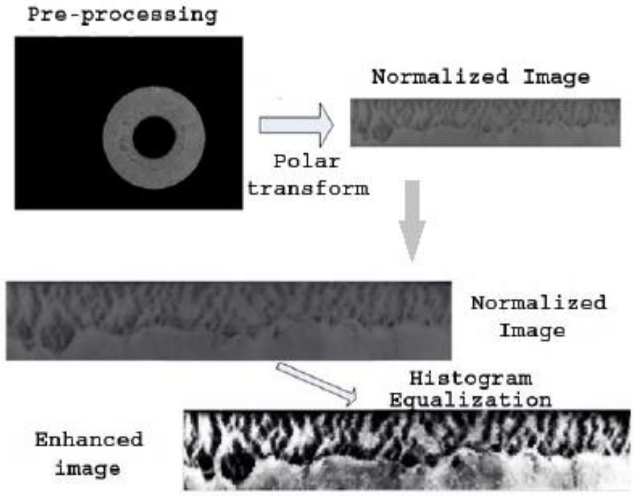

Figure 2.15: Preprocessing and normalizing iris[13]

At the part of feature extraction, in order to get the Zernike moment Z_{pq}, author used transformation $x' = \frac{(x-cx)}{M}$ and $y' = \frac{y-cy}{M}$, where the M was max(width, height) and *(x,y)* were original coordinates. When it satisfied the condition$(x')^2 + (y')^2 \leq 1$, the Zernike moments were computed by those new coordinate. Then they created the feature vector with different numbers of moments and order the order moments from lower to higher. [13]

After training and simulation, they gained the good results, i.e. by using Zernike moments is 99.36% with 130 moments while pseudo-Zernike is 99.89% with 90 moments.

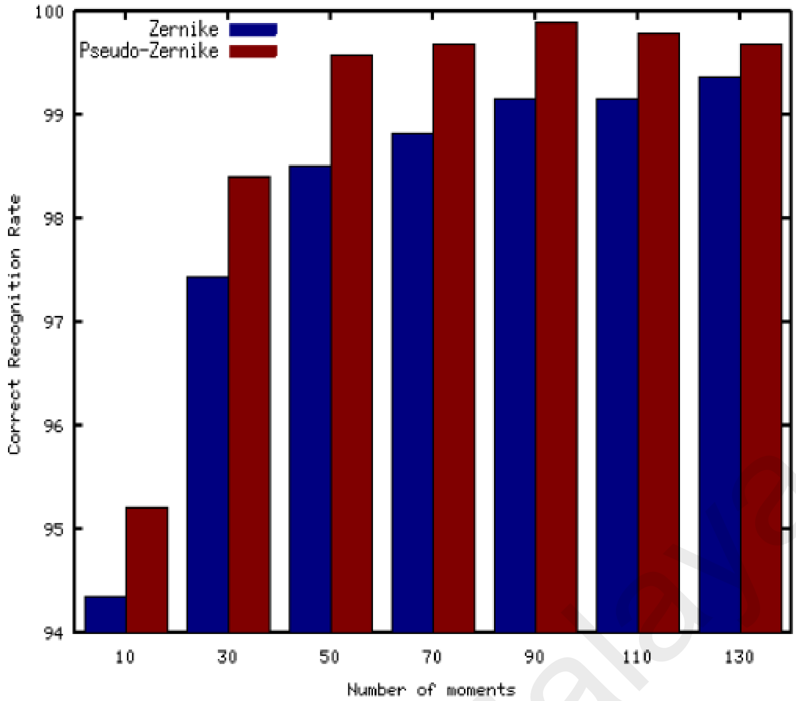

Figure 2.16: Chart of comparing successful rate of both Zernike and pseudo-Zernike [13]

2.43 Hierarchical Visual classification

HVC is a novel method to represent texture pattern of iris, it aims at encoding the texture primitives of iris images. HVC method is to integrate two Bag-of-words models together, which are Locality-constrained Linear Coding (LLC) and Vocabulary Tree (VT). HVC is able to takes advantages of both LLC and VC for expressing iris pattern accurately, and it employs a coarse-to-fine visual coding strategy. New experimental results show that it has the moderate processing speed compared to others technique, but accuracy of both in case of near iris recognition and far iris recognition. Furthermore, it is also capable of detecting the live iris.

This method was not commonly used, but thesis <*Iris Image Classification Based on Hierarchical Visual Codebook*> [14] described this method to extract feature of iris pattern in detail, the authors have also provided a chart to explain the flow of processes, as shown in figure 2.17.

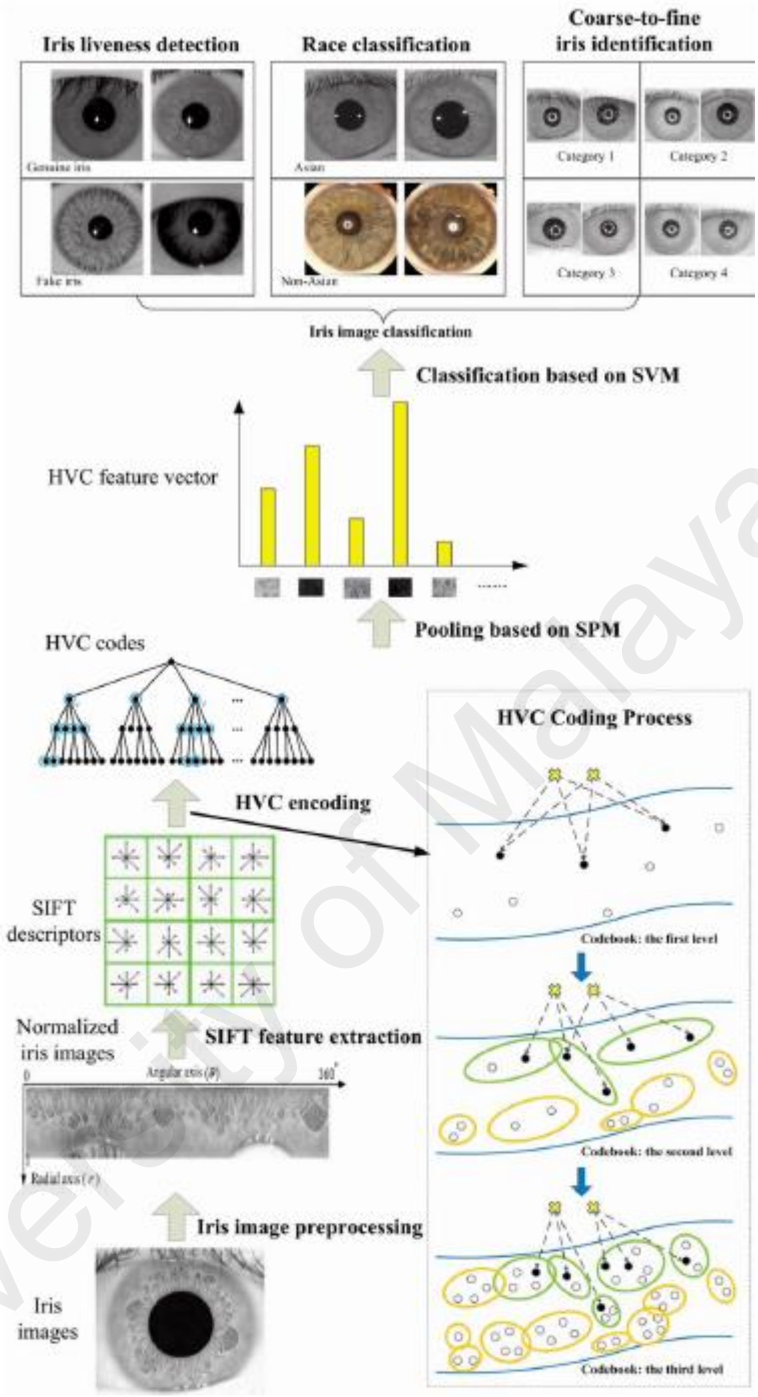

Figure 2.17: Simple process flow of HVC [14]

CHAPTER 3: METHODOLOGY

3.1 Introduction

Based on the discussion in Chapter 2 Literature Review, it clearly specifies the procedures of iris recognition. Convolutional neural networks have been used because of its high efficiency and accuracy. CNN is a deep learning algorithm, which is highly suitable for image recognition, and it will be described in detail in Chapter 3.3. CASIA-irisV4 has been chosen as the database for this study. It is published by The National Laboratory of Pattern Recognition of Chinese Academy of Science. It includes more than ten thousands iris images from 1050 eyes, which is sufficient for this study.

This study includes images processing and modelling in CNN. Due to the feature of CNN, some complex parts of images processing can be omitted, but for better enhance accuracy and efficiency, a simple images processing is still in need.

3.2 Images processing

Preprocessing

First of all, some suitable cases from CASIA-irisV4 was selected for this study. The reason of choosing them is mainly that iris patterns in these images are clearer. Cases selected are 002, 006, 012, 018, 119, 144, 160, 183, 192, 206, 216 ,239, 287, 294, 310, 333, 339, 381, 406, 407, all including both left and right eyes and 20 images of each eyes, totaling 800 images.

Because Images from Database CASIA-irisV4 are preprocessed by red channel, in this part, images were segmented to a suitable size. For avoiding the potential effects made by eyelid and others in orbit to the result of this study, segmenting images was only conducted along eyeballs, shown below.

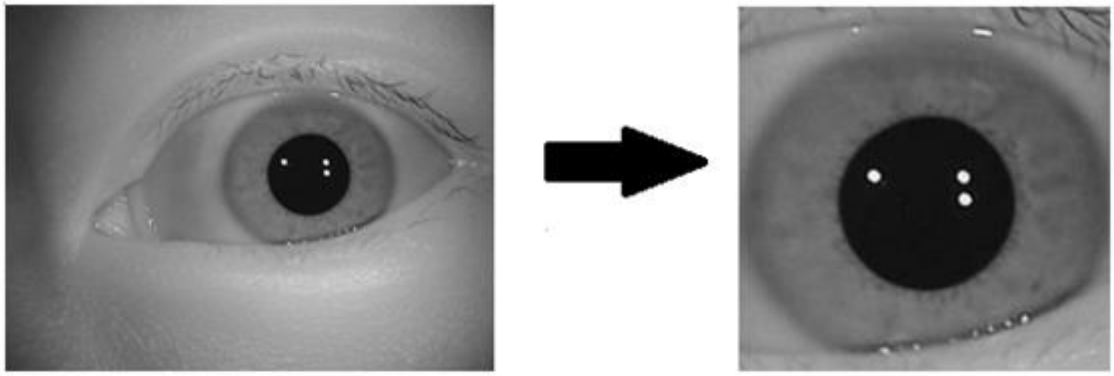

Figure 3.2: Simple segment of iris images

The complex segment process was not applied because of a high time cost as well as an unexpected lower recognition accuracy, it well be explained at Chapter 4.

3.3 Convolutional neural network

3.31 Structure of convolutional neural network

As shown in figure 3.5, a typical convolutional neural network is mainly consisted with input layer, convolutional layer, pooling layer (subsampling layer), fully connected layer and output layer [28].

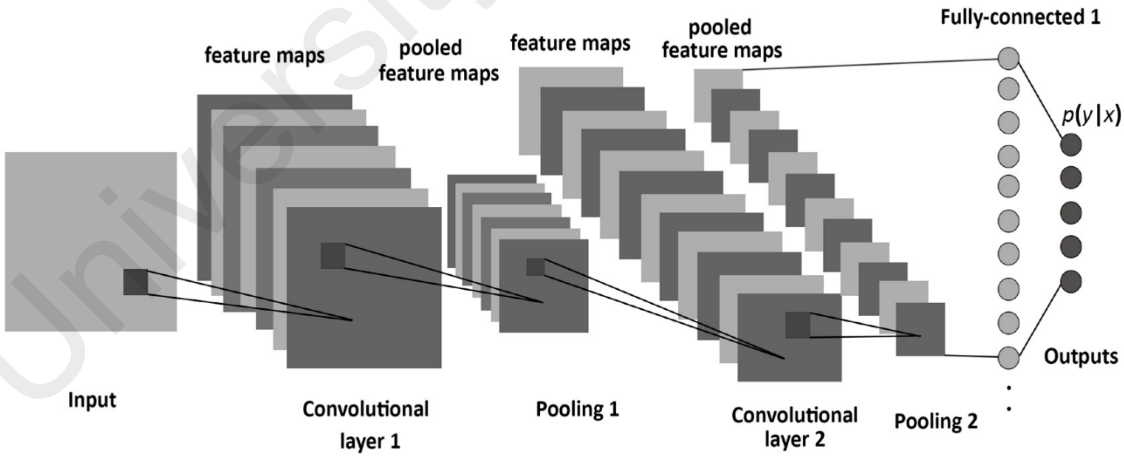

Figure 3.5: A structure of a typical CNN

Assuming the original input image is X and use x_j^l to represent the output maps of layer l of convolutional neural networks, if x_j^l is one of the convolutional layers, the equation to get it is that

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l)$$

(3.1)

Here k_{ij}^l is the weight of kernel or filter of layer l, M_j represents the selection of input maps, symbol * means operating convolution to both feature map of layer l-1 and filter matrix. Finally, x_j^l can be gained by adding an additive bias b_j^l to the convolution of feature map and weight.

Pooling layers are normally after convolutional layers. A pooling layer produces downsampled versions of the input maps, the main functions of pooling layers are downsampling the feature map of convolution layer, so if x_j^l is pooling layer, it is

$$x_j^l = f(\beta_j^l down\left(x_j^{l-1}\right) + b_j^l)$$

(3.2)

Passing through multiple convolutional and pooling layers alternately, convolutional neural networks use the fully connected network to classify the extracted features and get the Probability distribution Y based on inputs. As equation shown below, the function of convolutional neural networks is actually making the original input matrix transform to a new mathematical model Y to represent feature by multiple levels of data transformation or dimension reduction.

$$Y_j^l = P(L = l_l | X; (W, b))$$

(3.3)

The training purpose of convolutional neural networks is to minimise the loss function of network *L(W,b)*, *W* is weight for the system and b is additive bias. The error occurred between the expected value and the value calculated by loss function is called "residuals".

The common loss functions are also represented by the Mean Squared Error (MSE) function, the Negative Log Likelihood (NLL) function, etc.

$$MSE(W,b) = \frac{1}{|Y|}\sum_{l=1}^{|Y|}(Y(l) - \hat{Y}\,(l))^2 \tag{3.4}$$

$$NLL(W,b) = -1\sum_{l=1}^{|Y|} log\, Y(l) \tag{3.5}$$

In order to alleviate the problem of overfitting, the final loss function usually controls the overfitting of weights by adding the L2 norm, and controls the intensity of overfitting by the parameter λ (weight decay).

$$E(W.b) = L(W,b) + \tfrac{\lambda}{2}WW^T \tag{3.6}$$

During the training process, the gradient descent method is largely used as an optimization method. The residuals are backpropagated through a gradient descent, updating the training parameters (W and b) of each layer of the convolutional neural network layer by layer. The learning rate parameter (η) is used to control the strength of backward propagation of residuals[16].

$$W_l = W_l - \eta\frac{\partial E(W,\ b)}{\partial W_l} \tag{3.7}$$

$$b_l = b_l - \eta\frac{\partial E(W,\ b)}{\partial b_l} \tag{3.8}$$

3.32 Operational principle of convolutional neural network

The operational principle of convolutional neural networks can be mainly distinguished to three parts: network model definition, training network, and reasoning.

Network model definition. The network model is defined based on the network depth, the function of each layer of the network, and setting the parameters such as λ and η. There are many studies on the model design of convolutional neural networks, such as model depth, convolution step size, excitation function, etc.

Network training. training the parameters is done by the back propagation of residuals. However, problems such as overfitting in network training and gradient disappearance and explosion have greatly affected the performance of training. For solving the problems of training network, some effective methods are proposed, such as random initialization of network parameters based on Gaussian distribution, initialization using pre-trained network parameters, parameters for different layers of convolutional neural networks Independent and identically distributed initialization.

Reasoning. The process of reasoning convolutional neural networks is to convolute and subsample the input data, producing the feature maps of each layer, and finally the high-level reasoning is processed via fully connected layers. Outputs of this process can show clearly the probability distribution of original input, to speculate what the input exactly is.

To more specifically describe the operational principle of convolutional neural networks, the descriptions below explain in detail the function and working principle of each layer of convolution neural networks.

Convolutional layer: this layer can convolute the input with filters and output the features. As the figure shown in figure 3.6, the input is a 7x7 image, the filter is 3x3 volume, the process to get the first output number (shown as red) is convoluting the numbers in the first 3x3 of input with numbers in filter one by one and add them together. The next part moves one step to the second 3x3 image patch at right, the output number after convolution is shown as green in the output volume. Then every 3x3 image patches of input will convolute with filter and output to relative position. Finally, the output is a 5x5 volume. Normally there are multiple filters to convolute, so the next step is using different filters to gain different feature maps.

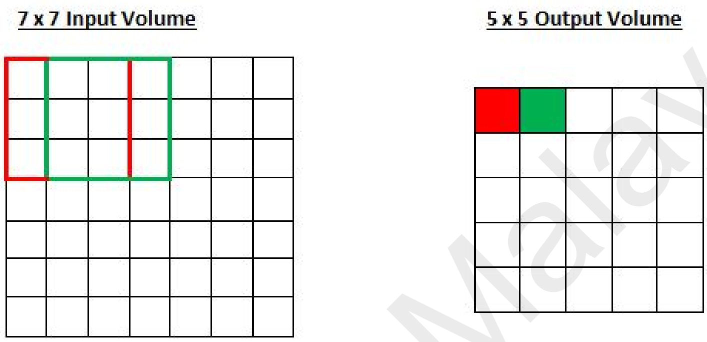

Figure 3.6: Operational principle of convolutional layers

Pooling layer (Subsampling layer): the function of pooling layers is basically subsampling input images, that is why a pooling layer can be called subsampling layer as well. Figure 3.7 shows a max pooling with 2x2 filter, max pooling is the method that compares the numbers in every 2x2 patches and outputs the maximum numbers of each patch. [17].

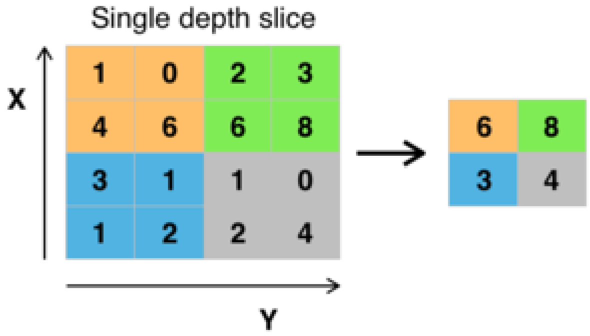

Figure 3.7: Operational principle of pooling layers

Fully connected layer: it can collect all features and output a probability distribution to the output layer. As the figure shown below in Figure 3.8, every block of this layer are fully connected with all feature maps of the previous layer (pooling layer 2), they are treated as a single list and all treated identically. Each block provides a probability value on what the input exactly is.

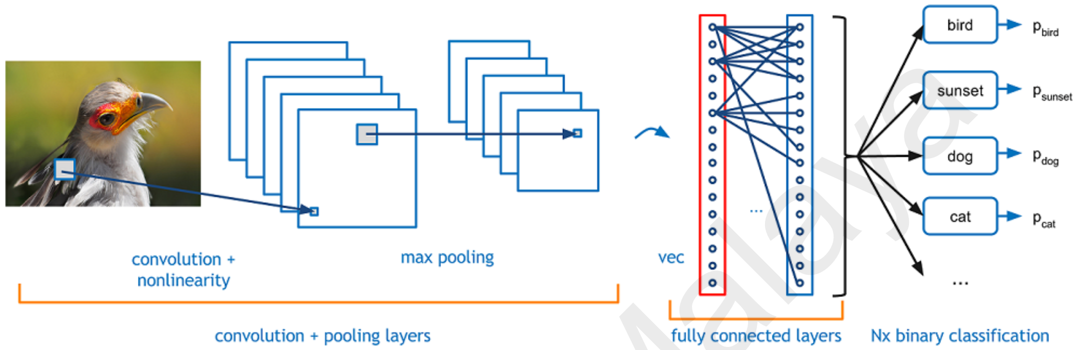

Figure 3.8: Operational principle of fully connected layers

3.33 Modeling and programming

For getting an ideal result to continue this study, MATLAB was used for programing the code in this study. After image processing and understanding foundation of convolutional neural networks, the main program was divided into 4 parts: data loading, constructing CNN, training and testing, outputting tested result.

Data loading

Computers are differentfromhumans when it "sees" an image, information that it loads is a group of matrices of numbers instead of colors. Therefore, transforming these images to a fixed form that can be received by computers as well as convolutional neural networks is a necessary part. A standard convolutional neural network normally accepts 28x28 size images as input, "imresize" function was applied to resize all input images.

After successfully loading input, the entire 800 input images of this study was separated into two parts. First, all inputs were combined into one series, using "randperm" function to randomly arrange them. The system selected first 700 images for training and the rest of them are used for testing. The reason why choosing random sampling all inputs instead of using fixed sample for training or testing is to show a more clear result without control.

Constructing CNN

The structure of a typical convolutional neural network is 6c-2s-12c-2s, which means two convolutional layers and two pooling layers [29]. The first convolutional layer includes six filters and output six feature maps, the second convolutional layer will receive the 2x2 subsampled form of these six feature maps from the first pooling layer. After convoluting with twelve filters of the second convolutional layer, they will be sent to the second pooling layer to proceed the second time of subsampling. This network structure will process an input image from 28x28 size to 12 feature maps and 4x4 size each, these 4x4 size images represent different features of input images, sent to fully connected layer and determined the tested result.

In this study, a typical 6c-2s-12c-2s convolutional neural network was constructed. From introduction of CNN, the output formula of both convolutional layers and pooling layers was mentioned, in order to set up a convolutional neural network, more calculation of parameters is still in need.

Sensitivities of units of convolutional layers is an important parameter to calculate the gradients of kernel weights. Meanwhile, the gradients are keys for CNN calculation in MATLAB. Therefore, in order to compute gradients, sensitivities are in need. From backpropagation algorithm, the formula of sensitivities is given, as shown below.

$$\delta_j^l = \beta_j^{l+1}(f'(u_j^l) \circ up(\delta_j^{l+1})) \tag{3.9}$$

where $up(\delta_j^{l+1})$ means the upsampling operation, δ_j^l is the sensitivities of convolutional layers. the upsampling operation can be also represented by Kronecker formula, their function is all tiles each pixel in the input *n* times in the output when the subsampling layer subsamples by a factor of *n*. Kronecker formula is shown below.[18]

$$up(x) \equiv x * l_{n\times n}$$

(3.10)

Based on the formula of sensitivities, the bias gradient can be immediately computed, shown below

$$\frac{\partial E}{\partial b_j} = \sum_{u,v} (\delta_j^l)_{uv}$$

(3.11)

Finally, gradients for kernel weights can be computed by using backpropagation, as shown below.

$$\frac{\partial E}{\partial k_{ij}^l} = \sum_{u,v} (\delta_j^l)_{uv} (p_i^{l-1})_{uv}$$

(3.12)

where $(p_i^{l-1})_{uv}$ is the patch in x_i^{l-1}, it is multiplied elementwise by k_{ij}^l during convolution for computing the element at *(u, v)* in the output convolution map x_i^l.[19]

The formula to get sensitivity maps is different when applying in pooling layers. Assuming the pooling layer are fully connected with the convolutional layer behind, the sensitivity maps of pooling layer can be computed by backpropagation equations, shown below.[20]

$$\delta_j^l = f'(u_j^l) \circ conv2(\delta_j^l, rot180(\delta_j^l), 'full').$$

(3.13)

In this equation, the kernel is rotated to make the convolutional function perform cross-correlation, "full" means a full convolution border handing, it will pad the missing inputs with zero in case of the number of inputs to a unit of layer is not the full size of kernels.

As the same with calculation of convolutional layers, the gradient of kernels can be computed with sensitivity maps and original downsampled maps.

$$\frac{\partial E}{\partial \beta_j} = \sum_{u,v} (\delta_j^l \circ d_j^l)_{uv}$$

(3.14)

where the d_j^l is downsampled maps, it can be computed by

$$d_j^l = down(x_j^{l-1})$$

(3.15)

These formulas are necessary for constructing the structure of convolutional neural networks in MATLAB. The completed code for convolutional neural network is able to adjust the training times (epoch), learning rate and batch size (code shown in appendix A).

Training and testing

Training convolutional neural networks is an efficient method to enhance the performance of convolutional neural networks, training times (epoch) is an important parameter that directly defines the accuracy during recognition, more training times will generally contribute to a high accuracy when testing. In this study, training times can be adjusted, it is flexible for modelling a high accuracy recognition.

During training, parameters of convolutional neural networks are initialized to different small random numbers, this is for ensuring network will not be a saturation condition as parameters oversize, meanwhile, the random numbers make sure the learning of network without problems. The training algorithm is similar with backpropagation, it mainly divides into two phases, one is feedforward pass and another one is backpropagation pass. Outputs of feedforward pass can be obtained by the equation below. The second phase using backpropagation is for figuring out the error between actual output O_p and expected output Y_p, accordingly to minimize the error and adjust the relative weights.

$$O_p = F_n(\dots(F_2(F_1(X_p W_1) W_2)) \dots W_n) \tag{3.16}$$

In the phase of feedforward pass, assuming the error of n training case is E^N, so the equation to obtain it is

$$E^N = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}(t_k^n - y_k^n)^2 \tag{3.17}$$

where t_k^n means the *k* dimension of case *n* at actual outputs, y_k^n means the *k* dimension of case *n* at expected outputs. Further, E^N can be also represented as the equation below when assuming there is only one case.

$$E^n = \frac{1}{2}\|t^n - y^2\|_2^2 \tag{3.19}$$

However, in a conventional neural network, it is necessary that the partial derivatives of every weight relative to error *E* must be calculated. assuming that it happens at layer l. so the output of layer l is.

$$x^l = f(u^l), with\ u^l = W^l x^{l-1} + b^l$$

(3.20)

where f means outputting activation function and W means matrices of weights, b means bias.

In the phase of back propagation pass, the sensitivity of bias mentioned above are needed to be considered, using it to represent the error of backpropagation, the equation is shown below

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial b} = \delta$$

(3.21)

Since $\frac{\partial u}{\partial b}$ equals to 1, So the sensitivity of bias are equivalent to the derivative of the error divided a unit's total input. Otherwise, the equation below can be used to represent the sensitivity of layer l.[21]

$$\delta^l = (W^{l+1})^T \delta^{l+1} \circ f'(u^l)$$

(3.22)

For the error function (l), the sensitivities for the neurons of output layer is gained with a slightly different form:[21]

$$\delta^L = f'(u^l) \circ (y^n - t^n)$$

(3.23)

Therefore, the delta rule for updating a weight is now available to every single neuron, equations are shown below.[21]

$$\frac{\partial E}{\partial W^l} = x^{l-1}(\delta^l)^T$$

$$\Delta W^l = -\eta \frac{\partial E}{\partial W^l}$$

(3.24)

The meaning is the partial derivative of the error E with weights in layer l is equivalent to multiplication cross of input with the sensitivity of every single neuron. Based on these equations, a toolbox has been introduced in for training CNN automatically.

Outputting tested result

In this part, an accuracy calculation function is required, Thus the code to calculate accuracy including total accuracy and specific accuracy was created, transforming the specific accuracy to a series for easily displaying with figure and bar char (code is shown in appendix A).

3.4 Summary

To enhance the accuracy of recognition, it is required to increase the train times, and the efficiency of recognition mostly depends on the quantity of input images and code. In this study, in order to gain a high accuracy as well as simplify the program and input images to raise efficiency, many parameters have been adjusted. In Chapter 4, the result of this study will be presented.

CHAPTER 4: RESULT AND DISCUSSION

4.1 Introduction

This chapter mainly introduces the results obtained from this study, defining advantages and weakness of this practical method, seeking a better method to enhance the accuracy as higher as possible at the mean time.

In this study, there were 800 images of eyes have been used, belonging 20 individuals and including both left and right eyes. During training phase, the system randomly selected 700 images as input for training, the rest 100 images were used for testing. Due to random selection, there were two cases of eyes were accidentally not in the test list. Therefore, real-time recognition was designed to recognize single to multiple cases immediately.

4.2 Result displaying

Firstly, compared both accuracy rate of segmented and unsegmented iris images, it is found that the accuracy rate of segmented iris images is slightly lower than unsegmented images. This work aims at developing a practical system with high recognition rate, therefore the results shown below are tested using unsegment iris images.

As mentioned above, the accuracy rate of recognition is immediately related to the quantity of training times (epoch).

Training times (epoch) was set to 100 times, the total accuracy rate was 0%, specific accuracy rates of each case were all 0%, results are shown below. It means 100 times training is not sufficient for this convolution neural network to understand and further to recognize iris, the figure of training MSE is shown below at Appendix C.

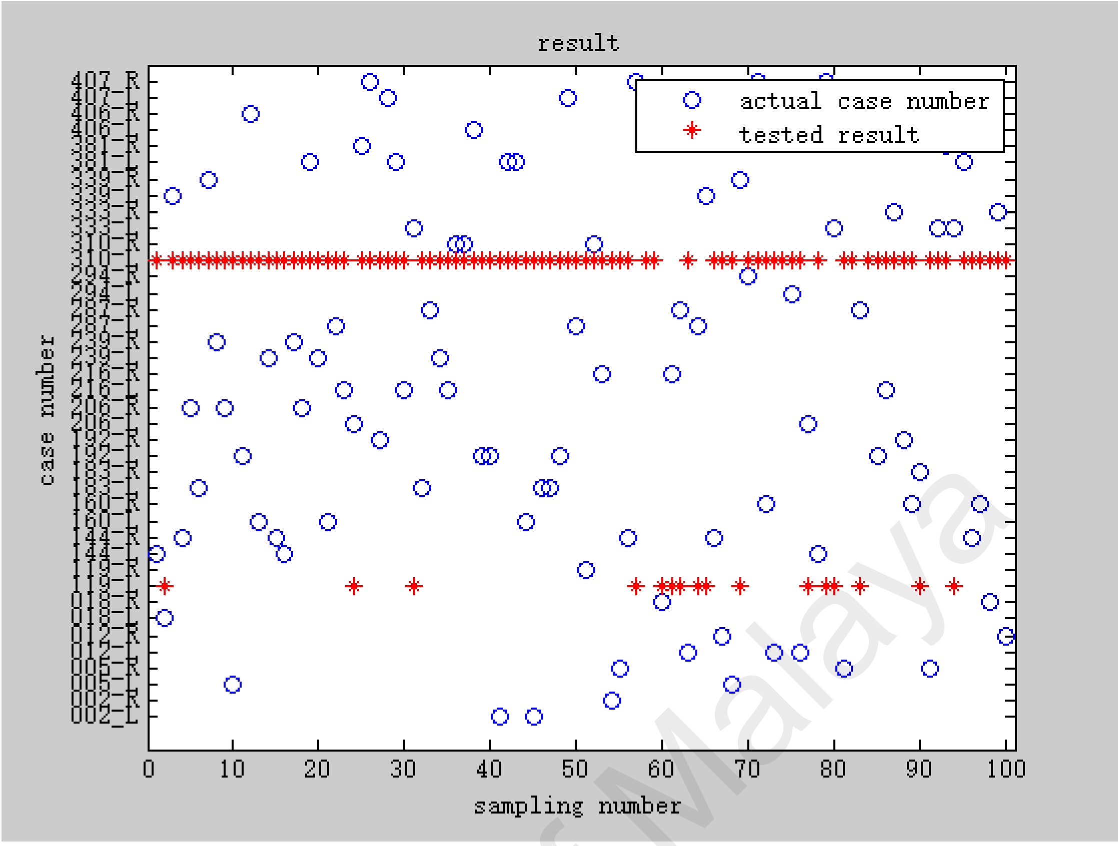

Figure 4.1: Tested result of 100 times trained CNN

Secondly, number of training times was adjusted to 500 times. Eventually, the total accuracy rate has increased to 48%, which means haft of tested data can match the actual case numbers.

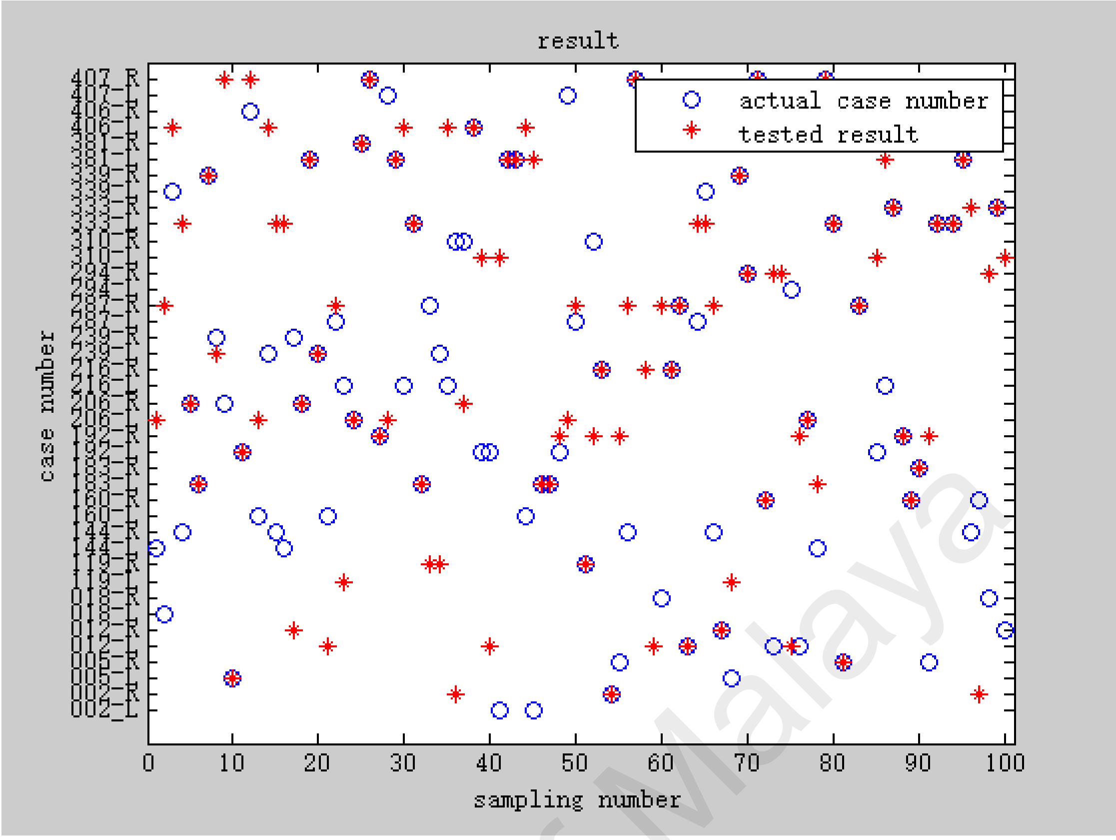

Figure 4.2: Tested result of 500 times trained CNN

The specific accuracy rates of each case were increased to different levels. Accuracy of 14 cases were able to achieve 100% and 13 cases were from 33.3% to 66.6%, but there were still 11 cases that the accuracy is 0%, results are shown below.

Table 1: Specific accuracy of each case in 500 times trained CNN

case	accuracy	case	accuracy	case	accuracy	case	accuracy
002L	0	144L	0	216L	0	333L	0
002R	100	144R	0	216R	0	333R	100
005L	50	160L	0	239L	0	339L	0
005R	33.33	160R	66.67	239R	0	339R	0
012L	33.33	183L	100	287L	0	381L	40
012R	50	183R	100	287R	0	381R	50
018L	0	192L	0	294L	0	406L	0
018R	0	192R	50	294R	0	406R	0
119L	X	206L	0	310L	X	407L	0
119R	100	206R	0	310R	0	407R	0

Further, the number of training times was increased to 1000 times. The total accuracy rate has shapely increased to 97%, in this case, a majority of tested data can match the actual case numbers, as shown in Figure 4.3.

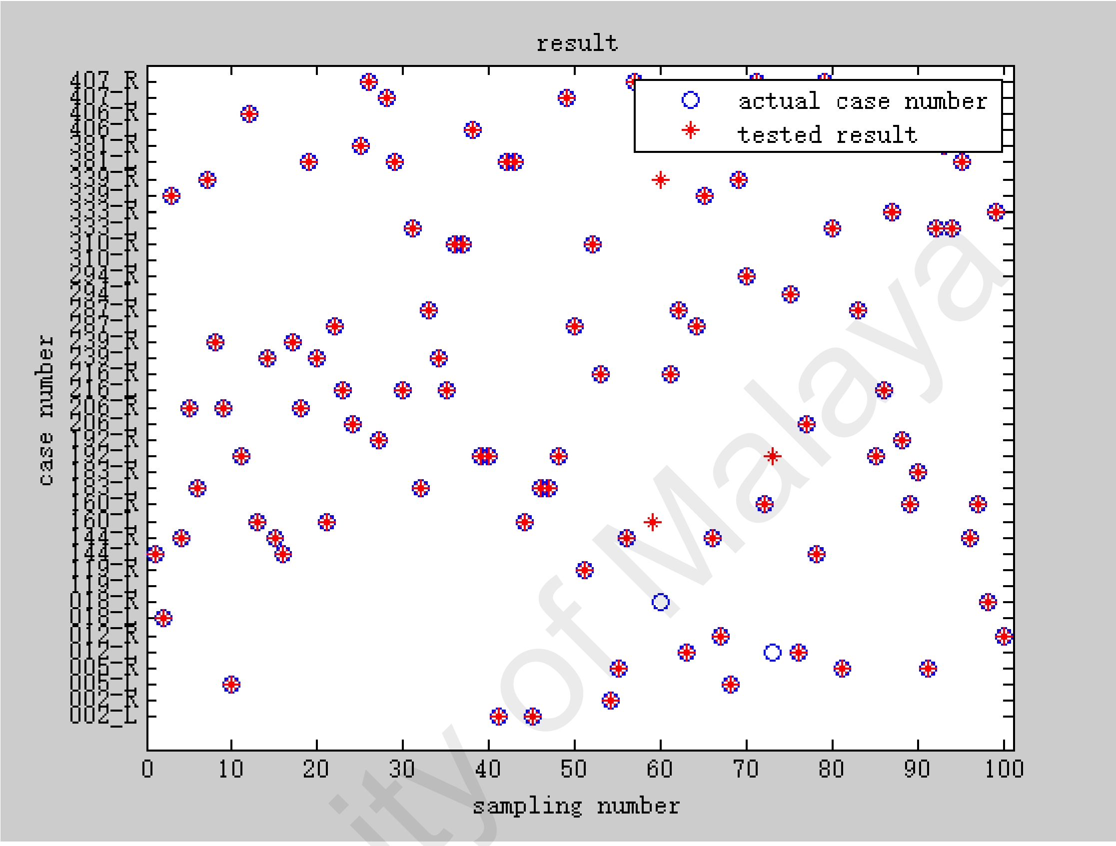

Figure 4.3: Tested result of 1000 times trained CNN

Meanwhile, accuracy rate of most of cases were 100%, there were only three cases whose accuracy rate is less than 100%. They are 012L with 66.67%accuracy rate, 018R with 50% accuracy rate and 406L with 75% accuracy rate.

Table 2: Specific accuracy of each case in 1000 times trained CNN

case	accuracy	case	accuracy	case	accuracy	case	accuracy
002L	100	144L	100	216L	100	333L	100
002R	100	144R	100	216R	100	333R	100
005L	100	160L	100	239L	100	339L	100
005R	100	160R	100	239R	100	339R	100
012L	66.67	183L	100	287L	100	381L	100
012R	100	183R	100	287R	100	381R	100
018L	100	192L	100	294L	100	406L	75
018R	50	192R	100	294R	100	406R	100
119L	X	206L	100	310L	X	407L	100
119R	100	206R	100	310R	100	407R	100

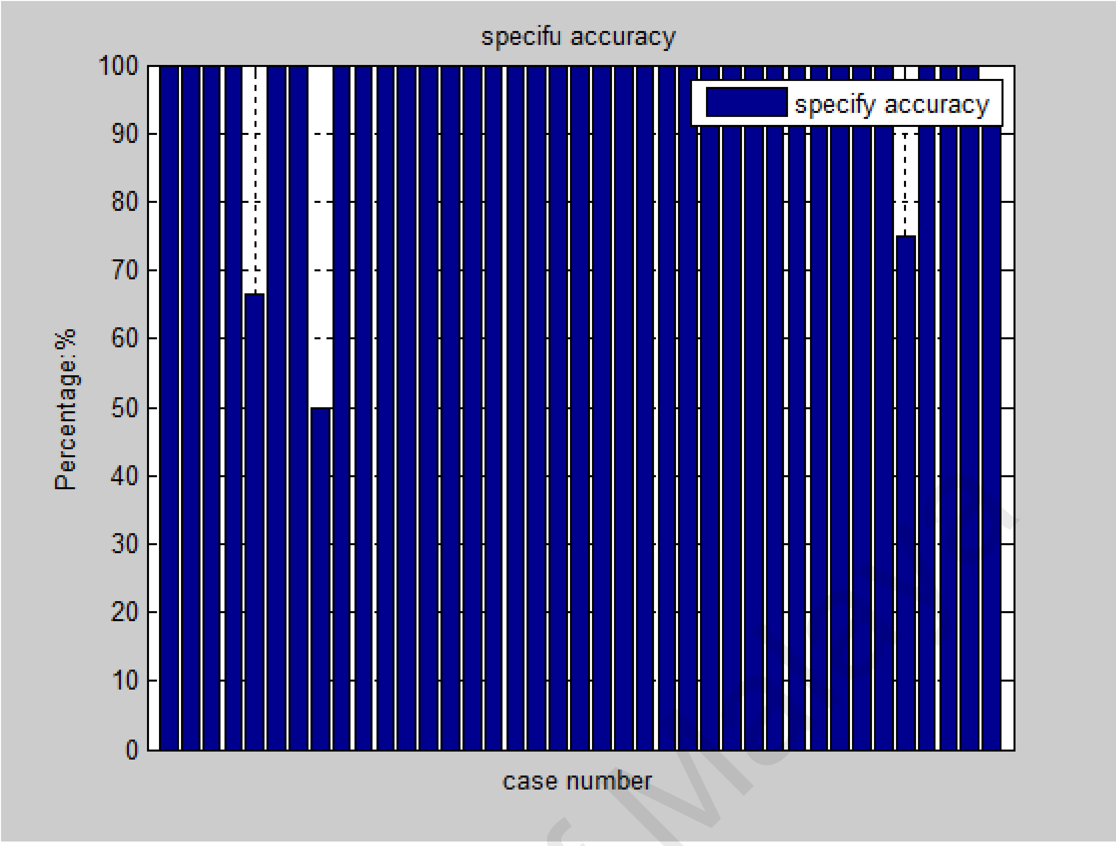

Figure 4.4: Specific accuracy rate of each case

Finally, in the condition when training times was 2000 times, the total accuracy increased to 99%, almost all tested result can match the actual case number, except one of case 406L.

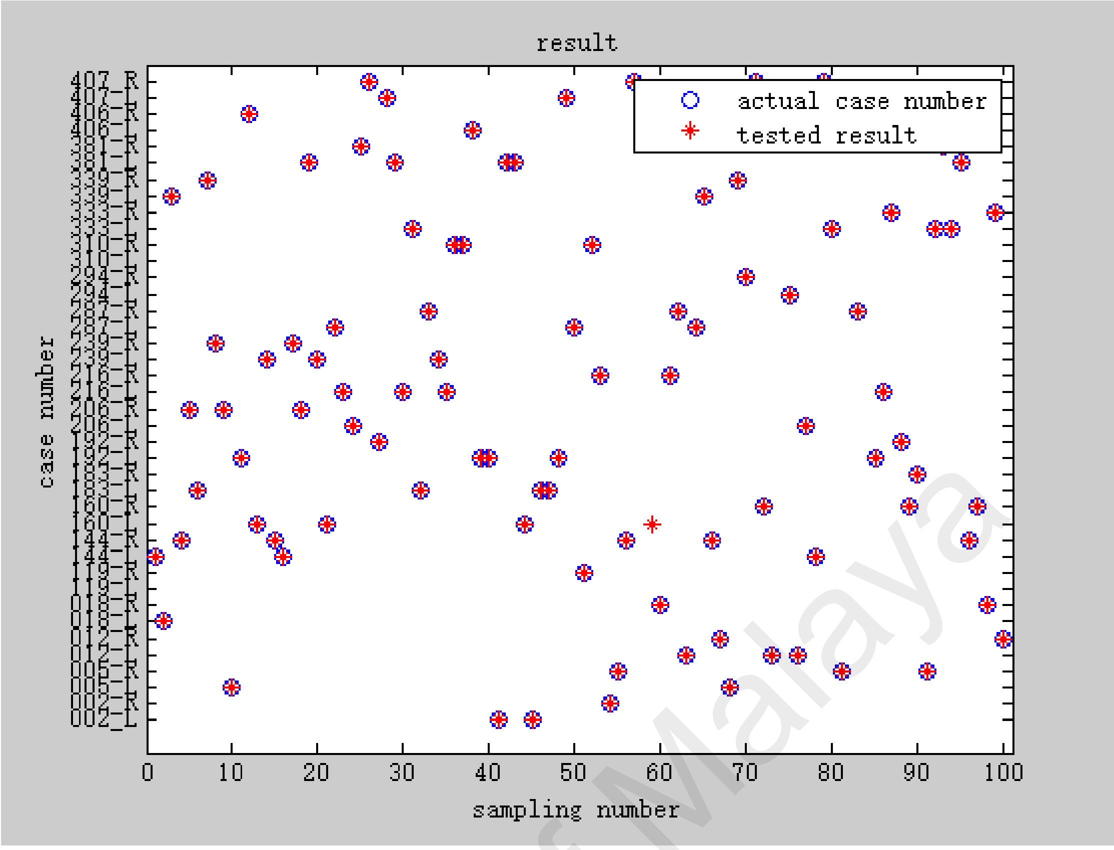

Figure 4.5: Tested result of 2000 times trained CNN

From the table below, it is clear that there was only one image that was not able to be recognized, i.e. 406L with 75% accuracy rate.

Table 3: Specific accuracy of each case in 2000 times trained CNN

case	accuracy	case	accuracy	case	accuracy	case	accuracy
002L	100	144L	100	216L	100	333L	100
002R	100	144R	100	216R	100	333R	100
005L	100	160L	100	239L	100	339L	100
005R	100	160R	100	239R	100	339R	100
012L	100	183L	100	287L	100	381L	100
012R	100	183R	100	287R	100	381R	100
018L	100	192L	100	294L	100	406L	75
018R	100	192R	100	294R	100	406R	100
119L	X	206L	100	310L	X	407L	100
119R	100	206R	100	310R	100	407R	100

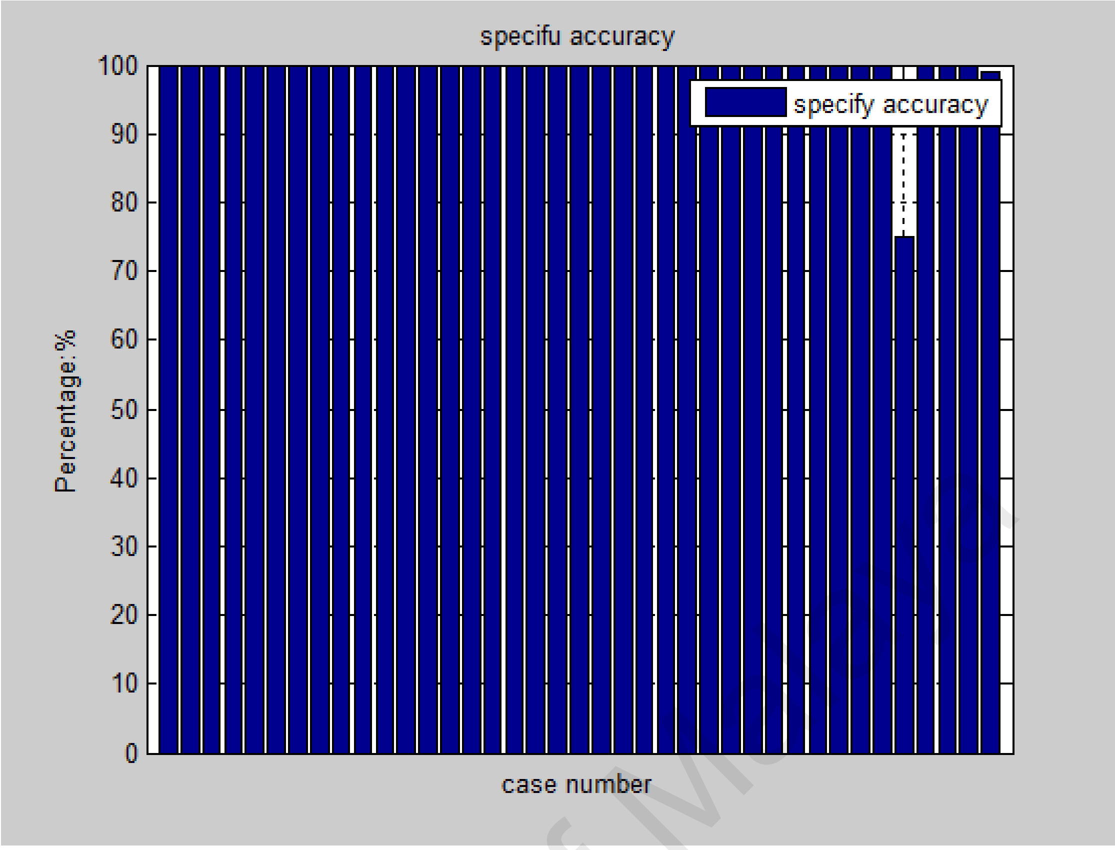

Figure 4.6: Specific accuracy rate of each case

In order to clearly understand the relationship between train times (epoch) and accuracy rate, convolutional neural network from 10 training times to 2000 training times had been tested, the results of different training times are summarized in Table 4. It is proven that the accuracy rate improves closely with increasing training times of convolutional neural network. (Table of specific accuracy of each case of different training times is presented in Appendix C).

Table 4: Accuracy rate of different numbers training

Epoch	10times	50times	100times	200times	300times	400times
Accuracy	0%%	0%	0%	0%	2%	8%
Epoch	500times	600times	700times	800times	1000times	2000times
Accuracy	48%	83%	92%	94%	97%	99%

Because of random selection, two cases were not found in the test list (case number 119L, 310L), for testing whether they can be recognized or not and further for improving the capability of real-time recognition of this system, a real-time recognition program based on this system as design to test (after 2000 times training).

At first, two images from both cases 119L and 310L were chosen, at the same time for further testing the real-time recognition, four images from others were randomly chosen. Marking them with the case number (cases selected are 002L, 002R, 005R, 018R, 119L, 144R, 160L, 183L, 192L,239L,239R, 287R, 294L, 310L, 339L, 381R, 406R, 407R), they were all put to TEST folder and be ready for testing, shown below as Figure 4.7.

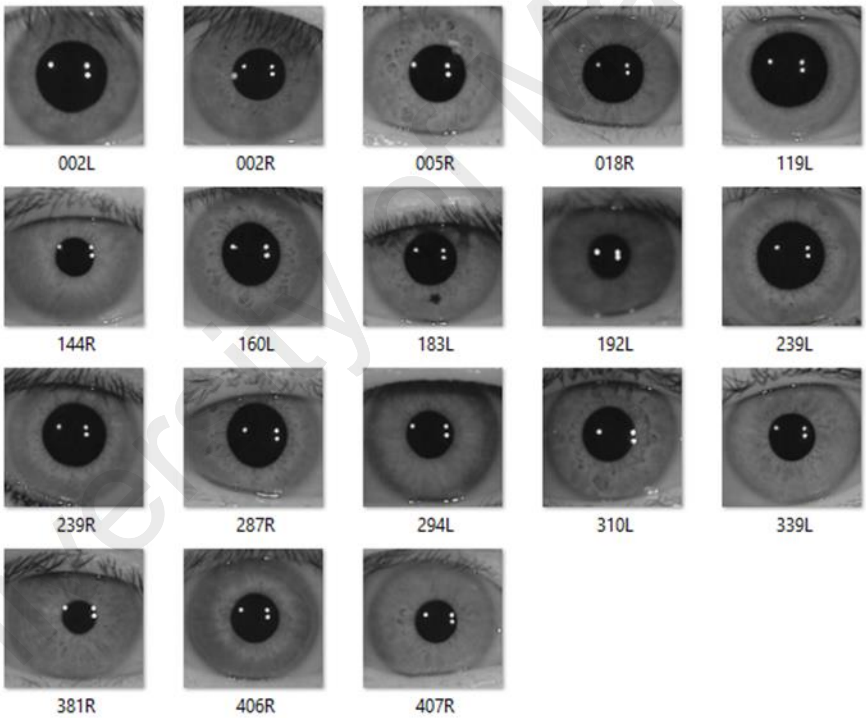

Figure 4.7: Randomly selected cases for the real-time recognition test

The system has a high recognition accuracy rate due to 2000 times training previously, so that these six images were able to be recognized correctly, and the result shown below in Figure 4.8.

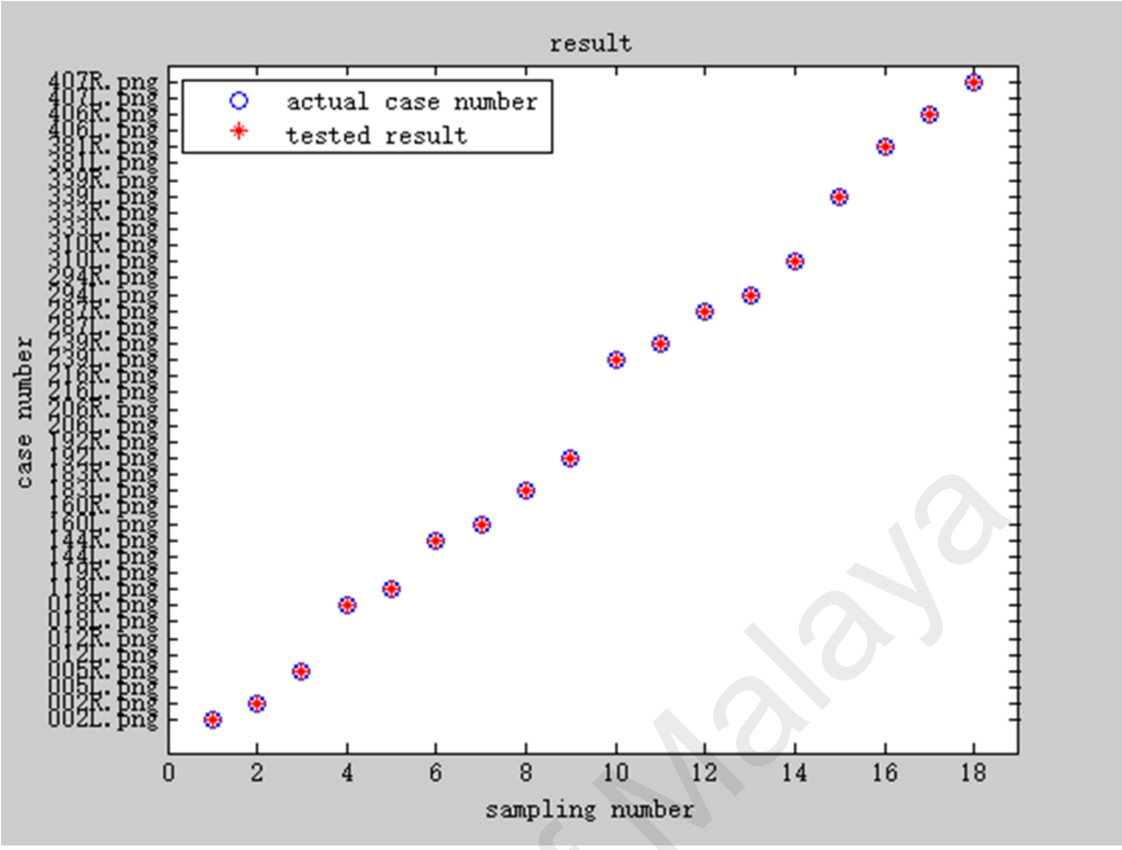

Figure 4.8: Random test of real-time recognition

4.3 Discussion

Results displayed at this chapter show that the system was operating as expected, both recognition tests were completed without mistakes. The high recognizing accuracy rate after 2000 times training is an advantage of this system. However, the time spent by each training is average at 1.05 second, thus training 2000 times takes around 35 minutes. In addition, this system needs a huge numbers of training, from results displayed in Figures, the accuracy of the first 400 times training were nearly 0%. After more than 700 times training, the recognition accuracy of the system barely surpassed 90%. These two points are closely related with the structure of convolutional neural network of this system. Therefore, this study will further focus on improving the structure for raising the efficiency of this system.

CHAPTER 5: CONCLUSION

5.1 Conclusion

This study aims at designing an iris recognition system using convolutional neural network, bringing together advantages from both iris recognition and artificial intelligence to create a practical system. As presented in the results, at this stage the combination of these both technologies have proven can be successfully employed and can be further applied to other fields in near future.

In this study, convolutional neural network contributes a number of benefits such as less artificial images process, efficiency features extraction and a high recognizing accuracy rate after huge numbers of training. Convolutional neural network is also considered to be robust for variations of automatically learned application by ensuring a large number of datasets are given for the training, so the quality of input images will not affect a lot for the recognition [31]. However, the disadvantages of convolutional neural network should be taken care of, i.e. its highly complex structure and long training time requirement make some negative effects on processing speed of this system.

5.2 Recommendation for future study

Suggestion and recommendation for future study is mostly focusing on improving the structure of convolutional neural network used in system, e.g. to develop a more powerful system with less training requirement and faster processing speed. Lastly, designing an input device for this system is also necessary so that this system can be completed to self-recording and self-recognition without manual operation or intervention.

REFERENCES

[1]Dong, X. and L. Chen (2018). "Ultrabroadband Plasmonic Absorber Based on Biomimetic Compound Eye Structures." IEEE Photonics Journal **10**(1): 1-7.

[2]E. Jung and K. Hong, "Automatic Retinal Vasculature Structure Tracing and Vascular Landmark Extraction from Human Eye Image," 2006 International Conference on Hybrid Information Technology, Cheju Island, 2006, pp. 161-167.

[3]Nguyen, K., et al. (2017). "Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective." IEEE Access **PP**(99): 1-1.

[4]Thompson, S. and G. Brat (2008). Verification of C++ Flight Software with the MCP Model Checker. 2008 IEEE Aerospace Conference.

[5]Soniya, et al. (2015). A review on advances in deep learning. 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI).

[6]Qiu, Y., et al. (2017). Pressure control of fuel pressure regulator based on BP neural network PID. 2017 International Conference on Advanced Mechatronic Systems (ICAMechS).

[7]D. P, U., et al. (2017). Artificial Intelligence Techniques Used to Detect Object and Face in an Image: A Review. 2017 3rd International Conference on Computational Intelligence and Networks (CINE).

[8]Aloysius, N. and M. Geetha (2017). A review on deep convolutional neural networks. 2017 International Conference on Communication and Signal Processing (ICCSP).

[9]Lin, S. Y., et al. (2017). A Dynamic Data-Driven Fine-Tuning Approach for Stacked Auto-Encoder Neural Network. 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE).

[10]Khan, M. F. F., et al. (2017). Iris recognition using machine learning from smartphone captured images in visible light. 2017 IEEE International Conference on Telecommunications and Photonics (ICTP).

[11] F. Alonso-Fernandez, R. A. Farrugia and J. Bigun, "Learning-based local-patch resolution reconstruction of iris smart-phone images," 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, 2017, pp. 787-793.

[12]Fitri, A., et al. (2012). Iris recognition method based on ordinal measure of discrete cosine transform coefficients. 2012 IEEE Symposium on Computers & Informatics (ISCI).

[13]Reyes-López, J., et al. (2011). Zernike's Feature Descriptors for Iris Recognition with SVM. 2011 30th International Conference of the Chilean Computer Science Society.

[14]Sun, Z., et al. (2014). "Iris Image Classification Based on Hierarchical Visual Codebook." IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(6): 1120-1133.

[15]Latha, L. and S. Thangasamy (2010). "A robust person authentication system based on score level fusion of left and right irises and retinal features." Procedia Computer Science **2**: 111-120.

[16]Daming Li, Lianbing Deng, Brij Bhooshan Gupta, Haoxiang Wang and Chang Choi*, "A Novel CNN based Security Guaranteed Image Watermarking Generation Scenario for Smart City Applications", Information Sciences, Online Published (2016 JCR IF :4.832)

[17] http://cs231n.github.io/convolutional-networks/

[18]Buccoli, M., et al. (2014). Unsupervised feature learning for bootleg detection using deep learning architectures. 2014 IEEE International Workshop on Information Forensics and Security (WIFS).

[19]Chen, W., et al. (2014). Door recognition and deep learning algorithm for visual based robot navigation. 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014).

[20]Kuang, P., et al. (2014). Preview on structures and algorithms of deep learning. 2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing(ICCWAMTIP).

[21]Zhang, S., et al. (2014). Learning high-level features by deep Boltzmann machines for handwriting digits recogintion. Proceedings of 2nd International Conference on Information Technology and Electronic Commerce.

[22]Czajka, A., et al. (2007). Iris recognition with match-on-card. 2007 15th European Signal Processing Conference.

[23]Isnanto, R. R. (2014). Iris recognition analysis using biorthogonal wavelets tranform for feature extraction. 2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering.

[24]Proença, H. (2011). Non-cooperative iris recognition: Issues and trends. 2011 19th European Signal Processing Conference.

[25]Yang, T., et al. (2014). Subregion mosaicking applied to nonideal iris recognition. 2014 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM).

[26]Cai, X., et al. (2016). Detecting Abnormal Behavior in Examination Surveillance Video with 3D Convolutional Neural Networks. 2016 6th International Conference on Digital Home (ICDH)

[27]How, D. N. T. and K. S. M. Sahari (2016). Character recognition of Malaysian vehicle license plate with deep convolutional neural networks. 2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS).

[28]Morchhale, S., et al. (2016). Classification of pixel-level fused hyperspectral and lidar data using deep convolutional neural networks. 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS).

[29]Turner, J. T., et al. (2016). SPARCNN: Spatially related convolutional neural networks. 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR).

[30]Jianqiang, X. and G. Capi (2017). Robot painting recognition based on deep belief learning. 2017 8th International Conference on Information, Intelligence, Systems &

Applications (IISA).

[31]Prasetio, M. D., et al. (2017). Deep belief network optimization in speech recognition. 2017 International Conference on Sustainable Information Engineering and Technology (SIET).