ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

SITI NUR ADHA BINTI AZIZAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

SESI PENGAJIAN: 2015/2016

Saya SITI NUR ADHA BINTI AZIZAN

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

✓	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
	TIDAK TERHAD	

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap: No. 11, Kampung Teloi Tua, 08200 Sik, Kedah Darul Aman

(Dr Zaheera Binti Zainal Abidin)

Tarikh : 25/08/2016

Tarikh : 25/08/2016

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM) ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa

ANALYSIS OF BEES ALGORITHM IN CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

SITI NUR ADHA BINTI AZIZAN

This report is submitted partial fulfillment of the requirement for the

Bachelor of Computer Science (Computer Security)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

TECHNICAL UNIVERSITY OF MALAYSIA MELAKA

2016

DECLARATION

I hereby declare that this project report entitled

ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ____________________ Date: 25/08/2016
(SITI NUR ADHA BINTI AZIZAN)

I hereby declare that I have read this project report and found this project is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Security) With Honours.

SUPERVISOR : ____________________ Date: 25/08/16
(DR. ZAHEERA BT ZAINAL ABIDIN)

DEDICATION

To Allah S.W.T, my beloved family, my respectful supervisor and last but not least my supportive friends.

ACKNOWLEDGEMENTS

First and foremost, I concede the surviving presence and the flourishing refinement of ALMIGHTY GOD for His concealed hand yet substantial supervision all through the session.

I am extremely indebted to my respected research team of FTMK and UTeM for rendering all the facilities for the successful completion of my project and give me an opportunity to gain more experiences in this site.

I express my heartiest thanks to my respected supervisor, my coordinator and my lecture guide, Dr. Zaheera Bt Zainal Abidin for the facilities and valuable guidance she provided in molding this project to a successful one.

I would also like to express my thanks to all other faculty staffs and members at Information and Communication Technology, UTeM for the valuable help provided by them.

Above all I would like to express my sincere gratitude and thanks to my parents and family for the physically and mentally support throughout this session. Not forgotten to all my friends for their valuable comments and suggestion for making this work a success.

ABSTRACT

This preliminary document covers the presentation phase of a final year BSc Computer Science Project major in Computer Security (Hons) aiming to investigate biometric technologies and modified one of several segmentation technique in Iris Recognition System which is The Bees Algorithm.

This project will focusing on enhanced bees algorithm technique on circular segmentation method to get high distortion template of iris images in iris recognition system. The introduction offers some background to the biometric and iris recognition system regarding to the project and establish the aims and objectives of the project overall. Following on from the introduction, the literature review presents a critique of research that provides the basic for this project. This material includes a number of texts, journals and research papers as well as additional information sourced from the Web. As drawn from the literature, the subject areas covered includes, history and background of biometrics, iris recognition system and approach, method and technique of segmentation, technological, social, environmental influences, including security, performance and testing.

ABSTRAK

Dokumen awal ini meliputi fasa satu pembentangan projek tahun akhir bagi Sarjana Muda Sains Komputer dalam Keselamatan Komputer (Kepujian) yang bertujuan untuk menganalisa teknologi biometrik dan melakukan beberapa perubahan dalam salah satu daripada beberapa teknik segmentasi dalam Sistem Pengiktirafan Iris “The Bees Algorithm”.

Projek ini akan memberi tumpuan kepada memantapkan teknik “Bees Algorithm” dalam kaedah segmentasi bulat untuk mendapatkan data penyelewengan yang tinggi bagi imej iris dalam sistem pengiktirafan iris. Pengenalan untuk projek ini menceritakan tentang sejarah sistem biometrik dan pengiktirafan iris mengenai projek dan matlamat serta objektif projek secara keseluruhan. Seterusnya, dalam bab yang kedua iaitu kajian literature, membentangkan kritikan terhadap penyelidikan yang telah dilakukan beberapa tahun yang lalu yang membantu menjadi asas untuk projek ini. Bahan ini termasuk beberapa teks, jurnal dan kertas penyelidikan serta maklumat tambahan yang diperoleh dari laman sesawang. Antara bahan yang dapat digunakan adalah kaedah dan teknik segmentasi, teknologi, sosial, pengaruh alam sekitar, termasuk keselamatan , prestasi dan ujian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Biometrics is a science that deals with the identification a person based on behavioral and physiological characteristics. For behavioral characteristics, the attitudes of a person like a gait, voice, and signature. While for physiological characteristics, are those which a person physically owns like fingerprints, iris patterns, face, ear shape, hand geometry, and retina-pattern. All the requirements for a characteristic to be biometrics are uniqueness, permanence, universality, and collectability. There are many applications that implement these biometrics systems such as credit card, computer systems security and access to any building in a secure way.

Biometrics also refers to the automatic identification of a person based on physiological or behavioral characteristic. This biometrics system and method of identification help and give several advantages over traditional methods involving ID cards, PIN numbers or password for various reasons. The advantages are when a person needs to be identified; a person just need and required being physically present at the point of identification and the identification technique are based on biometric system that obviates the need to remember a password or carry a token for identification.

In this era, with the internet into our everyday lives, it is needed to keep our sensitive and personal data protected and secured. To help in preventing an unauthorized access to ATMs, cellular phones, laptops, and computer networks, we can replace password PINs and start using biometrics PINs. Unlike biometric traits, PINs and passwords might be forgotten, and tokens like passports and driver's licenses may be forged, stolen or lost by accidently.

In this period, with the web into our ordinary lives, it is expected to keep our touchy and individual information ensured and secured. To help in keeping an unapproved access to ATMs, mobile phones, tablets, and PC systems, we can supplant secret word PINs and begin utilizing biometrics PINs. Not at all like biometric attributes, PINs and passwords may be overlooked, and tokens like travel permits and driver's licenses might be produced, stolen or lost by accidently.

Other biometric approaches are being developed and industrialized such as those based on gait (technique of walking), hand veins, ear recognition, facial thermo gram, DNA, odor and palm prints. Furthermore, nowadays in human identification, iris recognition has been widely used instead of fingerprint and face modalities due to its universality, uniqueness, acceptability and performance.

Iris recognition is an autonomous system that uses complex mathematical pattern recognition, image processing and machine learning techniques for measuring the human iris (Daugman, 1994). Inside the human iris, there are many unique features such as crypts, radial furrows, concentric furrows, collarets, freckles, pupils and pigment blotches which distinguish the genuine characteristics of a person, thus making it suitable for recognition purposes. However, the demand for high accuracy and high speed recognition in biometric system leads to continuous proposals of new iris recognition methods (Daugman,2007) since the arising of impostor users in the biometric system.

Impostor users can be categorized into two types. The first type (Type I) of impostor is when the biometric system rejects the genuine user who wants to access the system. Meanwhile, the second type (Type II) of impostor is someone who tries to penetrate the biometric systems and pretends to be the original person. Moreover, the rising number of impostor gives high error rates. High error rate in iris texture means the iris contains more noise, which is called noisy iris. There are two reasons for noisy iris to occur, which are the iris texture and the technical factor.

The first source of noisy iris comes from the iris texture itself that caused the constant change in its structure which is due to two conditions that are the human biological condition and the occlusion from eyelids and eyelashes (Rankin et al., 2013). Every human has different body condition and health. The iris texture constantly changes in slow motion based on human condition, such as aging, growth, emotion, diet, and eye surgery. In fact, the color of the iris texture may change due to inheritance and epigenetic diversity from different races. The ever-changing iris texture creates difficulties at the comparison phase to determine either the captured iris data are genuine or not. Previous studies show that lack of success was traced in 21% of intra-class comparisons cases, taken at interim of three and interim of six months (Fenker, Bowyer, Dame, & In, 2012). On the other hand, the occlusion from eyelids, eyelashes, and the presence of contact lenses or specular also contribute to the arising noise in the iris features since the crucial information is affected when the eyelashes cover some of the important information of the iris features.

The second source of noisy iris appears from technical factor in capturing the iris feature image. Some of the sources of noises are from the camera itself and the surrounding environment during the capturing process, such as indoor or outdoor, a distance between the camera and the eyes or source of light which are near infrared or LED and more. Other than that, the blurred and low-quality images of iris texture which is captured using 'off-angle' configuration, also add noise to the iris image (Schuckers et al., 2007).

Therefore, many researchers have done studies to overcome the noisy iris problem. There are numerous studies reported to address this issue, such as to remove the unwanted noise (Yu Chen et al., 2010), to enhance the noise level (R.P. Wildes et al., 1994), mask (Hollingsworth, Bowyer, & Flynn, 2011), segmentation process and feature selection (Garg & Parashar, 2012) however, the outcome still produce a noisy iris result. Among all proposed solution, in order to solve this problem is to ensure that only a certain unique part of the iris structure is used which remains unchanged for iris recognition. The unique part of the iris texture (Daugman, 1994) consists of crypts, furrows, collarets, pupil, freckles and blotches. Subsequently, some studies found that the micro characteristics in iris features have been mostly stable for recognition (Militello, Conti, Vitabile, & Sorbello, 2009; Conti, Milici, Sorbello, & Informatica, 2007). Nonetheless, the unique iris feature sustained only for a certain period of time (Flom & Safir, 1987; Baker, Bowyer, Flynn, & Phillips, 2012) and some other researchers reported that only to six years (Matey, Tabassi, Quinn, & Chumakov, 2013).

Studies on selecting the iris texture from its original structure have been gaining attention from some researchers (Yoon, 2005; Proença et al., 2006; Chen, Chou, Shih, Chen, & Chen, 2007; Sun & Tan, 2009; Han, Li, Qi, & Sheng, 2010; Tallapragada, 2012; Sharma, Mishra, & Dubey, 2013). Hence, feature selection is vital for choosing a subset features of available unique features by eliminating unnecessary features since the information of iris features obtained can be tremendously huge and consequently consumes a lot of computational resources (Garg & Parashar, 2012). In fact, in feature selection is observed that the existing method is lacking of the natural computational element in the iris recognition. Therefore, the unique iris feature selected is based on the best features points from the entire iris texture which is required in learning the changes or instability in iris texture intelligently.

The genuine user who has been claimed as an impostor (Type I) could be accurately recognized by the biometric system. In fact, the error rates can be reduced in solving the problem of noisy iris. Moreover, the crucial iris features from the unique iris and renews the information at the classifier which means that re-enrolment process can be reduced if there is an information change in iris texture.

1.2 Problem Statement (PS)

Iris Recognition is the most solid biometric framework since iris examples are special to every person and don't change with time. However, the current system in iris recognition has several limitations such as the noisy and robustness that may affect the accuracy of the result because of some criteria like human biological condition and the occlusion from the eyelids and eyelashes.

Nevertheless, in this project, the scope of this study is focusing on modified in iris template images with less noisy, robustness and intrusive performance. To obtain this all, Bees Algorithm has been used to fulfill the demand for higher accuracy.

Table 1.1: Summary of Problem Statements

PS	Problem Statement
PS1	Lack of natural computing method in detection of iris feature using circular segmentation.
PS2	The existence circular segmentation method unable to detect iris template.
PS3	The existence system of iris recognition still not fully in the network environment.

1.3 Project Question (PQ)

Table 1.2: Summary of Project Questions

PS	PQ	Project Question
PS1	PQ1	How to obtain iris template in high noise images with low frequency?
PS2	PQ2	How to detect iris template in circular segmentation method?
PS3	PQ3	How to create an iris recognition system in the network environment?

1.4 Project Objective (PO)

Based on the research questions in Section 1.3, the research objectives was formulated and was identified, which were described, as follows:

Table 1.3: Summary of Project Objectives

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To study Bees Algorithm on Circular Segmentation in Iris Biometrics.
PS2	PQ2	PO2	To modified the purpose of Bees Algorithm in Iris Biometrics.
PS3	PQ3	PO3	To implement the circular segmentation process using Bees Algorithm with Raspberry Pi in the network environment.

1.5 Project Scope

The scope for this studies covered the second generation of iris recognition in the biometric system. Iris is among the physiological traits that are understudied, after fingerprint and face. In general, iris features comprise of the pupil and its texture. The pupil dilation is due to the diameter and radius of the almost circle pupil shape. Meanwhile, the iris texture change due to the contraction of the iris muscle, which is caused by the amount of the light entering the eye. Iris textures are based on the authentic features, for examples furrow, crypts, ring, freckles and pigment blotches.

The change in pupil size and iris texture require a self-learning approach for feature selection using an optimization such as Ant Colony Optimization Algorithm in the extraction and the selection processes for iris recognition. This new approach demands a semi-supervised learning which requires the element of data mining area, in learning and training the data set using supervised learning classifier and testing with the real data set, or with the same classifier, or with unsupervised learning approach.

The outcome of this classification process produces a new approach of unique iris features pattern. In order to acquire large iris images, the public databases of the eye images were downloaded. Those databases are from CASIA, MMU, IITD, UBIRIS and CHUK. All of these databases are uses for further investigation in iris recognition system. The result of eye images then been stored in the format of (.bmp), (.png) and (.jpg).

1.6 Project Contribution (PC)

Table 1.4: Summary of Project Contributions

PS	PQ	PO	PC	**Project Contribution**
PS1	PQ1	PO1	PC1	To study the iris feature segmentation method in Bees Algorithm technique for Iris Biometrics.
PS2	PQ2	PO2	PC2	To analyze the performance of Bees Algorithm technique on circular segmentation in Iris Biometrics.
PS3	PQ3	PO3	PC3	The purpose of a new prototype of Iris Biometrics in the network environment.

1.7 Thesis Organization

Chapter 1: Introduction

The first chapter in this project is about to discuss the background. It defines the problem statement, project question, project objective, project scopes and project contribution.

Chapter 2: Literature Review

This chapter discusses about the previous works that are related with the project, the critical review and justification, also the purpose solution for the further project.

Chapter 3: Methodology

Chapter 3 is discuss about the methodology of the project which is explained the selected methodology for each stage and the project milestone.

Chapter 4: Analysis and Design

Chapter 4 is discusses about project analysis, requirement analysis, and prototype detailed design. This chapter is good for prototype development.

Chapter 5: Implementation

This chapter discuss about the prototype development environment setup, configuration management and implementation status.

Chapter 6: Testing

This chapter discuss about the test plan, test strategy, test design and test result and analysis.

Chapter 7: Project Conclusion

Chapter 7 is the last chapter that discuss about the project summarization, project contribution, project limitation and the future works.

1.8 Conclusion

This chapter explained the introduction about the project that be done later on. It is use as guide in order to start the project by defining the objectives, problem statement, problem contribution, scope and others. The iris recognition faces challenges in attaining the demand for accuracy, robustness and speed. The research on iris is at the cutting edge and iris recognition technique is more robust, high in accuracy and speed. While, the noisy in iris recognition demands a new intelligent and self-learning approach for iris recognition system, but the current approach that been discussed and tested still lacking on the feature of selection criteria and intelligence elements.

The next chapter explained about the literature review and methodology. The literature review presents a critique of research that provides the basic for this project.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Iris Recognition is a strategy for recognizing an individual by breaking down arbitrary example of iris. Iris Recognition is the most dependable system of individual check because of restrictive surfaces, non-intrusiveness, and steadiness all through the human life time, open endorsement, and accessibility of easy to understand catching gadgets. The part of an iris recognition framework is to remove, speak to and coordinate the textural many-sided quality present on the clear of the iris. Such a technique involves iris segmentation, feature extraction (encoding) and feature matching.

The history of iris recognition dates back to the nineteenth century. In 1882, Alphonse Bertillon started to conduct body measurement on police record cards for arrestee identification purposes. He was also the first to purpose properties of the human eye (color) for biometric recognition in 1886. The idea of using iris patterns for identification was born in 1936 with Frank Burch.

Among all biometric features, the pattern of an iris texture is assumed to be the most unique among different individuals. Contrasted with other biometric attributes, for example, fingerprints, the iris is a protected interior organ whose arbitrary surface is unpredictable, restrictive, and accepted to be extremely settled all through life. However, there are still ongoing researches on finding the best way to search unique iris features since iris image contains high noise.

In this chapter 2 is to discover the idea of iris recognition system, the related works and the previous works in iris recognition. One of the prime problems that exist in this system is noisy iris. The noisy iris is described as the condition where iris image contains high noise rates, which can be measured in false rejected rate (FRR). The high noise iris images, usually give the biometric systems to deliver erroneous results, leading to categorizations where the actual user is labeled as an impostor.

Therefore, this study focuses on a novel method, targeted at overcoming the aforementioned challenge and we use of bee algorithm based image retrieval (BA) technique as a successful method in recognizing in noisy iris. In section 2.3, we discussed the critical review of current problem of related study and justification has to be made in highlight the important research gap. Finally the proposed solution for project has to be made to achieve the aim of the project target.

2.2 Background Study

Iris recognition concept was initially proposed by an ophthalmologist, Dr. Frank Earl Burch in 1936. At that point, the beginnings of cutting edge mechanized iris identification were established in 1986, when Dr. Leonard Flom and Dr. Aran Safir walked in a patent for the principal iris acknowledgment framework, however with no calculation. Less than 10 years after the starter of the thought by Flom and Safir, John Daugman protected the primary contemporary robotized iris acknowledgment framework in 1994. It depended on doubly dimensionless directions for standardization, 2D Gabor channels as components and Hamming separations (HD) scores as comparator. The well-known iris recognition algorithms were suggested by (Daugman, 1994) and (Wildes,1997) and that has been triumph used for commercial applications, also used until today. For iris recognition system in biometrics, the Daugman's approach also has been used as a model of reference. The logical of using it is that it has been tested in the real environment, and involves enormous databases and also procedures worldwide. (Koh et al. 2010)

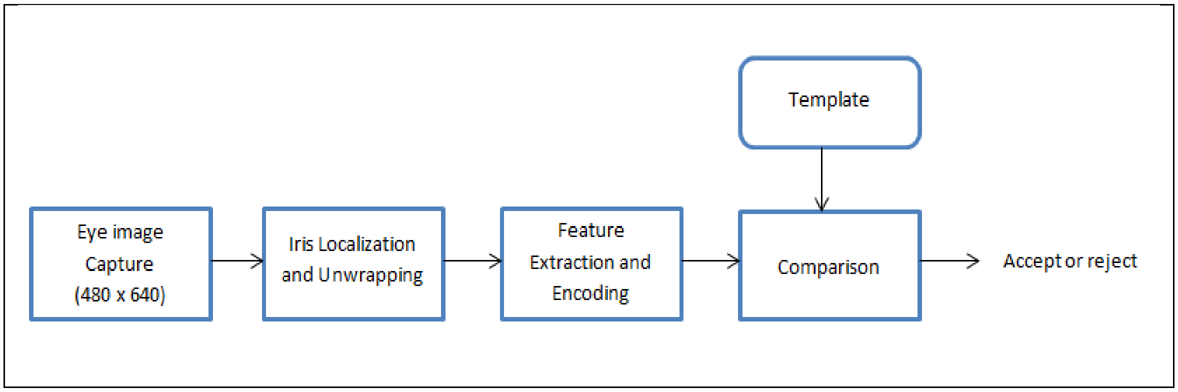


Figure 2.1: Daugman's Approach of Iris Recognition

In past years, the regularly expanding demand on biometric frameworks requested nonstop proposition of novel iris acknowledgment systems. Still, the handling chain of old-style iris acknowledgment frameworks has waited practically unaltered. In Daugman's approach, there are four leading phase in the existing iris recognition system. (Mahlouji & Noruzi 2012)

1. Iris image acquisition (Acquisition phase)
2. Image pre-processing (Segmentation and normalization phase)
3. Feature extraction (Extraction phase)
4. Comparison (Matching phase)

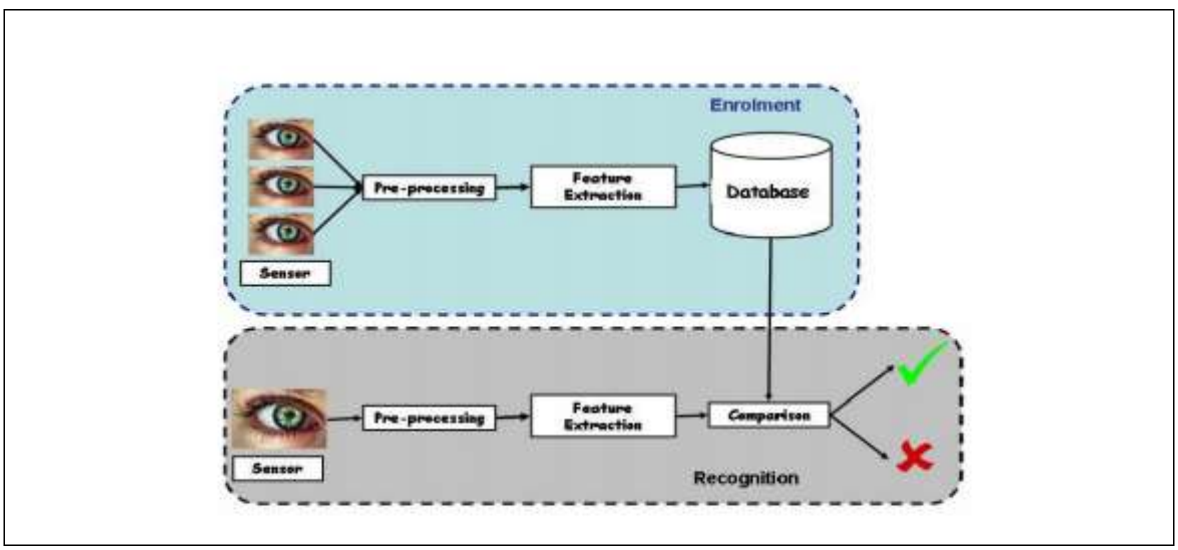


Figure 2.2: Biometric System Phase

2.2.1 Iris Image Acquisition and Iris Database (Acquisition phase)

Image acquisition shows a critical part in iris recognition. It involve hardware and software specification which the blurriness and off-angle image captured from camera or iris sensor and the environment contribute to either good or low quality like noisy iris image. In this stage, an image photo is taken form iris. Images that acquired by the camera sensor then stored into iris database.

According to the Mitre report, there are four different camera types for iris acquisition. Firstly, Access control which is typically wall-mounted cameras capturing both eyes simultaneously with a moderate focal volume and requiring active user cooperation. Secondly is Handheld, it is typically portable cameras with small focal volumes for single-eye capture and requiring attended capture. Third are Dual-eye visor camera and when the camera shielding both eyes from direct sunlight, requiring active cooperation and attended capture of both eyes simultaneously. Lastly, Stand-off portal camera, which is next-generation portal devices with large focal volume, full portrait capture and requiring less active user cooperation but still need user to look towards the camera.

After capturing iris images, the images need to be stored in one place, which called database. Database is a place provides for store and sharing images with other researcher and it is available to public for research and educational purpose. Databases that are mostly use in biometric are CASIA and UBIRIS. These database can be downloaded from the CASIA (Chinese Academy of Sciences' Institute of Automation) from China, and UBIRIS (Noisy Visible Wavelength Iris Image Database) delivered by SOCIA Lab – Soft Computing and Image Analysis Group, Department of Computer Science, the University of Beira Interior, Portugal. These both databases is trusted to be able to replicate the real environment in different condition and act as test for further testing, evaluation and validation for purpose approach in extraction and matching phase. (Koh et al. 2010)

2.2.2 Image pre-processing (Segmentation and normalization phase)

Iris image preprocessing including edge detection, contrast adjustment and multiplier, also contains segmentation and normalization steps and is a pivotal step for successful recognition. The objective of iris segmentation is too precisely and powerfully the iris in images of the human eye regardless of the presence of noise, such as eyelids, eyelashes and reflections for various individuals, sensors, and images properties. This process is to distinguishing pupillary and limbic boundaries, meanwhile normalization uses the found pupillary and limbic boundaries of the iris to map the annular texture in the rectilinear iris image into a doubly dimensionless polar format, originally proposed by Daugman’s.

There are few different strategies for iris segmentation. But, most approaches are only slight refinements of two approaches are; first, Integro-Differential Operator and second is Hough Transform (HT).

In iris segmentation, there are consists of circular segmentation and non-circular segmentation method. This is explained in details in further chapters.

2.2.2.1 Circular Segmentation Method

In the iris segmentation, the pupil and iris boundaries are localized and the process of contouring the iris starts from inner to outer part of the iris. There are a couple of types of segmentation which is the entire iris texture is segmented, whereas several studies show that partial segmentation is performed.

(John Daugman, 2007) has segmented the entire iris texture using the circular for,at of the iris texture regardless of the existence of noise, for example eyelids, eyelashes and occlusions, meanwhile the iris texture represents the person's vital information which was proposed by Paul Hough in 1962 (Gonzalez & Woods, 2002). On the other hand, (R.P Wildes et al, 1994) has segmented the whole iris texture by applying the mask and white noise to reduce the noise. In fact, noise reduction at segmentation lead to higher accuracy performance.

Two establish algorithm have been widely used in commercial iris biometric applications, which are Integro-Differential Operator and Hough Transform. Experiments using both algorithms have been tested by Zainal Abidin et al, (2012). The experiments showed that the Hough Transform (HT) performed better compared to Integro-Differential Operator (IDO) in terms of noise tolerance in noisy iris. No matter how, the existing techniques are incapable to cater the segment iris when it is off-angle and occluded. (Abidin 2012)

Figure 2.3: Circular Segmentation (Zainal Abidin et al, 2012)

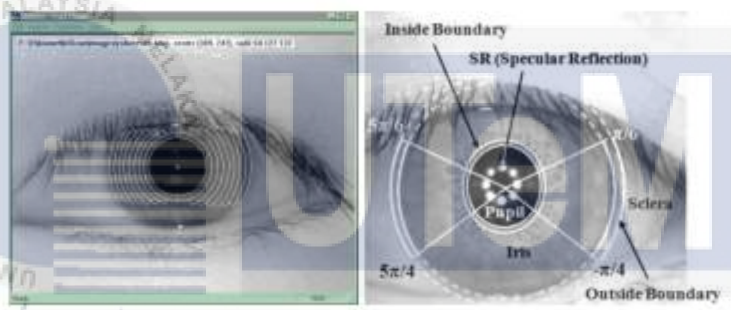
2.2.2.2 Non-Circular Segmentation Method

Studies found that the human eye is not a round shape. As claimed by De Marsico, Nappi and Riccio (2012), the segmentation process should use the non-circular format that follows the shape of the original iris. This is because, with this process, it produces higher accuracy in iris recognition, particularly when the iris was in high quality. Figure 2.4 (a) and (b) show the sample of non-circular format.

(a) (b)

Figure 2.4: (a) Non-Circular Segmentation (Yu Chen et al, 2010) and (b) (He, Tan, Sun, & Qiu, 2009)

The partial segmentation of iris recognition using inner boundary of the pupil's circle region has increased, and binary integrated edge intensity curve conquests the complications of eyelids occlusions (Patel, Modi, Paunwala & Patnaik, 2011) which important when noisy iris image is off angle and in a blurred condition. The off-angle and blurred iris textures are due to non-cooperative imaging environment. For non-cooperative environment, pulling and pushing method is applied in pupil localization. Figure 2.5 (a) and (b) show the sample.



(a) (b)

Figure 2.5: Partial Segmentation (a) http://www.engr.sjsu.edu/wbarret/ce195/sample.jpg and (b) (Kang, Byung Jun, Park, Yoo & Kiyoung, 2010)

2.2.3 Feature Extraction (Extraction Phase)

The motivation behind the element extraction module is to create a minimized representation of the biometric attributes so as to facilitate the coordinating procedure. In this stage, it's including clamor end from iris image and creating iris code. Numerous others highlight extraction strategies have been proposed for iris acknowledgment, for example, highlight extraction techniques recognized by the kind of produced highlight vectors; double and genuine esteemed procedures.

Compared to real-valued techniques, binary feature vectors follow Daugman's iris-code approach [1] supporting the matching process by quantizing intermediate features to get a more stable and compact template that can be effective compared. Despite the variety of available techniques, all main commercial systems apply Daugman's iris-code algorithm based on 2D Gabor wavelets. The 2D Gabor function is a product of an axis-stretched Gaussian and a complex plane wave, generalized by Daugman [2, 3] as follows:

$$G(x,y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]}e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]}$$

where (x_o, y_o) specify position in the image, (α, β) specify the effective width and length, and (u_o, v_o) specify modulation, which has spatial frequency $\omega_0 = \sqrt{u_0^2 + v_0^2}$. The odd symmetric and even symmetric 2D Gabor filters are shown in Figure 4.1.

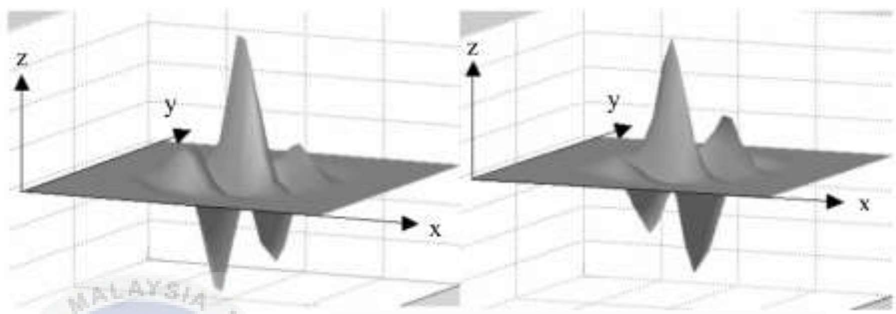


Figure 2.6: A quadrature pair of 2D Gabor filters (left) real component or even symmetric filter characterized by a cosine modulated by a Gaussian (right) imaginary component or odd symmetric filter characterized by a sine modulated by a Gaussian.

2.2.4 Comparison (Matching Phase)

Recognition of biometric features involves evaluation of feature vectors, which is two feature vectors are matched by executing a biometric comparator. This stage also involves re-enrolment process in acquiring the real-time iris texture template and compared with the stored iris texture template in the database.

Identification and verification are two modes of comparison. These two are involve in alignment and matching processes. Techniques used in matching include of Hamming Distance (HM), normalized correlation, Weighted Euclidean Distance, and Classifier (SVM, RVM and Adaboost).

Once the iris images are enrolled to a biometric system and matching with the reference data (templates) that are stored in database, biometric authentication can be performed in two different operation techniques:

1. *Verification (Comparison for One-to-One)*

    In confirmation mode, the subject is required to raise a character claim, for a case the subject cases to be a particular individual who has already enrolled with the framework. This should be possible by utilizing IDs, cards, or outer information or token or based one of kind identifiers.

2. *Identification (Comparison for One-to-Many)*

In identification mode, the subject is not required to raise a personality claim, case the extricated biometric template is contrasted with each and every format in the reference database. Once a rank 1 individual (the format which displays the most astounding level of comparability to the given one) is found, acknowledgment or dismissal is yielded in view of adequate threshold.

Biometric recognition is essentially a pattern recognition difficulty that require biometric signal, which is used to classify authentication attempts into the class of genuine or not genuine. On the authority of Daugman (2004), at least 50% of the iris must be visible, or an HD of 0.3 with a minimum radius of 70 pixels should be captured in determining the iris pattern genuine.

A few metrics present when measuring the performance of biometrics systems. Most important aspects include false rejection rate (FRR), false acceptance rate (FAR), and equal error rate (EER) [229]. Although the FRR explains the "proportion of verification transactions with truthful claims of identity that are incorrectly rejected", the FAR defines the "proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed" (ISO/IEC FDIS 19795-1). The genuine acceptance rate (GAR) is distinct as GAR = 1- FRR. As score distributions overlap, FAR and FRR intersect at a certain point (FRR=FAR), define the EER of the system. Regarding to intra and interclass accumulations generated by biometrics algorithm, FRRs and FARs are attuned by varying system thresholds.

In general, decreasing the FRR (corresponds to increasing the GAR) increases the FAR and vice versa. Sometimes, instead of FAR and FRR, rates called false match rate (FMR) and false non-match rate (FNMR) are reported. These rates are referring to similar errors, but do not include errors specific to biometrics [38], such as the failure-to-enroll rate (FTE), the proportion of people failing to enroll successfully.

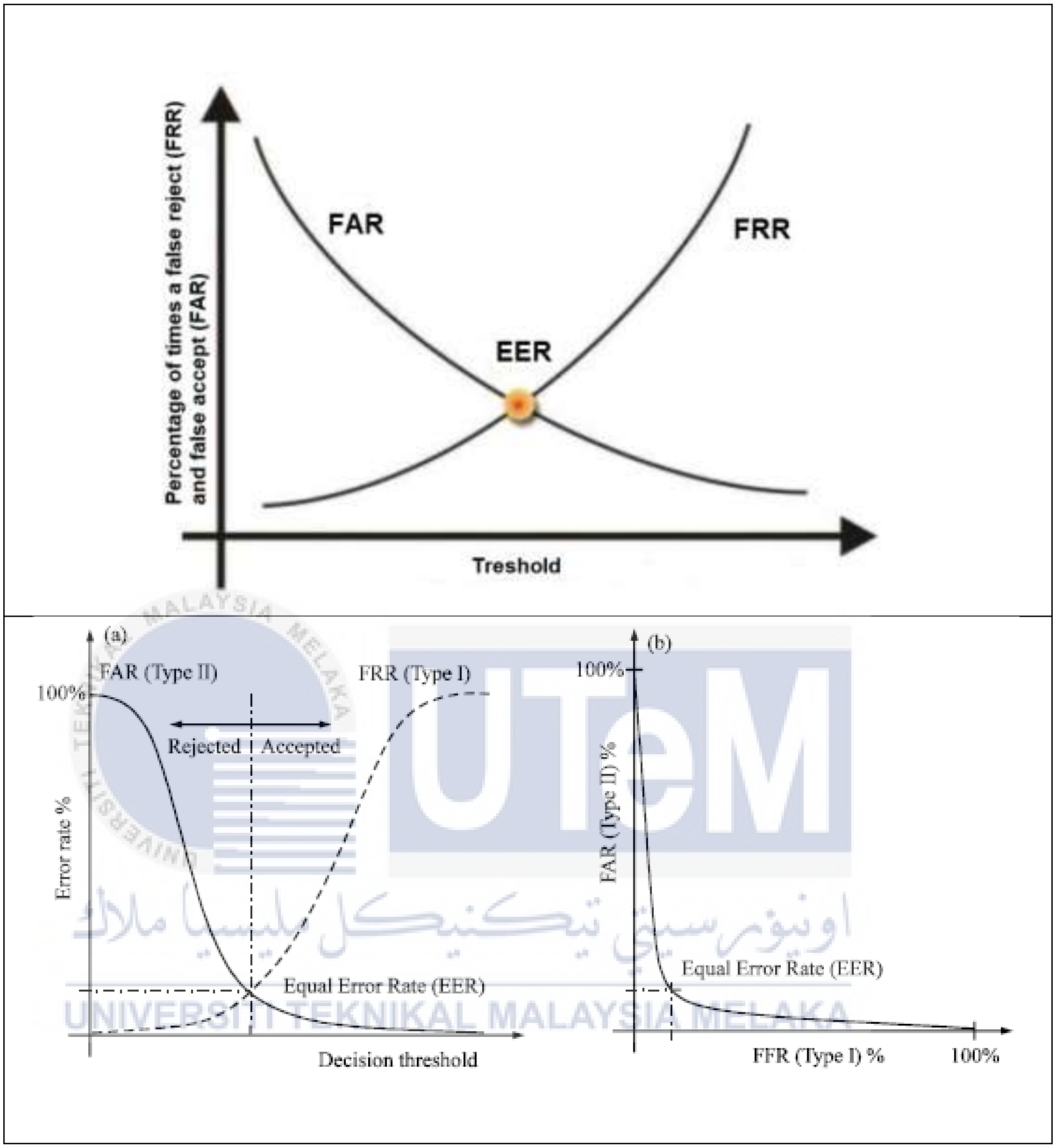


Figure 2.7: Performance of an (iris) biometrics system: Interrelation between false rejection (FRR), false acceptance rate (FAR) and equal error rate (EER)

2.3 Critical Review of Current Problem and Justification

Swarm Intelligence has ended up being a promising and strange zone in the field of nature roused procedures that are utilized to get to the base of enhancement issues at some phase in the previous decade. It is inspired by the common conduct of social living things. Swarm based improvement calculations hit upon arrangement by communitarian experimentation.

In journal of Artificial Bees or multilevel thresholding of iris image written by A. Bouaziz, A. Draa and S. Chikhi,(2015) CASIA and UBIRIS database had been used to improved iris detection rate. There are combinations of techniques that used in this experimental, which are used of two Artificial Bee Colony (ABC); Conventional Artificial Bee Colony and G-best guided Artificial Bee Colony (CS + PSO).

Therefore, there are other alternative ways to improved iris detection rate and more accurate for iris recognition by using algorithm, such as Gaussian Filter Algorithm, Masek Algorithm (circular Hough Transform), and also Integro-Differential Operator of Daugman. These algorithms can be apply and used to remove noise in iris recognition. (Abidin 2012)

In this journal, Masek Algorithm and Integro-Differential Operator of Daugman Algorithm had been used to improve iris recognition rate and both algorithm had been tested in both CASIA and UBIRIS database. As a result, Masek Algithm showed the highest accuracy for iris recognition in CASIA database which is 94.57%, while in UBIRIS database is about 93.77%. Different in Intergo-Differential Operator of Daugman Algorithm, UBIRIS database give better accuracy reading than CASIA which is 68.38% and CASIA 63.22%. (Bouaziz et al. 2015)

The advantage of this purpose is it helps in increase the accuracy in iris recognition by remove the noise of iris.

2.4 Bees Algorithm Proposed Solution

Improvement calculations are pursuit techniques where the objective is to locate an ideal answer for an issue, with a specific end goal to fulfill one or more target capacities, conceivably subject to an arrangement of imperatives. The fundamental Bees Algorithm and its enhanced adaptations are depicted and are executed so as to advance a few benchmark capacities, and the outcome is contrasted and those got with various streamlining calculations. This Bees Algorithm offering some favorable position over other enhancement techniques as indicated by the way of the issue and it is an improvement calculation enlivened by the characteristic scavenging conduct of bumble bees to locate the ideal arrangement. (Akay & Karaboga 2012)

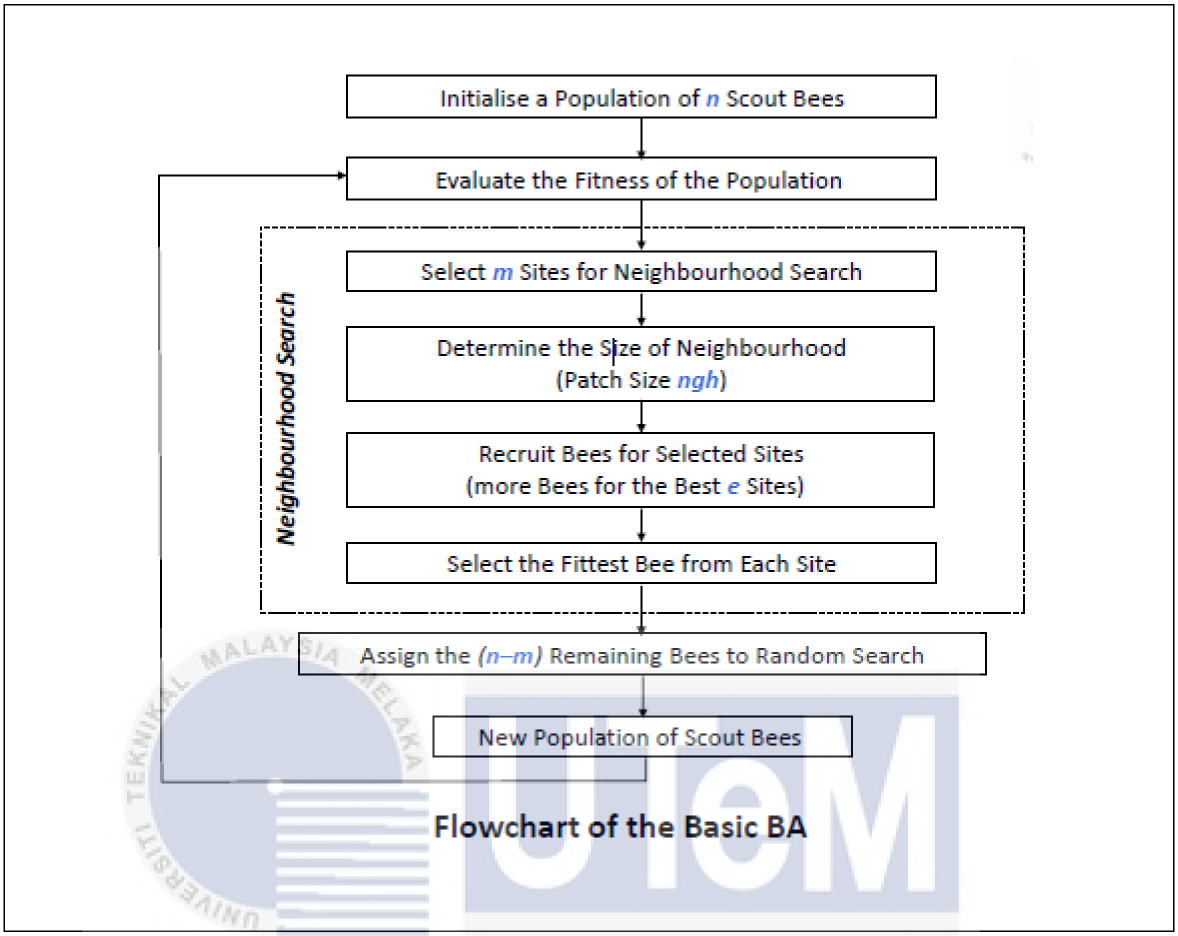


Figure 2.8: Flowchart of the basic Bees Algorithm

The Bees Algorithm has both neighborhood and worldwide pursuit capacity using abuse and investigation techniques, separately. The Bees Algorithm utilizes the arrangement of parameters given in Figure 2.9 and the pseudo-code of the calculation is given in Figure 3.0

Parameter	Symbols
Number of scout bees in the selected patches	*n*
Number of best patches in the selected patches	*m*
Number of elite patches in the selected best patches	*e*
Number of recruited bees in the elite patches	*nep*
Number of recruited bees in the non-elite best patches	*nsp*
The size of neighborhood for each patch	*ngh*
Number of iterations	*Maxiter*
Difference between value of the first and last iterations	*diff*

Figure 2.9: Basic parameters of Bees Algorithm

Generate the initial population size as *n*, set the best patch size as *m*, set the elite patch size as *e*, set the number of forager bees recruited to the of elite sites as *nep*, set the number of forager bees around the non-elite best patches as *nsp*, set the neighborhood size as *ngh*, set the maximum iteration number as *MaxIter*, and set the error limit as *Error*.

$i = 0$

Generate initial population.

Evaluate Fitness Value of initial population.

Sort the initial population based on the fitness result.

While $i \leq MaxIter$ *or* $FitnessValue_i - FitnessValue_{i-1} \leq Error$

1. $i = i + 1$;
2. Select the elite patches and non-elite best patches for neighborhood search.
3. Recruit the forager bees to the elite patches and non-elite best patches.
4. Evaluate the fitness value of each patch.
5. Sort the results based on their fitness.
6. Allocate the rest of the bees for global search to the non-best locations.
7. Evaluate the fitness value of non-best patches.
8. Sort the overall results based on their fitness.
9. Run the algorithm until termination criteria met.

End

Figure 3.0: Pseudo-code of Bees Algorithm

For condition in Bees Algorithm procedure, there are three stages which are instatement stage, utilized honey bees stage and Onlooker Bees stage. (Bouaziz et al. 2015)

1. Initialization phase

The colony of artificial bees includes three groups of bees, employed bees, onlooker bees and scout bees. The food source vectors are initialized by scout bees. The initialization formula is shown as follow;

$$x_{mi} = l_i + rand(0,1) * (u_i - l_i) \quad (1)$$

where $\vec{x}_m$ is a solution vector to be optimization problem, l_i and u_i are the lower and upper bound of the parameter , respectively.

2. Employed bees phase

Employed bees search for new food source with more nectar within the neighborhood of the food source (x_{mi}) in their memory. They evaluate the profitability (fitness) after they had find a neighbor food source. Formula (7) shows a way to determine the neighbor food source.

$$v_{mi} = x_{mi} + \phi_{mi}(x_{mi} - x_{ki})$$

(2)

Where $\vec{x}_k$ is a random chosen food source, *i* is a random parameter index and ϕ_{mi} is a random number within the range [-a, a]. Thereafter, the fitness is calculated and the formula is as below,

$$fit(\vec{x}_m) = \begin{cases} \frac{1}{1 + f_m(\vec{x}_m)} & if \quad f_m(\vec{x}_m) \geq 0 \\ 1 + abs(fm(\vec{x}_m)) & if \quad f_m(\vec{x}_m) < 0 \end{cases} \qquad (3)$$

Where $f_m(\vec{x}_m)$ is the objective function value of solution $\vec{x}_m$.

3. Onlooker bees phase

Employed bees share their food source information with onlooker bees in the hive by dancing, and then onlooker bees choose the food sources by the probability. The probability is calculated by fitness values provided by employed bees. The probability value pm with which $\vec{x}_m$ is selected by an onlooker bee could be calculated by equation (4),

$$p_m = \frac{f_m(\vec{x}_m)}{\sum_{m-1}^{SN} f_m(\vec{x}_m)}$$

(4)

After a food source $\vec{x}_m$ for an onlooker bee is probabilistically chosen, a neighborhood source $\vec{v}_m$ is determined by using equation (4), and its fitness value is computed. Hence, more onlookers are recruited to richer sources and positive feedback behaviour appears.

2.5 Iris Segmentation using Raspberry Pi

Raspberry Pi utilizes a standard mouse and console that are master card size and low in expense. It's known as charge card size on account of its capacity to connects to a PC screen or T. It is an able little gadget that empowers individuals of any age to investigate figuring, and to figure out how to program in dialects like Scratch and Python. In this project, Raspberry Pi is use as a testing platform of the algorithm proposed in this study (Balasubramani & Marcus 2013).

In this project research, Raspberry Pi 2 Model B been used. This model is the next generation of Raspberry Pi, and it traded the original Raspberry Pi 1 Model B+. This Raspberry Pi consumes a 900MHz quad-core ARM Cortex-A7 CPU and holds 1GB RAM.

This process takes parts in MABLAB to help us to acquire data from sensors and imaging devices that connected to Raspberry Pi. While connected, we can use the Raspberry Pi camera board, 12C interface, SPI interface, Serial Interface, GPIO and Linux system shell to captured iris images for segmentation process.

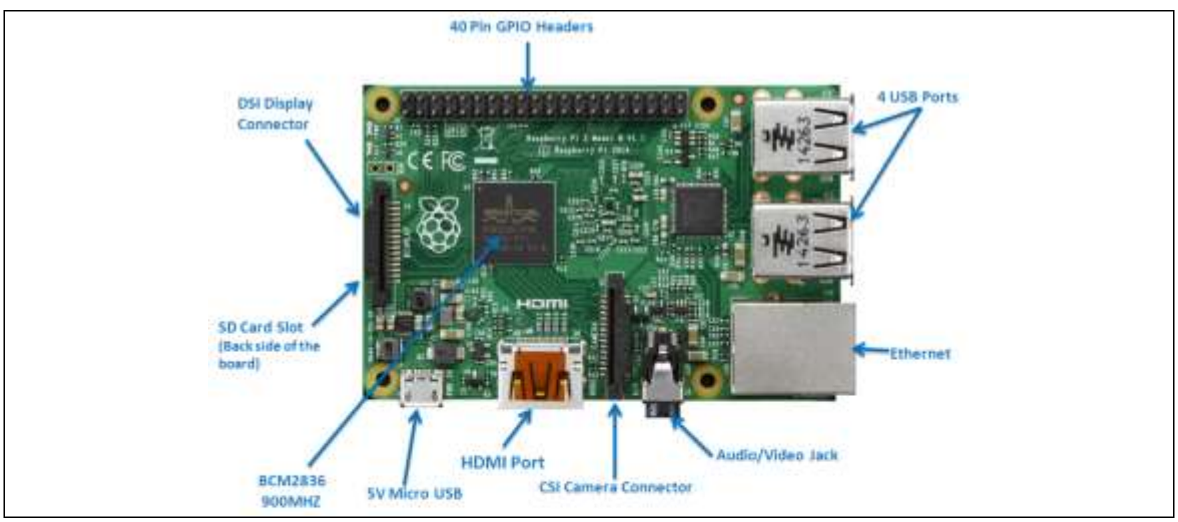


Figure 3.1: Raspberry Pi 2 Model B

2.6 Conclusion

In conclusion, we had recognize one of the problem in iris recognition which is noisy iris that may lead to low accuracy of recognition rate and. The iris feature change and occlusion is the leading problem in iris recognition.

In chapter 2, Bees Algorithm had been purpose as a technique to improve and increase the accuracy rate in iris recognition. Compared to other algorithms, Bees Algorithm are easy to use and has the ability for local search and global search. This Bees Algorithm also available for hybridization combination with other algorithm plus it is implemented with several optimization problems.

CHAPTER III

PROJECT METHODOLOGY

3.1 Introduction

Project methodology is a basic method also known as principle that needed to apply on a study of project. By following the method or principle, it clearly state out the goal of the project and we can see the right way to archive the project's objectives. The concepts and ways are logically related.

3.2 Methodology

Project methodology is developed in order to guide project to be conducted in a correct sequence. For this project, Daugman's approach has been used as model of reference for iris recognition system in biometrics.

In Daugman's approach, there are four main phases, which are image acquisition phase, segmentation phase, feature extraction phase and lastly is matching phase.

In image acquisition phase, the eye image is captured using camera sensor. While in segmentation phase, the pupil and iris are localized and then iris is unwrapped into normalized form. In third phase, it is process of removal of the unwanted information from the iris template and then encoding process by binarization of iris template. In matching phase, real iris is verified to identify if the user is accepted or rejected by the system. Figure 3.0 shows how Daugman's Approach works.

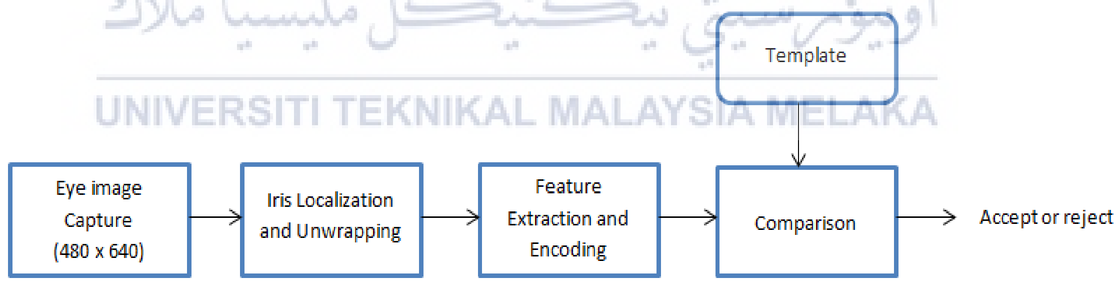


Figure 3.0: Daugman's Approach of Iris Recognition.

Phase 1: Eye Image Capture

Taking a photo from iris is underlying phase of an iris-based recognition system. It deals with capturing of a high quality image of the iris. Worries on the picture procurement rigs, acquire pictures with adequate determination and sharpness besides, a great complexity in the iris design with appropriate enlightenment. Very much focused without unduly obliging the administrator, the separation is up to 3 meter and close infrared camera or LED.

Achievement of other acknowledgment stages is dependent on the nature of the pictures taken from iris amid picture securing stage. Pictures accessible in CASIA database need appearance in understudy and iris regions since infrared was utilized for imaging. Furthermore, if obvious light is utilized amid imaging for those people whose iris is dull, a slight differentiation comes to presence amongst iris and understudy which makes it difficult to isolate these two ranges.

After capturing iris images, the images need to be stored in one place, which called database. Database is a place provides for store and sharing images with other researcher and it is available to public for research and educational purpose. Databases that are mostly use in biometric are CASIA and UBIRIS.

Phase 2: Iris Localization and Unwrapping

Iris localization is a procedure to seclude the iris district from whatever is left of the obtained image. Iris can be approximated by two circles, one for iris or sclera boundary and another for iris or pupil boundary.

This involve in segmentation stage which exact iris picture division assumes an essential part in an iris acknowledgment framework since accomplishment of the framework in up and coming stages is straightforwardly subject to the accuracy of this stage. The principle motivation behind division stage is to restrict the two irises limits in particular, internal limit of iris-student and external one of iris-sclera and to localize eyelids. Localization in segmentation stage includes three steps;

1. Localization of iris inner boundary (the boundary between pupil and iris).
2. Localization of iris outer boundary (the limbic border between sclera and iris).
3. Localization of boundary between eyelids and iris.

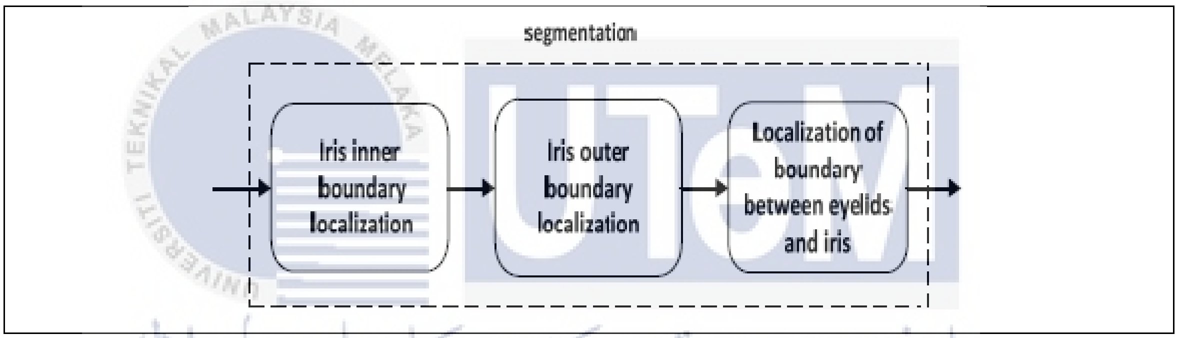


Figure 3.1: Block diagram of Segmentation Stage

Phase 3: Feature Extraction and Encoding

The purpose of the feature extraction module is to generate a compact representation of the biometric characteristics in order to alleviate the comparison (matching) process. In this stage, it's including noise removal from iris image and generating iris code. Many different feature extraction methods have been proposed for iris recognition such as feature extraction techniques distinguished by the type of generated feature vectors; binary and real-valued methods.

To separate the components, two-dimensional Gabor Filters are utilized, and by performing Gabor Filters to the image from various orientations, extreme element vector is acquired. In this stage, the measurements of the element vector extricated from iris range must be as little as could be expected under the circumstances. With respect to measurements of the image drawn, Wavelet change was performed so as to lessening the measurements in the way that imperative data existing in tissue can be protected regardless of cutting back image dimensions. Later, this picture is utilized to concentrate highlights vector.

Regarding to some occlusions that caused by eyelids and eyelashes in feature extraction that may cause error in segmentation stage, some parts of sclera that detected as iris area are need a measure be taken to remove these points form the feature extraction stage. To resolve this problem, encoding is needed. Encoding is use in matching stage and two outputs are generated in this stage, which the first output is belongs to transformation of iris to iris encoding and another output belongs to transforming iris noises into encodings.

Feature encoding was implemented by convolving the normalized iris pattern with 1D Log-Gaber wavelet. 2D normalized patterns are broken up into a number of 1D signal. Each row corresponds to a circular ring on the iris region. The angular direction is taken rather than the radial one, which corresponds to columns of normalized pattern. The features are extracted in codes of 0 and 1.

Phase 4: Comparison

For matching stage, Hamming Distance was chosen as a metric for recognition, and iris images are taken from template of database which is from CASIA database or UBIRIS database. In the event that the estimation of highlight vector in a point is equivalent to the estimation of other element vector in that point, digit 1 and on the off chance that they are not equivalent, digit zero is dispensed to that point and afterward the qualities allotted to the pixels are summed up and likeness rule of the two vectors is achieved by finding the best of Hamming separation understanding in taking after condition:

$$HD = \frac{1}{N}\sum_{i=1}^{N}(x_i \oplus y_i) \quad (1)$$

Figure 3.2: Hamming Distance equation

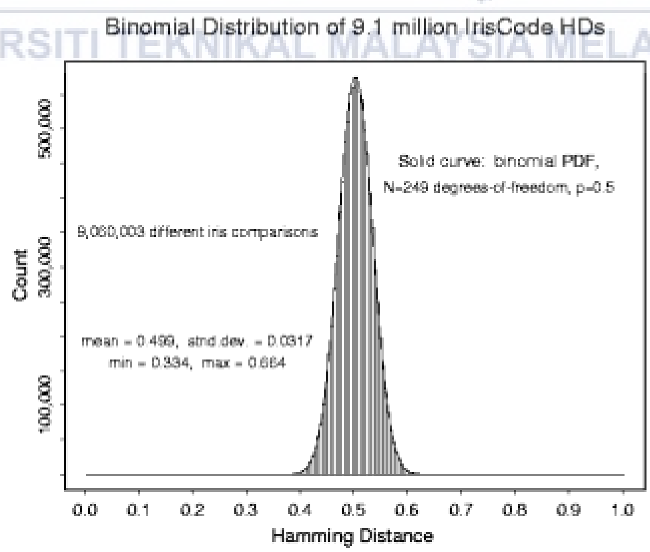


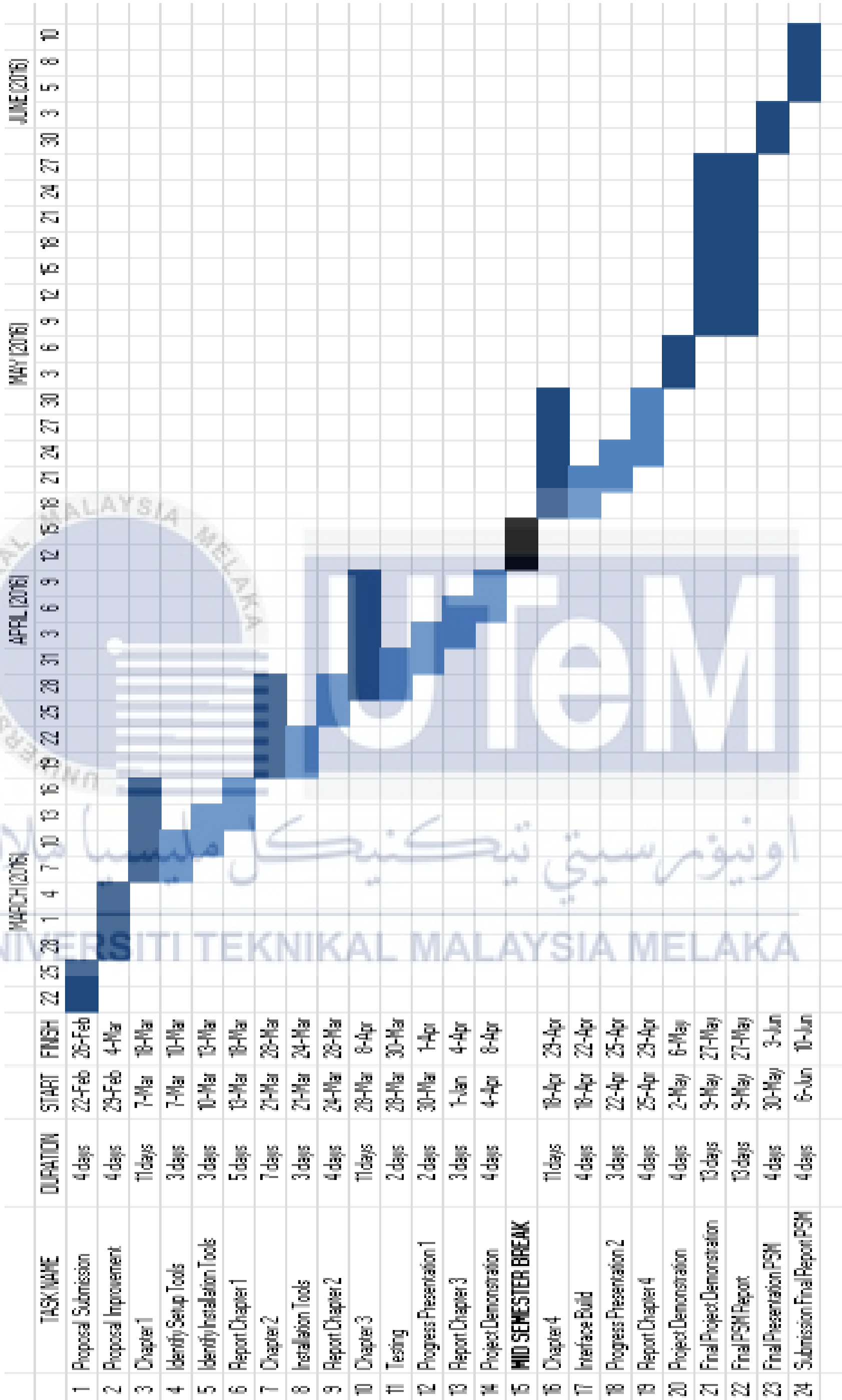
Figure 3.3: Binomial Distribution of Iris Code Hamming Distances

The result of this computation is then used as the goodness of match, with smaller values indicating better matches. If two patterns are derived from same iris, the hamming distance between them is close to 0 due to high correlation.

3.3 Milestone

Week	Date	Activities
1	22 – 26 February	- Proposal Submission & Presentation
2	29 – 4 March	- Proposal Correction & Improvement - Chapter 1
3	7 – 11 March	- **Chapter 1** - Identify Environment Setup - Documentation
4	14 – 18 March	- **Chapter 1 & Chapter 2**
5	21 – 25 March	- **Chapter 2** - Interface Preparation - Documentation
6	28 March – 1 April	- **Chapter 2 & Chapter 3** - Progress Presentation 1
7	4 – 8 April	- Project Demonstration & **Chapter 3** - **Chapter 4**
8		**MID SEMESTER BREAK**
9	18 – 22 April	- Project Demonstration - **Chapter 4**
10	25 – 29 April	- Project Demonstration & **Chapter 4** - Progress Presentation 2
11	2 – 6 May	- Project Demonstration
12	9 – 13 May	- Project Demonstration - PSM Report
13	16 – 20 May	- Project Demonstration - PSM Report
14	23 – 27 May	- Project Demonstration - PSM Report
15	30 – 3 June	- Final Presentation
16	6 – 10 June	- Submission Final Report PSM

3.4 Gantt Chart



3.5 Conclusion

This chapter explain the methodology on overall process occur in this project and relevant methodology that used regarding to iris recognition system. Milestone and Gantt chart help in development and act as a guideline to this project. There are four phase of iris recognition that involve in this project; Acquisition phase (iris image acquisition), Segmentation and normalization phase (image pre-processing), Extraction phase (feature extraction) and matching phase (comparison). These phases are important in iris recognition because it help in determine if the individual is genuine or not. In next topic, the design and analysis of the project be reviewed.

CHAPTER IV

DESIGN

4.1 Introduction

This chapter investigates and identifies the problems and the weakness of this project. The current problem been analyzed from various sources. This chapter also describes on the Requirement Analysis and High Level Design. Under the Requirement Analysis, Hardware and Software Requirement are covered.

4.2 Problem Analysis

The current system of iris recognition consists of acquisition, segmentation, normalization and matching. A comparison of iris template image from database and real time process is to determine either user is real or impostor.

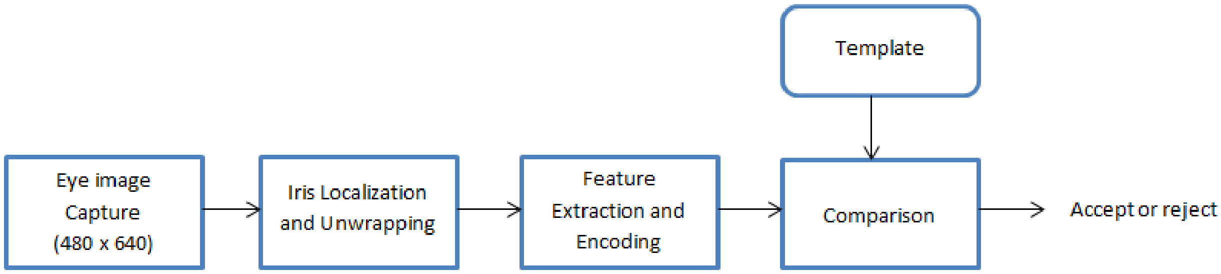


Figure 4.1: Current System of Iris Recognition

The current system has several limitations, for instance;

a) The Bees Algorithm requires new fitness tests on the new algorithm parameters to improve performance,
b) Slow down when used in sequential processing and slow to obtain accurate solutions,
c) A demand of a robust mechanism to detect the iris region within the iris image regardless the non-iris image (eyelashes, eyebrow, sclera)

However, in this project, the scope of this study is focusing on modify the algorithm for high accuracy in iris distortion template on iris recognition. To obtain less intrusive quality iris image template, circular segmentation has been used to fulfill the demand for modification of algorithm to archive high distortion template.

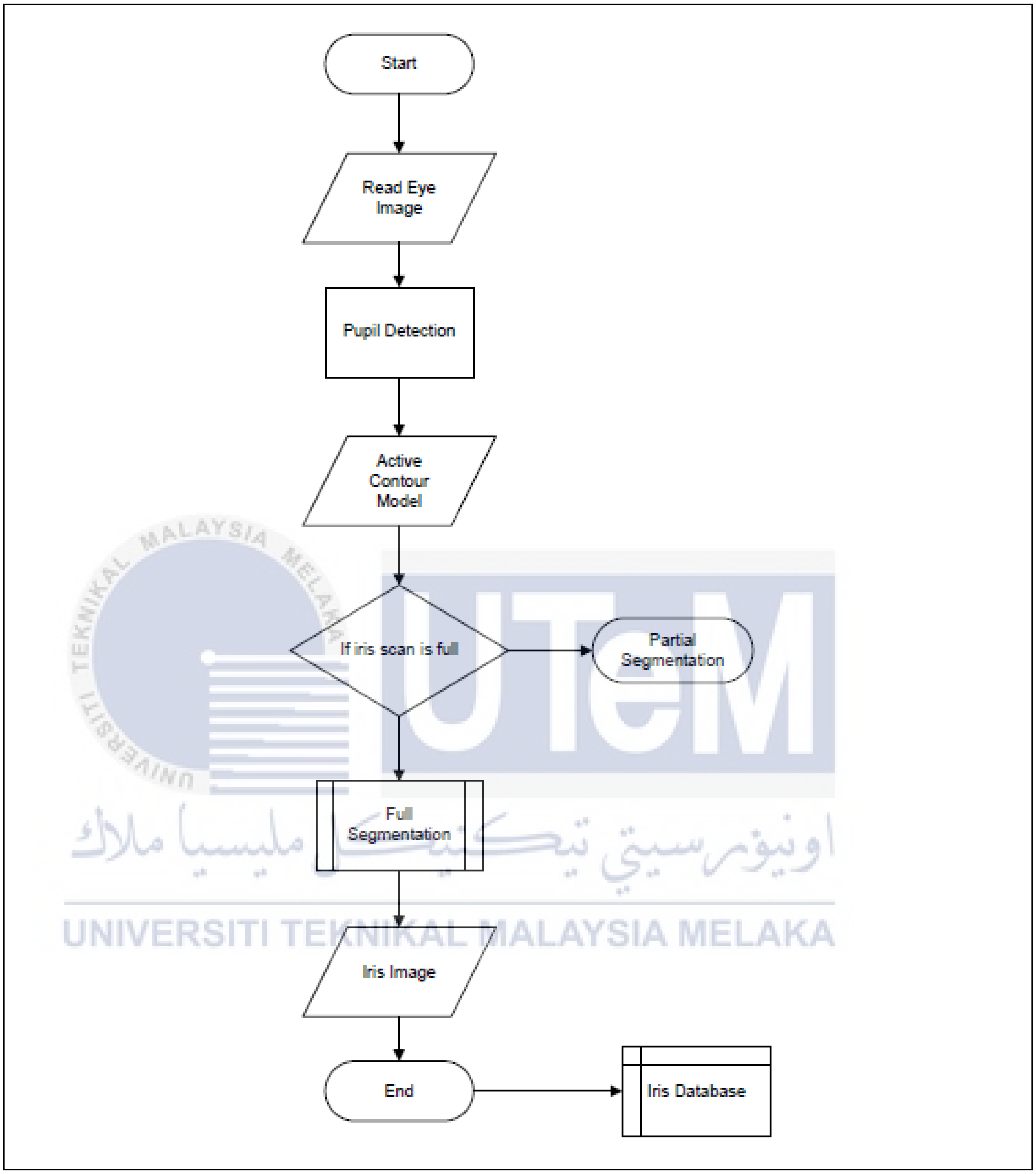


Figure 4.2: Flowchart of Segmentation

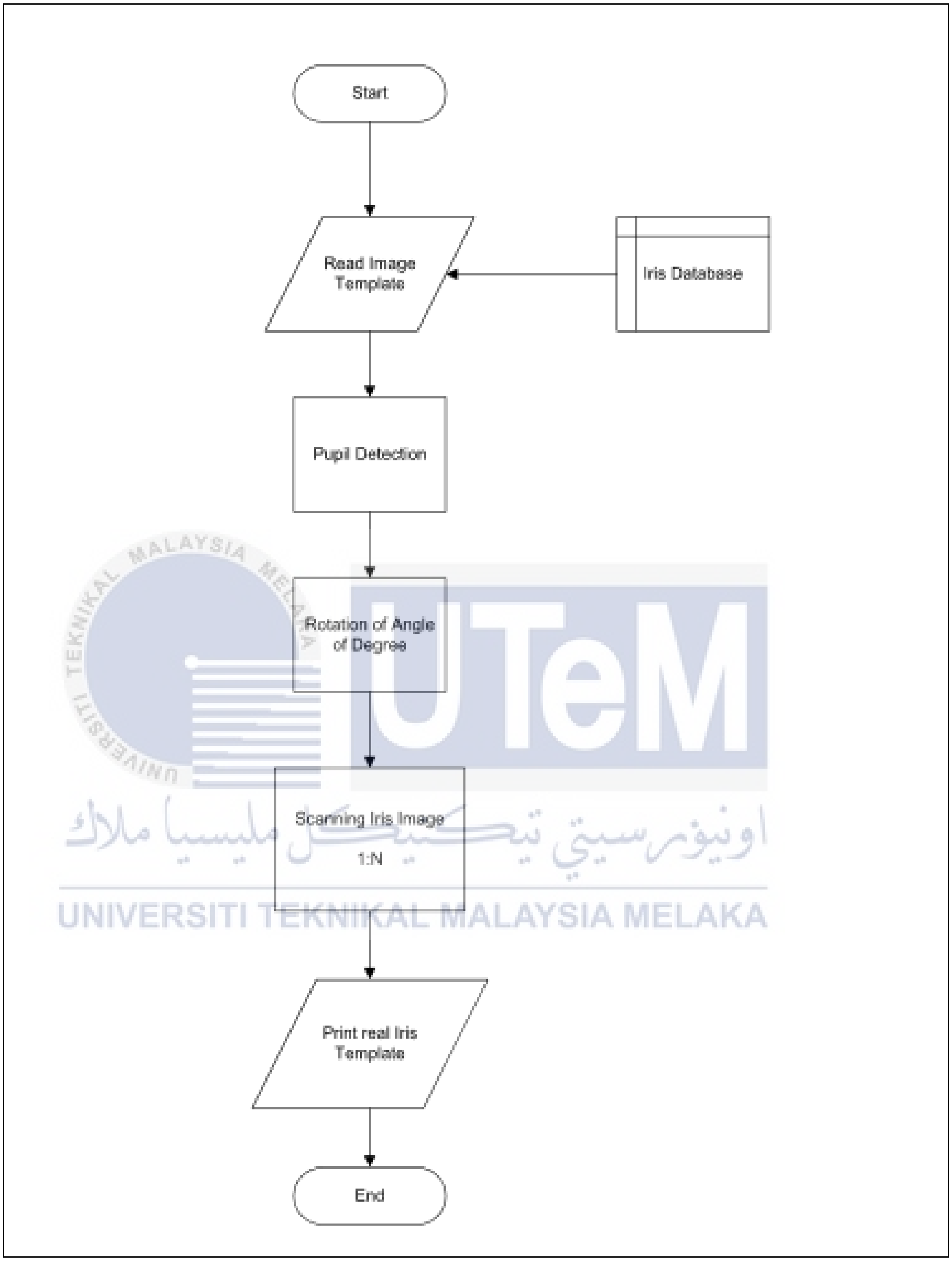


Figure 4.3: Flowchart of Full Segmentation

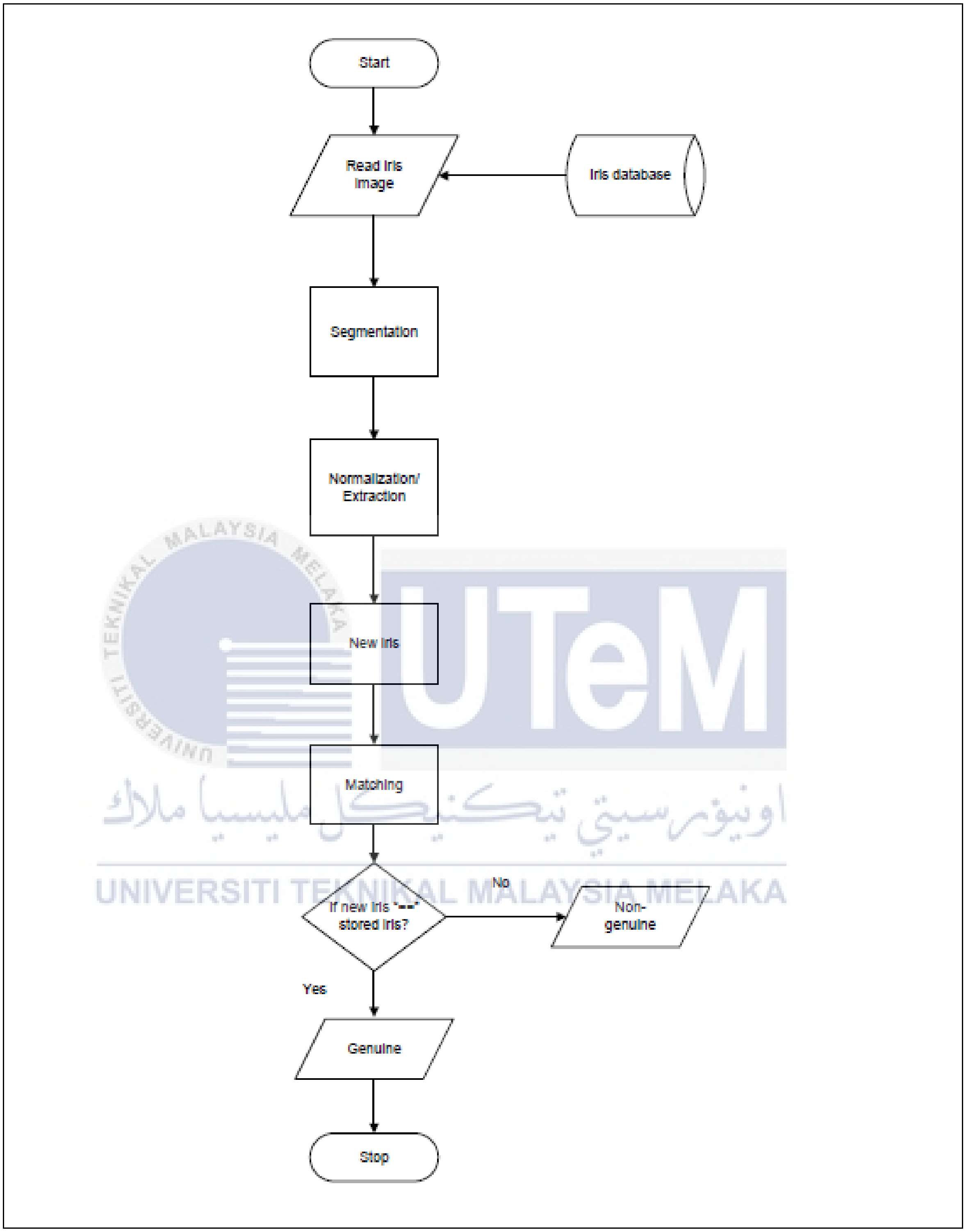


Figure 4.4: Iris Recognition Process

4.3 Requirement Analysis

4.3.1 Data Requirement

Generally, data that is required in the development of this prototype are images. There are a few of images that are needed, and it is all begin with the eye/iris images. This eye/iris images are captured recorded by using a special camera and then, the eye/iris images then be converted into other kind of images in the system. Next, the images are going through some process that is segmentation, extraction and matching before identity of a real person is realized.

Figure 4.5: IriShield™ -USB BK 2121U Camera

Product Name	IriShield-USB BK 2121U	
Descriptions	IriShield-USB Serial supports USB Interface, Security Infrastructure and various OS family, such as Windows, Linux, WinCE, Embedded Linux and Android	
Capture Mode	Auto capture	
Capture Distance	BK 2121U	From the front camera lens : 13.5 cm – 14.5 cm (5.3 inches-5.7 inches) (Optimal distance = 14 cm (5.5 inches) Focal depth = 1.0 cm (0.4 inch) Field of View = 3.3 cm x 2.4 cm at 15 cm (1.3 inches x 0.9 inch at 5.9 inches)
Image Format	IOS Standard 19794-6 (2005 & 2011), (640 x 480 Pixels, 8 bit Grayscale) Full support of K1,K2,K3,K7	
Sensor Resolution	VGA	
Dimensions	BK 2121U	IriShield-USB : 124 mm x 63.2mm x 42.5 mm (4.9 inches x 2.49 inches x 1.68 inches) Google : 200 mm x 145 mm x 72 mm (7.9 inches x 5.7 inches x 2.8 inches)
Power	BK 2121U	Single USB Bus Powered (DC +5±5%) (Max power consumption = 430 mA)
Illumination	Infrared LED	
Environmental	BK 2121U	Operating : 0 ℃ to +50 ℃ Storage : -20 ℃
Usage	Indoor Outdoor (avoid direct sunlight and bright reflections)	
Compliance & certificates	Eye safety standard (IEC 62471:2006-07), RoHS FCC-Class A * , IP54*	
Resolution	Spatial : ≥ 60% @ 4.0 Lp/mm, Pixel : ≥ 16 Pixels/mm	

Connectivity	USB 2.0
Security	PKI (2048-bit) and AES (256-bit); Z509 Certificate PFX/PKCS #12 Certificate, RSA key pair generated on-board
Host OS	Windows Family, Linux Family, WinCE, Embedded Linux, Android and Mac OS

4.3.2 Hardware Requirement

The hardware requirement for this prototype is a workstation with a good performance, and a specific camera for capturing eye images. Below is the specification of the workstations while the camera required is as explained.

- Workstations

1) Laptop

Type of Hardware	Description
Manufacturer	Fujitsu
CPU	Intel® Core™ 2 Duo T7250 @ 2.00GHz
Maximum Turbo Speed	2.00 GHz
Number of Cores	Dual-Core
System Type	32-bit Operating System
Chipset Type	Mobile Intel HM65 Express
Maximum Supported Size (Memory)	2GB
Memory Speed (RAM)	1066 MHz
Display	15.4 inch, 16:10, 1280x800 pixels
Widescreen	Yes
Graphic Processor	Integrated /On- Board Graphics
Graphic Adapter	ATI Mobility Radeon X1400, Intel Graphics Media Accelerator (GMA) 4500HD, Intel Graphics Media (GMA) 950, Intel Graphics Media Accelerator (GMA) X3100, NVIDIA GeForce 8400M G
Data Link Protocol (Networking)	Ethernet, Fast Ethernet,
Wireless LAN Supported	Yes

2) PC

Type of hardware	Descriptions
Manufacturer	Dell
Processor	Intel® Core™ i7-4790 Processor (Quad Core, 8MB, 3.60GHz w/HD4600 Graphics)
Operating System	Windows 7 Professional English, French, Spanish 64bit (Includes Windows 10 Pro License)
Security Software	No Anti-Virus SW
Memory	4GB (1x4GB) 1600MHz DDR3 Memory
Hard Drive	500GB 3.5inch SATA (7200 RPM) Hard Drive
Graphics Card	Intel® Integrated Graphics
Optical Drive	16X DVD-ROM
Digitally Delivered Software	NCR
Ports	4 External USB 3.0 Ports 2 Front 2 Rear 6 External USB 2.0 Ports 2 Front 4 Rear except USFF 4 Rear only and 2 Internal USB 2.0 MT only 45 - RJ Network Connector 1 Serial 1 VGA 2 DisplayPort 2 PS/2 MT/SFF only 2 Line-in stereo/MicrophOne 2 Line-out Headphone/speaker
Keyboard	US English (QWERTY) Dell KB212-B QuietKey USB Keyboard Black
Mouse	Dell MS111 USB Optical Mouse

3) Raspberry Pi 2 Model B

Type of hardware	Descriptions
Brand Name	Raspberry Pi 2 Model B
Processor	0.9 GHz ARM 7100
RAM	1 GB SDRAM
Number of USB 2.0 Ports	4
Hardware Platform	Linux
Operating System	Linux
Weight	3.2 ounces
Dimensions	5 x 4 x 3 inches
Color	CPU
Processor Brand	ARM
Computer Memory Type	DIMM
Power Requirements	5 volts
Support Operating System	Windows 8.1, Debian GNU/Linux, Fedora, Arch Linux, Risc Os
Micro SD card	8GB
Storage	microSD
Low-Level Peripherals	27 x GPIO, UART, SPI bus with two chip selects, +3.3V, +5V, Ground
Video System Feature	Broadcom VideoCore IV GPU Video/Audio Out via 4-pole 3.5mm connector, HDMI, or Raw LCD (DSI)

4.3.3 Software Requirement

MATLAB®	
Microsoft© Windows 7	
Microsoft© Office Visio	
Microsoft© Office Project	
Linux Operating System (Debian)	

MATLAB: Software that use high-level language and interactive environment for numerical computation, visualization and programming. Use to analyze data, creates applications and models and to develop algorithm. Multi-use software for a range of applications, such as image and video processing, test and measurement, signal processing and computational biology and finance.

Windows 7: This project focused only on windows 7 environment. This operating system (OS) is widely use and it is convenience.

Microsoft Visio: Used to draw diagram

Microsoft Project: Used for developing the Gantt chart

Linux Operating System: Used as interface of Raspberry Pi

4.4 High- Level Design

This topic describes the high level view of the system structure and User Interface Design.

4.4.1 System Architecture

System architecture shows the real time of the overall system and the way the workstations, PC and the raspberry Pi was connected. The Raspberry Pi acts as the main Hardware as all other hardware is connected to it. Figure 4.6 (a) below shows the actual environment that has been set up for this project. While in Figure 4.6 (b) and Figure 4.6(c) shows the connection between Raspberry Pi and other hardware such as camera, Ethernet cable, monitor, mouse and keyboard.

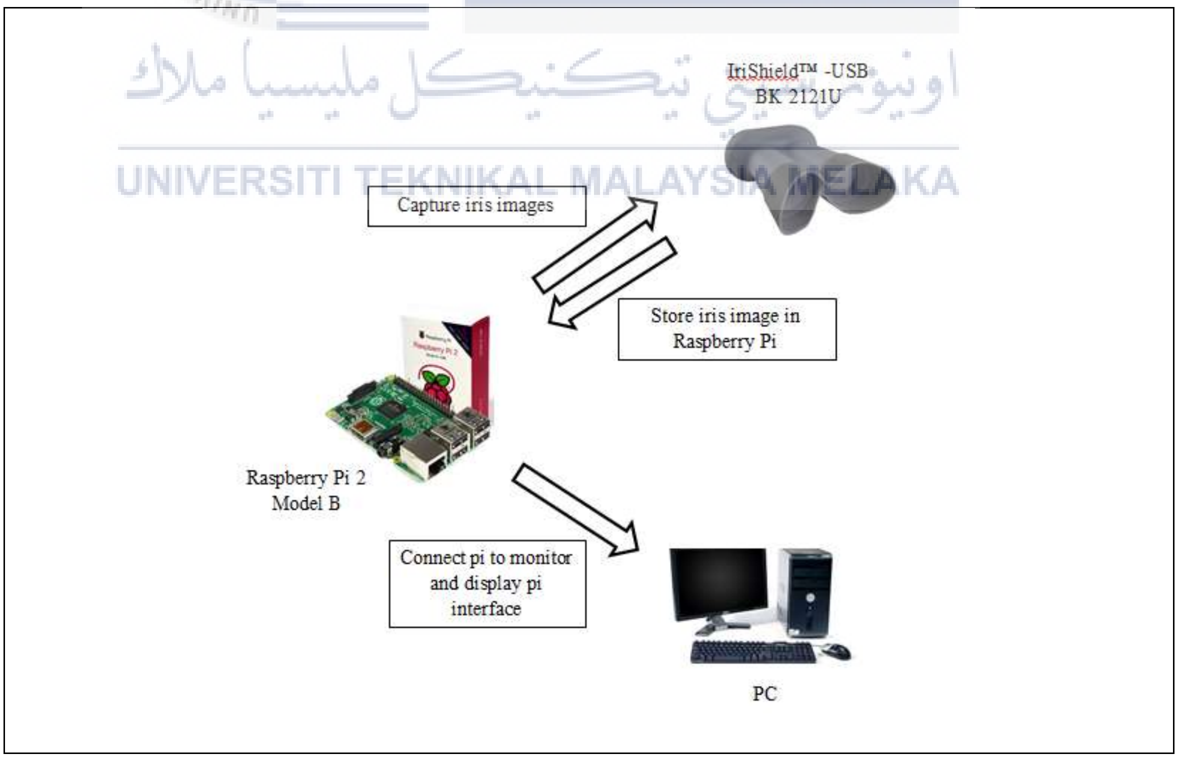


Figure 4.6(a): System Architecture

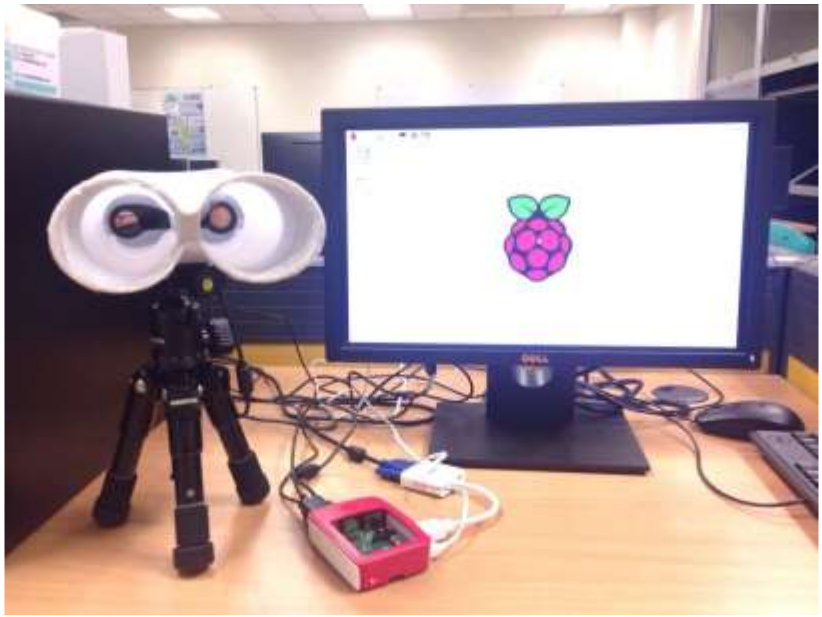


Figure 4.6(b): System Architecture

Figure 4.6(c): System Architecture

4.4.2 User Interface Design

User interface design is a design of the prototypes application which focus on user's requirement and user;s interface. The function of user interface design is to provide a simple and efficient user;s interface. The user interface in this prototype is simple and clear.

In order to capture iris image by using IriShield™ camera that remotely control through Raspberry Pi, the installation of Pygaze need to be done at first. Pygaze is an open source for eye-tracking software. It offers platform that is more user-friendly than the currently existing alternatives. The installation steps of Pygaze are as follows;

Step 1: Install the NumPy, SciPy, Matplotlib, PIL, PyGame, Psychopy and pyglet by typing following command in the order.

- *sudo apt-get install python-numpy python-scipy python-matplotlib python-imaging python-pygame psychopy python-pyglet*

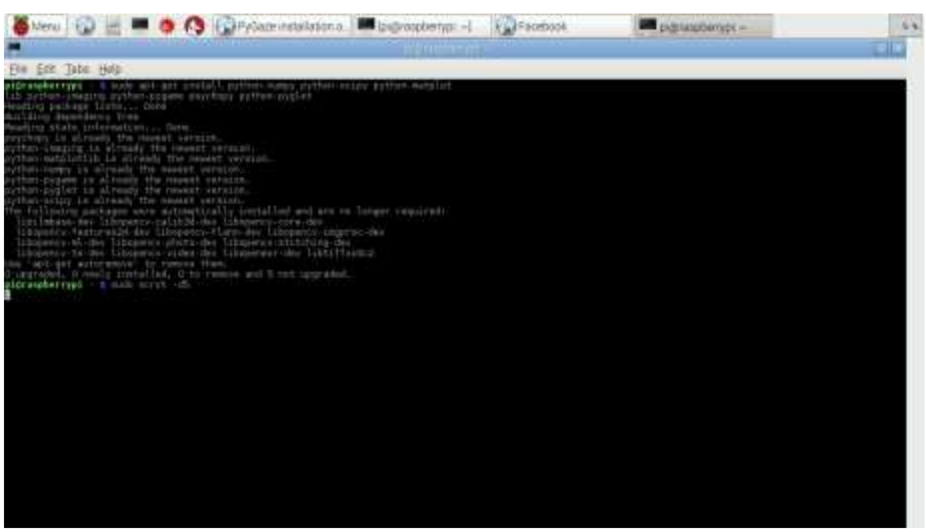


Figure 4.7: Installation Free Software

Step 2: Go to https://www.sr-support.com/forum.php and register with SR Research support forum to download pylink in the tar.gz file from the website.

Step 3: Install pylink by unpack the file in the terminal with this command:

- *tar -xvzf pylink4python2.7_linux_x32.tar.gz*

Step 4: Go into the unpacked folder "pylink4python2.7″:

- *cd pylink4python2.7*

Step 5: Copy the folder pylink2.7 to the folder [/usr/lib/pymodules/] and rename it to "pylink"

- *sudo cp -r pylink2.7 /usr/lib/pymodules/pylink*

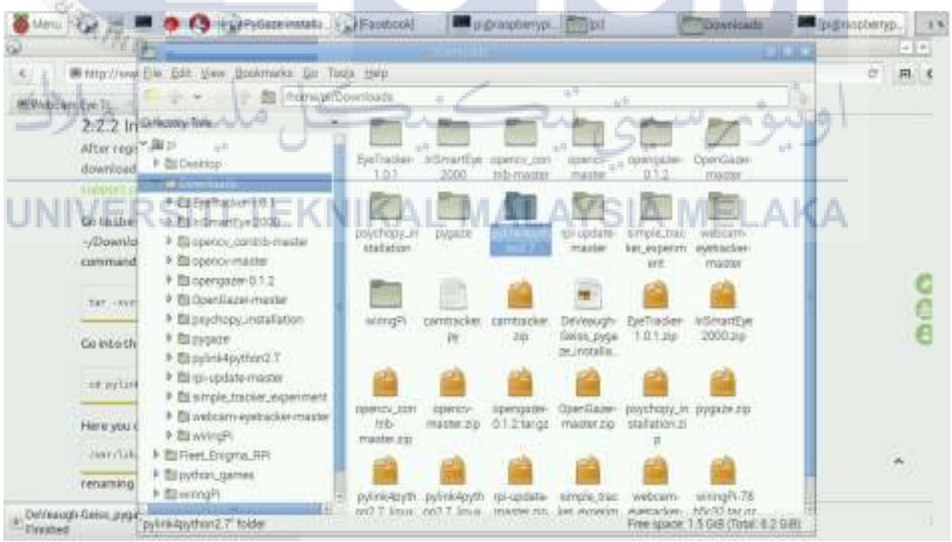


Figure 4.8: Installation of Pylink

Step 6: Then, download the Pygaze zip file from http://www.pygaze.org/downloads/. Unzip the file and copy to the site-packages directory. Then set the permissions. Type the following command.

- *unzip pygaze.zip*
- *sudo cp -r pygaze /usr/local/lib/python2.7/dist-packages/*
- *sudo chmod 2755 -R /usr/local/lib/python2.7/dist-packages/pygaze/*

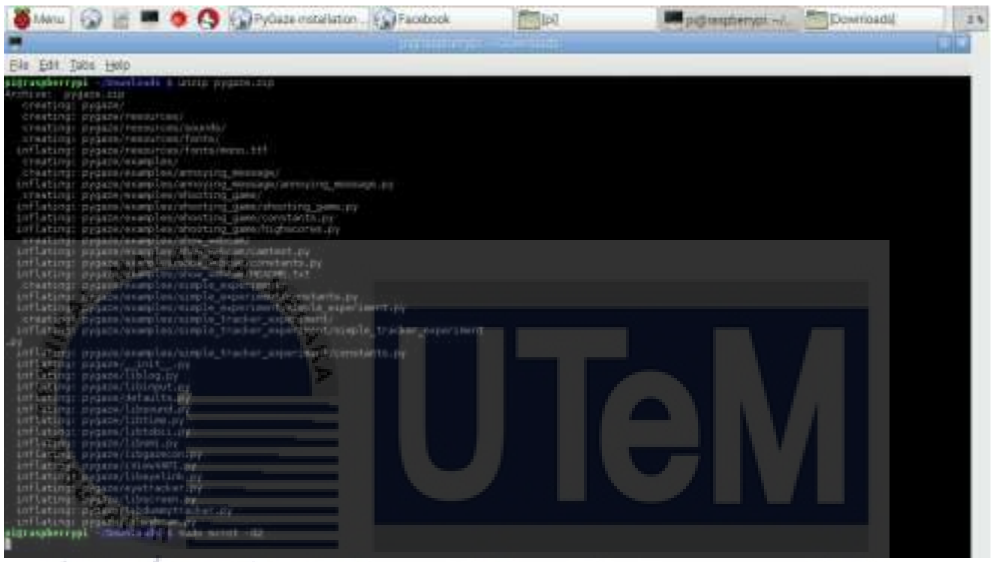


Figure 4.9(a): PyGaze Installation

Step 7: Installing PyGaze requires copying the PyGaze directory into Python's site-packages directory. In the terminal, type:

- *python*

Step 8: To check the site-packages directory, type

- *import site; site.getsitepackages()*

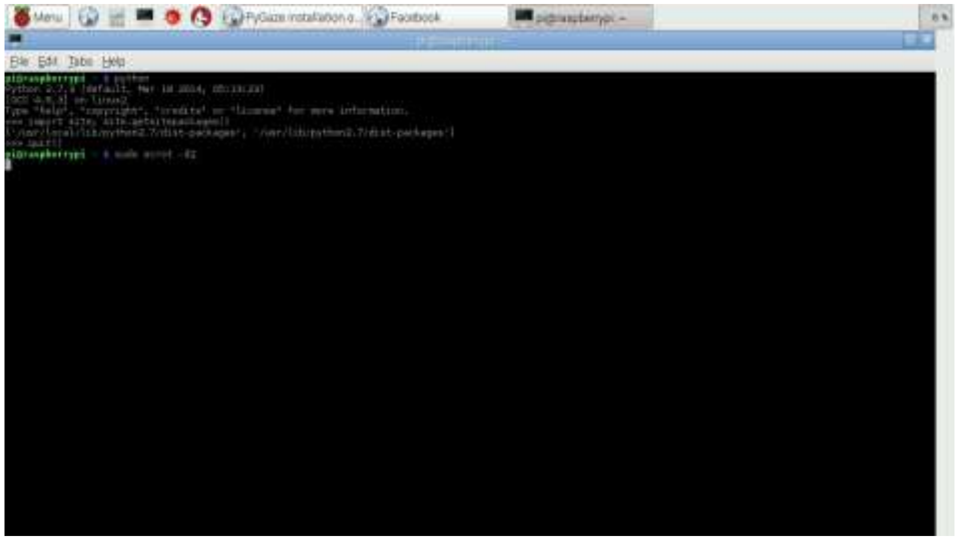


Figure 4.9(b): PyGaze Installation

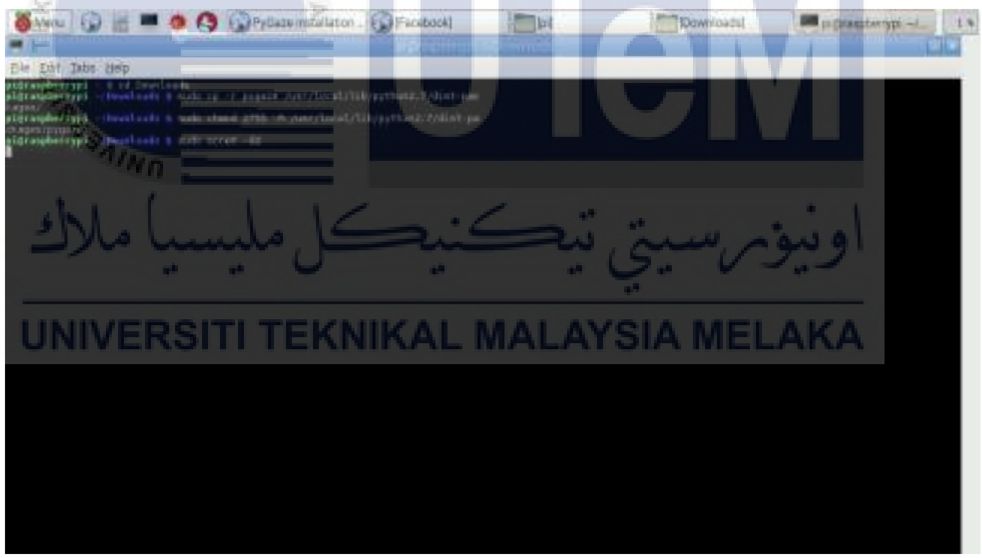


Figure 4.9(c): PyGaze Installation

Step 9: Test the installation by type the following command:

- *python*
- *import pygaze*

***** Pygaze is successfully installed because there is no error message.

Figure 5.0: Testing PyGaze

After finishing the installation of PyGaze, the IriShield™ -USB BK 2121U (camera) need to be tested to fulfill the requirement of setup for iris images tracking and capturing. For this project, we are used IriShield™ -USB BK 2121U camera to capture the iris images, then stored and process the images data in Raspberry Pi. While for the interface and software for the images capturing, we are using IriSmartEye2000 software that are connected and had been installed in Raspberry Pi.

Figure 5.1(a): Iris images

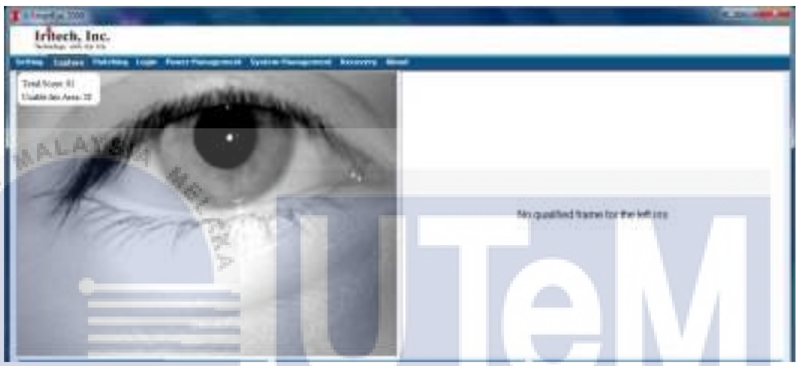


Figure 5.1(b): Iris image with low quality (noisy)

Iris image quality is a determining factor regarding the accuracy of iris-based identification. Higher quality images produce more accurate iris recognition results. Iris image quality could be affected by several factors including; poor capturing conditions, heavy occlusion by eyelid, eyelashes, or by foreign objects, image blurriness caused by excessive movement and iris nonlinear/non-uniform deformation due to pupil's constriction and dilation.

4.5 Conclusion

In this chapter, it covers three main topics which are the project analysis, requirement analysis and the prototype detailed design. In project analysis, it explains about the iris recognition process. Meanwhile, in the requirement analysis, it describes about the hardware and software used in this project and its specifications. For the design, it is conducted in Linux based operating system which is Debian in Raspberry Pi 2 Model B. by using the requirements and the design; it is able to develop a good design prototype for this project. For the next chapter, the implementation part of this project been discussed.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

Implementation is an important stage for developing a project. In this chapter, setup of the environment is discussed and how it is done for implement to this research. Regularly, the major work involve in implementing a project are the system coding, debugging and installing.

Firstly, the tools that been used to implement this research are determined, starting from the acquisition of iris images until the segmentation process without error. The iris biometrics process is running step by step based on the function module in this implementation stage. In this stage, MATLAB code is the main factor that determines the performance of the segmentation process in iris biometrics.

5.2 Environment Setup

The main environment for this research is MATLAB and all the iris images are processed using MATLAB software. In this research, MATLAB R2014b is used and the required operating system is Windows platform. The hardware that is used in this system is a personal workstation with standard performance as long as it run MATLAB and output result. After preparing the hardware, the entire software requirement been installed into the workstation.

For this project research, Bee Algorithm technique is the important and been highlight in order to analyze the performance in images based segmentation of iris biometrics. Bee Algorithm technique has been chooses to be implemented in this project research in order to find the best and optimal solutions for iris template in high noise images and low frequency.

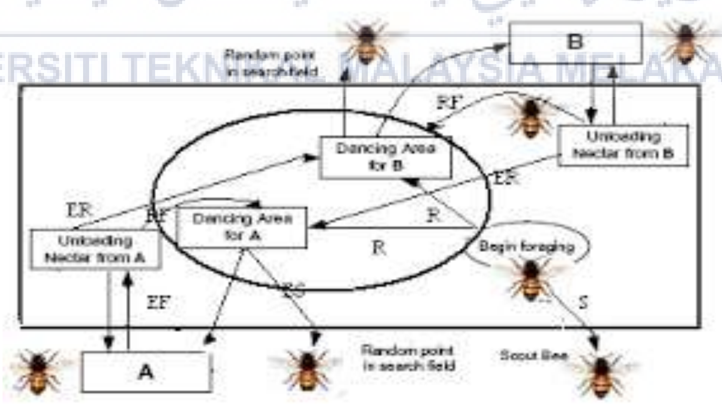


Figure 5.2: Show the typical behavior of honey bee foraging on finding the best and optimal solutions http://ceit.aut.ac.ir/~meybodi/paper/Aghazadeh-IPCSIT%202011-Singapur-2011.pdf

The Bees Algorithm is a population-based search algorithm inspired by the natural foraging behaviour of honey bees. In its basic version, the algorithm starts by scout bees being placed randomly in the search space. Then, the fitnesses of the sites visited by the scout bees is evaluated and Bees that have the highest fitnesses are chosen as “selected bees” and sites visited by them are chosen for neighbourhood search.

Then, the algorithm conducts searches in the neighbourhood of the selected sites, assigning more bees to search near to the best e-sites. Searches in the neighbourhood of the best e sites are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm.

The remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each of iteration, the colony has two parts, those that were the fittest representatives from a patch and those that have been sent out randomly. The algorithm performs a kind of neighbourhood search combined with random search and can be used for both combinatorial and functional optimization.

Lists of system coding used for segmentation process in this project:

1. Coding of Bees Algorithm for iteration
2. Coding of Bees Algorithm for track iris images
3. Coding of Bees Algorithm for time and success rate calculation

Table 5.1(a): Problem Definition for iteration

```
%% Problem Definition
CostFunction=@(x) Sphere(x);
nVar=5;
VarSize=[1 nVar];
VarMin=-10;
VarMax= 20;
```


Table 5.1(a) defined the problem definition for Iris Segmentation Process using Bees Algorithm. Variable "X" is defines as cost function, meanwhile the "nVar" is defines as number of decision variables and "VarSize" is the decision variables for matrix size. "VarMin" and "VarMax" are for lower and upper bound variables that setting up for threshold in Bees Algorithm.

Table 5.1(b): Bee Algorithm parameters for iteration

```
%% Bees Algorithm Parameters
MaxIt=100;
nScoutBee=10;
nSelectedSite=round(0.5*nScoutBee);
nEliteSite=round(0.4*nSelectedSite);
nSelectedSiteBee=round(0.5*nScoutBee);
nEliteSiteBee=2*nSelectedSiteBee;
r=0.1*(VarMax-VarMin);
rdamp=0.95;
```

In Table 5.1(b) Bees Algorithm procedure and source code had been executed into iris pictures in division process. There are the arrangements of parameters that utilized as a part of this procedure; "MaxIt" is alludes to the most extreme number of cycle used to locate the ideal and best hunt highlights in iris pictures and "nScoutBee" is characterizes as number of scout honey bees that required in seeking the iris highlights. Next, "nSelectedSite", "nEliteSite", "nSelectedSiteBee", "nEliteSiteBee" are allude to number of locales that include in seeking the components in iris pictures.

Table 5.1(c): Coding of Bees parameter for the main iteration process

```
%% Bees Algorithm Main Loop
for it=1:MaxIt
    for i=1:nEliteSite

        bestnewbee.Cost=inf;

        for j=1:nEliteSiteBee
            newbee.Position=PerformBeeDance(bee(i).Position,r);
            newbee.Cost=CostFunction(newbee.Position);
            if newbee.Cost<bestnewbee.Cost
                bestnewbee=newbee;
```

```
            end
        end

        if bestnewbee.Cost<bee(i).Cost
            bee(i)=bestnewbee;
        end

    end

    %
    for i=nEliteSite+1:nSelectedSite

        bestnewbee.Cost=inf;

        for j=1:nSelectedSiteBee
            newbee.Position=PerformBeeDance(bee(i).Position,r);
            newbee.Cost=CostFunction(newbee.Position);
            if newbee.Cost<bestnewbee.Cost
                bestnewbee=newbee;
            end
        end

        if bestnewbee.Cost<bee(i).Cost
            bee(i)=bestnewbee;
        end

    end

    %
    for i=nSelectedSite+1:nScoutBee
        bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
        bee(i).Cost=CostFunction(bee(i).Position);
    end

    %
    [~, SortOrder]=sort([bee.Cost]);
```

```
    bee=bee(SortOrder);
    %
    BestSol=bee(1);
    %
    BestCost(it)=BestSol.Cost;
    %
    disp(['Iteration ' num2str(it) ': Best Fitness = '
num2str(BestCost(it))]);
    %
    r=r*rdamp;

end
```

In Table 5.1(c), the coding showed the main loop where the operation and process in searching the features were happened.

Table 5.1(d): Coding of Bees parameter to display the result of iteration and fitness

```
%% Results
figure;
semilogy(BestCost,'LineWidth',2);
xlabel('No. of Iteration');
ylabel('Best Fitness');
```


Table 5.1(d) showed the coding used to display the graph result for the Best Fitness versus Number of Iteration. Result from this graph then be used for analyze and measure the performance, accuracy and speed of the process takes.

Table 5.2(a): Coding Bees Algorithm to track iris image

```
%
im = imread('S1019L01.jpg');
figure(1);
subplot(2,2,1);
imshow(im);
title('Original image');
%
subplot(2,2,2);
codes = locateCodes(im)
title('Tracked image');
```

Table 5.2(a) demonstrated the coding of Bees Algorithm that used to track iris pictures in division process. This coding is utilized to track first iris images, which is correct iris pictures. This coding then as of late uses to test different iris images in the database to see their exhibitions.

Table 5.2(b): Coding Bees Algorithm for track iris image after iteration process

```
im2 = imread('S1019L01Segment_n_100.jpg');
subplot(2,2,3);
imshow(im2);
title('Original Image after iteration');

subplot(2,2,4);
codes2 = locateCodes(im2, 'colMode', 1, 'thresh', 0.2)
title('Tracked image after iteration');
```

Table 5.2(b) showed Bees Algorithm source code that used to track iris images after iteration stage in segmentation process. The parameters of Bees Algorithm for this coding have been modified in order to get a clear and less blurriness result.

Table 5.3(a): Coding of Bees parameter to calculate success rate and times

```
% Set BA algorithm parameters
BAparams.BA = 10;
BAparams.MaxImp = 1,000;
BAparams.HMCR = 0.5;
BAparams.PAR = 0.4;
BAparams.b = (xU-xL)/1000;
```

Table 5.3(a) showed the parameters of Bees Algorithm technique that used to calculate the success rate of segmentation process and time taken for it to complete the process.

Table 5.3(b): Coding of Bees parameter to start the calculation

```
Nrep    = 1000;
NFE     = zeros(Nrep,1);
success = zeros(Nrep,1);
CPUtime = zeros(Nrep,1);

for i=1:Nrep
    [xbest,fbest,NFE(i),success(i),CPUtime(i)] ...
        = bee_stats(f,xL,xU,fmin,tol,BAparams);
end

NFE_valid = NFE; NFE_valid(~success) = [];
CPUtime_valid = CPUtime; CPUtime(~success) = [];
```

Table 5.3(c): Coding of Bees parameter to display the result of success rate and times

```
% Display results
fprintf('Success rate = %.0f%%\n', mean(success)*100)
fprintf('Mean NFE = %.0f; min NFE = %.0f; max NFE = %.0f; SD NFE =
%.0f\n', ...
    mean(NFE_valid), min(NFE_valid), max(NFE_valid), std(NFE_valid))
fprintf('Mean CPU = %fs; min CPU = %fs; max CPU = %fs; SD CPU = %fs\n',
...
    mean(CPUtime_valid), min(CPUtime_valid), max(CPUtime_valid),
std(CPUtime_valid))
```

Table 5.3(c) showed the coding used to display the result of success rate and times taken for one iris images to completed segmentation process.

5.3 Conclusion

Implementation phase is important in order to define how the project is implemented. The fifth chapter has reviewed an implementation that covered for environment setup as well as expected output of the prototype. Testing report been generated to record result that to be analyzed. Next, these projects proceed to testing which elaborate more in the sixth chapter.

CHAPTER VI

TESTING AND ANALYSIS

6.1 Introduction

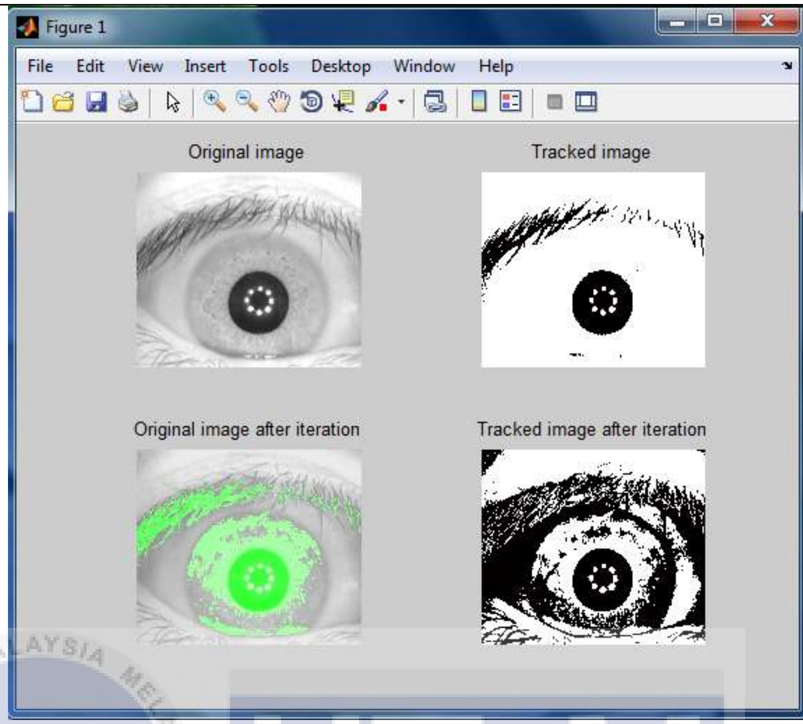
Testing process is a part of internal control review which accesses the actual practice follows or complies with prescribed conditions and procedure. One of aims for this testing process is to make sure whether the objective s is achieved. Test result also been discussed.

This testing and analysis chapter is described the last stage of this project research. In this stage, the segmentation process need to be built and tested to ensure this project meets the expectation in order to achieve the objectives.

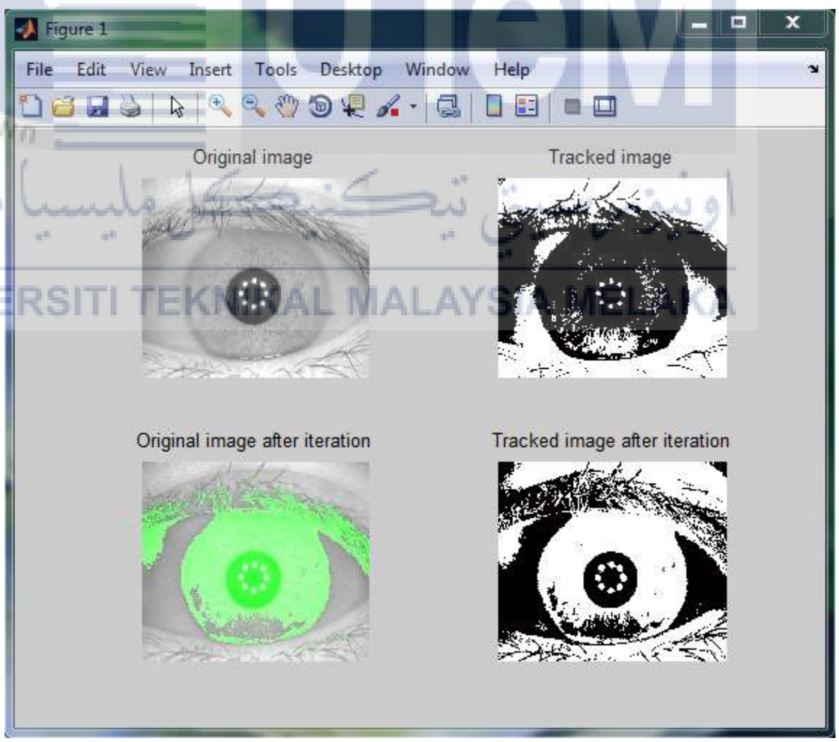
6.2 Result and Analysis

Test result is documented to ensure that the process is running and been tested accordingly by starting with acquisition stage, where templates of iris images have been collected to be tested in segmentation process, iteration process and also segmentation with Bees Algorithm. An iris image that undergoes this process is selected to test the performance of Bees Algorithm technique and the noise type also noise environment. At this stage, some modifications are made to correct the coding system and improve the segmentation process to get a better result.

After finish with the correction on the coding system, the calculation of iteration process with modified parameters being tabulated. This is to analyze the best iteration number and fitness between default parameter and modified parameters. To analyze the performance of this Bees Algorithm, a coding system been generate and modified to test the success rate and times takes for an iris image to complete the segmentation process.



(a)



(b)

Figure 6.1: (a) Iris Segmentation using default parameters in Bee Algorithm (b) Iris Segmentation using Iteration and modified parameters in Bee Algorithm

After done with the segmentation process of iris images, next we calculate the best fitness of Bees Algorithm. Best fitness is refers to the fitness of best number for scout bee in bees population (refers to Bees Algorithm). However, the average of best fitness is the mean of the fitness values across the entire bee population. For each generation, when the population changed, the average population fitness also changed.

In Swarm Based technique for Iris Biometrics, Bees Algorithm showed that the best fitness tends to get better as the iterations proceed. This happens quickly at first, and then slowing down as the algorithm finds better and better solutions that are harder to improve upon.

The average fitness is always less than or equal to the best fitness, and the difference between the two goes on decreasing over time, which is until the algorithm completely converges, and the population is containing copies of the same best number of scout bees in the population.

Graphs below show the performance of the Bee Algorithm through the calculation of iteration and best fitness. The parameters of Bee Algorithm are important in order to apply the coding into iris segmentation process. This calculation has been run for 200 times to collect the best iteration and best fitness for Bee Algorithm.

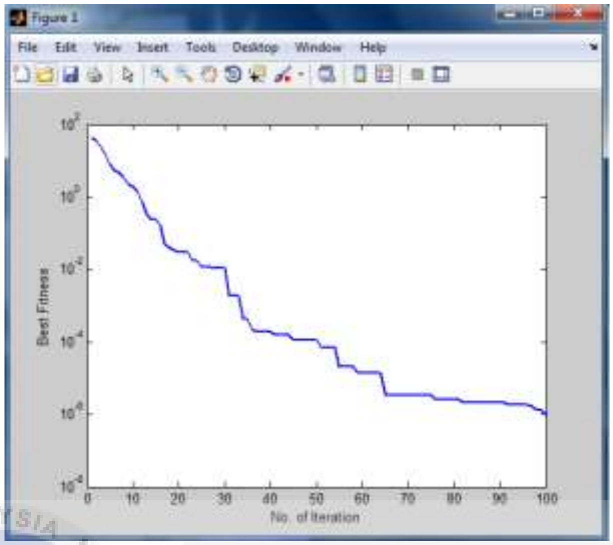
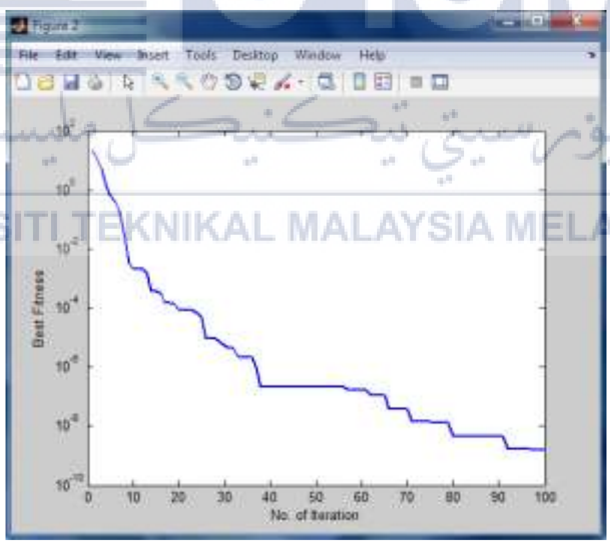
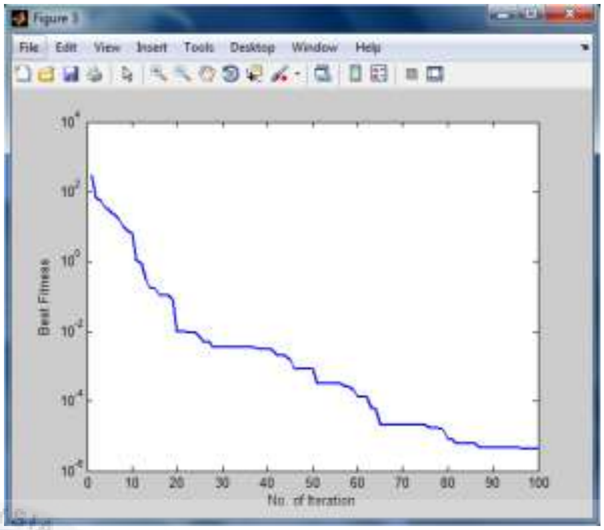
Parameters: MaxIt = 100 VarMax = 10 nScoutBee = 10  **(a)**
Parameters: MaxIt = 100 VarMax = 20 nScoutBee = 10  **(b)**

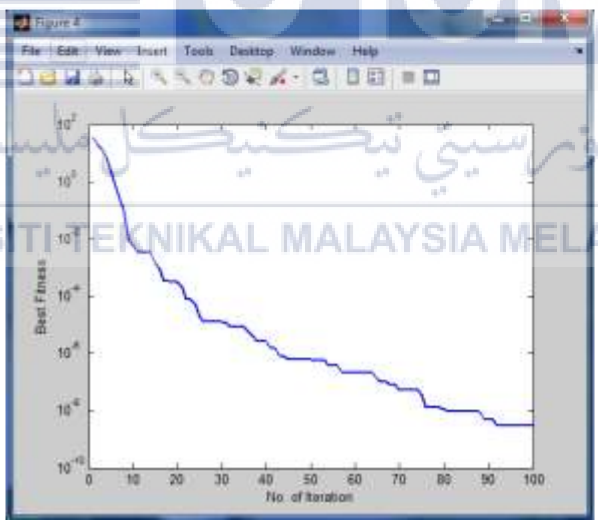
Figure 6.2: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm.

Parameters: MaxIt = 100
VarMax = 10
nScoutBee = 100



(a)

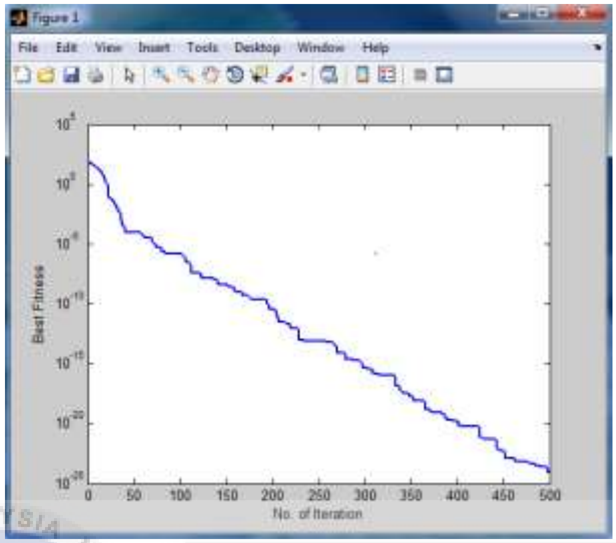
Parameters: MaxIt = 100
VarMax = 20
nScoutBee = 100



(b)

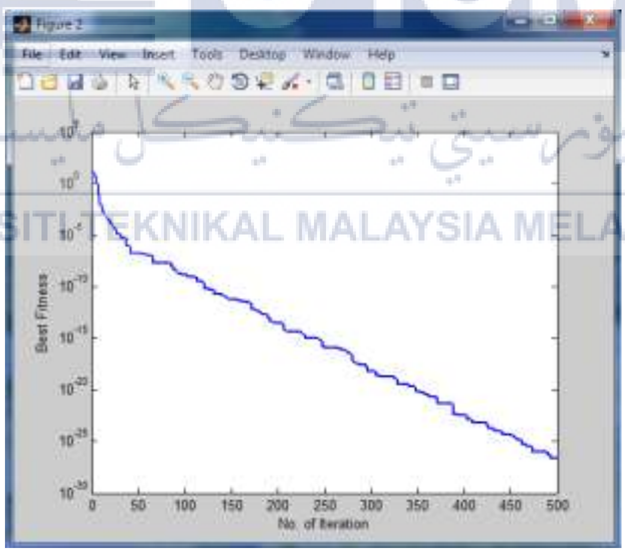
Figure 6.3: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm

Parameters: MaxIt = 500
VarMax = 10
nScoutBee = 10



(a)

Parameters: MaxIt = 500
VarMax = 20
nScoutBee = 10



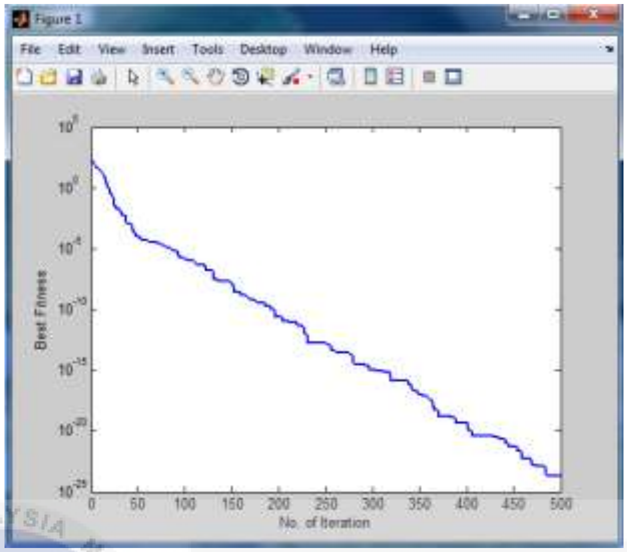
(b)

Figure 6.4: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm

Parameters: MaxIt = 500

VatMax = 10

nScoutBee = 100

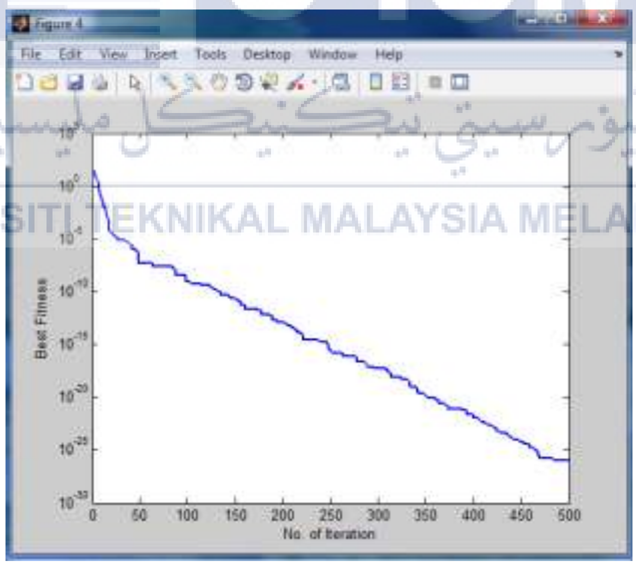


(a)

Parameters: MaxIt = 500

VatMax = 20

nScoutBee = 100



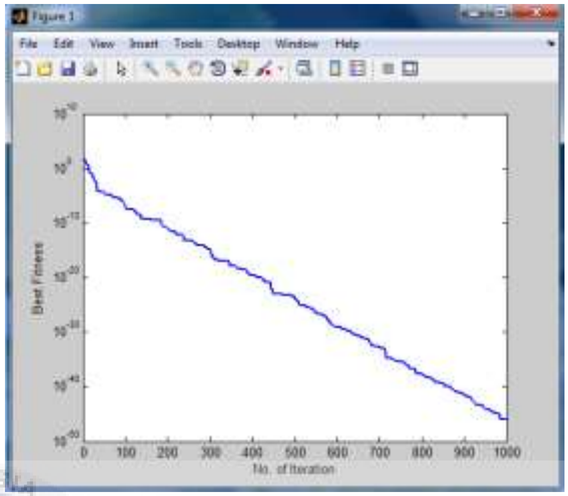
(b)

Figure 6.5: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm

Parameters: MaxIt = 1000

VarMax = 10

nScoutBee = 10

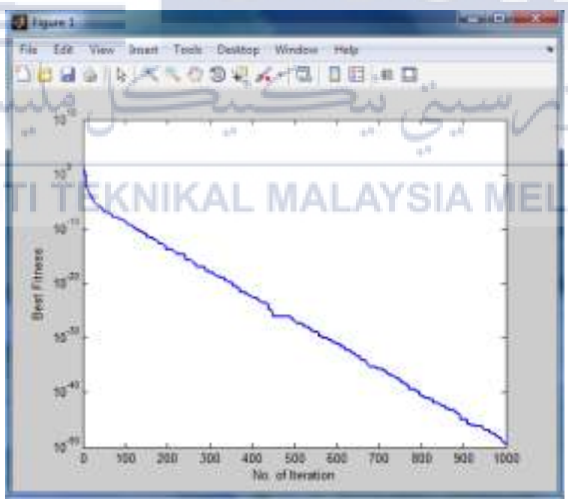


(a)

Parameters: MaxIt = 1000

VarMax = 20

nScoutBee = 10



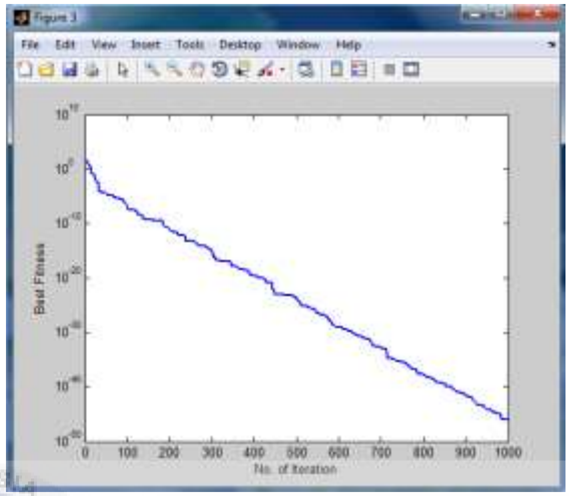
(b)

Figure 6.6: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm

Parameters: MaxIt = 1000

VarMax = 10

nScoutBee = 100

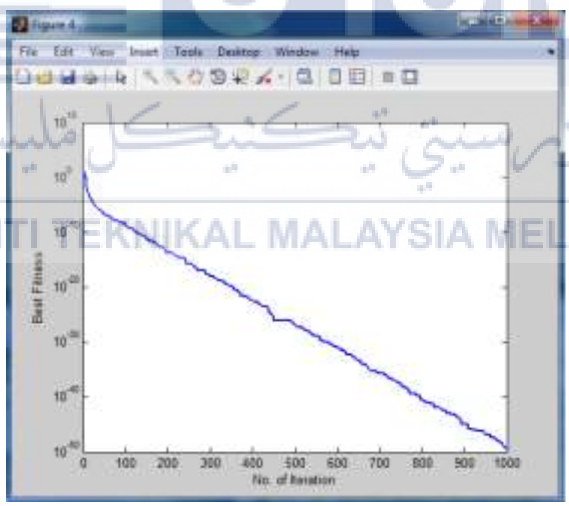


(a)

Parameters: MaxIt = 1000

VarMax = 20

nScoutBee = 100



(b)

Figure 6.7: (a) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using default parameters of Bees Algorithm (b) Graph Iris Segmentation based on Best Fitness vs. Number of Iteration using modified parameters of Bees Algorithm

Figure 6.2(a), 6.3(a), 6.4(a), 6.5(a), 6.6(a) and 6.7(a) showed the graph of best fitness for Iris Segmentation using default parameters of Bees Algorithm after the completed process of segmentation and iteration in order to test the performance of the technique. Parameters of Bees Algorithm are important in order to test and analysis the performance and accuracy of the technique in Iris Recognition system.

In the interim, for figure 6.2(b), 6.3(b), 6.4(b), 6.5(b), 6.6(b) and 6.7(b) it demonstrated the chart of best wellness for Iris Segmentation utilizing altered parameters of Bees Algorithm after the finished procedure of division and cycle with a specific end goal to test the execution of the system. Parameters of Bees Algorithm Technique that has been changed are the lower bound and upper bound to locate the best limit, likewise the quantity of scout honey bees itself. This been acclimated to test the extent of populace and the zone of iris seeking.

After plotted the diagram, it demonstrated that adjusted parameters have turned out to be better in term of precision and execution contrasted with default parameters of Bees Algorithm. From the outcome, it additionally demonstrates that adjusted parameters are speedier in location contrasted with default parameters that has been utilized.

Table 6.1(a): Result for calculation of Best Fitness by using default parameter Bees Algorithm

Method	Parameter		Max No. of Iteration	No. of Scout Bees									
				10	20	30	40	50	60	70	80	90	100
Bees Algorithm	VarMin	-10	100	9.3806E-07	3.1339E-08	3.1618E-08	8.072E-10	6.5832E-09	6.7646E-09	3.4791E-09	3.014E-09	1.6874E-09	1.589E-09
	VarMax	10	200	1.91E-11	1.25E-12	1.21E-12	8.63E-13	2.62E-13	6.28E-14	1.12E-13	1.37E-13	7.94E-14	5.42E-14
	nSelectedSite	round (0.5*nScoutBee)	300	2.94E-16	1.08E-16	1.09E-17	2.99E-17	5.32E-18	1.12E-17	4.24E-18	2.95E-18	1.09E-18	1.17E-18
	nEliteSite	round (0.4*nSelectedSite)	400	6.81E-21	4.91E-21	5.06E-22	7.27E-22	9.11E-23	1.14E-22	1.66E-22	5.02E-23	2.03E-23	4.67E-23
	nSelectedSiteBee	round (0.5*nScoutBee)	500	9.2447E-25	4.5749E-26	3.9829E-26	7.9144E-27	4.9894E-27	1.0766E-26	3.5266E-27	2.8703E-27	2.6309E-27	2.2025E-27
	nEliteSiteBee	2*nSelectedSiteBee	600	4.07E-29	1.96E-30	2.01E-30	8.26E-31	6.74E-31	4.14E-31	2.71E-31	1.07E-31	9.12E-32	4.07E-32
	r	0.1* (VarMax-VarMin)	700	1.15E-33	1.48E-34	5.24E-35	3.16E-35	1.43E-35	6.64E-36	7.80E-36	1.40E-36	3.28E-36	2.41E-36
	rdamp	0.95	800	1.99E-38	3.06E-40	3.58E-40	9.26E-40	4.86E-40	1.04E-40	7.79E-41	7.79E-41	1.04E-40	4.96E-41
			900	6.87E-43	1.80E-43	6.25E-44	2.41E-44	1.33E-44	5.53E-45	3.12E-45	4.34E-45	3.66E-45	7.31E-46
			1000	8.63E-47	5.50E-48	3.50E-48	1.69E-48	3.80E-49	3.88E-49	4.39E-49	9.07E-50	1.19E-49	4.64E-50

Table 6.1(b): Result for calculation of Best Fitness by using modified parameter Bees Algorithm

Method	Parameter		Max No. of Iteration	No. of Scout Bees									
				10	20	30	40	50	60	70	80	90	100
Bees Algorithm	VarMin	-10	100	4.39E-06	6.79E-08	2.63E-08	3.39E-09	3.80E-09	9.22E-09	4.55E-09	2.35E-08	3.89E-10	2.98E-09
	VarMax	20	200	2.17E-11	1.02E-11	3.27E-12	8.75E-13	1.17E-12	3.13E-13	5.60E-13	1.26E-13	1.85E-13	7.48E-14
	nSelectedSite	round (0.5*nScoutBee)	300	1.65E-15	1.39E-16	3.42E-17	3.69E-17	2.84E-17	1.13E-18	1.82E-17	1.09E-17	3.55E-18	2.99E-18
	nEliteSite	round (0.4*nSelectedSite)	400	4.73E-20	5.80E-22	9.26E-22	1.61E-21	7.48E-22	1.54E-22	2.65E-22	1.66E-22	6.66E-23	9.08E-23
	nSelectedSiteBee	round (0.5*nScoutBee)	500	1.93E-24	1.20E-26	3.30E-26	3.63E-26	2.69E-26	1.65E-26	1.94E-26	2.69E-27	2.25E-27	9.05E-27
	nEliteSiteBee	2*nSelectedSiteBee	600	5.87E-30	1.27E-29	6.91E-31	1.79E-30	6.38E-31	6.10E-31	4.94E-31	5.67E-31	2.74E-31	8.88E-32
	r	0.1* (VarMax-VarMin)	700	4.01E-33	4.51E-34	4.16E-35	4.13E-35	2.20E-35	1.53E-35	1.73E-35	1.42E-35	7.35E-36	2.82E-36
	rdamp	0.95	800	1.39E-37	3.37E-39	3.73E-39	1.54E-39	1.57E-39	1.74E-39	4.33E-40	2.05E-40	1.77E-40	1.41E-40
			900	3.60E-42	3.26E-44	5.62E-44	4.88E-44	1.88E-44	3.84E-44	1.03E-44	5.16E-45	5.25E-45	1.58E-44
			1000	8.63E-47	5.50E-48	3.50E-48	1.69E-48	3.80E-49	3.88E-49	4.39E-49	4.39E-49	1.19E-49	4.64E-50

Table 6.1(a) showed the result calculation for the Best Fitness using default Bees Algorithm. This calculation has been done for 100 times for each size of number scout bees, which is from 10 until 100. This calculation is tabled from the graph result of iris image that completed segmentation process using Bees Algorithm. For Table 6.1(b), the parameters of Bees Algorithm have been modified and adjust to test and analysis the performance, accuracy and speed of the technique. These modified parameters have been tested 100 times for each number of scout bees from 10 to 100. At 1000 number of iteration, both parameters showed the same reading. This is because it had over the fitting data. Furthermore, the reading and result showed that modified parameters are better in performance, accuracy and speed compared to default parameters.

Table 6.2(a): Success rate and time for iteration by using default parameter of Bees Algorithm on left iris image

Iris Image	Max No. of Iteration	Success Rate (%)	Mean CPU (s)	Min CPU (s)	Max CPU (s)	SD CPU (s)
S1019L01	100	3	0.012684s	0.005791s	0.028410s	0.000997s
	500	37	0.046462s	0.004883s	0.092047s	0.016898s
	1000	40	0.079780s	0.005624s	0.145326s	0.042215s

Table 6.2(b): Success rate and time for iteration by using default parameter of Bees Algorithm on right iris image

Iris Image	Max No. of Iteration	Success Rate (%)	Mean CPU (s)	Min CPU (s)	Max CPU (s)	SD CPU (s)
S1019R01	100	3	0.012518s	0.005994s	0.187884s	0.005644s
	500	37	0.044052s	0.004863s	0.069366s	0.016007s
	1000	40	0.077568s	0.005473s	0.165804s	0.040962s

Table 6.3(a): Success rate and time for iteration by using modified parameter of Bees Algorithm on left iris image

Iris Image	Max No. of Iteration	Success Rate (%)	Mean CPU (s)	Min CPU (s)	Max CPU (s)	SD CPU (s)
S1019L01	100	4	0.012611 s	0.006801 s	0.026323 s	0.001258s
	500	38	0.044534 s	0.004237 s	0.065315 s	0.017502s
	1000	45	0.074496 s	0.004527 s	0.124008 s	0.043909s

Table 6.3(b): Success rate and time for iteration by using modified parameter of Bees Algorithm on right iris image

Iris Image	Max No. of Iteration	Success Rate (%)	Mean CPU (s)	Min CPU (s)	Max CPU (s)	SD CPU (s)
S1019R01	100	4	0.012666 s	0.006804 s	0.036064 s	0.001371s
	500	38	0.044514 s	0.004298 s	0.063618 s	0.017491s
	1000	45	0.075582 s	0.004476 s	0.219450 s	0.044764s

Table 6.2(a) and (b) demonstrate the consequence of progress rate and times for an iris pictures after finished the division procedure utilizing Bee Algorithm with default parameters. While, for Table 6.3 (an) and (b) demonstrate the aftereffect of achievement rate and times for the same iris pictures after finished the division utilizing Bee Algorithm with adjusted parameters.

Table 6.4: Comparison of means and standard deviation for various techniques in iris biometrics

Method	No. of Iterations	Population Size/ No. of Scout Bees		
		20	40	60
PSO	100	1.39 × 10^1	9.97 × 10^−1	5.56 × 10^−2
		(5.81 × 10^1)	(2.12 × 10^−1)	(3.11 × 10^−2)
	500	3.11 × 10^−1	5.75 × 10−1	2.28 × 10−2
		(2.08 × 10^−1)	(4.37 × 10−1)	(4.45 × 10−2)
	1000	3.75 × 10^−1	3.75 × 10^−1	7.88 × 10^−3
		(2.81 × 10^−1)	(2.81 × 10^−1)	(4.34 × 10^−3)
GA	100	1.42 × 10^−1	7.22 × 10^−2	8.22 × 10^−2
		(2.20 × 10^−1)	(3.12 × 10^−2)	(4.43 × 10^−2)
	500	7.72 × 10−1	4.75 × 10−2	5.58 × 10−2
		(3.12 × 10−1)	(4.01 × 10−2)	(4.00 × 10−2)
	1000	3.40 × 10−1	2.12 × 10−2	7.71 × 10−3
		(5.11 × 10−1)	(3.36 × 10−2)	(2.09 × 10−3)
ABC	100	5.13 × 10−1	5.21 × 10−2	4.47 × 10−2
		(3.75 × 10−1)	(4.10 × 10−2)	(3.25 × 10−2)
	500	3.39 × 10−2	3.55 × 10−2	1.10 × 10−2
		(2.99 × 10−2)	(2.22 × 10−2)	(2.79 × 10−2)
	1000	1.01 × 10−2	1.25 × 10−2	7.26 × 10−3
		(2.95 × 10−2)	(2.57 × 10−2)	(3.77 × 10−3)
LEPSO	100	5.71 × 10−2	8.91 × 10−3	6.01 × 10−3
		(1.12 × 10−2)	(1.12 × 10−3)	(3.81 × 10−3)
	500	7.28 × 10−2	4.40 × 10−3	4.56 × 10−3
		(5.09 × 10−2)	(2.32 × 10−3)	(3.35 × 10−3)
	1000	3.78 × 10−3	1.01 × 10−3	8.21 × 10−4
		(2.01 × 10−2)	(1.78 × 10−3)	(5.55 × 10−4)
ABCDE	100	3.12 × 10−3	4.41 × 10−4	7.22 × 10−5
		(2.08 × 10−3)	(3.77 × 10−4)	(5.51 × 10−5)
	500	1.33 × 10−3	1.91 × 10−4	3.01 × 10−5
		(5.18 × 10−3)	(2.61 × 10−4)	(1.22 × 10−5)
	1000	9.01 × 10−4	9.44 × 10−5	5.56 × 10−6
		(4.22 × 10−4)	(3.17 × 10−5)	(1.01 × 10−6)
DEBCO	100	1.71 × 10−5	2.95 × 10−6	4.91 × 10−8
		(1.85 × 10−5)	(1.10 × 10−6)	(1.01 × 10−8)
	500	2.13 × 10−6	8.99 × 10−7	9.07 × 10−9
		(4.55 × 10−6)	(3.21 × 10−7)	(3.81 × 10−9)
	1000	1.88 × 10−7	3.35 × 10−8	5.77 × 10−9
		(3.35 × 10−7)	(1.23 × 10−8)	(2.71 × 10−9)
BA	100	7.26867 x 10-1	5.42845 x 10-1	6.61932 x 10-1
		3.573734	**2.601019**	**3.204368**
	500	2.12473 x 10-1	1.5203206 x 10-1	4.9192 x 10-2
		2.039999	**1.68353151**	**0.6296**
	1000	1.59887 x 10-1	1.04483 x 10-1	2.2047 x 10-2
		2.164114	**1.666123**	**0.3751165**

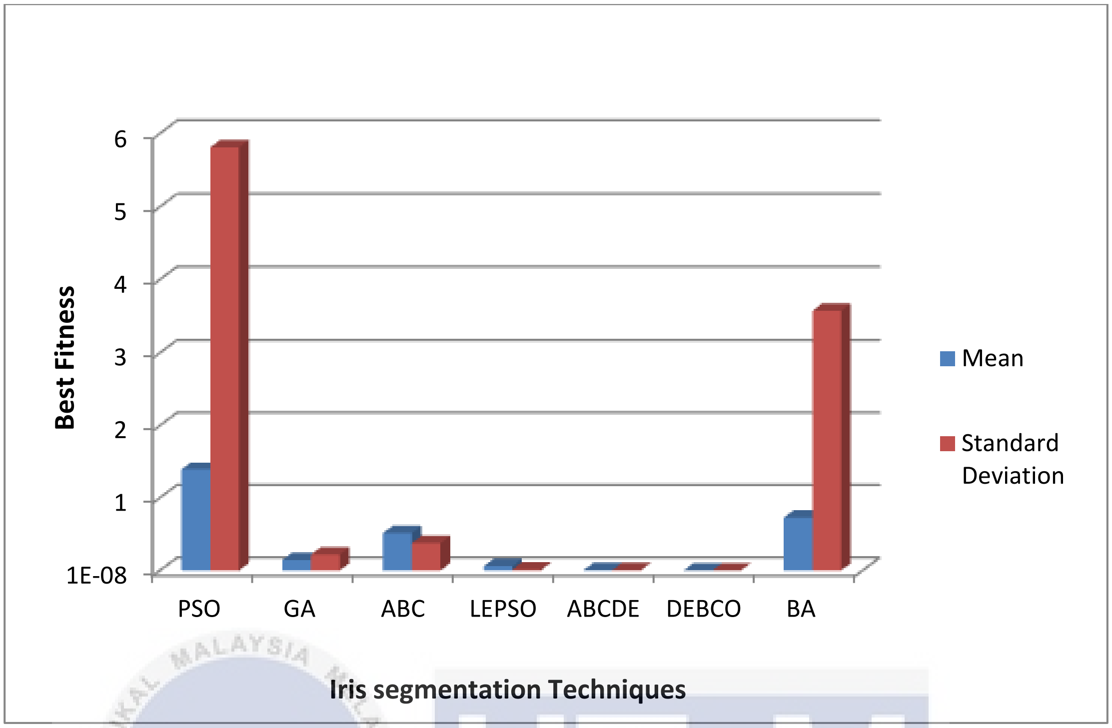


Figure 6.8: Graph for means and standard deviation for various techniques in Iris Biometrics

From Table (6.4) and figure (6.8), there are different strategies of Iris Recognition System that have been tried in Image Based Segmentation to examine their execution towards cycle number and wellness. As expressed in figure (6.8), Particle Swarm Optimization (PSO) procedure demonstrates the most noteworthy perusing for the standard deviation of wellness took after by Bee Algorithm (BA), Artificial Bee Colony (ABC) and Genetic Algorithm (GA). While, for the LEPSO, ABCDE and DEBCO the estimation of means and standard deviation are little. This is on the grounds that these three methods are half and half systems. While for PSO, BA, ABC and GA are standalone systems that require high number of cycle to locate the ideal arrangements. The quantity of emphasis additionally influence the season of looking the ideal arrangements. Despite the fact that mean and standard deviation are high yet it relies on upon the parameter settings for every strategy. Then again, if the mean and standard deviation are low, the pace of seeking the iris components is quicker and more vigorous.

From the result, it showed that the accuracy, performance and speed are influenced by a couple factors. There are iris images and noise type.

Iris Images

The demand for higher accuracy and high speed recognition in biometric system leads to continuous proposals of a new iris recognition method since the arising of impostor users in the biometric system. High error rate in iris texture means the iris contains more noise, which is called as noisy iris.

There are two reasons for noisy iris to occur;

a) The iris texture
b) The technical factor

The first source of noisy iris comes from the iris texture itself that caused constant change in its structure which is due to;

1) Human biological condition
2) The occlusion from eyelids and eyelashes

Every human has different body condition and health. The iris texture constantly changes in slow motion based on human condition, such as aging, growth, emotion, and diet also eye surgery. In fact, the color of the iris texture may change due to inheritance and epigenetic diversity from different races. The ever-changing iris texture creates difficulties at the comparison phase to determine either the captured iris data are genuine or not.

The second source of noisy iris appears from technical factor in capturing the iris feature image. Some of the sources of noises are from the camera itself and the surrounding environment during the capturing process, such as indoor or outdoor, distance between the camera and the eyes (short/remote) or source of light (near infrared/LED) and many more. Other than that, the blurred and low quality images of iris texture which is captured using 'off-angle' configuration, also add noise to the iris image.

6.3 Conclusion

To summarize this analysis, in this project there are two main factors that affect the result of iris recognition. Next chapter which is final chapter is 5conclude the overall about this project. Furthermore, the limitations of this project are elaborate properly in the chapter.

CHAPTER VII

PROJECT CONCLUSION

7.1 Introduction

In previous chapter, the testing and analysis of test results had been done. This chapter is discusses about project conclusion that conclude the whole project analysis and the achievement of objectives. This paper is the closing chapter for this project that includes the research contributions, research limitations and future works for this research.

7.2 Project Summarization

After conducting this research, the first objective (PO1) is to study the circular method of iris segmentation in biometrics. From the past researches that have been done and from the papers and journals gained, the history, the conceptual and how the segmentation is been implements are understood. Aided from MATLAB is also an important aspect here. The first objective is achieved.

The second project objective (PO2) is to modify the purpose of Bees Algorithm in Iris Biometrics has been achieved in Chapter 5 by using different parameters to test the Bees Algorithm technique on iris images. This is to see directly how the segmentation on iris image been done when the Bees Algorithm been apply. This process is done in MATLAB, where the iris images been captured, run, analyze, code and tested to see the performance. In this objective, it is important in identify the parameter and code that been used and modified to test the ability and it’s performance on new iris images.

To implement the circular segmentation process using Bee Algorithm with Raspberry Pi in network environment is the last objective for this project research (PO3). Because of the limitation space and small micro processing of Raspberry Pi, the whole process of iris recognition system can’t be embedded into the Raspberry Pi system. As an alternative way, the purpose of new prototype in Iris Biometrics involving Raspberry Pi and MATLAB are develop in order to run the segmentation process using Bee Algorithm.

7.3 Project Contribution

Table 7.1: Summary of Project Contributions

PS	PQ	PO	PC	**Project Contribution**
PS1	PQ1	PO1	PC1	To study the iris feature segmentation method in Bees Algorithm technique for Iris Biometrics.
PS2	PQ2	PO2	PC2	To analyze the performance of Bees Algorithm technique on circular segmentation method in Iris Biometrics.
PS3	PQ3	PO3	PC3	The purpose of new prototype in iris biometrics in network environment.

Every person and individuals is getting the benefits from this research to generate their ideas on doing more deep research on this biometrics technology especially on Circular Segmentation in Iris Biometrics, also on Bees Algorithms technique itself. Besides the research, the prototype development also can be implementing throughout this project that may be uses as a startup on developing a more complex prototype or a full biometric system to software developer or engineer. With such contributions, this surely gives something to individual(s) and also to organization(s) out there.

7.4 Project Limitation

During this project, surely there are few of limitations that occurred. However, the major restrains are as follows:

1. The movement and process of iteration and segmentation are not smooth
2. Iris image that been captured contains high noise
3. The segmentation process need to run part by part
4. The segmentation process can't be embedded into Raspberry Pi
   (This is because of limitation space and small micro processing of Raspberry Pi)

7.5 Future Works

In order to improve this project, there are several things that need be done. Firstly, the parameters of Bees Algorithms technique need to be analyze and modified more to improve the segmentations process and quality of the results. Next, the iris images contains high noisy and blurriness that need more features to reduce and clear the noise and blurriness in order to get a better result. Lastly, bigger space and more stable microprocessor of embedded system in Raspberry Pi are needed to run and test iris recognition system in network environment.

7.6 Conclusion

In a nutshell, this project has meets with all three objectives prescribed in the beginning. Throughout the process of developing this project, more knowledge on biometrics especially in Iris Biometrics has been gained. This is very useful in future with hoping that it be uses in real working environment.

References

[1] Daugman, J.: How iris recognition works. IEEE Trans. Circ. Syst. Video Tech. **14**(1), 21-30 (2004). Doi: 10.1109/ICIP.2002.1037952

[2] Daugman, J.: Two-dimensional spectral analysis of cortical receptive field profiles. Vis. Res. **20**(10), 847-856 (1980). doi: 10.1016/0042-6989(80)90065-6

[3] Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Am. A Opt. Image Sci. Vis. **2**(7), 1160-1169 (1985). doi: 10.1364/JOSAA.2.001160

Abidin, Z.Z., 2012. Iris Segmentation Analysis using Integro-Differential Operator and Hough Transform in Biometric System. , 4(2), pp.41–48.

Akay, B. & Karaboga, D., 2012. A modified Artificial Bee Colony algorithm for real-parameter optimization. *Information Sciences*, 192, pp.120–142. Available at: http://dx.doi.org/10.1016/j.ins.2010.07.015.

Balasubramani, K. & Marcus, K., 2013. A Comprehensive review of Artificial Bee Colony Algorithm. *International Journal of Computers & Technology*, 5(1), pp.15–28. Available at: http://cirworld.org/journals/index.php/ijct/article/view/1199.

Bouaziz, A., Draa, A. & Chikhi, S., 2015. Arti fi cial bees for multilevel thresholding of iris images. *Swarm and Evolutionary Computation*, 21, pp.32–40. Available at: http://dx.doi.org/10.1016/j.swevo.2014.12.002.

Koh, J.K.J., Govindaraju, V. & Chaudhary, V., 2010. A Robust Iris Localization Method Using an Active Contour Model and Hough Transform. *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp.2852–2856.

Mahlouji, M. & Noruzi, A., 2012. Human Iris Segmentation for Iris Recognition in Unconstrained Environments. , 9(1), pp.149–155.

Ali, E.M., Ahmed, E.S. & Ali, A.F., 2007. Recognition of human iris patterns for biometric identification. *Journal of Engineering and Applied Science*, 54(6), pp.635–651.

Bansal, J. et al., 2013. Memetic search in artificial bee colony algorithm. *Soft Computing*, 5(2), pp.1–18. Available at: http://dx.doi.org/10.1007/s00500-013-1032-8..

Daugman, J., 2007. New methods in iris recognition. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 37(5), pp.1167–1175.

K, M.S.T. & M, M.J.K., 2013. A Literature Review on Iris Segmentation Techniques for Iris Recognition Systems. , 11(1), pp.46–50.

Karaboga, D. & Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing Journal*, 8(1), pp.687–697.

Kıran, M.S. & Fındık, O., 2015. A directed artificial bee colony algorithm. *Applied Soft Computing*, 26, pp.454–462. Available at: http://www.sciencedirect.com/science/article/pii/S1568494614005304.

Koh, J.K.J., Govindaraju, V. & Chaudhary, V., 2010. A Robust Iris Localization Method Using an Active Contour Model and Hough Transform. *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp.2852–2856.

Min, T.H. & Park, R.H., 2008. Comparison of eyelid and eyelash detection algorithms for performance improvement of iris recognition. *Proceedings - International Conference on Image Processing, ICIP*, pp.257–260.

Nabti, M. & Bouridane, A., 2008. An effective and fast iris recognition system based on a combined multiscale feature extraction technique. *Pattern Recognition*, 41(3), pp.868–879.

Ozturk, C., Hancer, E. & Karaboga, D., 2015. A novel binary artificial bee colony algorithm based on genetic operators. *Information Sciences*, 297, pp.154–170. Available at: http://dx.doi.org/10.1016/j.ins.2014.10.060.

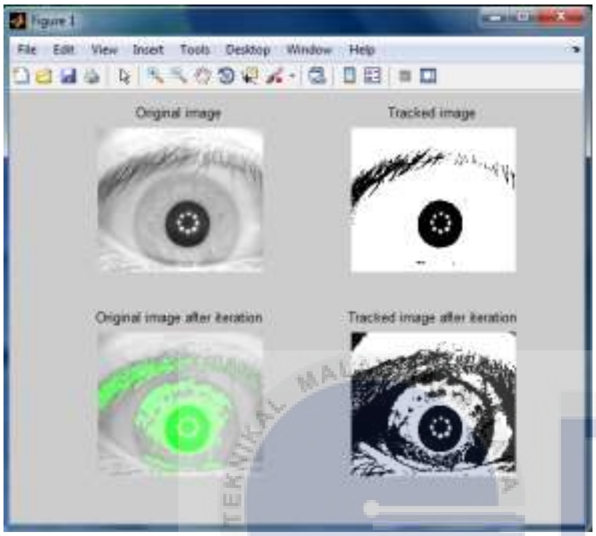
Pawar, M.K., Student, P.G. & Jhajjav, G., 2012. Iris Segmentation using Geodesic Active Contour for Improved Texture Extraction in Recognition. *International Journal*, 47(16), pp.40–47.

Yahya, A.E. & Nordin, M.J., 2010. Accurate iris segmentation method for non-cooperative iris recognition system. *Journal of Computer Science*, 6(5), pp.492–497.

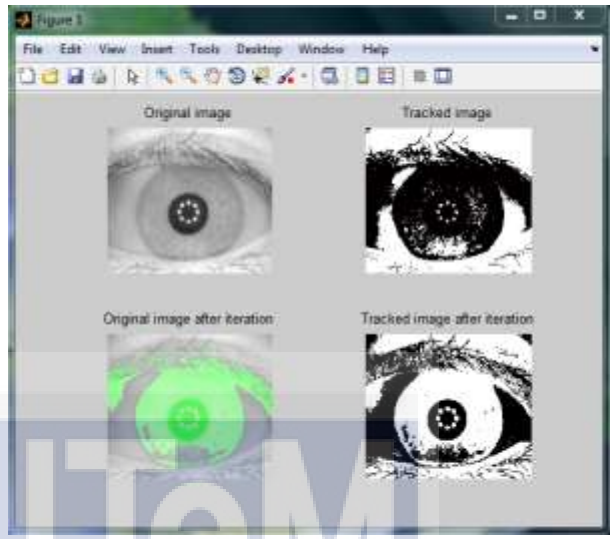
Zhai, Y.K. et al., 2010. A Novel Iris Recognition Method Based on the Contourlet Transform and Biomimetic Pattern Recognition Algorithm. *2010 Ieee 10th International Conference on Signal Processing Proceedings (Icsp2010), Vols I-Iii*, pp.1390–1393. Available at: <Go to ISI>://000296986200344.

Appendix A: Result of Iris Segmentation using Bees Algorithm

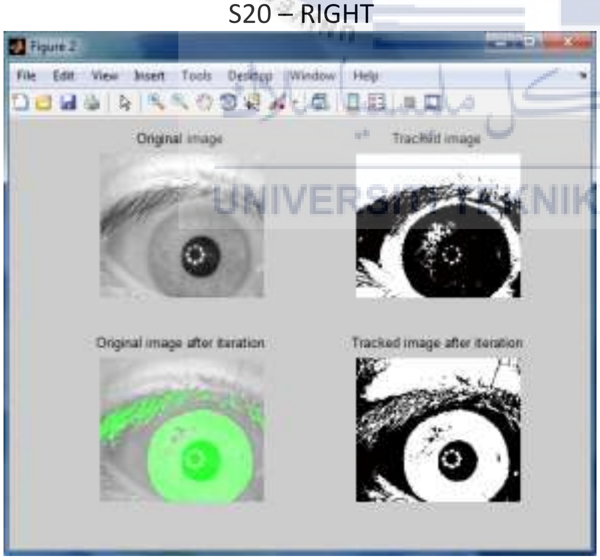
S19 – RIGHT



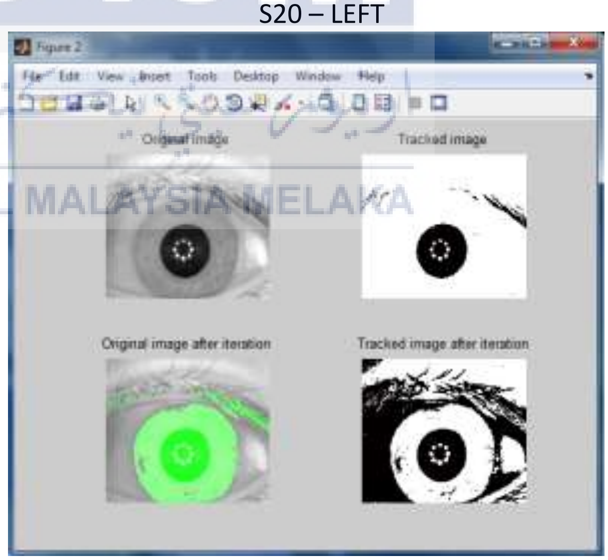
S19 – LEFT



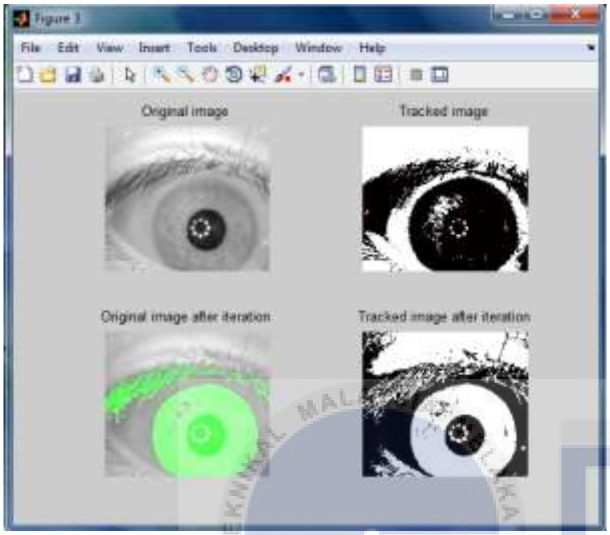
S20 – RIGHT



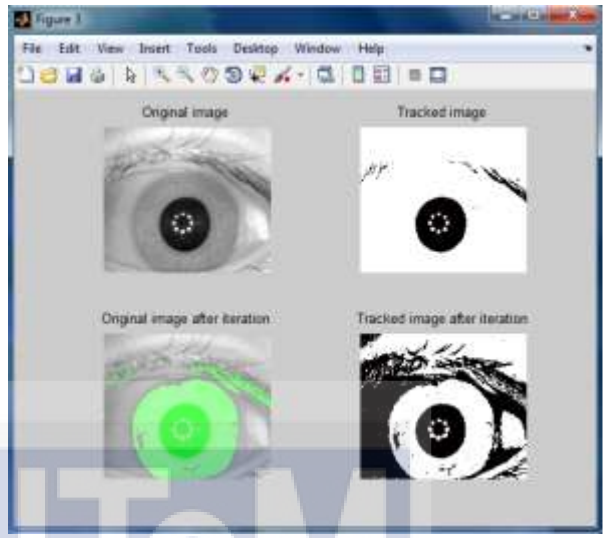
S20 – LEFT



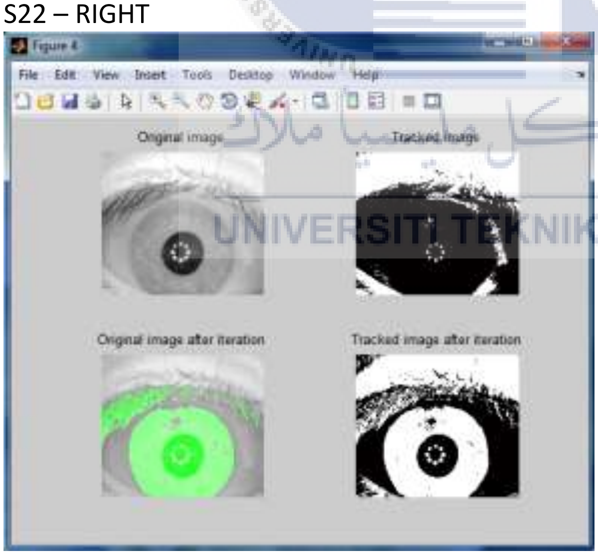
S21 – RIGHT



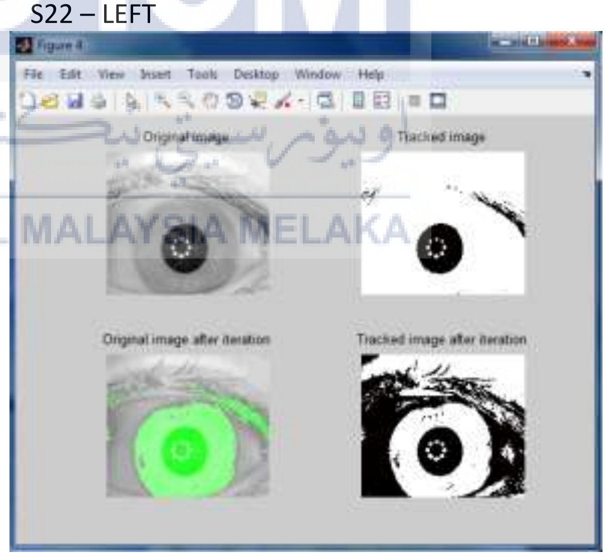
S21 – LEFT



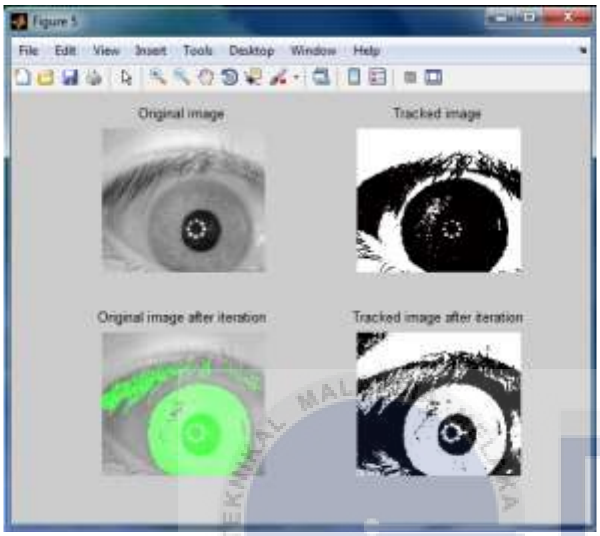
S22 – RIGHT



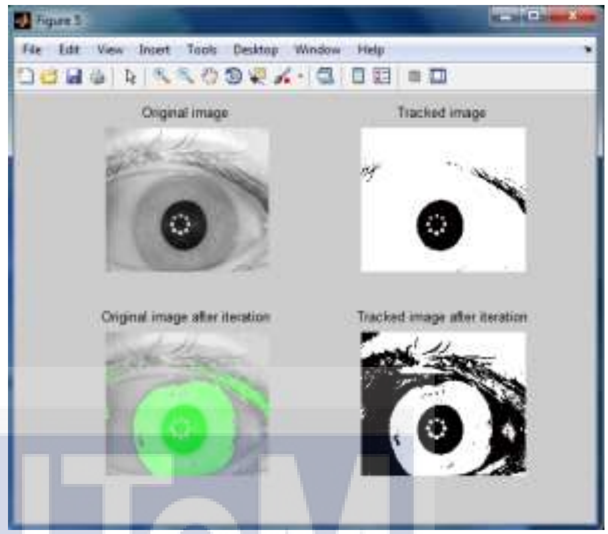
S22 – LEFT



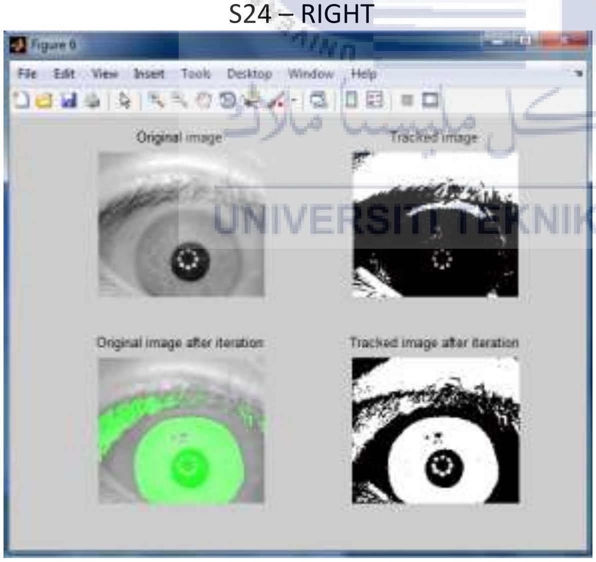
S23 – RIGHT



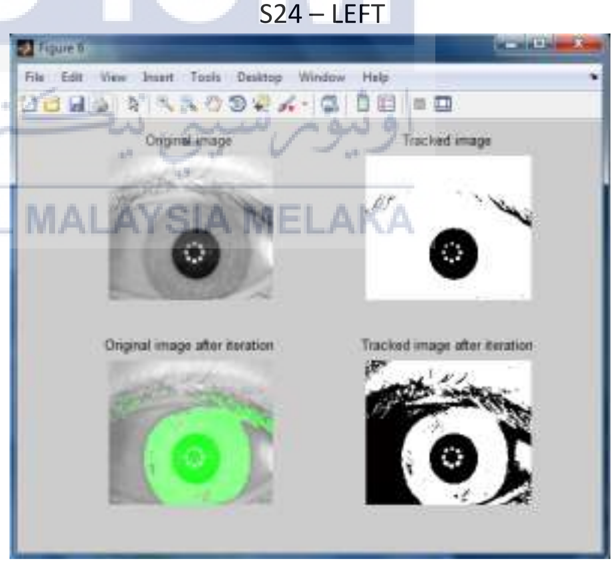
S23 – LEFT



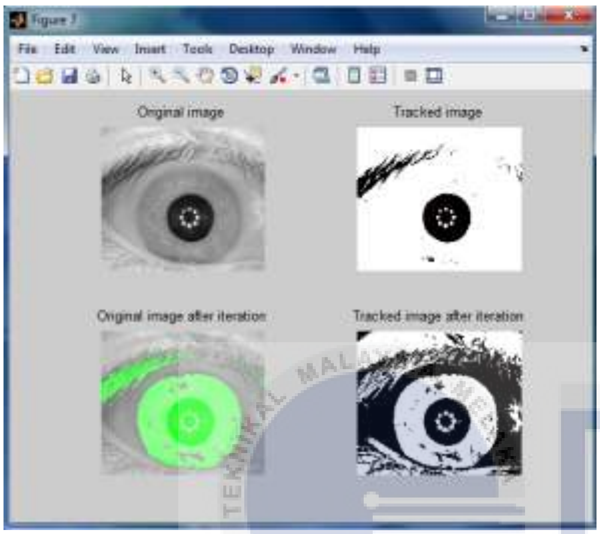
S24 – RIGHT



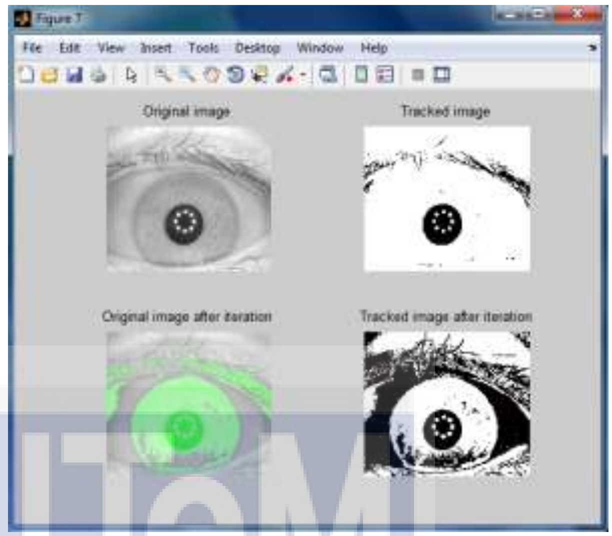
S24 – LEFT



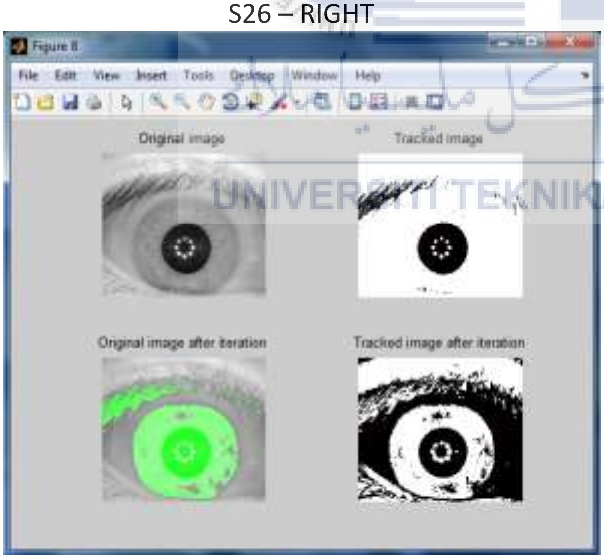
S25 – RIGHT



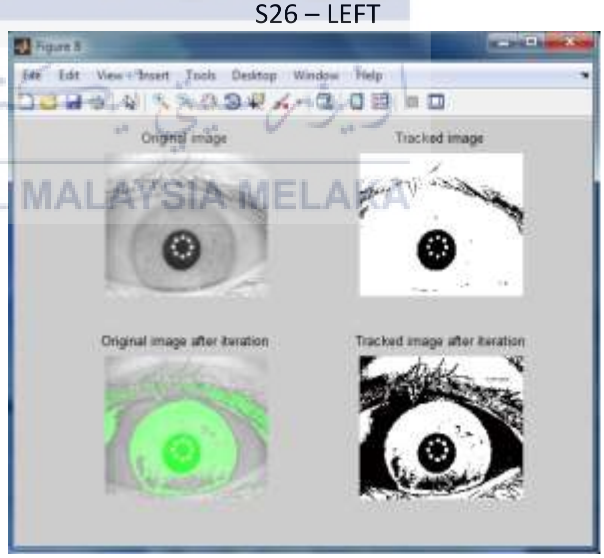
S24 – LEFT



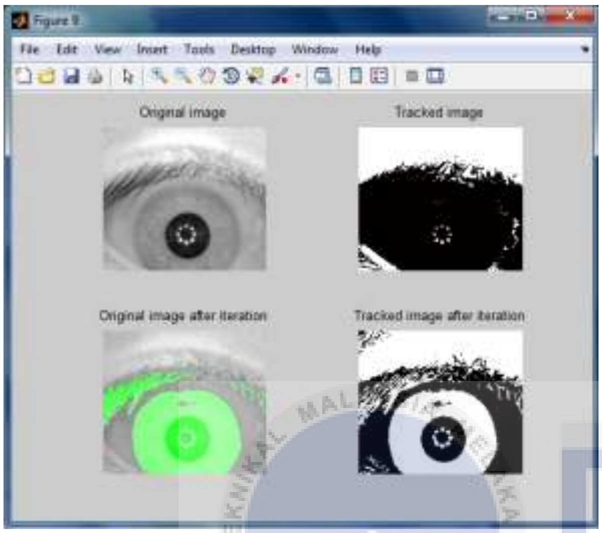
S26 – RIGHT



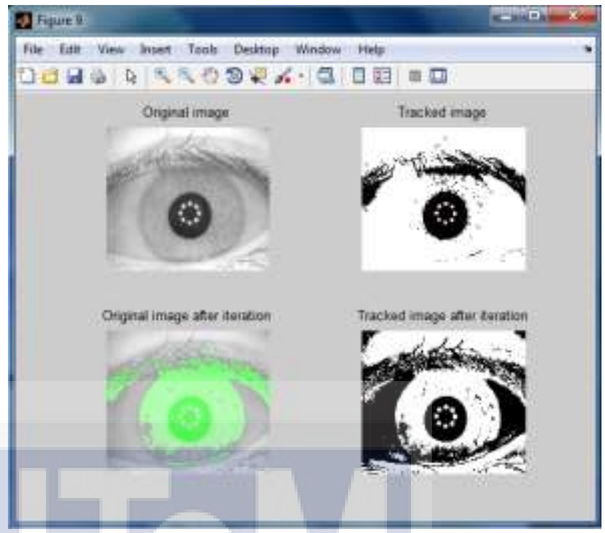
S26 – LEFT



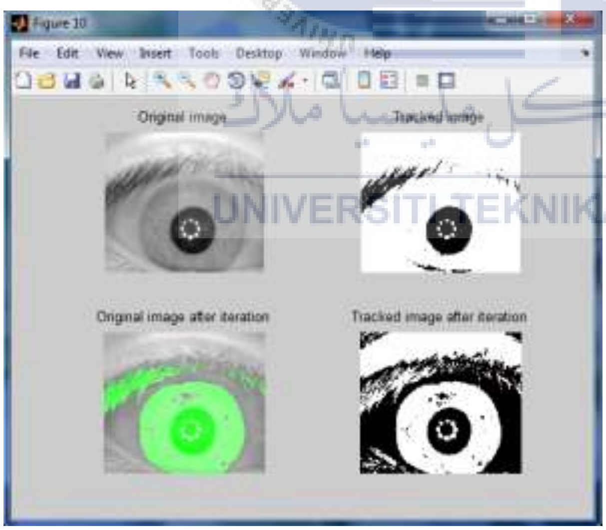
S27 – RIGHT



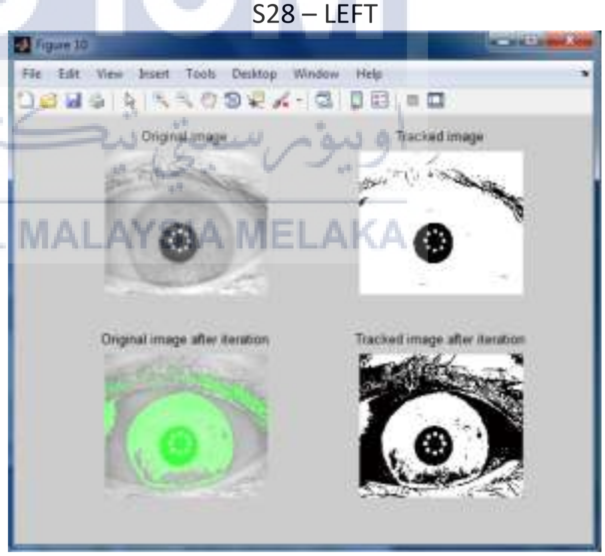
S27 – LEFT



S28 – RIGHT



S28 – LEFT



Appendix B: Source Code

```
% SITI NUR ADHA BT AZIZAN
% ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

%Read in example file
im = imread('Left.jpg');
figure(1);
subplot(2,2,1);
imshow(im);
title('Original image 1');

%Locate codes using the default values
subplot(2,2,2);
codes = locateCodes(im)
title('Tracked image 1');


%Read in second example file
im2 = imread('try.jpg');
subplot(2,2,3);
imshow(im2);
title('Original image 2');

%Locate codes in the image using some manual input values instead of
%defaults
subplot(2,2,4);
codes2 = locateCodes(im2, 'colMode', 1, 'thresh', 0.2)
title('Tracked image 2');
```



```
function R = locateCodes(im, varargin)

%
v = strcmp('vis', varargin);

if sum(v) == 0
    vis = 1;
else
    vis = cell2mat(varargin(find(v == 1) + 1));
end



%
colM = strcmp('colMode', varargin);

if sum(colM) == 0
    colMode = 0;
else
    colMode = cell2mat(varargin(find(colM == 1) + 1));
```

```
end


%
tagTh = strcmp('sizeThresh', varargin);

if sum(tagTh) == 0
    sizeThresh = 100;
else
    sizeThresh = cell2mat(varargin(find(tagTh == 1) + 1));
end


%
threshM = strcmp('threshMode', varargin);

if sum(threshM) == 0
    threshMode = 0;
else
    threshMode = cell2mat(varargin(find(threshM == 1) + 1));
end


%
bradleyP = strcmp('bradleyFilterSize', varargin);

if sum(bradleyP) == 0
    smP = [15 15];
else
    smP = cell2mat(varargin(find(bradleyP == 1) + 1));
end


%
bradleyT = strcmp('bradleyThreshold', varargin);
if sum(bradleyT) == 0
    brT = 3;
else
    brT = cell2mat(varargin(find(bradleyT == 1) + 1));
end


%
if ndims(im) > 2
    GRAY = rgb2gray(im);
elseif ndims(im) == 2
    GRAY = im;
end


%
th = strcmp('thresh', varargin);
```

```
if sum(th) == 0
    thresh=graythresh(GRAY);
else
    thresh = cell2mat(varargin(find(th == 1) + 1));
end


%
if threshMode == 0
    BW=im2bw(GRAY, thresh);
elseif threshMode  == 1
    BW = bradley(GRAY, smP, brT);
end


%
if colMode == 1 && vis == 1
    imshow(GRAY);
end

if colMode == 0 && vis== 1
    imshow(BW);
end

if colMode == 2 && vis == 1
    imshow(im);
end


%
trackM = strcmp('robustTrack', varargin);

if sum(trackM) == 0
    trackMode = 0;
else
    trackMode = 1;
    imo = cell2mat(varargin(find(trackM == 1) + 1));
end


%
listM = strcmp('tagList', varargin);

if sum(listM) == 0
    listMode = 0;
    validTagList = [];
else
    listMode = 1;
    validTagList = cell2mat(varargin(find(listM == 1) + 1));
end


%
cornerSize = 10;
```

```
%%
R = regionprops(BW, 'Centroid','Area','BoundingBox','FilledImage');

%%

if numel(sizeThresh) == 1

    R = R([R.Area] > sizeThresh);

elseif numel(sizeThresh) == 2

    R =  R([R.Area] > sizeThresh(1) & [R.Area] < sizeThresh(2));

else

    disp('sizeThresh has an incorrect numbers of elements: Please supply
either a single number or a two-element numeric vector');
    return;

end

if isempty(R)

    disp('No sufficiently large what regions detected - try changing
thresholding values for binary image threshold (thresh) or tag size
(sizeThresh)');
    return

end

%%
for i = 1:numel(R)

    try
        warning('off', 'all');
        [isq,cnr] = fitquad( R(i).BoundingBox, R(i).FilledImage);
        warning('on', 'all');
        R(i).isQuad = isq;

    catch

        R(i).isQuad = 0;
        continue

    end

    if isq

        R(i).corners = cnr;

    end
end
```

```
R = R(logical([R.isQuad]));

%%

if isempty(R)

    disp('No potentially valid tag regions found')
    return

end

for i=1:numel(R)

    corners = R(i).corners;
    cornersP = [corners(2,:) ;corners(1,:)];
    tform = maketform('projective', cornersP',[ 0 0;  1 0;  1  1;  0 1]);
    udata = [0 1];  vdata = [0 1];

    hold on

    for bb = 1:4

        if vis ==1
            plot(cornersP(1,bb), cornersP(2,bb),'g.', 'MarkerSize',
cornerSize)
        end

    end

    %
    x = [5.5/7 4.5/7 3.5/7 2.5/7 1.5/7];
    xp = [repmat(x(1), 5, 1) ;repmat(x(2), 5, 1);repmat(x(3), 5,
1);repmat(x(4), 5, 1);repmat(x(5), 5, 1)];
    P = [xp  repmat(x,1,5)'];
    f = [ 0 0;  0 1;  1  1;  1 0];
    pts = tforminv(tform,P);
    pts = round(pts);
    R(i).pts = pts;

    hold on;


    %
    ptvals = [];

    for aa = 1:numel(pts(:,1))

        cur = pts(aa,:);
        cur = fliplr(cur);

        try
```

```
            ptvals(aa) = BW(cur(1),cur(2));
            %
        catch

            continue

        end

    end


    %
    if numel(ptvals) == 25

        if trackMode == 0

            code =
[ptvals(1:5);ptvals(6:10);ptvals(11:15);ptvals(16:20);ptvals(21:25)];
            code = fliplr(code);
            [pass code orientation] = checkOrs25(code);
            %
            R(i).passCode = pass;
            R(i).code = code;
            R(i).orientation = orientation;

        elseif trackMode == 1

            [pass code orientation] = permissiveCodeTracking(imo, pts);
            R(i).passCode = pass;
            R(i).code = code;
            R(i).orientation = orientation;

        end

    else
        R(i).passCode = 0;
        R(i).code = [];
        R(i).orientation = NaN;
    end




end


%%
R = R([R.passCode]==1);


%
for i=1:numel(R)
    %%
```

```
    R(i).number = bin2dec(num2str(R(i).code(1:15)));

    %
    corners = R(i).corners;
    cornersP = [corners(2,:) ;corners(1,:)];
    tform = maketform('projective', cornersP',[ 0 0;  1 0;  1  1;  0 1]);
    udata = [0 1];  vdata = [0 1];



    %%
    or = R(i).orientation;
    if or == 1
        ind = [1 2];
    elseif or == 2
        ind = [2 3];
    elseif or ==3
        ind = [3 4];
    elseif or ==4
        ind = [1 4];
    end

    frontX = mean(cornersP(1,ind));
    frontY = mean(cornersP(2,ind));

    R(i).frontX = frontX;
    R(i).frontY = frontY;
    %
end

%%

if ~isempty(validTagList)

    R = R(ismember([R.number], validTagList));

    if isempty(R);
        disp('No Valid Tags Found');
    end

end



%%

if vis==1
    for i = 1:numel(R)
        corners = R(i).corners;
        cornersP = [corners(2,:) ;corners(1,:)];
        text(R(i).Centroid(1), R(i).Centroid(2), num2str(R(i).number),
'FontSize',30, 'color','r');
        hold on
        for bb = 1:4
            plot(cornersP(1,bb), cornersP(2,bb),'g.', 'MarkerSize',
cornerSize)
```

```
        end

        plot(R(i).frontX, R(i).frontY, 'b.', 'MarkerSize', cornerSize);
    end
end

R = rmfield(R, {'FilledImage', 'isQuad', 'passCode'});
hold off;
%%
```

```
% SITI NUR ADHA BT AZIZAN
% ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS

clc;
clear;
close all;

%% Problem Definition

CostFunction=@(x) Sphere(x);

nVar=5;

VarSize=[1 nVar];

VarMin=-10;
VarMax= 10;

%% Bees Algorithm Parameters

MaxIt=1000;

nScoutBee=100;

nSelectedSite=round(0.5*nScoutBee);

nEliteSite=round(0.4*nSelectedSite);

nSelectedSiteBee=round(0.5*nScoutBee);

nEliteSiteBee=2*nSelectedSiteBee;

r=0.1*(VarMax-VarMin);

rdamp=0.95;

%% Initialization

%
empty_bee.Position=[];
empty_bee.Cost=[];

%
bee=repmat(empty_bee,nScoutBee,1);

%
for i=1:nScoutBee
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
    bee(i).Cost=CostFunction(bee(i).Position);
```

```
end

%
[~, SortOrder]=sort([bee.Cost]);
bee=bee(SortOrder);

%
BestSol=bee(1);

%
BestCost=zeros(MaxIt,1);

%% Bees Algorithm Main Loop

for it=1:MaxIt

    %
    for i=1:nEliteSite

        bestnewbee.Cost=inf;

        for j=1:nEliteSiteBee
            newbee.Position=PerformBeeDance(bee(i).Position,r);
            newbee.Cost=CostFunction(newbee.Position);
            if newbee.Cost<bestnewbee.Cost
                bestnewbee=newbee;
            end
        end

        if bestnewbee.Cost<bee(i).Cost
            bee(i)=bestnewbee;
        end

    end

    %
    for i=nEliteSite+1:nSelectedSite

        bestnewbee.Cost=inf;

        for j=1:nSelectedSiteBee
            newbee.Position=PerformBeeDance(bee(i).Position,r);
            newbee.Cost=CostFunction(newbee.Position);
            if newbee.Cost<bestnewbee.Cost
                bestnewbee=newbee;
            end
        end

        if bestnewbee.Cost<bee(i).Cost
            bee(i)=bestnewbee;
        end

    end
```

```
    %
    for i=nSelectedSite+1:nScoutBee
        bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
        bee(i).Cost=CostFunction(bee(i).Position);
    end

    %
    [~, SortOrder]=sort([bee.Cost]);
    bee=bee(SortOrder);

    %
    BestSol=bee(1);

    %
    BestCost(it)=BestSol.Cost;

    %
    disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);

    %
    r=r*rdamp;

end

%% Results

figure;
%plot(BestCost,'LineWidth',2);
semilogy(BestCost,'LineWidth',2);
xlabel('No. of Iteration');
ylabel('Best Fitness');
```



```
% SITI NUR ADHA BT AZIZAN
% ANALYSIS OF BEES ALGORITHM ON CIRCULAR SEGMENTATION IN IRIS BIOMETRICS
%
%------------------------------------------------------------------------

function seg =
localized_seg(I,init_mask,max_its,rad,alpha,method,FigRefreshRate,display)

  %
  if(~exist('alpha','var'))
    alpha = .2;
  end
  %
  if(~exist('method','var'))
    method = 2;
  end
  %
  if(~exist('display','var'))
    display = true;
  end
```

```
 if(~exist('FigRefreshRate','var'))
   FigRefreshRate =20;
end
%
I = im2graydouble(I);
%
[dimy dimx] = size(I);
if(~exist('rad','var'))
  rad = round((dimy+dimx)/(2*8));
  if(display>0)
    disp(['localiztion radius is: ' num2str(rad) ' pixels']);
  end
end

%
phi = mask2phi(init_mask);

%--main loop
for its = 1:max_its

  %
  idx = find(phi <= 1.2 & phi >= -1.2)';
  [y x] = ind2sub(size(phi),idx);

  %
  xneg = x-rad; xpos = x+rad;
  yneg = y-rad; ypos = y+rad;
  xneg(xneg<1)=1; yneg(yneg<1)=1;
  xpos(xpos>dimx)=dimx; ypos(ypos>dimy)=dimy;

  %
  u=zeros(size(idx)); v=zeros(size(idx));
  Ain=zeros(size(idx)); Aout=zeros(size(idx));

  %
  for i = 1:numel(idx)
    img = I(yneg(i):ypos(i),xneg(i):xpos(i));
    P = phi(yneg(i):ypos(i),xneg(i):xpos(i));

    upts = find(P<=0);
    Ain(i) = length(upts)+eps;
    u(i) = sum(img(upts))/Ain(i);

    vpts = find(P>0);
    Aout(i) = length(vpts)+eps;
    v(i) = sum(img(vpts))/Aout(i);
  end

  %
  switch method
   case 1,
    F = -(u-v).*(2.*I(idx)-u-v);
   otherwise,
    F = -((u-v).*((I(idx)-u)./Ain+(I(idx)-v)./Aout));
  end
```

```
    %
    curvature = get_curvature(phi,idx,x,y);

    %
    dphidt = F./max(abs(F)) + alpha*curvature;

    %
    dt = .45/(max(dphidt)+eps);

    %
    phi(idx) = phi(idx) + dt.*dphidt;

    %
    phi = sussman(phi, .5);

    %
    if((display>0)&&(mod(its,FigRefreshRate) == 0))
      showCurveAndPhi(I,phi,its);
    end
  end

  %
  if(display)
    showCurveAndPhi(I,phi,its);
  end

  %
  seg = phi<=0;

%----------------------------------------------------------------------
%----------------------------------------------------------------------
%----------------------------------------------------------------------
%----------------------------------------------------------------------

function showCurveAndPhi(I, phi, i)
  imshow(I,[]); hold on;
  contour(phi, [0 0], 'g','LineWidth',4);
  contour(phi, [0 0], 'k','LineWidth',2);
  title(['Localized Region Based Active Contour Segmentation ',num2str(i) '
Iterations']); hold off;drawnow;

%
function phi = mask2phi(init_a)
  phi=bwdist(init_a)-bwdist(1-init_a)+im2double(init_a)-.5;

%
function curvature = get_curvature(phi,idx,x,y)
    [dimy, dimx] = size(phi);

    %
    ym1 = y-1; xm1 = x-1; yp1 = y+1; xp1 = x+1;

    %
```

```
    ym1(ym1<1) = 1; xm1(xm1<1) = 1;
    yp1(yp1>dimy)=dimy; xp1(xp1>dimx) = dimx;

    %
    idup = sub2ind(size(phi),yp1,x);
    iddn = sub2ind(size(phi),ym1,x);
    idlt = sub2ind(size(phi),y,xm1);
    idrt = sub2ind(size(phi),y,xp1);
    idul = sub2ind(size(phi),yp1,xm1);
    idur = sub2ind(size(phi),yp1,xp1);
    iddl = sub2ind(size(phi),ym1,xm1);
    iddr = sub2ind(size(phi),ym1,xp1);

    %
    phi_x  = -phi(idlt)+phi(idrt);
    phi_y  = -phi(iddn)+phi(idup);
    phi_xx = phi(idlt)-2*phi(idx)+phi(idrt);
    phi_yy = phi(iddn)-2*phi(idx)+phi(idup);
    phi_xy = -0.25*phi(iddl)-0.25*phi(idur)...
             +0.25*phi(iddr)+0.25*phi(idul);
    phi_x2 = phi_x.^2;
    phi_y2 = phi_y.^2;

    %
    curvature = ((phi_x2.*phi_yy + phi_y2.*phi_xx -
2*phi_x.*phi_y.*phi_xy)./...
              (phi_x2 + phi_y2 +eps).^(3/2)).*(phi_x2 + phi_y2).^(1/2);

%
function img = im2graydouble(img)
  [dimy, dimx, c] = size(img);
  if(isfloat(img))
    if(c==3)
      img = rgb2gray(uint8(img));
    end
  else
    if(c==3)
      img = rgb2gray(img);
    end
    img = double(img);
  end

%
function D = sussman(D, dt)
  %
  a = D - shiftR(D);
  b = shiftL(D) - D;
  c = D - shiftD(D);
  d = shiftU(D) - D;

  a_p = a;  a_n = a;
  b_p = b;  b_n = b;
  c_p = c;  c_n = c;
  d_p = d;  d_n = d;
```

```
  a_p(a < 0) = 0;
  a_n(a > 0) = 0;
  b_p(b < 0) = 0;
  b_n(b > 0) = 0;
  c_p(c < 0) = 0;
  c_n(c > 0) = 0;
  d_p(d < 0) = 0;
  d_n(d > 0) = 0;

  dD = zeros(size(D));
  D_neg_ind = find(D < 0);
  D_pos_ind = find(D > 0);
  dD(D_pos_ind) = sqrt(max(a_p(D_pos_ind).^2, b_n(D_pos_ind).^2) ...
                     + max(c_p(D_pos_ind).^2, d_n(D_pos_ind).^2)) - 1;
  dD(D_neg_ind) = sqrt(max(a_n(D_neg_ind).^2, b_p(D_neg_ind).^2) ...
                     + max(c_n(D_neg_ind).^2, d_p(D_neg_ind).^2)) - 1;

  D = D - dt .* sussman_sign(D) .* dD;

%
function shift = shiftD(M)
  shift = shiftR(M')';

function shift = shiftL(M)
  shift = [ M(:,2:size(M,2)) M(:,size(M,2)) ];

function shift = shiftR(M)
  shift = [ M(:,1) M(:,1:size(M,2)-1) ];

function shift = shiftU(M)
  shift = shiftL(M')';

function S = sussman_sign(D)
  S = D ./ sqrt(D.^2 + 1);
```

```
function [xbest,fbest,Imp,success,CPUtime] = ...
    harmony_stats(f,xL,xU,fmin,tol,HSparams)

BA     = BAparams.BA;
MaxImp = HSparams.MaxImp;
HMCR   = HSparams.HMCR;
PAR    = HSparams.PAR;
b      = HSparams.b;

%
N    = length(xL);
HM   = zeros(HMS,N);
F    = zeros(HMS,1);
xnew = zeros(1,N);

for j = 1:HMS
    HM(j,:) = xL + (xU-xL).*rand(1,N);
    F(j)    = f(HM(j,:));
end
success = false;

for j = 1:MaxImp


    for i = 1:N

        if rand < HMCR

            xnew(i) = HM(ceil(rand*HMS),i);
            if rand < PAR


                xnew(i) = xnew(i) + (2*rand-1)*b(i);
                xnew(i) = min(max(xnew(i),xL(i)),xU(i));
            end
        else

            xnew(i) = xL(i) + rand*(xU(i)-xL(i));
        end
    end



    fnew = f(xnew);
    [fworst,idxworst] = max(F);
    if fnew < fworst
        HM(idxworst,:) = xnew;
        F(idxworst)    = fnew;
    end



    [fbest,idxbest] = min(F);
    if abs(fbest-fmin) <= tol
        success = true;
        break
```

```
    end

end


xbest = HM(idxbest,:);

Imp = j;
CPUtime = toc;
```

```
%
f = @test2;
xL = [-10 -10];
xU = [10 10];

%
HSparams.HMS = 10;
HSparams.MaxImp = 100;
HSparams.HMCR = 0.5;
HSparams.PAR = 0.4;
HSparams.b = (xU-xL)/1000;

%
fmin = 0;
tol = 1e-6;

%
Nrep    = 1000;
NFE     = zeros(Nrep,1);
success = zeros(Nrep,1);
CPUtime = zeros(Nrep,1);

%
for i=1:Nrep
    [xbest,fbest,NFE(i),success(i),CPUtime(i)] ...
        = harmony_stats(f,xL,xU,fmin,tol,HSparams);
end

%
NFE_valid = NFE; NFE_valid(~success) = [];
CPUtime_valid = CPUtime; CPUtime(~success) = [];

%
fprintf('Success rate = %.0f%%\n', mean(success)*100)
fprintf('Mean NFE = %.0f; min NFE = %.0f; max NFE = %.0f; SD NFE = %.0f\n',
...
    mean(NFE_valid), min(NFE_valid), max(NFE_valid), std(NFE_valid))
fprintf('Mean CPU = %fs; min CPU = %fs; max CPU = %fs; SD CPU = %fs\n', ...
    mean(CPUtime_valid), min(CPUtime_valid), max(CPUtime_valid),
std(CPUtime_valid))
```