IRIS RECOGNITION BY USING DEEP LEARNING CLASSIFIER AND FRACTAL FEATURE EXTRACTOR

KUA CHOON WEN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

IRIS RECOGNITION BY USING DEEP LEARNING CLASSIFIER AND FRACTAL FEATURE EXTRACTOR

KUA CHOON WEN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2021

DECLARATION

I declare that this report entitled “Iris Recognition by using Deep Learning Classifier and Fractal Feature Extractor” is the result of my own work except for quotes as cited in the references.

Signature :

Author : Kua Choon Wen……………………………

Date : 25 June 2021……………………………

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : ……………………………………

Supervisor Name : Dr. Norazlina Binti Adb. Razak

Date : 25 June 2021

DEDICATION

I would like to dedicate this project to my beloved family.

ABSTRACT

Iris recognition system is a reliable biometric technology due to its high uniqueness and performance. This system provides high security service since the iris is unique and hard to be fake. This project is aimed to build an iris recognition system by using MATLAB and analyse the performance in terms of accuracy and recognition speed. The flow of iris recognition system starts with image acquisition and followed by edge detection. In edge detection, the centre of iris, radius of inner and outer iris boundary is extracted. Then, the information is used to segment the iris from iris image. The normalization is then used to normalize the segmented iris to rectangular block. The fractal feature which is lacunarity is extracted and inputs to the convolutional neural network. In the proposed iris recognition system, 50 iris images are trained, and another 50 iris images are tested. The proposed iris recognition system recognizes 49 iris images which is 98% of accuracy with an average recognition time of 0.2s approximately. Therefore, this proves that iris recognition system is reliable due to its high accuracy and fast recognition time.

ABSTRAK

Sistem pengiktirafan iris adalah teknologi biometrik yang boleh dipercayai disebabkan oleh keunikan dan prestasi yang tinggi. Sistem ini menyediakan perkhidmatan securiti yang tinggi kerana iris unik dan sukar dipalsukan. Projek ini bertujuan untuk membina sistem pengiktirafan iris dengan menggunakan MATLAB dan menganalisis prestasi dari segi ketepatan dan kepantasan pengiktirafan. Aliran sistem pengiktirafan bermula dengan pemerolehan imej diikuti dengan pengesanan tepi. Dalam pengesanan tepi, pusat iris, jejari sempadan iris dalaman dan luar diekstrakan. Lepas itu, maklumat tersebut digunakan untuk menyegmentasikan iris dari imej iris. Normalisasi kemudian digunakan untuk menormalkan iris yang tersegmentasi menjadi blok segi empat. Lacunariti yang mewakili ciri fraktal telah diekstrakan dan dimasukkan ke rangkaian saraf konvolusional. Dalam sistem pengiktirafan iris, 50 imej iris telah dilatih dan 50 imej iris telah diuji. Sistem pengiktirafan iris telah berjaya mengenali 49 imej iris iaitu ketepatan 98% dengan menggunakan masa pengiktirafan purata lebih kurang 0.2s. Oleh itu, ini membuktikan bahawa sistem pengiktirafan iris boleh dipercayai kerana ketepatannya tinggi dan pengiktirafan yang laju.

ACKNOWLEDGEMENTS

First of all, I want to express my deep and sincere gratitude to my project supervisor, Dr. Norazlina Binti Abd Razak, who gives me the opportunity to do this project. She also gives me a lot of guidance, knowledge, information, caring and tolerance which gives me motivation and effort in this project. Without Dr. Norazlina, I would not have chance to have the knowledge of this field and even interest in this field.

Moreover, I want to thank to my family especially my parents. They always give their unlimited encouragement and motivation to me to accomplish this project. Also, I would like to thank to my friends who give me the helping hand and helping ear.

Furthermore, I would like to express my gratitude to my project panels, PM Dr. Wong Yan Chiew and Pn. Siti Aisah Binti Mat Junos @ Yunos, because they give me valuable comments and suggestions to improve my project and thesis.

Last but not least, I would also like to thank to the lecturers of Faulty of Electronic Engineering and Computer Engineering who gives me knowledge since I am year one student. And thanks to the one and all who directly or indirectly lend their hand for supporting and assisting me to accomplish this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	:	Convolutional Neural Network
CRR	:	Correct Recognition Rate
DBC	:	Differential Box Counting
FD	:	Fractal Dimension
GUI	:	Graphical User Interface
ROI	:	Region of Interest
(x_c, y_c)	:	Coordinates of centre of pupil
r_i	:	Radius of inner iris boundary
r_o	:	Radius of outer iris boundary

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Biometric identification technology is the technology that identify a person based on his or her physical and behavioural characteristics that can distinguish him or her from others. For example, fingerprint, DNA, eyes, gesture, sound and many more. These characteristics are called biometric identifiers.

Iris recognition is one of the reliable biometric technologies that can identify or authenticate a person through his or her iris. Iris recognition is said to be the biometric technology with high universality, uniqueness, permanence, performance and circumvention [1]. Although the development of iris recognition system is still immature and unpopular in public but can believe that the efforts in invention of a user-friendly and mature iris recognition system is underway.

1.2 Project Background

Nowadays, biometric identification is widely used as identity recognition since its reliability is high and it is unique between individuals. This identification method brings convenience to the community and in various fields. For example, DNA and fingerprint identification are widely used in crime scene investigation, and fingerprint or face recognition are implemented as a password to unlock smart phone. This type of identification is really bringing the world into a new era.

In this new era, fingerprint and face recognition are widely used in the activity of public. For example, both of the identification can be used as the authentication of credit cards and password of phone. However, iris recognition is still not widely implemented and used in technology. Iris recognition is one of the reliable biometric technology because the human iris is also externally visible and yet is protected by the cornea. Hence, the pattern of iris is unique and remains unchanged throughout the life. This system can replace the password or PIN number that may be easily stolen and forgotten. If we compared to other biometric identification, iris recognition system can give higher performance and lesser rejection due to its uniqueness, permanence, and stability.

Hence, this project is planned to design a program of iris recognition system which extract the fractal feature from one's iris and apply the deep learning neural network to identify the person. This project also analyses the iris recognition performance and time taken to identify the person.

1.3 Problem Statements

Password and PIN numbers are normally used as identity identification to access to personal account and devices. But it is insecure because it required the individual to remember which can be forgotten and even be stolen by hacker easily. In face recognition's problems, there are some difficulties to recognize the person if his/her display various facial expression, angle, pose, make up and even face changed due to aging and plastic surgery. So, as alternative, iris recognition system with high accuracy and fast recognition time is proposed to identify an individual.

1.4 Objectives

- To implement an iris recognition system that extract the fractal feature of iris and apply deep learning neural network to identify the person.
- To analyse the performance of iris recognition system and time taken to identify the person.

1.5 Research Outline

Figure 1.1 shows the flow chart of iris recognition system. First the iris image is acquired by the iris recognition system. Next, in pre-processing stage, the important information which is centre, radius of inner and outer iris boundary is extracted for segmentation. In segmentation, the iris part is extracted from the iris image. Then, the segmented iris image is normalized from annular shape into rectangular block in the normalization step. The fractal feature is extracted in feature extraction step and it is used to train or test the deep learning neural network for classification.

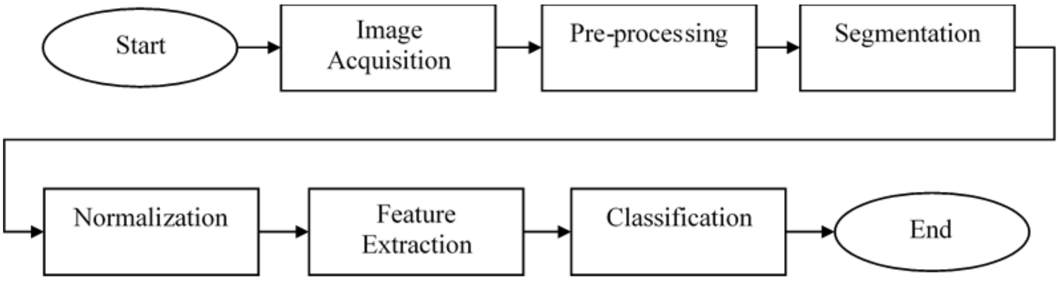


Figure 1.1: Flow Chart of Iris Recognition System

1.6 Scope of Project

Since this project is mainly implementation the program of iris recognition system, the MATLAB is used to make simulation throughout the project. In MATLAB, Image Processing Toolbox will be installed to process the image of iris like iris acquisition, edge detection, fractal feature extraction, and others. After the process of fractal feature extraction, a neural network is set up which is used for classification of iris. Hence, Deep Learning Toolbox will be installed in MATLAB. A user interface (GUI) will be built to monitor the process and performance of iris recognition system. The GUI will be built by using App Designer provides by MATLAB. The performance of the iris recognition system which is accuracy and recognition time will be analysed.

For the iris images used in implementation and simulation of iris recognition system, the CASIA Iris Image Database V4.0 (CASIA-IrisV4) is downloaded and the suitable subset which is CASIA-Iris-Interval is chosen. From the subset, 10 persons are randomly chosen and each person contains 10 iris images. Thus, a total of 100 iris images are used in the iris recognition system.

1.7 Structure of Thesis

This thesis consists of five chapters. Chapter 1 describes the project background, problem statements, objectives, scope of work and expected outcome of the project. Chapter 2 mainly discusses the background study on the biometric identification technology, human iris, iris recognition system, and the existing approaches of iris recognition system as well as the performances. In the background study of iris recognition system, the typical stage of iris recognition system, strengths and weakness are studied and reviewed. In Chapter 3, the resources of iris images and implementation flow of iris recognition system are briefly explained. The results are presented and discussed in Chapter 4. In Chapter 5, the project is concluded and the recommendations of the project for future works are proposed.

CHAPTER 2

BACKGROUND STUDY

2.1 Biometric Identification Technology

Based on Oxford English Dictionary, identity is defined as the characteristics, feelings, or belief that make people different from others. Identity is a term that have various meaning from different field. In psychology, identity describes as the character of a person that can distinguish from different person [2]. Also, identity can be just simply represented by one's personal information like name, age, gender, birthday date, and so on. In biometric, the identity of human is represented by the physical and behavioural characteristics that are different and never repeated from others. For example, DNA, fingerprint, palm print, face, iris pattern, voice and many more. These characteristics are known as biometric identifiers.

Biometric is defined as the measurement and calculation of the characteristics of human body part or behaviour. Biometric identification is used to identify individual based on his/her distinguishable physical and behavioural characteristics [3]. Since the biometric identity is distinctive from others and unchanged throughout the life, so it is more reliable and stable.

Since past days, people are using two kinds of traditional identity identifications which are token-based and knowledge-based identification [4]. Token-based identification is the objects that can identify the person such as identity card, passport, driving license, credit cards and so on. These objects are important because they act as the proof to one's identity. On the other hand, knowledge-based identification is using a bunch of alphabet, number, or sentence as a personal identification like password, PIN numbers or answers to security questions. This kind of identification like password requires the individual to remember since it is not an object that can be carried by individual unless the password is recorded.

However, these two kinds of identity identification cannot truly represent the identity of the person because they are not a part of human that distinctive to others. There are a lot of disadvantages by using them. The token-type identity may be missing, drop, forgotten and stolen by others while the PIN number can be forgotten easily and even hack by fraud. Both token-based and knowledge-based identification cannot be differentiated whether he or she is the authorized person or a fraud who use the token or knowledge of the authorized person [4].

Therefore, in these few years, the technology of biometric identification is developed rapidly to replace the traditional identity identifications. This is because the biometric identifiers have higher reliability and stability. The biometric identifiers are the things that cannot be stolen like credit cards or password. There are a lot of biometric identification technology that already widely use in the world. One of the technologies is fingerprint recognition. The fingerprint recognition is used as personal identification for centuries. This technology is mainly used in the forensic science associated with the DNA recognition since the fingerprints and DNA are unique to each person. The fingerprints of twins and even each finger of the same person is different. In these years, the fingerprint recognition is implemented into mobile devices which we can just unlock the phone by our own finger.

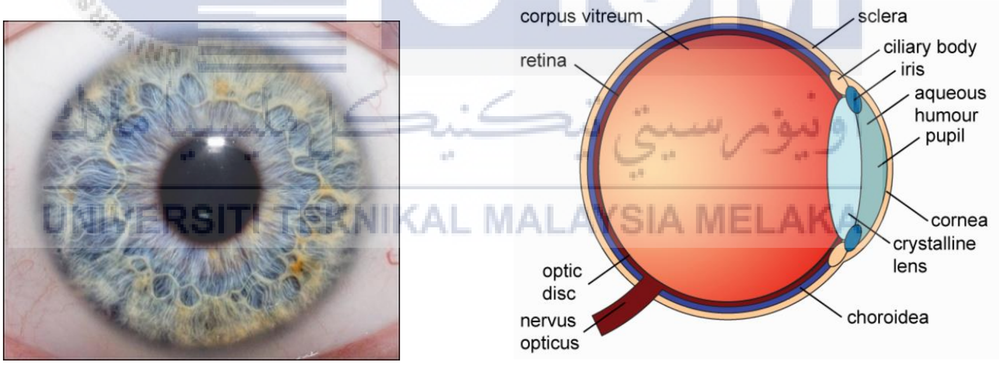
Moreover, the face recognition is one of the common methods of identification since the acquiring of face image is easy. The approaches to face recognition are basically based on the shape and location of face attributes like eyes, nose, lips, eyebrows, chin, face shape and their spatial relationships [1]. But face is changeable during lifetime. There are difficulties for current face recognition system to recognize the person if his/her display various facial expression, angle, pose, make up and even face changed due to aging and plastic surgery.

Last but not least, the iris recognition is a biometric identification technology that still have not widely use in the world. But iris recognition gives high accuracy and speed in personal identification. This is because the uniqueness and permanence of the iris is very high. The fractal of iris is very distinctive from every individual and even the iris of twins is unidentical. Although the early iris recognition system was immature and expensive, the hard work in development on more user-friendly and cost-effective iris recognition system are ongoing.

In conclusion, as the technology develops, the traditional identity identification is taking over by biometric identification gradually. This is because the universality, uniqueness, permanence, and reliability of the biometric identification technology are on another level. It brings much more convenience to the community and greatly decrease the cases of fraud since biometric identifier cannot be stolen or hack by others.

2.2 Human Iris

Iris is the part of human eye that responsible to adjust the size of pupil to control the amount of light reaching the retina. As the eye image in Figure 2.1 (a), the annular part that surrounded the black pupil is the iris. The iris will not damage or deform because of the cornea protect the iris from outside and the aqueous humour hold the pattern of iris as shown in Figure 2.1 (b). Eye colour that we can view from external is basically the pigmentation of iris.



(a) Front View [5] (b) Anatomy of the Eye [6]

Figure 2.1: Human Eye

As shown in Figure 2.2, the anterior layer of the iris consists of two main parts: pupillary zone and ciliary zone, which is separated by collarette [7]. Pupillary zone is as flat region extends from the border of pupil to the collarette while ciliary zone extends from collarette to the rest of iris.

The ciliary zone is the part that creates the fractal feature of the iris because of the existing of crypts and/or pigment spot. The crypts are the pits that appear near to the collarette and the periphery of iris, while the pigment spots are the random black spots appear in the ciliary zone.

The black lines in the ciliary zone are the contraction furrows which control the size of the pupil. When the person is in the bright place, the contraction furrows of the iris is relaxed. So, the radius of pupil become smaller and decreased the amount of light reached to the retina. Same to the condition when the person in the dark place, the contraction furrows of the iris is contracted to expand the size of iris.

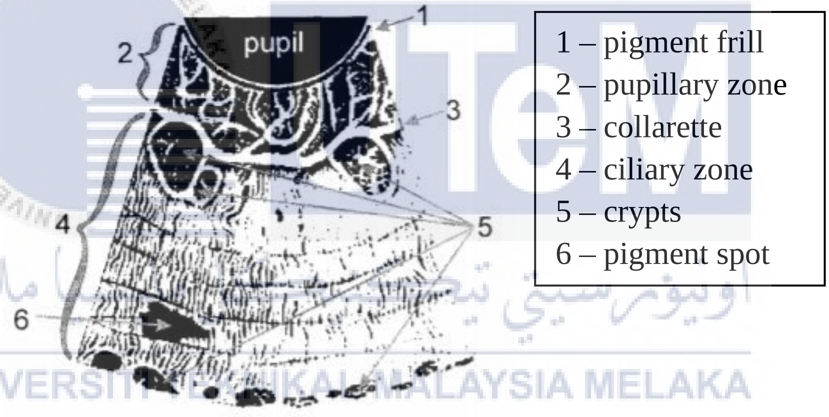


Figure 2.2: Anterior Layer of Human Iris [7]

From the anatomy of iris, iris is one of the most protected part in human body. Since the iris is protected by the eyelid, cornea and aqueous humour, the pattern of iris is protected and preserved. Additionally, the iris is not easy to be contaminated by liquid or material, and the features will not change due to aging or vision problems. This means the permanence and stability of iris is high.

Furthermore, the iris minutiae are chaotic morphogenesis because these random patterns of iris depend on initial conditions in embryonic genetic expression. The limitation of partial genetic penetrance assures that the iris of identical twins and both eyes of the person have different features [8]. Hence, the uniqueness of the iris is extremely high. Based on Williams G.O., the probability of finding two identical irises is estimated at 1 in 10^{72}, while the population of the earth is 7.594 billion (7.594×10^{9}) [8].

2.3 Strength and Limitation of Iris Recognition System

Iris recognition has numerous advantages compared to other biometric identification technology. The most significant strength of iris recognition system is the uniqueness. The feature of iris is developed during the embryonic gestation and there are a huge number of degrees of freedom that makes the feature of iris. It is extremely impossible to find the same iris from two different persons. Even both identical twins and both eyes of the same person have different iris pattern.

Next, since the uniqueness of iris is very high, this creates high security to iris recognition system. This is because the iris is extremely hard to forge unlike fingerprint. Fingerprint can be forged easily because it is exposed externally and can be collected from the things he or she touched, which is opposite from the iris. The features of irises are hard to be collected because they are the organ than is protected by cornea and aqueous humour.

Moreover, the iris recognition system has faster performance. User can finish the whole iris identification process for the first time by just a few seconds and even faster for next identification [9]. The iris recognition system is also user friendly since the user do not require any other skills to operate it. The user just needs to stay still to be scanned by the camera. This strength also led to another strength which is hygiene. The user does not require any physical contact on the device but just scan the iris by the camera with a distance. Unlike the fingerprint and palmprint recognition that require user to press the finger or palm on the scanning device that may not be sanitized after used.

Apart from the strength, the iris recognition system has the limitations too. Firstly, the iris scanner is expensive compared to other biometric identification system. This is because the invent and manufacture of iris scanner is still unmatured. The iris recognition system not yet to be widely use by public is also one of the factors that cause the equipment of iris recognition system expensive. Besides, this system is not suitable to be use in the forensic because the iris biometric will not left in the crime scene as evidence [10].

Furthermore, the iris recognition system may encounter the difficulties in iris detection. For instance, the reflection of light on the cornea or spectacles. Although the iris can be scan for the person who wears the spectacles or clear contact lenses, but the results may affect by the reflection of light. The reflection of light will cover some area of iris feature and affect the accuracy of the system. In addition, all fractal of iris is covered up if the user is wearing coloured contact lenses. The iris may also partially block by the eyelids which cause the difficulties in detection of whole iris.

In summary, there are a lot of advantages of iris recognition system and sure has its disadvantages too. However, these disadvantages can be solved future when the iris recognition system is released and used widely. This is how a system developed from the immature to a mature system.

2.4 Typical Flow of Iris Recognition System Implementation

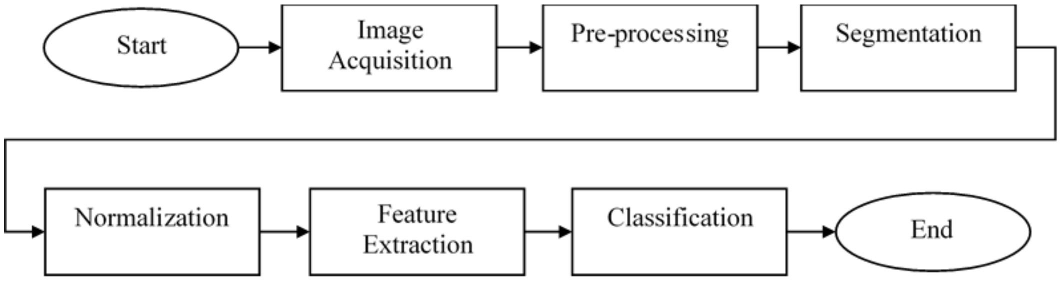


Figure 2.3: Typical Flow Chart of Iris Recognition System [11]

Figure 2.3 shows the typical flow of iris recognition system. First is image acquisition. The iris images need to be acquired from the capture of camera or databases. In iris acquisition, the iris images must be clear and sharp especially the pattern and boundaries of iris. A set of iris images is collected from each person for the purpose of training and testing through the neural network later.

Next stage is the pre-processing process. This process is aimed to improve or enhance the raw iris images, then extract the information of iris. First, the input iris images are standardized into a standard size, but this can be skipped if the iris images are taken from the same camera with the same resolution. The colour iris images must convert into grayscale so that the edge detection is more accurate. This is because the RGB colours contain three luminance values while grayscale has only one luminance. Next, the information such as centre of iris, inner radius and outer radius of iris are extracted.

In the segmentation process, the non-iris area is removed and the iris part is segmented from the iris image. Each annular shape iris image has different thickness due to the deformation of iris in various condition. To transform all segmented iris image in standard form, the normalization is performed. All segmented annular iris images are transformed into rectangular shape with standard size.

After normalization, the feature of iris has to be extracted prior to training process or as the input of the classification algorithm. So, the normalized iris images are enhanced to make a strong contrast from other images. The extracted feature must be unique because it represents the identity of every person.

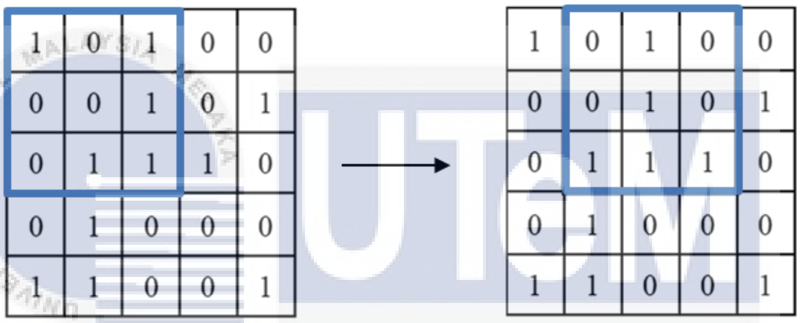
Last stage is classification. There are two steps in classification which are evaluation of similarity and recognition decision [12]. In first step, the similarities of information in test image are evaluated with stored information and resulting the similarity value. Then, in recognition decision step, the acceptance or rejection decision is performed based on a threshold. For the failed recognition, the classification algorithm is modified and trained again.

2.5 Fractal Dimension and Lacunarity

Hausdorff-Besicovitch (HB) dimension is used to estimate geometrical complexity of an object or an image [13]. The set with HB dimension is greater than topological dimension is called fractal set. The fractal dimension (FD) is first calculated by Mandelbrot to estimate the coastline length [14]. FD can also use in texture estimation, fractal feature extraction, and so on. Based on Mandelbrot, the FD is estimated by least square linear regression. The slope of the linear fitted line is then the FD.

In fact, the iris images from different persons will have similar FD. In order to differentiate the iris images based on FD, the lacunarity of the FD is computed. According to Mandelbrot [14], the origin word of lacunarity, *lacuna*, means gap in Latin. A fractal is called *lacunar* if the gaps are wide.

Lacunarity is a measurement of inhomogeneity of fractal in the image [15]. The more inhomogeneous the fractal of the image, the higher the lacunarity. Gliding box lacunarity [16, 17] is one of the methods used to obtain the lacunarity of the image or FD.



1	0	1	0	0
0	0	1	0	1
0	1	1	1	0
0	1	0	0	0
1	1	0	0	1

1	0	1	0	0
0	0	1	0	1
0	1	1	1	0
0	1	0	0	0
1	1	0	0	1

Figure 2.4: Algorithm of Gliding Box Lacunarity

Figure 2.4 shows the how gliding box lacunarity works. First, the box with size $r \times r$ is specified and place in the upper right corner of the image or matrix with size of $M \times M$. Then the total number of boxes $N(r)$ and probability distribution $Q(S,r)$ is expressed as:

$$N(r) = (M - r + 1)^2 \tag{2.1}$$

$$Q(S,r) = \frac{n(S,r)}{N(r)} \tag{2.2}$$

where $n(S,r)$ is the number of boxes of size r containing S occupied sites. Next, the first and second moments of the distribution is determined as equation (2.3) and (2.4). Finally, the lacunarity is obtained by equation (2.5).

$$Z^{(1)} = \sum S \; Q(S, r) \qquad (2.3)$$

$$Z^{(2)} = \sum S^2 \; Q(S, r) \qquad (2.4)$$

$$\Lambda(r) = \frac{Z^{(2)}}{(Z^{(1)})^2} \qquad (2.5)$$

If express in statistics way, the $Z^{(1)}$, $Z^{(2)}$ and $\Lambda(r)$ are then expressed as:

$$Z^{(1)} = \overline{s(r)} \qquad (2.6)$$

$$Z^{(2)} = s_s^2(r) + \left[\, \overline{s(r)}\, \right]^2 \qquad (2.7)$$

$$\Lambda(r) = \frac{s_s^2(r)}{\left[\, \overline{s(r)}\, \right]^2} + 1 \qquad (2.8)$$

where $\overline{s(r)}$ and $s_s^2(r)$ is mean and variance, respectively. After the lacunarity $\Lambda(r)$ in the box is obtained, the box is moved to the right and repeat the calculation until the box is moved to the end of the image.

2.6 Convolutional Neural Network

Convolutional neural network (CNN) is deep learning classifier that is widely used in various fields such as computer vision, pattern recognition, artificial intelligence and many more. CNN is suitable to do classification with images as inputs as well as the features or data in 2D or 3D form. Define the layers of CNN is the first step in neural network architecture. CNN mainly consists of input layer, hidden layers and output layer. In the hidden layers consists of convolutional layers, pooling layers, fully connected layers. A basic CNN architecture is shown as Figure 2.5 and each CNN layers are described as below:

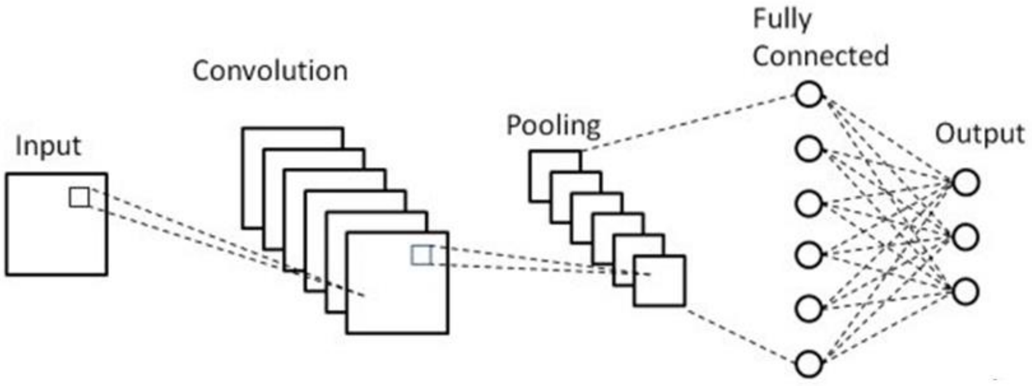


Figure 2.5: Basic CNN Architecture

1. **Input Layer**

There are various types of input can be inputs to the CNN depends on the application that are working on. For the image recognition application, the image input layer is used. The input size in image inputs layer must be specified with height, width, and number of channels. The number of channels is specified based on the types on images. For the grayscale images, the number of channels is 1, the number of channels for RGB images is 3.

2. **Convolutional Layer**

Convolutional layer is the first layer in hidden layer of CNN. In this layer, the sliding window which is convolutional filter is applied to the inputs. The inputs convolve with the filter and then pass the output to the next layer. If the image is RGB image, the filter with 3 channels is applied in the convolution. The size of the filter is necessary to specified because it will affect the output of the layers. Let the input image has a size of $M \times N$ and the size of filter is $m \times n$, then the output of this layer has the size of $(M - m + 1) \times (N - n + 1)$. Stride is another option can be specified in the convolutional layer. Stride is the step size of the filter moves. Considered the previous condition and stride of x is specified, the size of the output of this layer is $\left(\frac{M-m}{x} + 1\right) \times \left(\frac{N-n}{x} + 1\right)$. A CNN can consist of one or more convolutional layers depends on the size of inputs and complexity of the data.

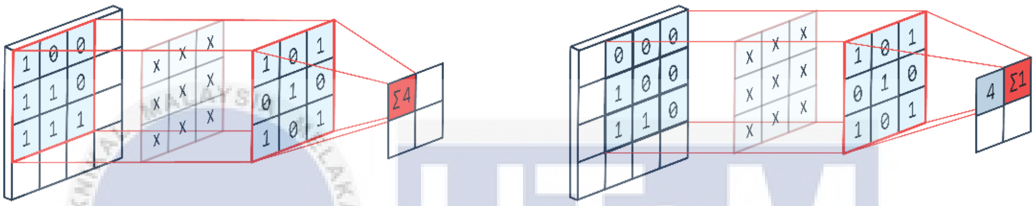


Figure 2.6: Convolutional Layer

3. Pooling Layer

In pooling layer, the down sampling of input is performed. The pooling size is required to specify. There are two types of pooling layer – max pooling layer and average pooling layer. For max pooling layer, the maximum value which covered in the pooling region is taken; while for average pooling layer, the average value in the pooling region is calculated. Figure 2.7 shows an example of both pooling layers. Let the input size is $M \times N$ and size of the pooling region is $p \times p$, the output of the pooling layer is $M/p \times N/p$.

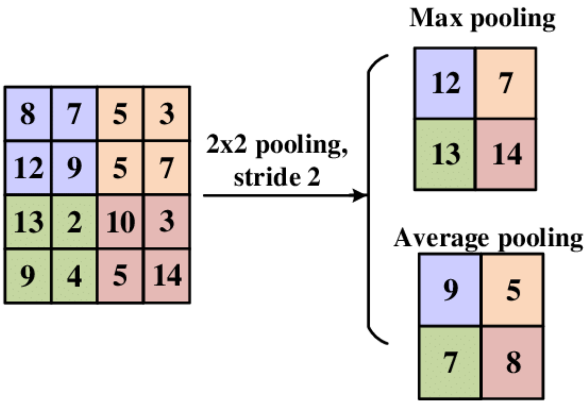


Figure 2.7: Example of Max and Average Pooling Layer

4. Fully Connected Layer

The fully connected layer is basically the last layer of hidden layers and usually used in classification applications. All features learned in the previous layers are combined and fully connected to this layer. If there is more than one fully connected layers are specified, the output size of last layer must same to the number of classes in classification.

2.7 Previous Works on Feature Extractor in Iris Recognition System

Table 2.1 shows the benchmarks of the previous works on the methods of feature extraction in iris recognition system.

Table 2.1: Benchmark of Previous Works on Feature Extractor

Author	Year	Feature Extraction	Related Work
Daouk *et al.* [18]	2002	Haar Wavelet	Apply Haar Wavelet to obtain the 5-level wavelet tree of iris image. These levels include horizontal, vertical and diagonal coefficients. Six distinct coefficients that can represent the core iris pattern are picked and combined as feature vector.
Masek [19]	2003	1D Gabor Filtering	Broken up the 2D image into 1D signals and convolute with 1D Gabor wavelets. The output is quantized to four levels. Then, encode the output into a bitwise template.
Hong-Ying *et al.* [20]	2005	Variation fractal based on steerable pyramid	Set high-threshold and low-threshold. Changes falling between both thresholds are counted. The variant fractal dimension is then calculated. Steerable pyramid analyse and obtain the fractal feature information of iris.
Umer *et al.* [21]	2015	TCM with ordered pattern book	Compute covariance matrix and eigen vectors. Then ordering the code words in pattern book. Then compute TCM and calculate five texture features for each image.
Khotimah and Juniati [22]	2018	Box Counting	Cover the image with size r that change from 1 to 2^n. Then number of boxes that containing the object, N(r), is calculated. Create the straight line from log(1/r) and log(N(r)) and calculate the slope which is fractal dimension.
Khanam *et al.* [23]	2019	Haar Wavelet	Extract features from wavelet change. The feature vector with range -1 to 1 is sign quantized, which 0 and 1 are represented positive and negative respectively.

2.8 Previous Works on Classifier in Iris Recognition System

Table 2.2 shows the benchmarks of the previous works on the methods of classification in iris recognition system.

Table 2.2: Benchmark of Previous Works on Classifier

Author	Year	Classification	Related Work
Daouk *et al.* [18]	2002	Hamming distance (HD)	Calculate HD value by using coefficient of two test image. Based on John Daugman, iris images are same if HD value not more than 0.32 [24].
Hong-Ying *et al.* [20]	2005	Non-symmetrical SVM	Include a constant as the error-punishment for the mistake in classification in equation of classification hyperplane, H. The position of H varies for different vectors.
Roy *et al.* [25]	2011	SVM	Build left and right iris recognition system by two-class approach which is intraclass and interclass difference vector using SVM classifier. Then, train and test the classifier using leave-one-out cross validation.
Khotimah and Juniati [22]	2018	K-Nearest Neighbors (KNN)	KNN assume that similar data are near. Select a value of K and calculate the Euclidean distance between testing and training data. Arrange the Euclidean distance value in ascending order and classify the testing data to majority K label.
Wai *et al.* [11]	2019	Neural Network	Neural network is trained with supervised learning algorithm. In testing stage, the correct recognition is saved into neural network. While the image that failed to recognize is used to trained neural network again.
Tobji *et al.* [26]	2019	MCNN	Iris image is resized N times as the inputs of MCNN. Each CNN contains 6 layers to extract the fractal features. Results in 1 x 1 convolutional maps. After extraction, the classification is performed on classification layer to classify the images.

2.9 Performance of Previous Works

The performance of each previous work is also reviewed as the reference for the decision making of accuracy and recognition speed of iris recognition system. Table 2.3 shows the Correct Recognition Rate (CRR) and time taken for recognition of the iris recognition system of previous works.

Table 2.3: Performance of Previous Works

Author	**Year**	**Evaluation Protocol**	**Performance**	
			Accuracy (%)	**Recognition Time (s)**
Daouk *et al.* [18]	2002	60 iris images	93	31
Hong-Ying *et al.* [20]	2005	756 iris images	99.5	-
Roy *et al.* [25]	2011	Cross-validation	97.21	0.995
Kerim and Mohammed [27]	2014	-	99.4	< 2
Bharath *et al.* [28]	2014	4:1 (training:testing)	84.17	0.44
Umer *et al.* [21]	2015	1386 iris images, leave-one-out cross validation	100	0.98
Al-Waisy *et al.* [29]	2017	1680 iris images from 120 subjects	100	0.89
Khotimah and Juniati [22]	2018	60 iris images	92.63	-
Tobji *et al.* [26]	2019	7:3 (training:testing)	95.63	-
Wai *et al.* [11]	2019	25 iris images (3 training:2 testing)	80	0.02

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will present the methodology to implement the iris recognition system. In this project, MATLAB application is used to implement the iris recognition system. The implementation of iris recognition system is separated into few parts as shown in Figure 3.1. Each part of the implementation is explained in detailed in each sub chapter. After the implementation of iris recognition system, a GUI is built to monitor the process and performance of iris recognition system.

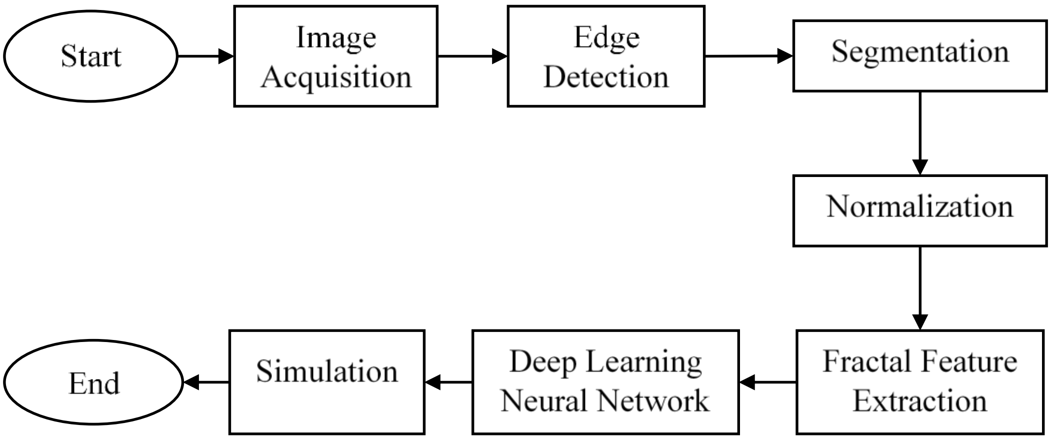


Figure 3.1: Flow Chart of Implementation of Iris Recognition

3.2 Image Acquisition

3.2.1 Requirements of Iris Image

Image acquisition is claimed as the most important step in this project since the upcoming process are highly depends on the quality of the iris image. The iris image used in the iris recognition system must be clear and sharp [11]. The boundary between iris and pupil, and boundary between iris and sclera must be clear. The fractal of the iris must also be clear because this is the most important part that represent as the identity of the person.

In addition, the iris image is very hard to capture since the lighting, distance and the reflection of light will affect the quality of the iris image. These problems are hard to remove by program of MATLAB and will affect the iris fractal feature extraction. Moreover, the grayscale iris images are used instead of RGB iris images because RGB colours contain three luminance values while grayscale has only one luminance. So, it is easier to do image processing on grayscale images with lesser arguments compared to RGB images.

3.2.2 CASIA Iris Image Database V4.0 (CASIA-IrisV4)

CASIA Iris Image Database (CASIA-Iris) has been developed by the research group of Center for Biometrics and Security Research (CBSR). The CASIA-Iris has been updated from CASIA Iris Image Database V1.0 (CASIA-IrisV1) to CASIA Iris Image Database V4.0 (CASIA-IrisV4) since 2002 and all databases are free to download from the website of CBSR. The latest database which is CASIA-IrisV4 contains a total of 54,609 iris images and is categorized into 6 subsets as shown in Table 3.1 and Figure 3.2.

Table 3.1: Characteristics of CASIA-IrisV4 Subsets

Name of Subset	**Sensor**	**Characteristics**
CASIA-Iris-Interval	CASIA close-up iris camera	Both boundaries between pupil, iris and sclera, and the fractal feature of iris are extremely clear.
CASIA-Iris-Lamp	OKI IRISPASS-h	Deformation of iris is captured by the variation of visible illumination.
CASIA-Iris-Twins	OKI IRISPASS-h	Collection of iris images from 100 pairs of twins.
CASIA-Iris-Distance	CASIA long-range iris camera	Long range and high quality of face images with iris.
CASIA-Iris-Thousand	Irisking IKEMB-100	Collection of iris images from 1000 subjects. Part of the subjects are wearing spectacles.
CASIA-Iris-Syn	CASIA iris image synthesis algo	Contains 10,000 artificial iris images that synthesized from the real iris images. The artificial iris images are embedded into the real eye image to look more realistic.

Note: Data are from the website of CBSR [30]

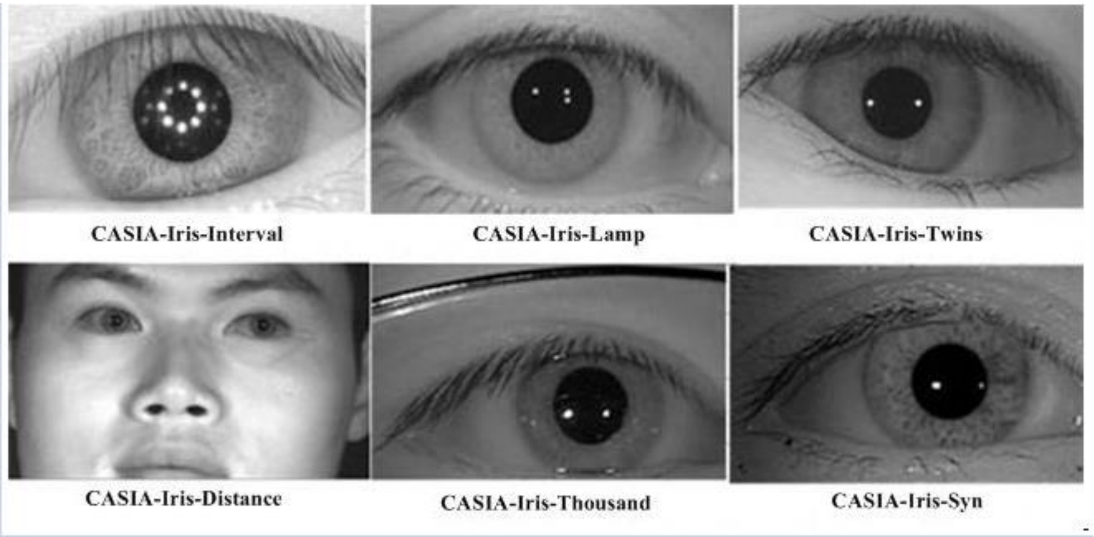


Figure 3.2: 6 Subsets of CASIA-IrisV4

From Table 3.1, we choose CASIA-Iris-Interval subset as the training and test images in the iris recognition system because the requirements for image acquisition are met. The iris images from this subset have higher quality and the iris patterns are also extremely clear. These iris images are taken by using self-designed circular Near-Infrared Light Emitting Diode (NIR LED) with a suitable luminance, so it does not cause the reflection of light on the iris and at the same time capture the details and boundaries of the iris. In this subset, the type of iris images is jpg and have a standard size of 280×320.

In this project, 10 persons with 10 iris images each are used. From each 10 iris images, 5 iris images are used to train the neural network and 5 iris images are used as test image to analyse the performance of the designed iris recognition system. In short, 100 iris images of 10 persons are used, which is split into 50 iris images for training and another 50 iris images for testing.

3.3 Edge Detection

3.3.1 Inner Iris Boundary

In the edge detection part, the inner iris boundary is detected first. The inner iris boundary is referred to the boundary between pupil and iris. Hence, the centre and radius of inner iris boundary is the same with the centre and radius of pupil. Pupil of eye is normally a circle, therefore, to detect a circle, circular Hough transform is used. Circular Hough transform is a function in Image Processing Toolbox of MATLAB which detect circles in the image. The object polarity which is one of the function's input argument is set to dark to ensure that only the black pupil is detected in the image. After the pupil is detected, the coordinates of the pupil centre its radius are returned as output. The centre of iris is assumed to be the same as centre of pupil.

3.3.2 Outer Iris Boundary

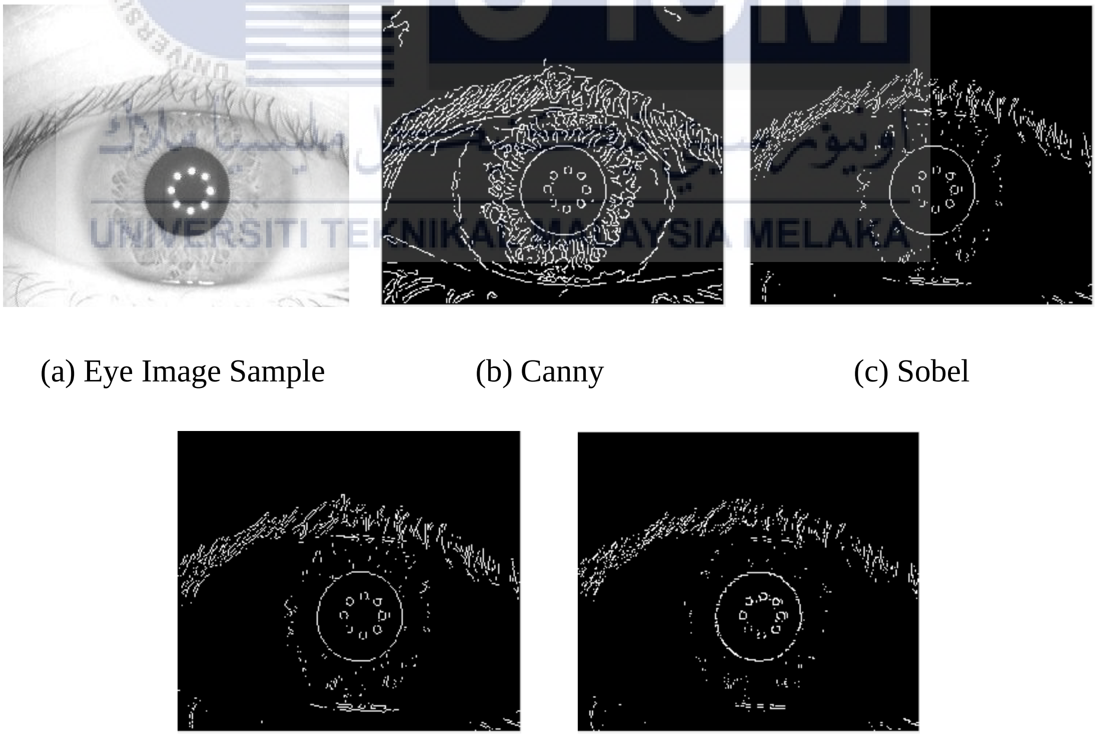
3.3.2.1 Edge Detection and x increment

There are several edge detection algorithms can be used to detect edge in the image. Table 3.2 shows the descriptions of each edge detection algorithms and Figure 3.3 shows the output images of the edge detection algorithms.

By the analysis of output images in Figure 3.3, the Canny edge detection algorithm is chosen as the edge detector in the iris recognition system. This is because, as shown in Figure 3.3, the Canny method gives more details compared to other algorithms and only Canny method detected the outer iris boundary.

Table 3.2: Characteristics of Edge Detection Operators [31]

Algorithm	Description
Canny	Find edges by looking for local maxima of the gradient of input image. The gradient is calculated by using the derivative of Gaussian filter. This method is less likely to be disturbed by the noise and detected weak edges.
Sobel	Find edges at the maximum gradient of the input image. The derivative of Sobel approximation is used.
Prewitt	Find edges at the maximum gradient of the input image. The derivative of Prewitt approximation is used.
Roberts	Find edges at the maximum gradient of the input image. The derivative of Roberts approximation is used.

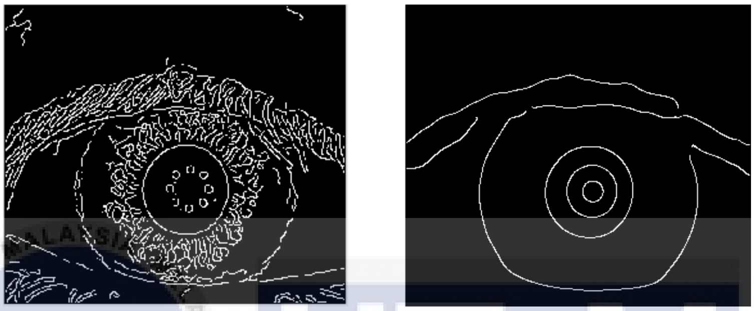


(a) Eye Image Sample (b) Canny (c) Sobel

(d) Prewitt (e) Roberts

Figure 3.3: Edge Detection Algorithms

Two arguments are included into the edge detection function which is sensitivity thresholds and standard deviation of Gaussian filter. Gaussian filter is specified to a higher sigma to remove the unused edges or noise edges. For sensitivity thresholds, the edges that are lower than the low threshold value are neglected, and the edges that higher than the high threshold value are preserve. Figure 3.4 shows the example of unused edges removal.



(a) Before (b) After

Figure 3.4: Noise Removal in Canny Edge Detector

Next, x increment method is used to determine the outer boundary of iris. First, set the x-value, x_o, as equation (3.1):

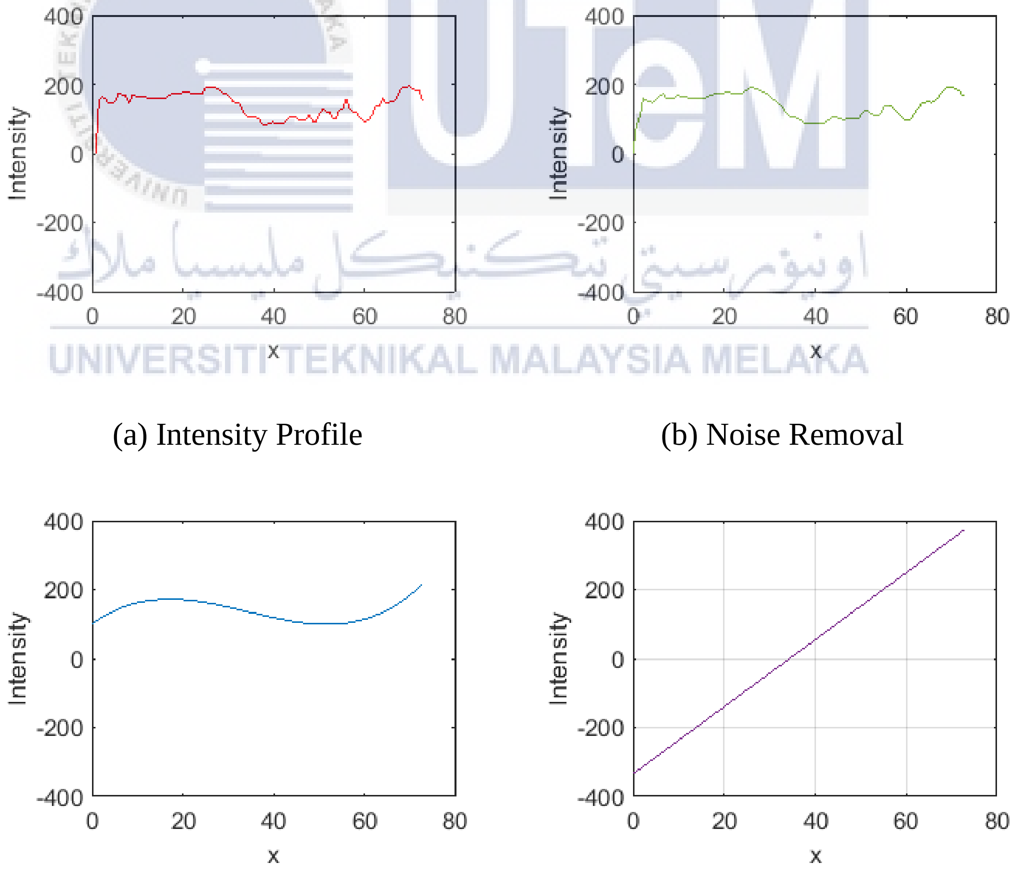
$$x_o = x_c + r_i + 20 \tag{3.1}$$

where x_c is x-coordinate of the centre of pupil and r_i is the radius of inner iris boundary. The 20 pixels are added to keep away from the detected inner iris boundary. The x_o is then start increase by 1 if the pixel is black and stop when the pixel is white. After the x increment stops, the radius of outer iris boundary, r_o, is calculated as equation (3.2):

$$r_o = |x_o - x_c| \tag{3.2}$$

3.3.2.2 Intensity Profile

Intensity profile is an alternative method when x increment method is failed. Figure 3.5 shows the plot to determine the outer boundary of iris by using intensity profile. First, the horizontal intensity profile (Figure 3.5(a)) which align with centre of pupil and across the outer iris boundary is determined. Next, the noisy data in intensity profile is smoothed (Figure 3.5(b)) and fitted by cubic polynomial (Figure 3.5(c)). The second derivative of cubic polynomial is performed to obtain the corresponding linear polynomial (Figure 3.5(d)). The root of the linear polynomial is the x-coordinate of the outer iris boundary, x_o. Lastly, the radius of outer iris boundary, r_o, can be calculated as equation (3.2).



(a) Intensity Profile

(b) Noise Removal

(c) Cubic Polynomial

(d) Linear Polynomial

Figure 3.5: Plot of Intensity Profile Method

3.3.3 Verification

After the coordinate of pupil centre (x_c, y_c), radius of inner iris boundary r_i, and radius of outer iris boundary r_o are extracted, two circles are drawn by using this three information to visualize the result of edge detection. Figure 3.6 shows an example of success edge detection.

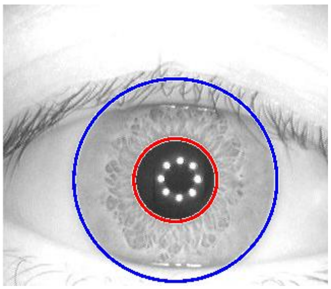


Figure 3.6: Result of Edge Detection

3.4 Segmentation

Before segmentation is performed, the distance between each pixel (i, j) is computed as equation (3.3):

$$d = \sqrt{(i - x_c)^2 - (j - y_c)^2} \qquad (3.3)$$

Since iris part is between outer boundary and inner boundary of iris. For every pixel, the distance which is larger than r_o and smaller than r_i is ignored. Only the pixels with distance between r_o and r_i are remained. Figure 3.7 shows an example of segmentation of iris part from iris image.

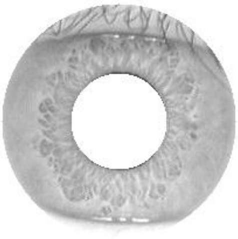


Figure 3.7: Segmentation of Iris Part

3.5 Normalization

After the segmentation, the segmented iris image is in ‘doughnut’ shape or annular shape. Since the radius of iris is varying due to the deformation of iris in different condition, the thickness of the segmented iris images is different. Therefore, the normalization of segmented iris image is carried out to standardize the size of iris image.

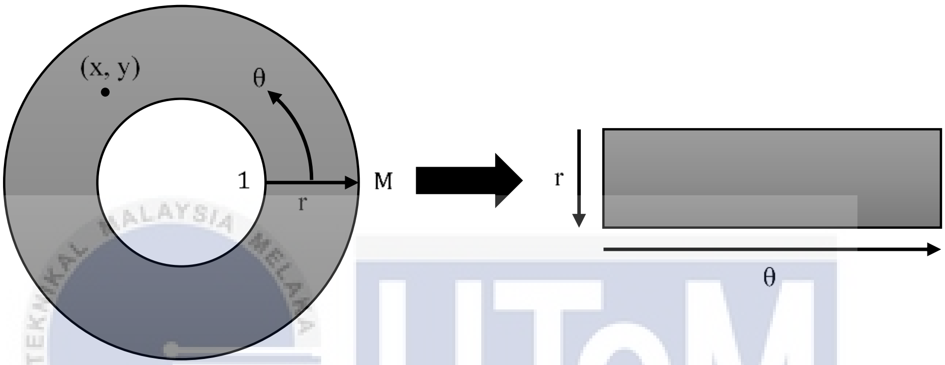


Figure 3.8: Daugman’s Rubber Sheet Model

Daugman’s Rubber Sheet Model [32] is the most effective way to do normalization which introduced by Daugman. This model normalizes the segmented iris image from annular shape into rectangular block as shown in Figure 3.8. The dimension of the resulting rectangular iris image is in polar coordinate (r, θ) where the range of r is between $[1, M]$ and θ is $[0, 2\pi]$. The expression from Cartesian coordinates (x, y) to polar coordinate (r, θ) is expressed as:

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

with

$$x(r,\theta) = (1-r)x_i(\theta) + rx_o(\theta) \tag{3.4}$$

$$y(r,\theta) = (1-r)y_i(\theta) + ry_o(\theta) \tag{3.5}$$

where (x_i, y_i) and (x_o, y_o) is the coordinate of inner iris boundary and outer iris boundary along the θ direction which is expressed as:

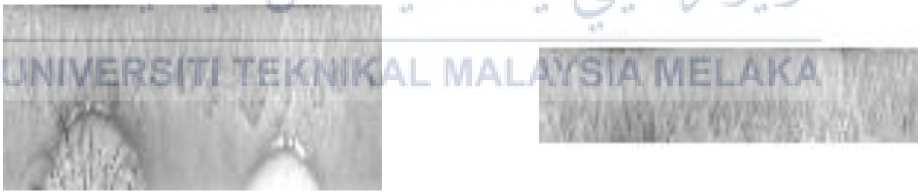
$$x_i(\theta) = x_c + r_i \, cos(\theta) \tag{3.6}$$

$$y_i(\theta) = y_c + r_i \, sin(\theta) \tag{3.7}$$

$$x_o(\theta) = x_c + r_o \, cos(\theta) \tag{3.8}$$

$$y_o(\theta) = y_c + r_o \, sin(\theta) \tag{3.9}$$

In this proposed iris recognition system, the normalized iris image is standardized to 64×128 pixels. Figure 3.9(a) shows an example of normalized iris image. In the normalized iris image, the upper half as shown in Figure 3.9(b) contains the fractal of iris which is the region of interest (ROI) that is used to extract the fractal feature. While the lower half of the normalized iris image contains eyelid and eyelashes which is unusable. Hence, the normalized iris image is divided into half and the upper part with the size of 32×128 pixels is proceeded to fractal feature extraction step.



(a) Normalized Iris Image (b) ROI

Figure 3.9: Normalization and ROI of Iris Image

3.6 Fractal Feature Extraction

In this section, the normalized iris image is acquired and the fractal dimension (FD) of the image is computed. For the FD estimation, Differential Box Counting (DBC) method [33, 34, 35, 36] is used. By using DBC, a box is represented as a grey level intensity. So, each of the pixels are estimated by boxes which represented the pixels' intensity. The 2D image is then represented by boxes that required to cover the intensity of image. Figure 3.10 shows an example of image with 10 × 10 pixels that cover by boxes.

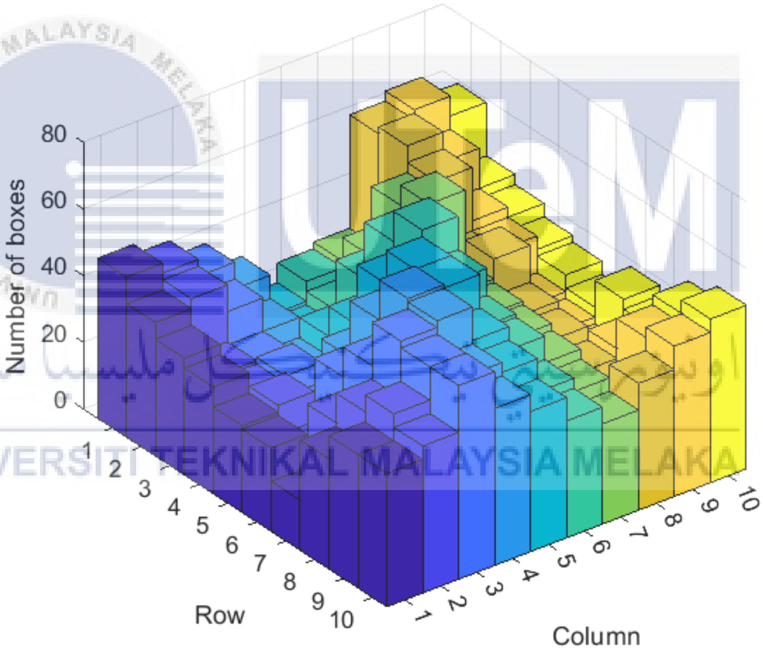


Figure 3.10: Example of Image Covered by Boxes

To obtain the number of boxes necessary to cover up the image with size of $M \times N$, a kernel of size $r \times r$ is created with function (3.10):

$$w(s,t) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} floor\left(\frac{g_{max} - g_{min}}{r}\right) + 1 \qquad (3.10)$$

where

$$r = 2, 3, \dots, j$$

$$a = b = ceil\left(\frac{r-1}{2}\right) \qquad (3.11)$$

The kernel with scaling factor r from 2 to j is applied on every pixel of normalized iris image, I. The number of boxes, N_d, for each r is then obtained by equation (3.12):

$$N_d(x, y, r-1) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) I(x+s, y+t) \left(\frac{j}{r}\right)^2 \qquad (3.12)$$

Next, the FD is estimated by using least squares linear regression as Figure 3.11. First, the logarithm of r and logarithm of $N_d(x, y, r-1)$ are computed and stored in the form as expression (3.13) and (3.14). Both m and $n(x,y)$ are in the form of matrix. The length of matrix m is $j-1$ while the size of matrix $n(x,y)$ is same to the normalized iris image which is $M \times N$. In $n(x,y)$, each element is an array with length $j-1$ which compromised the logarithm of $N_d(x, y, r-1)$ from $r = 2$ to j. For example, the first element of n which is $n(1,1)$ stores all first element in logarithm of $N_d(x, y, r-1)$ in array form.

$$m = log(r) = \begin{bmatrix} log(2) & log(3) & \cdots & log(j) \end{bmatrix} \qquad (3.13)$$

$$n(x,y) = \begin{bmatrix} log(N_d(x,y,1)) & log(N_d(x,y,2)) & \cdots & log(N_d(x,y,j-1)) \end{bmatrix} (3.14)$$

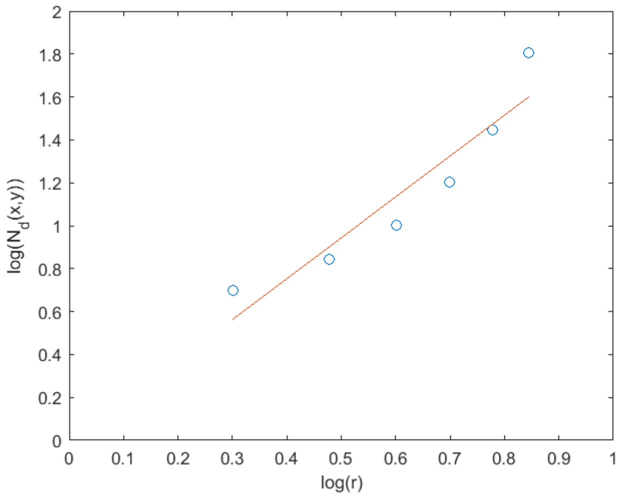


Figure 3.11: Estimation of $FD(x,y)$ by using Least Squares Linear Regression

After that, the sum of squares $\mathrm{SS_{mm}}$ and $SS_{mn}(x,y)$ is computed as equation (3.15) and (3.16). Lastly, the slope of least squares linear regression line which is $FD(x,y)$ is obtained by compute the equation (3.17).

$$SS_{mm} = \sum m^2 - \frac{(\sum m)^2}{j-1} \tag{3.15}$$

$$SS_{mn}(x,y) = \sum m \cdot n(x,y) - \frac{\sum m \cdot \sum n(x,y)}{n} \tag{3.16}$$

$$FD(x,y) = \frac{SS_{mn}(x,y)}{SS_{mm}} \tag{3.17}$$

In the proposed iris recognition system, the scaling factor r is set to the range from 2 to 7. Since the normalized iris image has a size of 32 × 128 pixels, the resulting FD is also in 32 rows × 128 columns.

The sliding box lacunarity [35, 16, 17] with window size $r \times r$ is applied to the FD. The function of the window is expressed as (3.18) which is equivalent to variance over mean squared.

$$Lacunarity, L(x,y) = \frac{\frac{1}{r^2}\sum_{s=-a}^{a}\sum_{t=-b}^{b} FD(x+s,y+t)^2}{\left(\frac{1}{r^2}\sum_{s=-a}^{a}\sum_{t=-b}^{b} FD(x+s,y+t)\right)^2} \quad (3.18)$$

In the proposed iris recognition system, the size of sliding box lacunarity r is set to 5 and the a and b is determine as equation (3.11).

3.7 Deep Learning Neural Network

Since the extracted fractal feature is in 2D, I use Convolutional Neural Network (CNN) as the classification algorithm of iris recognition system. The designed CNN consists of 10 layers. The input size of the CNN is set as the size of normalized iris image which is 32×128. The first 2D convolution layer with 3×3 filter and max-pooling layer with pool size of 2×2 and stride of 2 is specified. Next, the second 2D convolution layer with 4×4 filter and the second max-pooling layer with the same argument as first max pooling layer is specified. After that, the fully connected layer combines all of the features into 10 classes since iris images of 10 persons are used. Finally, the output layer which contains softmax layer and classification layer are specified.

Figure 3.12 shows the architecture of the CNN. After the layers are defined, the training options for CNN is set up. The solver is specified as "adam" and the maximum number of epochs for training is set to 500.

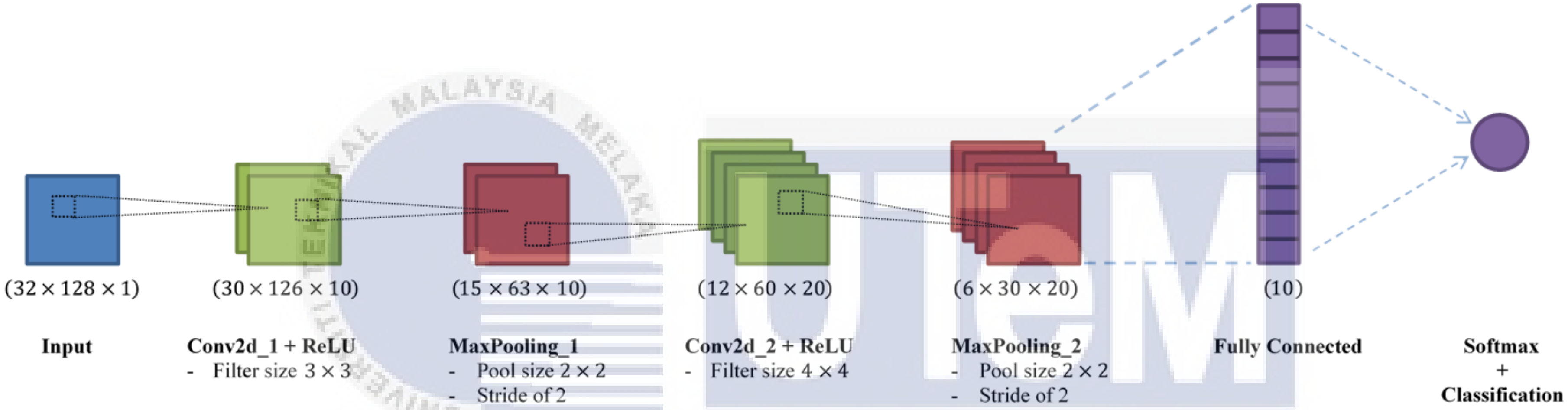


Figure 3.12: Architecture of CNN for Iris Recognition System

3.8 Simulation

Figure 3.13 shows the flow chart to train and test the neural network. Before the simulation, the 50 train iris images undergo the process of edge detection, segmentation and normalization. Then, the 50 normalized iris images are saved into a file. When the simulation starts, then the 50 normalized iris images are read from the file. This can save the processing time before the CNN is started to train. The normalized iris images are then undergoing fractal feature extraction process and use to train the CNN.

Next, the test iris images are used to test the CNN. In the testing process, if the performance of recognition is bad or failed, the argument of CNN is changed and retrain. The trained CNN that gives that best classification results are saved.

After the CNN is saved, all test iris images are used to test the whole iris recognition system. The performance of the iris recognition system is analysed in two aspects which are accuracy and recognition speed. The CRR is the measurement of accuracy of the iris recognition system which is calculated as equation (3.19).

$$CRR = \frac{Number\ of\ test\ images\ recognize\ correctly}{Total\ number\ of\ test\ images} \times 100\% \qquad (3.19)$$

Different to the train images, the test images are going through the process from image acquisition to classification without saving the normalized iris image. This is because the recognition time are counted from the image acquisition until the classification is done. Each recognition time of the test image is recorded and the average recognition time is calculated. The recognition speed is then analysed by the average recognition time. Based on the performance of previous works on iris recognition system as Table 2.3, the parameter for the performance of the iris recognition system is decided as Table 3.3.

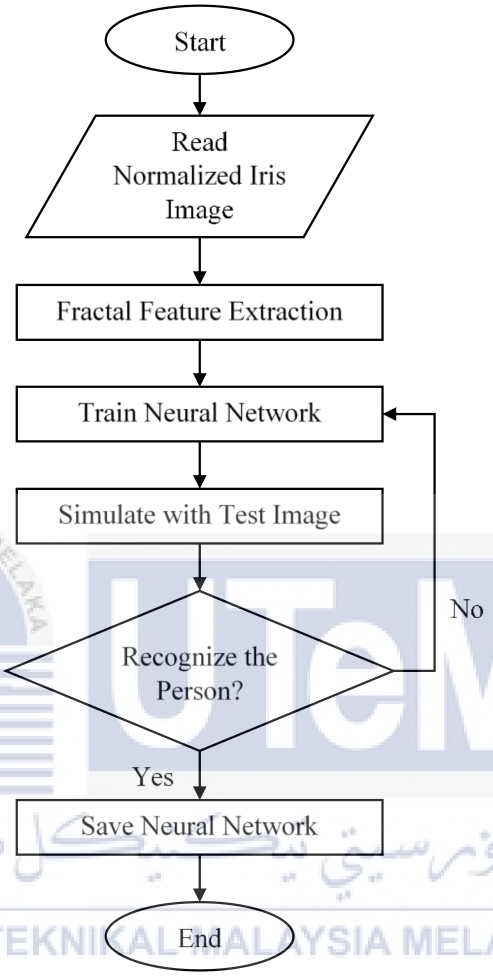


Figure 3.13: Flow Chart for Neural Network Training and Simulation

Table 3.3: Parameter for Performance of Iris Recognition System

(a) Accuracy

Accuracy	**CRR (%)**
High	> 90
Medium	80 – 90
Low	< 80

(b) Recognition Speed

Speed	**Average Recognition Time (s)**
Fast	< 1.0
Medium	1.0 – 2.0
Slow	> 2.0

3.9 Graphical User Interface (GUI)

To monitor the performance of the iris recognition system, a GUI is designed and built by using App Designer provides in MATLAB. The GUI is designed with three parts which is manual, automatic and result. In manual part, push buttons are inserted for every process. After the push button is push, the image of each process will show. So, when testing the coding with iris images, the output images of each process can be monitor and observe whether there is an error. On the other side, automatic part is faster version of manual part. After open the image, only one push button is allowed to push to start recognition. By using both parts, the recognition results and time taken will show in the result screen. When want to start a new recognition, a push button can be pushed to clear all the images, results and saved data in the GUI.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Edge Detection and Segmentation

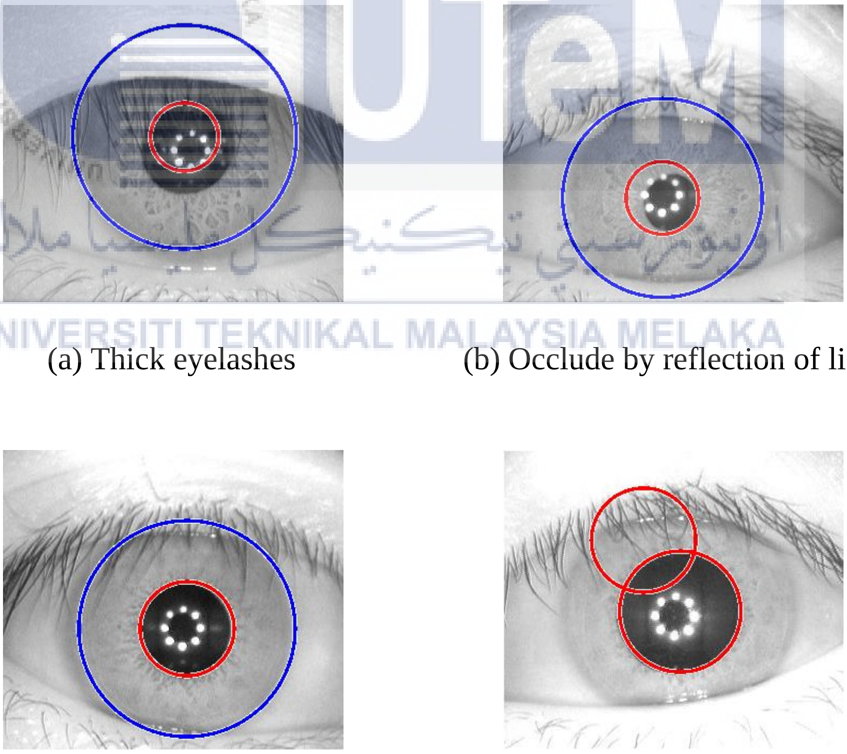
There are 100 iris images used in the iris recognition system which is 50 train images and 50 test images. After edge detection and segmentation process, both inner and outer iris boundary of all iris images are detected, and the iris images are segmented perfectly.

The iris images in the CASIA-Iris-Interval subset from CASIA-IrisV4 are tested. Most of the iris images can detected the boundaries and segmented successfully. However, the proposed edge detection algorithm is also failed at some problematic iris images. These are the reasons that cause the edge detection is failed:

1. Thick eyelashes that cover the pupil.
2. Reflection of light occludes the pupil boundary.
3. Pupil is not at the centre of iris.
4. Very small difference of between grey level intensity of iris and sclera.

5. Misdetection by circular Hough Transform.

The first two problems are mainly because of occlusion of pupil. In the proposed edge detection algorithm, the pupil is detected by using circular Hough transform and extract the radius and centre in the first place. Then the radius of outer iris boundary is determined based on the centre of pupil. If the pupil is occluded by thick eyelashes, eyelid or reflection of light, the circular Hough transform is failed to work, the radius and centre of pupil are not extracted. Therefore, the inner iris boundary is failed to detect. Since the centre of pupil are absent, the outer iris boundary is also failed to detect. Hence, the iris part is failed to segment from the iris image. Figure 4.1(a) and Figure 4.1(b) are the examples of the two problematic iris images as described.



(a) Thick eyelashes

(b) Occlude by reflection of light

(c) Pupil not centred

(d) Misdetection

Figure 4.1: Examples of Problematic Iris Images

For the third problem, the pupil is not centred means the thickness on both sides of iris are different. This with cause some part of the iris that is on the thicker side are not segmented. The iris image as shown in Figure 4.1(c) is one of the examples that the pupil is not at the centre of iris. However, the segmentation as Figure 4.1(c) is acceptable or is called false accept because the part of iris that is not segmented does not include fractal of iris which is the ROI. Also, the thinner side is fully segmented and the ROI is fully included in the segmented area.

Next, the iris images with the fourth problem will failed in the detection of outer iris boundary. The Canny edge detector is hard to detect the edge and also the intensity profile is failed to work since the step edge are not obvious. This problem is rarely happened because most of the iris and sclera has obvious difference in term of grey level intensity. Unless the iris of the person is light in colour while the sclera is deeper than ordinary colour. When transform into grey scale, the grey level intensity will then become very close.

The last problem is misdetection by circular Hough transform. This misdetection includes detect pupil in incorrect position and detected more than one circle in the iris image. If the pupil is detected incorrectly, then the information which is radius and centre of pupil are incorrect. The first and second problems are the main cause of this problem. For another misdetection which is extra circles detection, in this case, the information of the extra circles is extracted. This causes the error on outer iris boundary detection because of more than one set of pupil information. Figure 4.1(d) is the example which detected an extra circle apart from the pupil.

4.2 Graphical User Interface (GUI)

Figure 4.2 shows the GUI of iris recognition system. When the GUI is start up, the saved CNN is loaded and the GUI is ready to use. In manual part, when the “1. Insert Image” push button is clicked, a file selection window is opened to choose the image as input. Next, when the push buttons in the order of “2. Edge Detection”, “3. Segmentation” and “4. Normalization” is clicked, the corresponding output images are shown in the axes area. Then, “5. Fractal Feature Extraction” push button is clicked. After the “DONE!” word is shown, then the “Start Recognize” push button is clicked. The name of the person and the recognition time is shown in the result screen. Figure 4.3 shows the demonstration of the manual part of GUI.

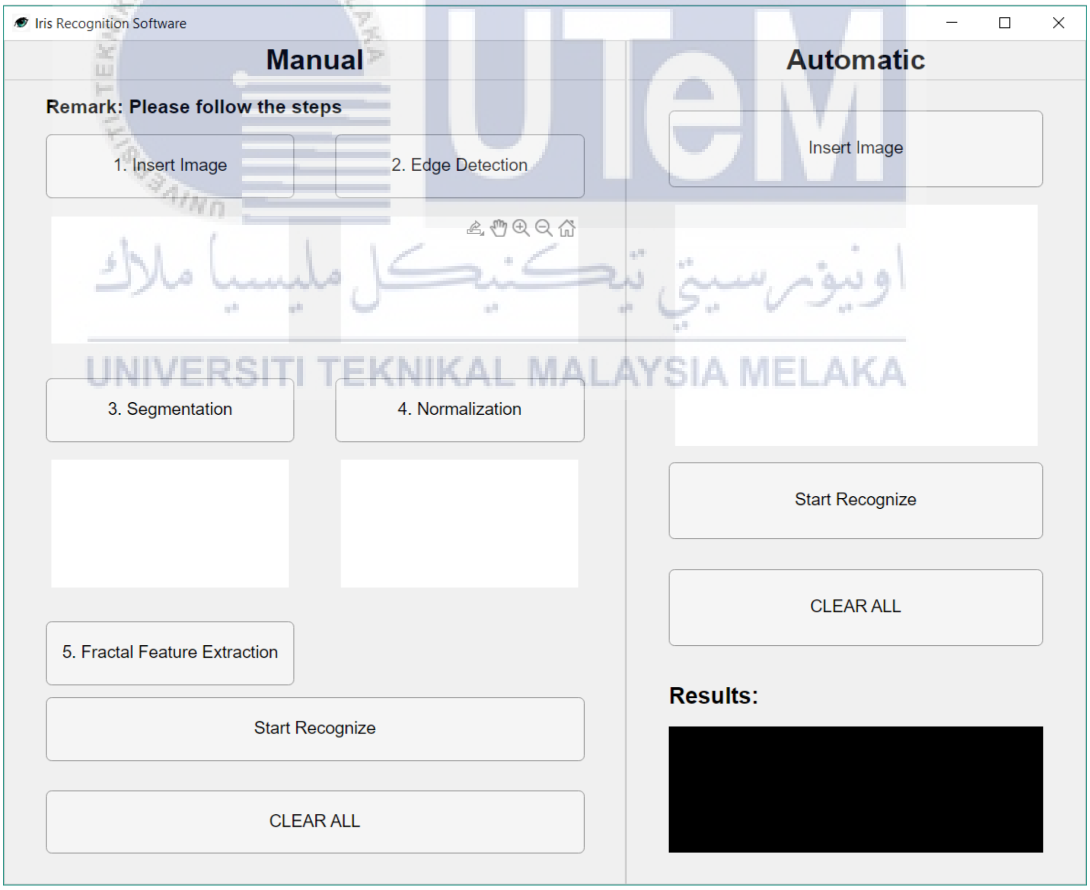
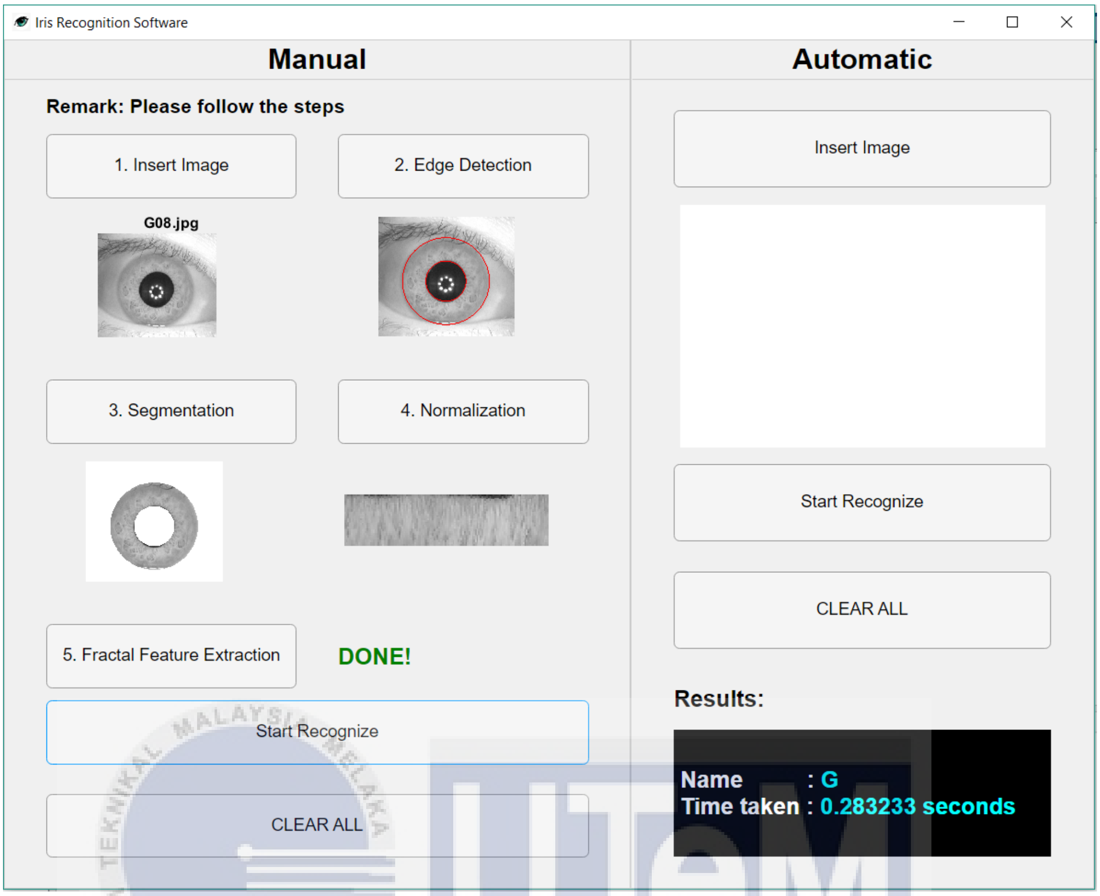


Figure 4.2: GUI of Iris Recognition System



Remark: Please follow the steps

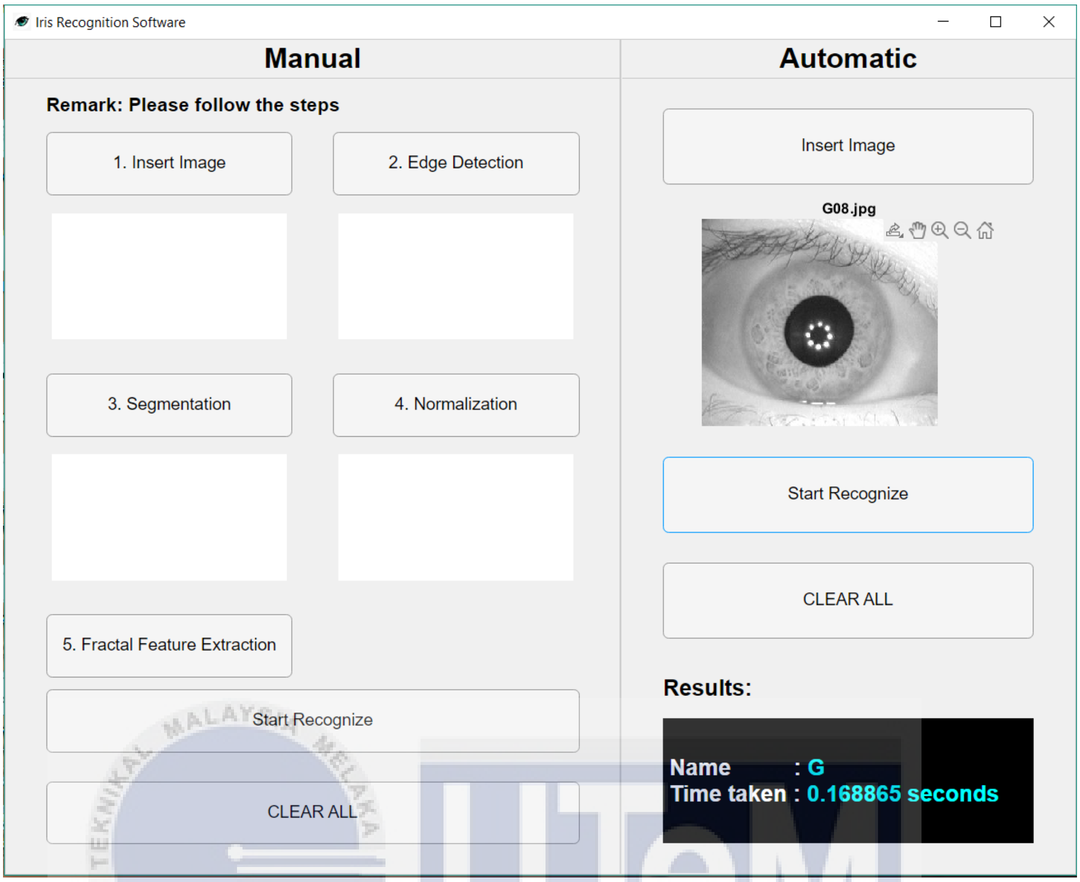
DONE!

Results:

```
Name       : G
Time taken : 0.283233 seconds
```

Figure 4.3: Demonstration on Manual Part of GUI

To start a new recognition, “CLEAR ALL” push button must be pushed to clear all data. For the automatic part as shown in Figure 4.4, the image can be chosen by click the “Insert Image” push button and start the recognition by clicking “Start Recognize” push button. The results are also shown in the result screen.



```
Name       : G
Time taken : 0.168865 seconds
```

Figure 4.4: Demonstration on Automatic Part of GUI

From Figure 4.3 and Figure 4.4, the recognition time for manual part is higher than automatic part. This is because the time taken to show the images are included. The recognition time for the automatic part is started when the “Start Recognize” push button is clicked until the classification is done.

In short, the manual part is used to monitor and observe the output images of each process while for the automatic part, the time taken to recognize the test images are recorded to calculate the recognition time and analyse the performance of the iris recognition system.

4.3 Performance of Iris Recognition System

In the proposed iris recognition system, the iris images are undergone the process from image acquisition to fractal feature extraction. Then, the extracted fractal features are used to train and test the CNN. The CRR is analysed and recognition time is taken from image acquisition stage to the end of classification.

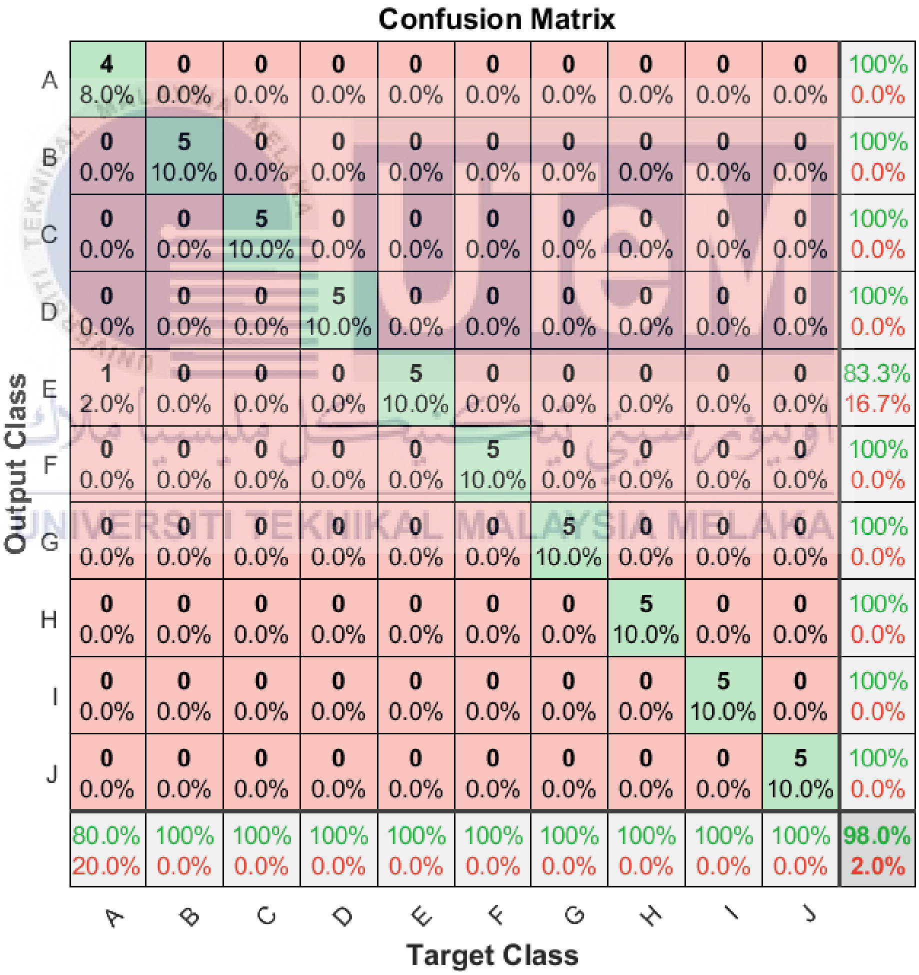
The performance of iris recognition system is analysed in another two cases. First case which is Case A, the iris images undergo the process from image acquisition to classification except fractal feature extraction process. This means in Case A, the fractal features of normalized iris images are not extracted and the normalized iris images itself are used to train and test the CNN. The CRR and recognition time of this case are calculated.

Second case which is Case B, the iris recognition system only classifies the raw iris images by using CNN without going through edge detection, segmentation, normalization and fractal feature extraction. In the CNN, the input size of input layer is changed to 280×320, the arguments in both 2D convolution layers are specified to filter size of 2×2 and stride of 2, while other layers keep unchanged. After the CNN is trained by raw iris train images, the raw iris test images are used to test the iris recognition system. The CRR and recognition time are then analysed.

Lastly, the performances of the proposed iris recognition system and the two cases are compared and discussed.

4.3.1 Proposed Iris Recognition System

After the extracted fractal features of 50 test images are tested in the proposed iris recognition system, the classification confusion matrix is plotted as shown in Figure 4.5. From the classification confusion matrix, all test images are recognized correctly except of one image. One of the iris images belongs to person A is recognized incorrectly as person D. The green value in bottom right corner of the classification confusion matrix is the CRR of the proposed iris recognition system which is 98%.



Confusion Matrix

Output Class \ Target Class	A	B	C	D	E	F	G	H	I	J	
A	**4** 8.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
B	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
C	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
D	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
E	**1** 2.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	83.3% 16.7%
F	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
G	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
H	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
I	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	**0** 0.0%	100% 0.0%
J	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 10.0%	100% 0.0%
	80.0% 20.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	**98.0%** **2.0%**

Figure 4.5: Classification Confusion Matrix

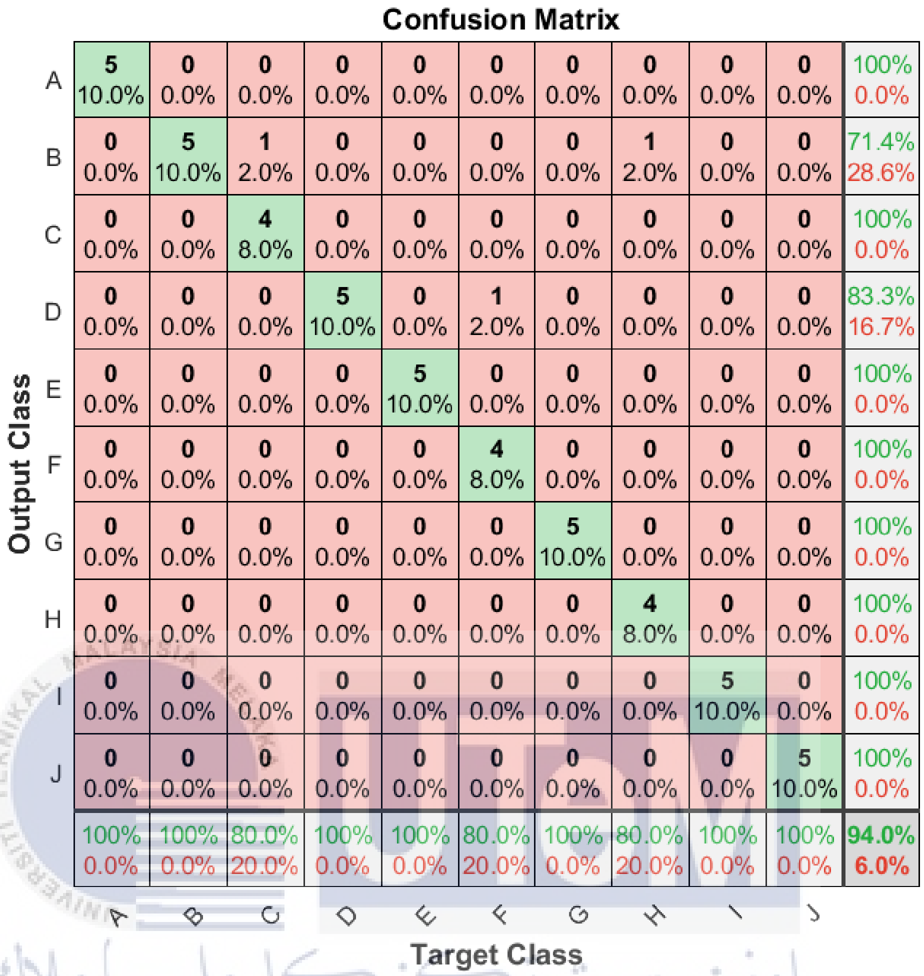
For each test process, the recognition time are taken from image acquisition to the end of classification. The recognition time of each test image is recorded as Table A.1 in Appendix A and the average recognition time is calculated. After calculation, the average recognition time is 0.1704s.

Thus, refer to Table 3.3, the performance of the proposed iris recognition system is high accuracy and fast.

4.3.2 Iris Recognition System that Classifies based on Normalized Iris Images (Case A)

In Case A, the 50 normalized iris train images are read from the file and used to train the CNN. Then, the iris recognition system is tested by 50 normalized iris test images. Figure 4.6 shows the classification confusion matrix. From the classification confusion matrix, there three iris images are misclassified. Thus, by referring to Table 3.3(a), the iris recognition system without fractal feature extraction has 94% of CRR which is considered as high accuracy.

In each of the testing process, the recognition time is taken start from image acquisition, edge detection, segmentation, normalization until the end of classification. Table A.2 in Appendix A shows the recognition time of each normalized iris test image. From the table, the average recognition time is 0.1631s which is considered as fast referring to Table 3.3(b).



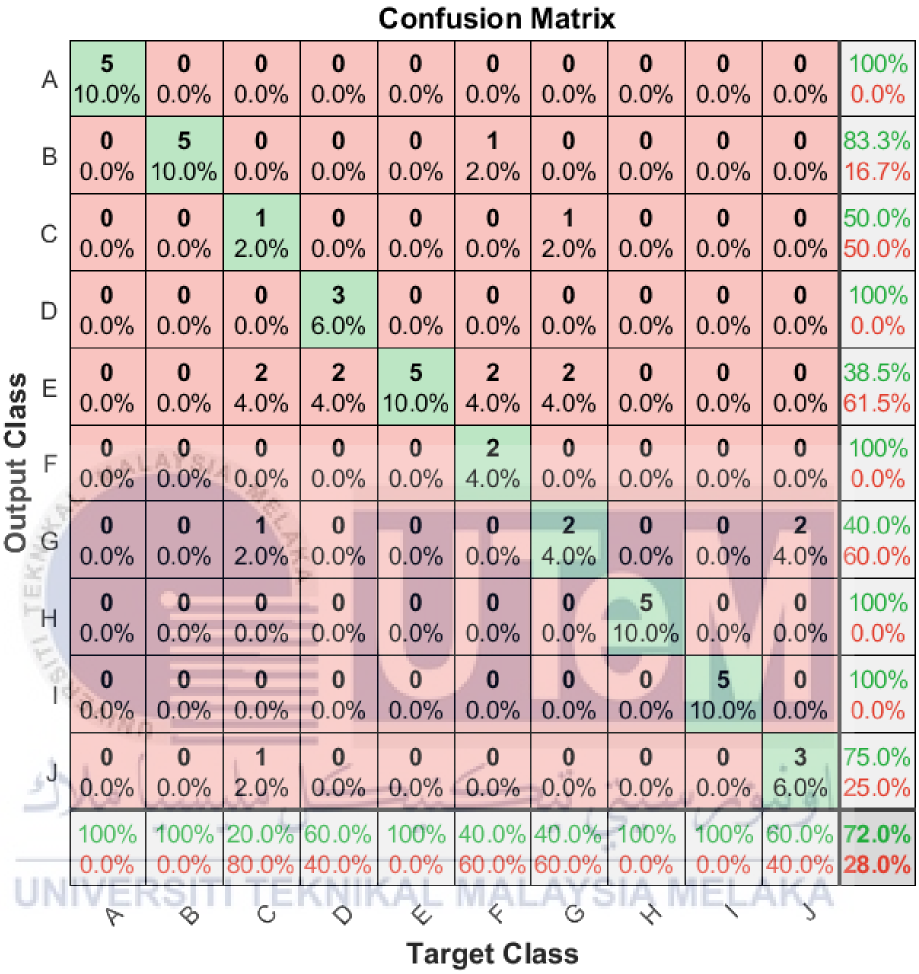
Confusion Matrix

Output Class \ Target Class	A	B	C	D	E	F	G	H	I	J	
A	5 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
B	0 0.0%	5 10.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	71.4% 28.6%
C	0 0.0%	0 0.0%	4 8.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
D	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	83.3% 16.7%
E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 8.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
H	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 8.0%	0 0.0%	0 0.0%	100% 0.0%
I	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	100% 0.0%
J	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	100% 0.0%
	100% 0.0%	100% 0.0%	80.0% 20.0%	100% 0.0%	100% 0.0%	80.0% 20.0%	100% 0.0%	80.0% 20.0%	100% 0.0%	100% 0.0%	**94.0%** **6.0%**

Figure 4.6: Classification Confusion Matrix based on Normalized Iris Image

4.3.3 Iris Recognition System that Classifies based on Raw Iris Images (Case B)

In Case B, after the CNN is trained, the 50 raw iris test images are tested in the iris recognition system. Figure 4.7 is the classification confusion matrix that shows the classification results. From the classification confusion matrix, only person A, B, E, H and I are successfully recognized through the raw iris images while others do not recognize more than 3 raw iris images. The CRR shows in the bottom right corner is 72% which is low accuracy referring to parameter of performance in Table 3.3(a).


Confusion Matrix
Output Class
Target Class

	A	B	C	D	E	F	G	H	I	J	
A	5 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
B	0 0.0%	5 10.0%	0 0.0%	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	83.3% 16.7%
C	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
D	0 0.0%	0 0.0%	0 0.0%	3 6.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
E	0 0.0%	0 0.0%	2 4.0%	2 4.0%	5 10.0%	2 4.0%	2 4.0%	0 0.0%	0 0.0%	0 0.0%	38.5% 61.5%
F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 4.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
G	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	2 4.0%	0 0.0%	0 0.0%	2 4.0%	40.0% 60.0%
H	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	0 0.0%	100% 0.0%
I	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 10.0%	0 0.0%	100% 0.0%
J	0 0.0%	0 0.0%	1 2.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 6.0%	75.0% 25.0%
	100% 0.0%	100% 0.0%	20.0% 80.0%	60.0% 40.0%	100% 0.0%	40.0% 60.0%	40.0% 60.0%	100% 0.0%	100% 0.0%	60.0% 40.0%	**72.0%** **28.0%**

Figure 4.7: Classification Confusion Matrix based on Raw Iris Image

The recognition time of each testing process of raw iris image is taken and recorded as Table A.3 in Appendix A. The average recognition time calculated from Table A.3 is 0.0133s which is considered as fast recognition speed refer to Table 3.3(b).

4.3.4 Comparison and Analysis

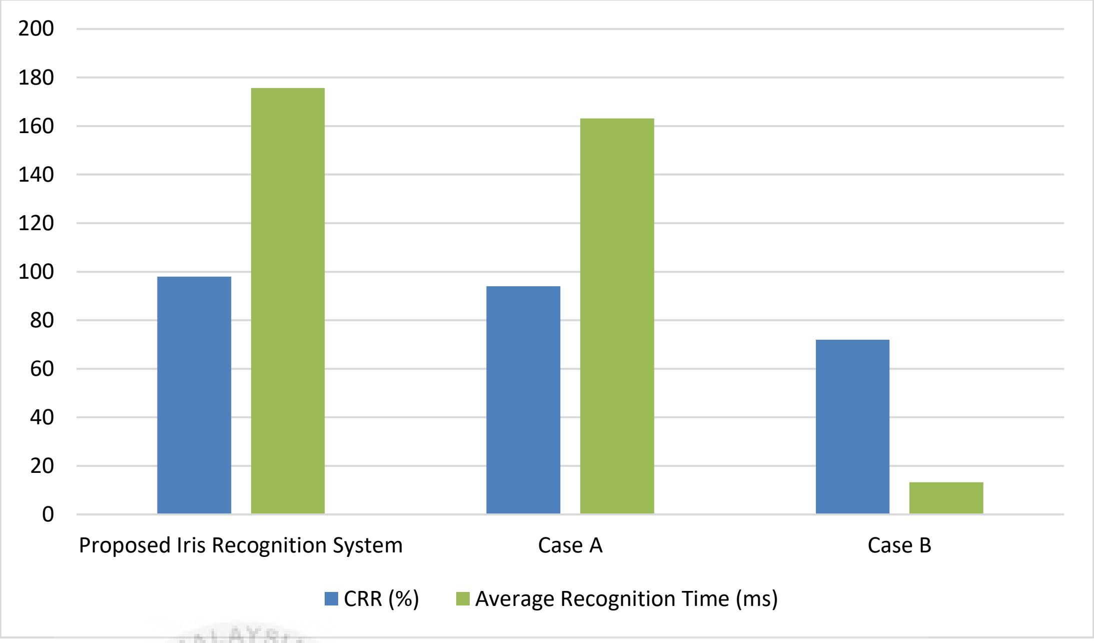


Figure 4.8: Bar Chart of CRR and Average Recognition System of Iris Recognition Systems

Table 4.1: Performance of Iris Recognition Systems

Case	Performance	
	Accuracy	Recognition Speed
Proposed iris recognition system	High	Fast
Case A (Refer Chapter 4.3.2)	High	Fast
Case B (Refer Chapter 4.3.3)	Low	Fast

The CRR and average recognition time of the proposed iris recognition system and the two cases is shown as shown in Figure 4.8 and the performance is tabulated in Table 4.1. From the bar chart, the proposed has the highest CRR value which is highest accuracy while Case B has the lowest accuracy. This proves that the iris recognition system with normalization and fractal feature extraction has better performance than the raw iris image.

Since the iris of the same person are full of uncertainty and variety in different conditions. With the aid of normalization, the iris that deformed in different conditions will standardize into a standard size. By comparing the accuracy of Case A and Case B, the accuracy is increased 22% when the normalization is included into iris recognition system. If comparing the accuracy of proposed iris recognition system and Case A, proposed iris recognition system has the higher accuracy than Case A. This proves that using the extracted feature from normalized iris image can improve the accuracy of recognition rather than using normalized iris image.

However, if compare the performance of iris recognition system in terms of recognition speed, Case B is the fastest. This is because the iris images do not go through edge detection, segmentation, normalization and fractal feature extraction process. Although the recognition speed of Case B is fast, but its accuracy is low. Case A is also having faster recognition speed than proposed iris recognition system because the normalized iris images in Case A do not undergo fractal feature extraction. But the difference of average recognition time between proposed iris recognition system and Case A is 0.0126s which means the fractal feature extraction process only takes 0.0126s to complete. Therefore, the proposed iris recognition system has better overall performance compared to Case A and Case B. Although its recognition time is longer than Case A and Case B, but the recognition speed is still considered as fast.

To conclude, the proposed iris recognition system has high accuracy and fast recognition speed. Thus, the second objective is achieved.

4.4 Comparison to Previous Works

By comparing the accuracy of proposed iris recognition system to the previous works as shown in Table 2.3, the accuracy of the proposed iris recognition system is 1% to 2% lower than some previous works [20, 21, 27, 29]. This is because these previous works use large number of iris image in the iris recognition system, while the proposed iris recognition system only use 100 iris images. The accuracy of the proposed iris recognition is better than [18, 22, 26, 25] may because of the feature extraction algorithm and classifier used in proposed iris recognition have better efficiency and give more accurate results.

By comparing the recognition time used to identify the person, the proposed iris recognition system with recognition time of 0.1704s has a significant difference compared to all previous works in Table 2.3. The recognition time of Umer *et al.* [21] and Al-Waisy *et al.* [29] are almost 1 second. This may because the iris images used in training and testing are more than 1300 iris images. While the number of iris images involved in proposed iris recognition system is 100 which is lesser than both works.

4.5 Limitation

With the high accuracy and fast recognition time, the proposed iris recognition system has a limitation on edge detection. Although the edge detector can detect the iris boundary in iris images accurately, some of the iris images are failed to do edge detection by both methods because of the problem listed in Chapter 4.1. When the edge detection is failed, the whole flow of iris recognition is stopped.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this project, the iris recognition system is built and worked successfully in identifying the identity of person through the iris. The edge detector algorithms, segmentation and normalization in the program is also provide high efficiency and accuracy in iris recognition. In the proposed iris recognition system, the lacunarity which is the fractal feature of the iris is extracted without any problem. Next, the deep learning classifier used in proposed iris recognition system, CNN, classifies through the extracted fractal features of iris image. A GUI is created to help in the monitoring of output image in each process. The automatic part of the iris recognition system GUI is used to test the 50 iris images. The results especially the recognition time of each test results are then recorded as shown in Appendix A.

By testing with 50 test images, the accuracy of proposed iris recognition system is 98% and the average recognition time is 0.1704s. By referring to Table 2.3, the proposed iris recognition system is concluded has high accuracy and fast recognition speed.

In Chapter 4.3, the proposed iris recognition system is proved to has a higher accuracy compared to the iris recognition system that recognize person through raw iris image and iris recognition system that trained and used the CNN by using normalized iris images without extracts the fractal feature. Although the recognition time of proposed iris recognition system is slightly slower, the difference of recognition time compared to others are not significant.

Lastly, in conclusion, the objectives of this project as mentioned in CHAPTER 1 are achieved. The iris recognition system is built and implemented with fractal feature extractor and identified the persons through deep learning neural network successfully. The performance of iris recognition system in terms of accuracy and recognition time is analysed and concluded that the iris recognition system has high and fast performance.

5.2 Recommendation

Since a big portion of recognition time is spent in edge detection compared to algorithms in other process, the edge detection of the iris recognition system is recommended to change to a better algorithm which is more accurate and faster. It is even better if the edge detection algorithm can solve the problem mentioned in Chapter 4.1.

Next, since the lacunarity is the only extracted fractal feature from normalized iris image, it is possible to reach a higher accuracy if more fractal feature is extracted. But the more fractal features are extracted, the longer the time to train the CNN. In addition, improving the architecture of CNN is also a possible way to gain 100% CRR.

Finally, I suggest that to add a security function. For example, when the person is recognized wrongly through the iris image, the function returns a failed message to the user to prevent the user access to another person’s ID.

REFERENCES

[1] Jain, A., Hong, L., & Pankanti, S. (2000). Biometric Identification. *Communications of the ACM*, *43*(2), 90-98.

[2] Blue, J., Condell, J., & Lunney, T. (2018). A Review of Identity, Identification and Authentication. *International Journal for Information Security Research (IJISR)*, *8*(2), 794-804.

[3] Jain, A. K., Bolle, R., & Pankanti, S. (Eds.). (2006). *Biometrics: personal identification in networked society*, *479*. Springer Science & Business Media.

[4] Miller, B. (1994). Vital signs of identity [biometrics]. *IEEE spectrum*, *31*(2), 22-30.

[5] Trader, J. (2012, July 30). *Black Hat Iris Biometrics Attacks Don't Tell The Whole Story*. Retrieved from https://www.m2sys.com/blog/biometric-hardware/black-hat-iris-biometrics-attacks-dont-tell-the-whole-story/.

[6] *ANATOMY OF THE EYE*. (n. d.). Retrieved from https://redroverventures.org/our-ventures/education/eye-health/anatomy-of-the-eye/.

[7] Muroo, A. (2000). The human iris structure and its usages. *Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium*, 87-95.

[8] Williams, G. O. (1996, October). Iris recognition technology. *1996 30th Annual International Carnahan Conference on Security Technology*, 46-59.

[9] Mehedi. (2018, January 28) *25 Advantages and Disadvantages of Iris Recognition*. Accessed from https://biometrictoday.com/25-advantages-disadvantages-iris-recognition/#:~:text=Compared%20to%20other%20modalities%2C%20it,few%20of%20its%20many%20uses.

[10] Majekodunmi, T. O., & Idachaba, F. E. (2011). A Review of the Fingerprint, Speaker Recognition, Face Recognition and Iris Recognition Based Biometric Identification Technologies. *Proceedings of the World Congress on Engineering, 2*.

[11] Wai, M. M. M., Aung, N. P., & Htay, L. L. (2019). Software Implementation of Iris Recognition System using MATLAB. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, *3*(5), 290-295.

[12] Rajesh, T., Karnan, M., & Sivakumar, R. (2014). Performance Analysis of Iris Recognition System - A Review. *International Journal of Computer Science & Information Technology*, 39-50.

[13] Chaudhuri, B. B., & Sarkar, N. (1995). Texture Segmentation Using Fractal Dimension. *IEEE Transactions on pattern analysis and machine intelligence*, *17*(1), 72-77.

[14] Mandelbrot, B. B., & Mandelbrot, B. B. (1982). *The fractal geometry of nature* (Vol. 1). New York: WH freeman.

[15] Smith Jr, T. G., Lange, G. D., & Marks, W. B. (1996). Fractal methods and results in cellular morphology—dimensions, lacunarity and multifractals. *Journal of neuroscience methods*, *69*(2), 123-136.

[16] Plotnick, R. E., Gardner, R. H., Hargrove, W. W., Prestegaard, K., & Perlmutter, M. (1996). Lacunarity analysis: a general technique for the analysis of spatial patterns. *Physical review E*, *53*(5), 5461-5468.

[17] McIntyre, N. E., & Wiens, J. A. (2000). A novel use of the lacunarity index to discern landscape function. *Landscape Ecology*, *15*(4), 313-321.

[18] Daouk, C. H., El-Esber, L. A., Kammoun, F. D., & Al Alaoui, M. A. (2002). Iris recognition. *IEEE ISSPIT*, *4*, 558-562.

[19] Masek, L. (2003). *Recognition of human iris patterns for biometric identification*. [Doctoral dissertation, Master's thesis, University of Western Australia].

[20] Hong-Ying, G., Yue-Ting, Z., & Yun-He, P. (2005). An iris recognition method based on multi-orientation features and Non-symmetrical SVM. *Journal of Zhejiang University-Science A*, *6*(5), 428-432.

[21] Umer, S., Dhara, B. C., & Chanda, B. (2016). Texture code matrix-based multi-instance iris recognition. *Pattern Analysis and Applications*, *19*(1), 283-295.

[22] Khotimah, C., & Juniati, D. (2018). Iris recognition using feature extraction of

box counting fractal dimension. *Journal of Physics: Conference Series*, *947*(1).

[23] Khanam, R., Haseen, Z., Rahman, N., & Singh, J. (2019). Performance analysis of iris recognition system. *Data and Communication Networks*, 159-171.

[24] Daugman, J. (2009). How iris recognition works. *The Essential Guide to Image Pprocessing*, 715-739.

[25] Roy, K., Bhattacharya, P., & Suen, C. Y. (2011). Iris recognition using shape-guided approach and game theory. *Pattern Analysis and Applications*, *14*(4), 329-348.

[26] Tobji, R., Di, W., & Ayoub, N. (2019). FMnet: iris segmentation and recognition by using fully and multi-scale CNN for biometric security. *Applied Sciences*, *9*(10), 2042.

[27] Kerim, A. A., & Mohammed, S. J. (2014). New iris feature extraction and pattern matching based on statistical measurement. *Int. J. Emerg. Trends Technol. Comput. Sci*, *3*(5), 226-231.

[28] Bharath, B. V., Vilas, A. S., Manikantan, K., & Ramachandran, S. (2014, December). Iris recognition using radon transform thresholding based feature extraction with Gradient-based Isolation as a pre-processing technique. *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, 1-8.

[29] Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S., & Nagem, T. A. (2018). A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications*, *21*(3), 783-802.

[30] CBSR (2005). *Center for Biometrics and Security Research*. Retrieved from http://www.cbsr.ia.ac.cn/china/Iris%20Databases%20CH.asp.

[31] The MathWorks, Inc, (2006). *Help Center: edge*. Retrieved from https://www.mathworks.com/help/images/ref/edge.html.

[32] Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, *15*(11), 1148-1161.

[33] Sarkar, N., & Chaudhuri, B. B. (1992). An Efficient Approach to Estimate Fractal Dimension of Textural Images. *Pattern Recognition*, *25*(9), 1035-1041.

[34] Sarkar, N., & Chaudhuri, B. B. (1994). An Efficient Differential Box-Counting Approach to Compute Fractal Dimension of Image. *IEEE Transactions on Systems Man and Cybernetics*, *24*(1), 115-120.

[35] Al-Kadi, O. S., & Watson, D. (2008). Texture Analysis of Aggressive and Nonaggressive Lung Tumor CE CT Images. *IEEE Transactions on Biomedical Engineering*, *55*(7), 1822-1830.

[36] Al-Kadi, O. S. (2020). Prediction of FDG-PET stage and uptake for non-small cell lung cancer on non-contrast enhanced CT scans via fractal analysis. *Clinical imaging*, *65*, 54-59.

[37] on2way (2014, November 9), *CSDN: "hongmo shibie (wu): hongmo fenge yu tuxiang guiyihua" 虹膜识别（五）：虹膜分割与图像归一化 [Iris recognition(5): Iris segmentation and normalization]*. Retrieved from https://blog.csdn.net/on2way/article/details/40948657.

[38] Al-Kadi, O. S. (2014, January 6). *Fractal Dimension*. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/44951-fractal-dimension. [Accessed 18 March 2021].

APPENDICES

Appendix A. Recognition Time of Test Image in Iris Recognition System

Table A.1: Recognition Time of Each Test Image simulated in Proposed Iris Recognition System in seconds(s)

Test Image # Person	1	2	3	4	5
A	0.1496	0.1318	0.1265	0.1307	0.1296
B	0.1673	0.1975	0.1587	0.2017	0.1570
C	0.1768	0.1822	0.1940	0.1948	0.1770
D	0.1490	0.1591	0.1621	0.1623	0.1482
E	0.1726	0.1673	0.1843	0.1572	0.1979
F	0.1716	0.1746	0.1672	0.1630	0.1663
G	0.1647	0.1538	0.1517	0.1654	0.1487
H	0.1749	0.1802	0.1811	0.1999	0.1779
I	0.1937	0.1945	0.1939	0.1933	0.1983
J	0.1739	0.1822	0.1718	0.1683	0.1752

Table A.2: Recognition Time of Each Normalized Iris Test Image simulated in Case A in seconds(s)

Test Image # Person	1	2	3	4	5
A	0.2154	0.1346	0.1210	0.1225	0.1208
B	0.1562	0.1864	0.1528	0.1958	0.1509
C	0.1893	0.1702	0.1883	0.1889	0.1730
D	0.1366	0.1522	0.1511	0.1457	0.1418
E	0.1592	0.1560	0.1752	0.1504	0.1819
F	0.1523	0.1592	0.1582	0.1755	0.1555
G	0.1493	0.1370	0.1470	0.1677	0.1595
H	0.1615	0.1686	0.1687	0.1932	0.1613
I	0.1878	0.1861	0.1787	0.1821	0.1847
J	0.1689	0.1646	0.1618	0.1550	0.1549

Table A.3: Recognition Time of Each Raw Iris Test Image simulated in Case B in seconds(s)

Test Image # Person	1	2	3	4	5
A	0.2160	0.0267	0.0214	0.0112	0.0109
B	0.0086	0.0087	0.0085	0.0085	0.0085
C	0.0083	0.0083	0.0085	0.0084	0.0084
D	0.0085	0.0083	0.0085	0.0084	0.0084
E	0.0084	0.0085	0.0085	0.0087	0.0082
F	0.0083	0.0083	0.0084	0.0083	0.0085
G	0.0082	0.0083	0.0083	0.0083	0.0083
H	0.0083	0.0084	0.0083	0.0083	0.0083
I	0.0083	0.0085	0.0084	0.0088	0.0083
J	0.0083	0.0083	0.0083	0.0083	0.0082

Appendix B. MATLAB Scripts

B.1 MATLAB Script 1

This MATLAB scripts includes image acquisition, edge detection, segmentation, normalization and save normalized iris train images.

```
cd 'Iris Image';
list_str = ls;
list_cell = cellstr(list_str);
[row_a,col_a] = size(list_cell);
cd '..';


for i = 3 : row_a          % Person
    % Image Acquisition
    Person   =   sprintf('C:\\Users\\Asus\\Desktop\\Iris   Recognition
              System\\Iris Image\\%s',list_cell{i});
    im_data{i} = imageDatastore(person);
    IM{i} = readall(im_data{i});
    for j = 1 : 5          % Iris Train Images
    % Segmentation
    [row,col] = size(IM{i}{j,1});           % Image size
    [XC,YC,RI,RO,centers] = EdgeDetection(IM{i}{j,1},col);     % call
    Edge Detection function
    IM_seg = Segmantation(XC,YC,RI,RO,IM{i}{j,1},row,col);     % call
    Segmentation function

    % Normalization
    [IMrect] = Normalization(RI,RO,XC,YC,IM_seg);
    % Write out normalized iris image
    save_file = sprintf('Iris Image\\%s',list_cell{i});
    cd (save_file);
    save_file_name = sprintf('Normalized %s0%d.jpg',list_cell{i},j);
    imwrite(IMrect,save_file_name);
    cd('..\\..');
    end
end


%% Edge Detection Function
function [XC,YC,RI,RO,centers] = EdgeDetection(IM,col)
% Iris Inner Boundary Detection
[centers,RI] = imfindcircles(IM,[30 150],'ObjectPolarity','dark',...
                'Sensitivity',0.9,'Method','twostage');
XC=round(centers(1));
YC=round(centers(2));

% Iris Outer Boundary Detection
IM2 = medfilt2(IM,[5 5],'symmetric');    % Blur the eyelash
edge_IA = edge(IM2,'canny',[0.1 0.3],7); % Canny Edge Detector

% x increment
XO = XC+RI+20;
while edge_IA(YC,XO)==0
    XO = XO+1;
    RO = XO-XC;
    if XO == col
        RO = 0;
        break
    end
end
```

```
% Intensity Profile Method as alternative method if x increment is
failed
if RO == 0
     [cx,cy,c] = improfile(IM,[0 (XC-RI-20)],[YC YC],100);
                  %intensity profile across iris outer boundary
    smooth_c = smoothdata(c);           %smooth noisy data
    [p,~,mu] = polyfit(cx,smooth_c,3);  %fit data by cubic polynomial
    p2 = polyder(polyder(p));           %obtain linear polynomial
    XO = roots(p2)*mu(2)+mu(1);         %destandardize root
    RO = abs(XC-XO);                    %radius of outer boundary
end

%% Segmentation
function IM = Segmantation(XC,YC,RI,RO,IM,row,col)
% if (dis > outer boundary) or (dis < inner boundary) => white(255)
for i = 1 : col          %x
    for j = 1 : row      %y
        dis = sqrt((i-XC).^2+(j-YC).^2);
        if dis > RO || dis < RI
            IM(j,i) = 255;
        end
    end
end

%% Normalization [37]
function IMrect = Normalization(RI,RO,XC,YC,IM)
% size of rectangular block
M = 2^6;            %width = RO-RI ~= 2^6 = 64
N = 2^7;            %length = pi*(RO+RI)/2 ~= 2^7 =128
alpha = 2*pi/N;     %divide 2pi into 128 samples
```

```
% determine coordinates of inner boundary(xi,yi) and outer
boundary(xo,yo)
for k = 1 : N
    xi(1,k) = XC + RI*cos(k*alpha);
    yi(1,k) = YC - RI*sin(k*alpha);
    xo(1,k) = XC + RO*cos(k*alpha);
    yo(1,k) = YC - RO*sin(k*alpha);
end

% obtain polar coordinates (x_rect,y_rect)
for r = 1 : M
    x_rect(r,:) = round(xi(1,:) + (xo(1,:)-xi(1,:))*r/M);
    y_rect(r,:) = round(yi(1,:) + (yo(1,:)-yi(1,:))*r/M);
end

% rebuild image by using polar coordinates (x_rect,y_rect)
% ROI => upper half of rectangular (M/2)
for i = 1 : M/2
    for j = 1 : N
        IMrect(i,j) = IM(y_rect(i,j),x_rect(i,j));
    end
end
```

B.2 MATLAB Script 2

MATLAB script 2 consists of function of fractal feature extraction, script of training of CNN and script of testing the iris recognition system with test images.

```
%% Feature Extraction [38]
function L = FeatureExtraction(IM)
[M,N] = size(IM);       % size of normalized iris image


% Differential Box Counting
for r = 2:7    %scaling factor
    rc = @(x) floor((max(x)-min(x))/r)+1;  % function of kernel
    F = colfilt(IM,[r r],'sliding',rc);     % sliding window
    Nd{r} = double(F*49/(r^2));      % number of boxes required to
                                     cover the intensity
    logNd{r} = log10(Nd{r});         % logarithm of Nd
end


% Fractal Dimension
% Obtain slope by least squares linear regression
x = log10(2:7);                   % x=log(r)
SSxx = dot(x,x) - (sum(x)^2)/6;   % sum of squares of xx
for i = 1 : M
    for j = 1 : N
        y = [logNd{7}(i,j),logNd{6}(i,j),...
             logNd{5}(i,j),logNd{4}(i,j), ...
             logNd{3}(i,j),logNd{2}(i,j)];      %y = log(Nd{2:7})
        %SSxy = sum of squares of xy
        SSxy = dot(x,y)-(sum(x)*sum(y))/6;
        FD(i,j) = SSxy/SSxx;    %slope = SSxy/SSxx
    end
end
```

```
% Fractal Feature
r = 5;       % sliding window size (r x r)
FDsum = colfilt(FD,[r r],'sliding',@sum);       % Sum of IM in r x r
                                                   window
FDsq = FD.^2;
FDsumsq = colfilt(FDsq,[r r],'sliding',@sum);  % Sum of square of IM
                                                   in r x r window
L = (FDsumsq./r^2)./((FDsum./r^2).^2)-1;       % Lacunarity


%% Convolutional Neural Network
cd 'Iris Image';
list_str = ls;
list_cell = cellstr(list_str);
[row_a,~] = size(list_cell);
cd '..';

count = 1;
for j = 1 : 5        % iris images
    for i = 3 : row_a       % Person
        % Read normalized iris image from saved file
        save_file = sprintf('Iris Image\\%s',list_cell{i});
        cd(save_file);
        f_name = sprintf('Normalized %s0%d.jpg',list_cell{i},j);
        IM = imread(f_name);
        cd ('..\\..')



        % Feature Extraction
        Lac = FeatureExtraction(IM);   %Call feature extraction func
        FFL(:,:,:,count) = Lac;        %4D inputs for input layer
        target(count,1) = list_cell{i};  %target corresponding to i/p
        count=count+1;
    end
end
target = categorical(cellstr(target)); %convert array to categorical

% Architecture of CNN
layers = [ ...
    imageInputLayer([32 128 1],'Name','Input')   % 32*128

    convolution2dLayer(3,10,'Name','Conv1')      % 30*126
    reluLayer
    maxPooling2dLayer(2,'Stride',2)              % 15*63

    convolution2dLayer(4,20,'Name','Conv2')      % 12*60
    reluLayer
    maxPooling2dLayer(2,'Stride',2)              % 6*30

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer
    ];
options = trainingOptions('adam','MaxEpochs',500);

net = trainNetwork(FFL,target',layers,options);     % Train CNN
CNN = net;
save CNN          % save CNN
```

```
%% Test Overall Iris Recognition System with Test Images
cd 'Iris Image';
list_str = ls;
list_cell = cellstr(list_str);
[row_a,~] = size(list_cell);
cd '..';
test_im_num = ['06';'07';'08';'09';'10'];

count = 1;
for i = 3 : row_a      % Person
    for j = 1 : 5      % iris images
        % Classify Test Image One by One
        tStart = tic;  % start timer
        cd('Test Image');
        save_file_name = sprintf('%s%c%c.jpg',list_cell{i},...
                          test_im_num(j,:));
        im = imread(save_file_name);                 %Image acquisition
        cd ('..')
        [line,row] = size(im);
        [XC,YC,RI,RO,~] = EdgeDetection(im,row);         %Edge Detection
        seg = Segmantation(XC,YC,RI,RO,im,line,row);    %Segmentation
        rect = Normalization(RI,RO,XC,YC,seg);          %Normalization
        fflac = FeatureExtraction(rect); %Fractal Feature Extraction
        output = classify(net,fflac)';   %Classification by CNN
        tEnd = toc(tStart);              %Stop timer

        % save fraction feature as 4D inputs and target class for the
        plot of confusion matrix
        FractalFeatureLac(:,:,:,count) = fflac;
        target_test(count,1) = list_cell{i};
        count=count+1;
        % save the recognition time of each test image in array form
        time_all(i-2,j) = tEnd;
     end
 end
 target_test = categorical(cellstr(target_test));

 XTest = FractalFeatureLac;        % FF of 50 test images
 YTest = target_test;              % Target of 50 test images
 YPred = classify(net,XTest);      % Classification Results
 plotconfusion(YTest,YPred);       % Plot confusion table
 time_mean = mean(time_all,'all'); % average recognition time
```