PERFORMANCE EVALUATION OF FOUR DISTANCE CLASSIFIERS IN ANT COLONY OPTIMIZATION-BASED GABOR FEATURES FOR FACIAL RECOGNITION

By

ARO, TAYE OLADELE
(99/55EK027)
B.Sc. (2003), M.Sc. (2007) (Unilorin)

A Thesis Submitted to the Department of Computer Science, Faculty of Communication and Information Sciences in Partial Fulfilment of the Requirements for the Award of the Degree of Doctor of Philosophy in Computer Science

APRIL, 2018

CERTIFCATION

This is to certify that this thesis has been read and approved as meeting the requirements of the Department of Computer Science, University of Ilorin, Ilorin, Nigeria for the Award of Doctor of Philosophy degree in Computer Science.

Dr. Oluwakemi C. Abikoye (Supervisor)	Date
Dr. D. R. Aremu (Head of Department)	Date
Dr. Tinuke O. Oladele (Postgraduate Coordinator)	Date
(External Examiner)	Date

DEDICATION

This research is dedicated to Almighty God (the most faithful heavenly father who can never fail in all his promises) and to my father, Late Mr. Ekundayo Samuel Aro.

DECLARATION

I, ARO, TAYE OLADELE with Matriculation Number 99/55EK027 hereby declare that this thesis entitled Performance Evaluation of Four Distance Classifiers in Ant Colony Optimization-Based Gabor Features for Facial Recognition is a record of my research work. It has neither been presented nor accepted in any previous application for higher degree. All sources of information have been specifically acknowledged.

In addition, the research work has been ethically approved by the University Ethical Review Committee

----------------------------------	---------------------------
Aro T. O	Date

AKNOWLEDGEMENTS

First and foremost, I give glory, praise and adoration to Almighty God for this rare opportunity bestowed upon me to carry out and also complete this study. I would like to express my depth-felt gratitude to my dynamic supervisor Dr Oluwakemi C. Abikoye for her patience, motivation, enthusiasm, excellent guidance and constant constructive criticism rendered in order to give this study a worthy output. It was a great privilege and honour to work and study under her supervision. I could not have imagined having a better supervisor and mentor for my Ph.D. study. My appreciation also goes to Dr. D. R. Aremu, the Head of Department of Computer Science, for all his efforts and encouragements during the course of the study. I am extremely grateful to Professor B. A. Oluwade, who reviewed several drafts of chapters of my thesis and provided assistance and valuable comments on my technical writing

I acknowledge the incessant contribution of my wonderful friend Dr. A. O. Bajeh towards the success of this study. I am grateful to the entire lecturing staff of Computer Science Department for their full support and advice: Dr. R. G. Jimoh, Dr. P. B. Shola, Dr. A. O. Babatunde, Dr. R. O. Oladele, Dr. O. A. Ameen, Dr. (Mrs). T. O. Oladele, Dr. (Mrs) A. G. Akintola, Dr. (Mrs). M. A. Mabayoje, Dr. A. R. Ajiboye, Dr. (Mrs) G. B. Balogun, Dr. (Mrs) F. E. Usman-Hamza, Mr. A. O. Balogun, Mr. P. Sadiku, Mr. H. A. Mojeed, Mrs. S. A. Saliu and all non-academic staff.

Special thanks to Dr. A. L. Azeez, Department of Mass Communication, University of Ilorin, for his advice and adequate guidance during the period of this study. I am also indebted to the following academic staff of University of Ilorin for their support and concern: Prof. A. T. Ande, Dr. O. D. Owolabi, Dr. A. Tella, Dr. O. B. Akinola and Dr. O. A. Iyiola.

I won't forget to appreciate all my family members, most especially my mother, Mrs F. O. Aro for her prayers and the great work done to give me quality educational background in life. Mum, may Almighty God bless you, I extend my thanks to my loving and caring parents, Mr & Mrs R. O Alabi, for their ceaseless prayers, sacrifices and contributions to the success of this research. I am also thankful to my following brothers and sisters: Engr. Olaolu A. Alabi, Mr. Babatunde Aro, Mr. Oluwashola Alabi, Mr. Ayotunde Aro, Mr & Mrs Ajao, Mr & Mrs Aina, Mr. Bukunmi Alabi and Inspector Seun Alabi.

Finally, in-depth gratitude to my darling wife, Mrs Odunola Mary Aro, and amazing children, Araoluwa Faith Aro, Anuoluwakiitan Emmanuel Aro and Oluwadarasimi Joshua Aro, for their understanding, tolerance and unflinching supports throughout the time of this research. I say a big thank you to all the people who have supported me either directly or indirectly to get this study accomplished

ABSTRACT

In face recognition system, several techniques have been proposed for extraction of facial features such as Local Binary Pattern, Gabor-filters, Elastic Bunch Graph Matching, Principal Component Analysis and Hidden Markov Models. Gabor-filters, among other feature extraction techniques, remain a powerful and useful tool in image processing. Its optimal functionality as feature extraction in face recognition is traceable to its biological importance and computational properties. In spite of all the distinctive characteristics of Gabor-filters, this technique suffers high feature dimensionality. This study therefore aimed at reducing the high dimensionality of Gabor features. The objectives were to: (i) extract facial features using Gabor-filters; (ii) optimize the Gabor features extracted with Ant Colony Optimization (ACO); (iii) perform facial image matching with the use of some selected distance classifiers; Chebysev, City-block, Mahalanobis and Euclidean; and (iv) evaluate the performance based on classification accuracy, classification time, sensitivity, specificity and error rate.

The facial features was extracted using Gabor-filters with 5 scales and 8 orientations, then the extracted features were optimized by applying ACO on the Gabor features to obtain the optimal features. The optimized features was passed into selected distance classifiers. The performance evaluation of the proposed system was done using two face image datasets; Locally Acquired Face Image Database (LAFI) and Olivetti Research Laboratory Database (ORL).

The findings of the study were that:

(i) gabor feature vectors were obtained for face image representation;
(ii) optimal features with relevant and discriminant information were produced;
(iii) the optimized features performed efficiently with some selected distance classifiers;
(iv) the best classification accuracy of 97.14% was obtained in Mahanolobis of image size (150x150) for LAFI Database, while classification accuracy of 95.71% was achieved in Mahanolobis (150x150), Euclidean (150x150), City-block (75x75, 100x100, 150x150) for ORL database;
(v) reduced classification time of 0.42507secs was obtained in Mahanolobis (125x125) for LAFI Database and 0.40422secs was obtained in Mahalanobis (125x125) for ORL Database;
(vi) the best sensitivity of 98.33% was obtained in Mahanolobis image size of (150x150), City-block (125x125) for LAFI, while the same percentage of 98.33% in Euclidean (150x150) for ORL;
(vii) the best specificity of 90% was achieved in Mahanolobis image size of (150x150), Euclidean (75x75, 100x100, 125x125), Chebysev (75x75, 125x125) and City-block (75x75); and
(viii) the best error rate of 2.86% was achieved in Mahanolobis of image size 150x150 for LAFI Database and 4.29% was obtained in Mahanolobis (150x150), Euclidean (150x150) and City-block (75x75, 100x100) for ORL database.

The study concluded that the high dimensionality of Gabor features was well reduced and optimized by Ant Colony Optimization Algorithm. The performance of optimized Gabor features with the selected distance classifiers recorded better experimental

results. Thus, the study recommended ACO as an effective feature optimization method for Gabor-features based face recognition system.

TABLE OF CONTENTS

CHAPTER THREE: RESEARCH METHODOLOGY

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS AND NOTATIONS

ACO:	Ant Colony Optimization
ABCO	Artificial Bee Colony Optimization
ACOGF:	Ant Colony Optimization Gabor Features
BCO	Bat Colony Optimization
DCT:	Discrete Cosine Transform
DR	Dimensionality Reduction
DWT:	Discrete Wavelet Transform
DNA:	Deoxyribonucleic Acid
FE:	Feature Extraction
FS:	Feature Extraction
LAFI	Locally Acquired Face Image
LBP:	Local Binary Pattern
LDA:	Linear Discriminant Analysis
HMM:	Hidden Markov Model

MD	Manhattan Distance
ORL	Olivetti Research Laboratory
PCA:	Principal Component Analysis
PIN:	Personal Identification Number
PSO	Particle Swarm Optimization
SIA	Swarm Intelligence Algorithm

CHAPTER ONE

INTRODUCTION

The chapter starts with discussion on some fundamental concepts and principles use as background to the study. Also, it mentions the problem statement, aim and specific objectives of the study, significance of the study, scope of the study. Finally, the chapter gives definition of some operational terms and thesis layout.

1.1 Background to the Study

As electronic transactions increase, information and digital data become more predominant in the society. In the past, people have used different techniques to secure their vital documents on computer systems with methods such as highly encrypted passwords, access codes and personal identification numbers (PINs) (Shende & Sarode, 2016). All these techniques are conventional methods of authentication that can either be possession based techniques (token like keys, cards or badges) or knowledge based techniques (access to physical or virtual domain based on password and PINs) (Jafri & Arabnia, 2009). These methods are subject to several fundamental flaws. They are many a times difficult to remember, easily guessed or forcefully possessed by unauthorized users. The security breaches of these approaches showed that they are unable to satisfy requirements for security of information in technological society. The several draw backs in these techniques led to an authentication based on individual distinctive biological characteristics measurement known as biometrics.

Biometrics is an automated method that involves the recognition of an individual based on a feature vector derived from individual biological traits. Biometric has become the most recent promising technique of recognition (Parmar & Mehta, 2013). Face recognition is a physiological approach of biometrics for verifying and identifying a person from digital image or video using dataset of face images (Joshi & Deshpande, 2015). The recognition of a face image is achieved by comparing selected facial features from database of stored face images in order to identify the input image (Barbu, 2010). This biometric technology is one of the most successfully representative applications in computer vision that has received a great interest in commercial and law enforcement domains such as human-computer interaction, access control, digital libraries, telecommunication and security systems (Shen & Bai, 2006; Jin & Ruan, 2009; Haider, Bashir, Sharif, Sharif & Wahab, 2014).

Facial recognition is user friendly; generally acceptable biometric recognition technique with high accuracy and low intrusiveness (Kashem, Akhter, Ahmed, & Alam, 2011). The face recognition system compared with other biometric methods does not require the complete cooperation of the test subject to work. A properly designed facial recognition systems installed in public environments, such as seaports, and airports can identify individuals among the crowd unaware of the existence of the system.

Other biometrics such as fingerprint, iris, retina, vein, Deoxyribonucleic acid (DNA), odour and voice recognition cannot perform this kind of mass identification. Biometric techniques that engage different people using the same particular equipment with direct contact, such as finger print, hand vein and palm print to capture their biometric data, indirectly expose the user to transmission of germs from other users. Face recognition approach is completely less intrusive and does not bear any health hazards (Bakshi & Singhal, 2014).

One of the main challenges in facial recognition algorithm is how to describe and extract accurately the features for facial image representation (Tan & Triggs, 2007). In pattern recognition system, such as face recognition, the extraction of features is considered to be difficult problem because best features need to be extracted to have minimized classification error and execution time (Shah, Sharif, Raza & Azeem, 2013). The feature extraction method involves the formation of new set of features from original facial image simply to reduce the feature measurement cost, increase classifier efficiency, and allow higher recognition accuracy (Chitra & Balakrishnan, 2012). The most essential and unique features are extracted from localized image in feature extraction phase (Kaur & Rajput, 2013). Feature extraction represents the most important phase of face recognition due to the direct dependency of accuracy of any face recognition model on the level of extracted features from face region (Shebani, 2015).

A number of algorithms have been employed for facial feature extraction such as Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), Independent Component Analysis (ICA), Local Binary Pattern (LBP), Hidden Markov Model (HMM) Elastic Bunch Graph Matching (EBGM), Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) (Jin & Ruan, 2009). Gabor-filter approach among other algorithms has gained much attention in image processing and pattern recognition (Bhuiyan & Liu, 2007). This technique remains a powerful and useful tool in image coding and image processing (Hafez, Selim, & Zayed, 2015). Its optimal functionality in face recognition is linked to its biological importance (

similarity to the receptive fields of simple cells in primary visual cortex) and computational properties (optimal for calculating local spatial frequencies) (Shen & Ji, 2009). Gabor filters have the ability of obtaining multi-orientation features from a facial image at several scales. This technique has achieved great success and considered as one of the best technique for face representation (Bouzalmat, Belghini, Zarghili, Kharroubi & Majda, 2011; Bhele & Mankar, 2012). This technique has been used in several applications like texture segmentation, edge detection, face recognition and discrimination in image processing (Grigorescu, Petkov & Kruizinga, 2002; Bellakhdhar, Loukil, & Abid, 2013; Kaur, Kaur & Randhawa, 2014). Among various local feature representations, Gabor-filters have been recognized as one of the most successful local feature extraction methods due to their robustness against distortions caused by variations in scale, translation and rotation (Bouzalmat et al., 2011).

It has been established that Gabor-filters have the ability to extract maximum information from local image regions (Deng, Jin, Zhen & Huang, 2005; Mounika & Reddy, 2012). Gabor-filters representation of face images exhibit salient properties such as spatial localization, spatial frequency and orientation selectivity (Dai & Qian, 2004). The discriminative features extracted from face images using Gabor-filters is roust against expression and illumination changes (Elgarrai, Meslouhi, & Allali, 2014). Gabor-filtered images are capable of capturing relevant frequency spectrum in order to extract features aligned at specified orientations to recognition of Region of Interest (ROI).

Despite the several overwhelming achievements of Gabor-features method in recognizing face images with different pose, illumination and expression (Shen & Bai, 2006). This technique suffers high feature dimension (Hafez et al., 2015), the dimension of the feature vectors extracted by applying Gabor-filters to the whole face image through convolution is very high (Jin & Ruan, 2009). The dimensionality of the input face images is usually very large and constitutes difficulties when performing classification on the original images. The general approach when employing Gabor filters for recognition of face involves the construction of filter bank with filters of different scales and orientations to filter a given face image with all filters from the bank. Thus, this approach makes the dimensionality of Gabor features very high and resulting to high computational complexity (Vinay, Shekhar, Murthy & Natarajan, 2015).

In pattern recognition, orthogonality remains the most important component. However Gabor-filters of different filters from the filter bank are not orthogonal to one another (Struc & Pavesic, 2009). Hence, this makes the final information contained in this technique to be redundant, and this further affects the classification accuracy of any classifier used on Gabor features for recognition.

In this study, a meta-heuristic optimization algorithm, which is also a population-based optimization algorithm, was introduced for the feature selection (Deng et al., 2005; Kumar, Shaikh, Katwate, & Jamdar, 2014). Ant Colony Optimization is an iterative probabilistic meta-heuristic algorithm and a nature inspired computational methods (Dorigo & Blum, 2005). This algorithm with a distinguishing feature of constructive random search space procedure using an indirect memory which is referred to as pheromone was employed to get the optimal features from Gabor feature dimensions before classification process (Al-ani, 2007). Finally, some selected distance measure classifiers; Euclidean, City block, Mahanolobis and Chebysev were applied on the computed optimized Gabor features for classification.

1.2 Statement of the Problem

Finding robust features that represent face image is identified to be a difficult task due to dimensionality problem (high feature dimension) (Pal, Chourasia & Ahirwar 2013). A face image is generally represented by huge dimensional feature vectors containing image pixel values that summarize the underlying content of a local region using local representation (Zhen, 2013). In computer vision, raw image data cannot be used directly owing to high dimensionality. The high feature dimension problems include consumption of large storage space, long computational time and misclassification (Gandhe, Talele & Keskar, 2007). Gabor-filters is a widely used technique for facial feature extraction (Shen & Bai, 2006), but the representation of a given face image using a bank of Gabor filters with different orientations and scales leads into an explosion of the image pixel space of the original face image dimensions.

Some dimensionality reduction methods only perform the function of feature extraction. They produce the linear combination of original image features without considering the irrelevancy and redundancy of features. The inclusion of irrelevant and redundant features increases the size of search space which as a result gives a prolong recognition time and makes generalization more difficult (Liu & Yu, 2005).

Several dimensionality reduction approaches such as down sampling, feature selection techniques, reduction of filter bank parameters and subspace projection methods have been applied to reduce computational complexity of Gabor feature into low dimension before being passed into a classifier (Struc, Gajsek & Pavesic, 2009). The down sampling technique involves the use of only selected feature points, but the final output consists of high number of image feature matrix which could cause the loss of feature distinct information and may eventually affect classification accuracy (Vinay et al., 2015).

Subspace projection technique involves the mapping of high dimensional features into lower dimensions: - Principal Component Analysis (PCA) had been used by many researchers to construct subspace for representing feature class but principal components which are the largest eigen vector of the co-variance matrix generated are not always the optimal features in lower dimension for classification purposes (Han & Kim, 2005; Babatunde, Olabiyisi, Omidiora & Ganiyu, 2015).

The Independent Component Analysis (ICA), which is a generalization of PCA, requires high computational time during training (Déniz, Castrillón & Hernández, 2003). The Linear Discriminant Analysis (LDA) for feature selection requires much computation time and its performance can also be degraded when sample (feature) size is large (Bhuiyan & Liu, 2007). It is therefore very important to select relevant features in facial recognition in order to have better accuracy in classification phase (Aghdam, Ghasem-Aghaee & Basiri, 2009).

With the feature selection, the complexity and computational cost of classifier can be reduced by minimizing the number of features to be used into measurable forms while still maintaining acceptable recognition accuracy (Miche, Bas, Lendasse, Jutten & Simula, 2007; Hira & Gillies, 2015). Obtaining an optimal feature subset in feature selection turns out to be usually intractable (Yu & Liu, 2004), and many other problems related to feature selection shown to be non- deterministic polynomial hard problem (NP) (Imani et al., 2012).

There is the need to apply meta-heuristic optimization technique in order to avoid prohibitive complexity for optimal feature selection. Among feature selection methods, the population-based meta-heuristic optimization approaches such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO) have received great attention (Aghdam et al., 2009). The meta-heuristic feature-based selection algorithms produce effective solution by using knowledge from previous iteration. Thus, this study applies ACO meta-heuristic optimization technique to select optimal features from Gabor features due to its uniqueness as a constructive search optimization algorithm that optimizes features using indirect communication with its search environment via pheromone in a phenomenon called stigmergy which is not commonly available in any other meta-heuristic (Saraç, Ay & Özel, 2014).

1.3 Aim and Objectives of the Study

The aim of this study is to reduce the high dimensionality of Gabor features for facial recognition system and the specific objectives of the study are to:

(i) extract facial features with Gabor filters of 8 orientations and 5 scales,

(ii) optimize the Gabor features extracted with Ant Colony Optimization Algorithm,

(iii) perform facial image matching with the use of some selected distance classifiers; City-block, Chebyshev, Malahanobis and Euclidean for face image classification,

(iv) implement the proposed system in Matlab 2016A and evaluate the performance based on classification accuracy, classification time, sensitivity, specificity and error rate.

1.4 Significance of the Study

Dimensionality reduction through the selection of the most relevant features from the Gabor feature dimensions using Ant Colony Optimization Algorithm contributed to the elimination of less important, unnecessary features and also helps to depict a face image in a relatively lower dimensional feature subspace without losing much information from the extracted Gabor features for face representation.

The optimization of Gabor features reduce high level of memory usage, increase classification accuracy and further makes face image recognition system possible in a reasonable computation time. The introduction of some selected distance classifiers for classification in the facial recognition system allows a further study to be conducted on how to know the effect of optimized Gabor features on distance classifiers. This also reduces the complexity that might occur in choosing a robust distance measure metric to classify any Gabor feature-based face recognition system.

1.5 Scope of the Study

The optimized Gabor features facial recognition system target in minimising the influence of redundant features, irrelevant features and computational complexity which are part of the major factors affecting classification accuracy of Gabor-based face recognition system. The study only employed Gabor-filters of 5 scales and 8 orientations for feature extraction, then later applied ACO meta-heuristic as a feature selection technique to adequately transform high Gabor feature dimension into low feature subspace. The study selected just only four distance classifiers (Mahanolobis, City-block, Euclidean and Chebyshev) to classify the optimized Gabor features using distance measure metrics.

1.6 Thesis Layout

The remaining chapters of this thesis are organized as follows:-

Chapter Two: include the related literatures on several dimensionality techniques applied Gabor feature recognition system, discusses various biometric methods, feature selection techniques and swarm intelligence optimization algorithms.

Chapter Three: presents the approach employed in the study, the image datasets with their different preprocessing methods, dimensionality reduction methods which involve the feature extraction and selection using Gabor-filters and ACO.

Chapter Four: Discusses and also presents the experimental results and analysis of this study.

Chapter Five: Gives the summary, conclusion of this study based on the results and provides recommendations for future research.

1.7 Definition of Operational Terms

Adaptive Histogram Equalization (AHE): It is a computer image processing technique used to improve contrast in images before extraction is being performed.

Ant Colony Optimization: this is a meta-heuristic approach for solving hard combinatorial optimization problems. It is a nature-inspired optimization algorithm used to find solution to hard combinatorial problem.

Automated Teller Machine: is an electronic banking outlet, which allows customers to complete basic transactions without the aid of a branch representative or teller. Anyone with a credit card or debit card can access most ATMs.

Biological Characteristics: these are unique features used for human identification purpose

Biometrics: This is a name given to the branch of information technology that deals with measuring and analyzing body characteristics of features such as fingerprints, eyes, voices, veins and face pattern for distinguishing between an authorized person and impostor.

Curse of Dimensionality: It is a term used to describe or define the difficulty and increase in cost of computing as the number of features increases.

Dimensionality Reduction: It is a process of reducing the number of random variables under consideration via obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

Discrete Cosine Transform (DCT): It is a novel approach for illumination normalization, it keeps facial features intact while removing excess lighting variations

Face Recognition: This is a biometric method of identifying an individual by comparing real-time capture or digital image data with the stored record of particular person. Face recognition is commonly applied for security purposes, but are increasingly being used in other applications.

Facial Features: The most interesting or distinguishing elements or structures of a face image such as eye, nose, long chin and thick eyebrows

Feature Extraction: It is a process in pattern recognition system in which a new set of important or relevant features is formed or created from localized image.

Feature Selection: A technique commonly used in machine learning and pattern recognition for reducing the dimension of feature space, it obtains a subset of relevant features from a large number of features to construct robust learning models.

Feature Vector: This is a vector containing multiple elements about an object commonly used to represent numeric or symbolic characteristics called feature of an object in a mathematical and easily analyzable method.

Gabor-filter: It is a linear filter named after Dennis Gabor used purposely for extraction of features in pattern recognition, image processing and edge detection.

Generalization: this is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data in machine learning applications and statistical learning theory. It is also known as the out-of-sample error.

Gray Level Transformation: It is the visual appearance of an image which is generally characterized by two properties; brightness and contrast

Histogram Equalization: It is a method in image processing of contrast adjustment using the image's histogram.

Illumination Normalization: It is a process for controlling the variation in lighting in order to make the image appear stable under different lighting conditions.

Image Database: This is collection of image information that is organized so that it can be easily accessed, managed and updated.

Image pixel: It is the smallest unit of a digital image or graphic that can be displayed and represented on a digital display device.

Impostor: A person who submits a biometric sample in an attempt (either intentional or unintentional) to gain access to a system using the identity of another enrollee.

Independent Component Analysis (ICA): This is a computational method for separating a multivariate signal into additive subcomponents.

Linear Discriminant Analysis (LDA): This is a generalization of Fisher's linear discriminant commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications.

Local Binary Pattern (LBP): It is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

Meta-heuristic: A computational technique that optimizes a problem by iteratively trying to improve a candidate solution. It allows to find the best solution over discrete search space.

Principal Component Analysis (PCA): It is one of the statistical techniques frequently used in signal processing for feature extraction, data dimension reduction and data de-correlation process.

Pattern Recognition: A branch of artificial intelligence concerned with the classification or description of observations. It is one of the four cornerstones of computer science that is involved in finding the similarities or pattern among small, decomposed problems that can help solve more complex problems more efficiently.

Optimal Features: These are the best subsets which contain the least number of features that most contribute to effectiveness and recognition accuracy of facial recognition system.

Stigmergy: This is an indirect mode of communication in which ants being distinct from each other tries to contact with each other through producing and reacting with the stimuli.

Uncontrolled Environment: It is an environment that does not have a predefined arrangement of subjects and capture devices.

Region of Interest (ROI): This is a portion of an image that you want to filter or perform some other operation on by creating a binary mask, which is a binary image that is the same size as the image you want to process.

CHAPTER TWO

REVIEW OF RELATED LITERATURE

This chapter borders on the theoretical background to biometric, face representation technique, recognition system, dimensionality reduction techniques and meta-heuristic algorithm-ACO. Other discussions include image processing, classification techniques and studies on dimensionality reduction of Gabor features are present.

2.1 Facial Recognition System

The automated face recognition is an important application which has received huge attention in biometrics, computer vision and pattern recognition (Rady, 2011). Woody Bledsoe, Helen Chan Wolf and Charles Bisson in the year 1960 designed the first semi-automated facial recognition programs. The programs required the system administrator to locate features such as eyes, ears, nose and mouth on the photograph. The calculated distances and ratios to popular reference points were compared with reference data. Goldstein, Harmon and Lesk (1971) in Bell Laboratories used twenty-one specific subjective features such as eyebrow weight, nose length, colour of hair and lip thickness to recognize face images using pattern classification techniques. Kanade (1973) came up with the first complete automatic face recognition system. He designed a face recognition program which was run in a computer system. The algorithm extracted sixteen facial parameters automatically. He compared the automatic extraction method to manual extraction. Result of 45-75 % correct identification obtained. Sirovich and Kirby (1987) considered eigenfaces, a standard linear algebra method for face recognition problem. They developed recognition method based on the Principal Component Analysis (PCA). The main objective of their research was to represent a face image in lower dimension without actually losing much relevant information. The recognition model showed a great improvement, because less than one hundred values were required to accurately code a suitable aligned and normalized face image. Turk & Pentland (1991), motivated by the work of Kirby and Sivorich, found out that while using eigenfaces methods, the residual error could be employed to detect faces in images. This discovery enabled reliable real-time automated face recognition systems. Although the method was affected by environmental conditions, it paved way for great concern in the continuation of automated face recognition systems development.

Many techniques have been used in face recognition, which has resulted to several algorithms. Some of these are neural network, LDA, ICA and their derivatives. The Face Recognition Technology Evaluation (FERET) sponsored by the Defence Advanced Research Products Agency (DARPA) from 1993 through 1997 strengthened the development of face recognition algorithms and technology by assessing the prototypes of face recognition systems (Moon & Phillips, 2001). It changed the face recognition technology from its infancy level to commercial products. The Face Recognition Vendor Tests (FRVT) performed in the year 2000, 2002, and 2006 built upon the work of FERET. The major goals of FERET were to assess the capabilities of commercially available facial recognition systems and to educate people on how to accurately present and analyze results. The FRVT 2002 was designed to measure technical progress, evaluate performance on real-life large scale databases and introduce new experiments for better understanding of face recognition performance.

The Face Recognition Grand Challenge (FRGC) evaluated the modern face recognition algorithms available. The high-resolution face images, 3D face scans, and iris images were used in the tests. The results showed that the new algorithms are ten times more accurate than the face recognition algorithms of 2002 and hundred times more accurate than those of 1995. Some of the algorithms outperformed human participants in recognizing faces and could also distinctively identify identical twins. In society nowadays, several companies are using face recognition in their products. The method of detecting people and analyzing their gesture is also being employed in automotive industry. The framework of face recognition is shown in Figure 2.1

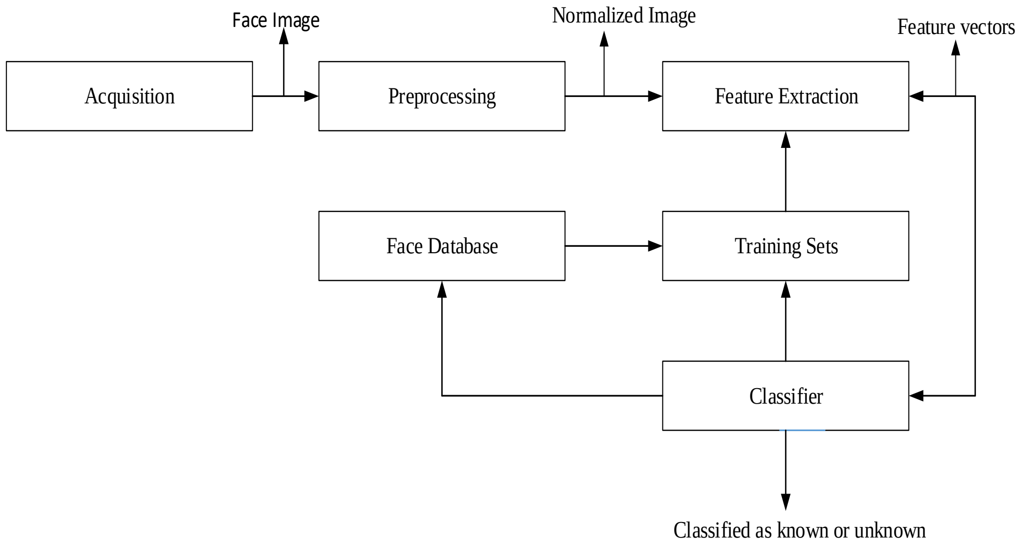


Figure 2.1: Process of Face Recognition (Chitra & Balakrishnan, 2012)

2.2 Methods of Biometric Technologies

The different variety of systems requires a more reliable personal recognition schemes to confirm or determine the identity of a particular person requesting for their services. The main goal of these schemes is to enable that services are accessible only to authorized users. Applications of such a system include mobile phones, automated teller machine (ATM), secure electronic banking, computer systems, laptops, health, credit cards and social services. Biometrics is process that involves the automatic recognition of individuals based on the feature vectors derived from biological characteristics (Jain, Ross, & Prabhakar, 2004) There are various methods of biometric recognition of individual, these are: fingerprint, retina scan, iris scan, signature dynamic, body odour, DNA, speech, vascular and Hand Geometry

2.2.1 Fingerprint Technique

This biometric system is the oldest of all biometric methods (Srivastva & Singh, 2011) It uses distinctive features of fingerprint to identify or verify individual. Fingerprint consists of ridges and valleys of a human finger. Fingerprints are easily deposited on suitable surfaces (such as glasses, metal and polished stone) by the natural secretions of sweat from the eccrine glands that are present in epidermal ridges. The traditional fingerprint technique uses the ink to obtain the finger print onto a piece of paper.

The piece of paper is then scanned with traditional scanner. The modern method uses live finger print readers, which are based on thermal, optical, silicon or ultrasonic principles (Jain, Ross, & Pankanti, 2006). They are two commonly used methods for finger print matching; minutiae based and correlation based. Minutiae based techniques find the minutiae points first and then map their relation placement on the finger. Correlation based techniques need the precise location of a registration point

(Cappelli, Maio, Maltoni, Wayman, & Jain, 2006). The fingerprint technology is a widely deployed biometric method in physical and logical access applications. The fingerprint pattern is shown in Figure 2.2

Figure 2.2: Human Finger Patterns (Jain et al., 2006)

2.2.2 Retina Scan

This recognition is based on the blood vessel pattern in the retina of the eye, as the blood vessels at the back of the eye have unique pattern from eye to eye and person to person. Due to the complex structure of the capillaries that supply the retina with blood, each individual's retina is distinctive in nature as shown in Figure 2.3. The network of blood vessels in the retina is not entirely genetically determined and thus even identical twins do not possess similar pattern structure (Singhal, Gupta, & Garg, 2012). A Retina recognition method is performed by casting an unperceived beam of low intensity infrared light into person's eye as they look through the scanner's eye piece. The beam of light traces a standardized path on the retina. A coupler is used to read the blood vessel patterns. This recognition method is suited for environments requiring maximum security, such as government, banking and military.

Figure 2.3: Image of Retina (Bhattacharyya, Ranjan, & Choi, 2009)

2.2.3 Iris Scan

Human iris is a thin circular structure in the eye which is responsible for controlling the diameter and size of the pupils. It also controls the amount of light which is allowed through to retinal in order to protect the eye's retina. The iris recognition is an automated method of biometric identification which uses mathematical pattern recognition techniques on video images of one or both of the irises of an individual eye, where complex random patterns are unique, stable and can be seen from some distance (Abikoye, Sadiku, Adewole, & Jimoh, 2014). Iris recognition uses video camera technology with subtle near infrared illumination to acquired images of the detail-rich, intricate structure of the iris which are visible externally. Figure 2.4 shows the diagram of human iris

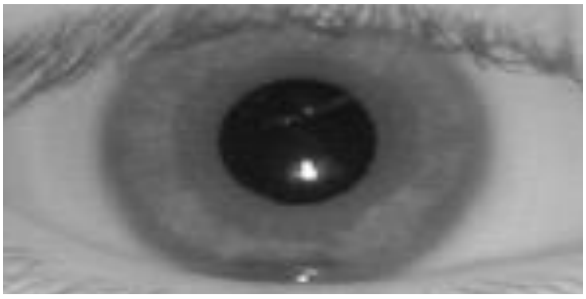


Figure 2.4: Diagram of Human Iris (Homayon & Cruz, 2015)

2.2.4 Signature Dynamic

This technology is based on the dynamics of marking the signature, rather than a direct comparison of the signature itself. The dynamics is measured as a means of the pressure, direction, acceleration and the length of the strokes, dynamics number of strokes and their duration. The most apparent and important of this is that a fraudster cannot glean any information on how to write the signature by simply looking at one that has been previously written. There are several devices used to capture the signature dynamics. These are either traditional tablets or special purpose devices. Tablets capture 2D coordinates and the pressure. Special (distinct) pens have the ability of capturing movements in all three dimensions. In tablet, the resulting digitalized signature looks different from the usual user signature as illustrated in Figure 2.5, and also during signing the user does not see what he or she has written.

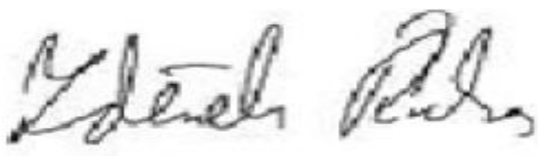


Figure 2.5: A Signature Taken Using Tablet (Bhattacharyya et al., 2009)

2.2.5 Body Odour Technique

It is well known that each individual emanates an odour that is characterized by its chemical composition and this could be applied to distinguish different people. Human body dynamically produces distinctive patterns of volatile organic compounds under various living conditions such as hormonal status, eating, sexual activities, drinking or health (Wongchoosuk, Lutz & Kerdcharoen, 2009).

The component of the odour emitted by a human body is unique to a particular individual. The odour biometric technology is based on the fact that virtually each human smell is unique. The smell is captured by sensors that are capable to obtain the odour from non-intrusive parts of the body such as the back of the hand. These are extracted by the system and converted into template. With body odour analysis, diagnosis of some diseases or activities can be deduced (Bhattacharyya et al., 2009).

2.2.6 DNA (Deoxyribonucleic Acid)

This technology uses the Deoxyribonucleic acid, a molecule that contains biological information of living organisms, which is also composed of chemical building blocks known as nucleotides as shown in Figure 2.6. DNA can be collected from any number of sources like blood, hair, fingernails, mouth swabs, blood stains, saliva, straws and any number of sources attached to human body. The technique is a new biometric technology predominantly used in extreme security domains (Jain et al., 2004). It is difficult to forge this method due to the high level of uniqueness and the probability of two persons having the exact same DNA profile except for the fact that it is the same for identical twin (Afolabi & Akintaro, 2017).

It requires expensive equipment in order to break down the DNA successfully, analyze the unique features of DNA and create a DNA profile. It is commonly used in forensic applications for person recognition. The system needs to get physical samples (hair or blood) from users to collect their DNA data. This technique requires long period of time to go through all the processes of creating a DNA profile for each individual and verifying each individual's DNA profile.

Figure 2.6: Strand of DNA (Bhable, Kayte, Maher, Kayte & Kayte, 2015)

2.2.7 Voice / Speech

Speech recognition technique is the verification of an individual's voice by simply tracing the pattern of voice (Srivastva & Singh, 2011a). There are various speech recognition methods; among those are the acoustics phonetic pattern comparisons and automatic speech recognition method (Uddin, Sharmin, Hasnat, Ahmed, & Hasan, 2011). The physical features of voice are related to the appendages that produce its sound. These features include the vocal tracts, mouth, nasal cavities, and lips. Also, on the other hand, the behavioural features of voice are related to the emotional and physical states of the speaker. Conventionally, voice-based verification techniques can be grouped into two major types: text-dependent and text-independent methods. In text-dependent techniques, the individuals are verified by speaking a fixed predetermined phrase, while in the text-independent techniques no constraints occur about what must be spoken.

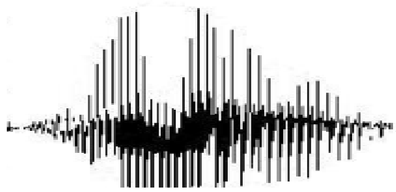


Figure 2.7: Diagram of Speech Recognition Pattern (Jain et al., 2006)

2.2.8 Vascular (Vein) Recognition

Under the hands are various network of veins, the patterns of these veins are unique for each individual. Vascular recognition uses optical scanning technology to capture vein images in the palm, finger or eyeball (Prasanthi, Hussain, Kanakam, Chakravarthy, 2015). This biometric technology works by recognizing the subcutaneous (under the skin) vein patterns in an individual's hand as shown in Figure 2.8. When a user's hand is placed on a scanner, a near-infrared light maps the location of the veins. The red blood cells present in the veins absorb the rays and show up on the map as black lines, whereas the remaining hand structure shows up as white. After the vein template is extracted, it is compared with previously stored patterns and a match is made. Various categories of vein recognition method, include finger vein, wrist vein, palm, and backhand vein recognition. The basic concept of scanning remains the same with each of these techniques.

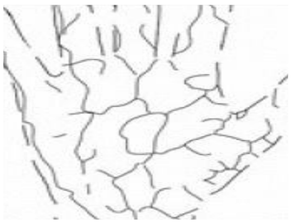


Figure 2.8: Diagram of Hand Vein Structure (Jain, 2004)

2.2.9 Hand Geometry Recognition

These recognition methods are based on a number of estimations or measurements taken from the palm and fingers structure, including its shape, size, thickness, lengths, widths of the fingers in different places (Sanchez-avila & Gonzalez-marcos, 2000). It is based on the fact that individual's hand is shaped differently and does not change after certain age (Das & Meshram, 2013). This biometric technique is simple, inexpensive and relatively easy to use. Environmental conditions such as dry weather and individual anomalies like dry skin do have any negative impacts on the recognition accuracy of hand geometry based technology (Uddin et al., 2011). The image of hand acquired is shown in Figure 2.9

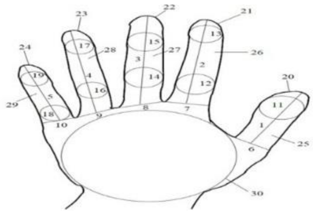


Figure 2.9: Structure of Hand Geometry (Bakshe & Patil, 2015)

2.3 Facial Representation Techniques

This is the derivation of most relevant features from original facial images to describe faces in order to develop an effective facial recognition system(Shan, Gong, & McOwan, 2009). The method applied in the representation of a facial image determines the success of detection and identification algorithms (Wahid, 2013). An image is viewed at the entry-level to confirm whether a given image represents a face; the image is transformed or processed (rotated or resized) to have the same position with images in the database system. Facial representation remains an important aspect in facial biometric recognition system. Finding efficient and discriminative facial representations that are resistant to large variations in illumination, pose, expression, ageing, partial occlusion other changes remain a great problem in facial recognition system (Flynn, 2006). The techniques developed for facial representation can be classified into two main categories: Appearance based and local based method (Feature based) (Zou, Ji, & Nagy, 2007).

2.3.1 Appearance Based Approach (Holistic Method)

This technique has attracted great interest from a broad range of research fields such as pattern recognition, computer vision, machine mining and biometrics (Beham & Roomi, 2012). The global facial region is considered as input data into face catching machine. Appearance based methods project images from higher to lower dimensional subspace and then classify such images based on similarity or distance with known image (Struc & Pavesic, 2009). In facial recognition system appearance based techniques are not only computationally expensive but also do not perform successfully under large variations in pose, scale and illumination. There are many techniques under this method like Principal Component Analysis (PCA), Discrete Wavelet Transform (DWT), Independent Component Analysis (generalization of PCA) and Linear Discriminant Analysis (LDA) (Bala, Singh, & Meena, 2016).

(a) Principal Component Analysis (PCA)

This is a statistical linear approach that is referred to as Karhunen-Loeve Transformation (KLT) or Hostelling Transform that optimally reduces redundancy (Singh, Sharma & Rao, 2011; Karamizadeh, Abdullah, Manaf, Zamani, & Hooman, 2013), PCA is used for feature extraction and data representation in computer vision and pattern recognition such as face recognition (Yang, Zhang, Frang & Yang, 2004). PCA is a commonly employed method for reducing number of variables in face recognition (Paul & Sumam, 2012; Bahurupi & Chaudhari, 2012). It searches for a set of representative projection feature vectors such as that the projected samples retain most information about original samples (Bala et al., 2016). The PCA method applies a vector space transform to reduce the dimensionality of large database. Applying mathematical projection that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components (Javed, 2013).

(b) Independent Component Analysis (ICA)

Independent Component Analysis is a computational method for revealing hidden factors that underlie sets of random variables, signals and measurements (Dangi &

Ahmad, 2009). It is a generalization of PCA, which attempts to find much relevant information contained in high-order relationship among the image pixels (reduces the second order and higher order dependencies in input samples) (Alyasseri, 2015). The technique has been successfully applied to several problems such as locating hidden features in financial data and face recognition (Deniz, Castrillon & Hernandez, 2003).

(c) Linear Discriminant Analysis (LDA)

It is known as fisherface in an appearance-based technique, LDA is a powerful technique for data reduction and feature extraction used for the development of facial recognition model (Malik & Bansal, 2016). It yields a robust representation that linearly transforms the original data space into a low-dimensional feature with focus on most discriminant features (perform dimensionality reduction while preserving as much of class discriminatory information as possible) (Bhele & Mankar, 2012). This method gives significance to those feature vectors in underlying space that best describe the best discriminate among classes rather than best describing data. LDA projects a face image from high dimensional image space to a low-dimensional image by computing transformation that maximizes the between class scatter while minimizing the within class scatter.

(d) Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) is an extensively used signal analysis tool for feature extraction, den-noising applications, image compression and texture classification due to its effective ability for multi-resolution decomposition analysis (Wadkar & Wankhade, 2012). Two-dimensional DWT is implemented as a set of filter banks, comprising high-pass and low-pass filters. The final result obtained is a decomposition of the input image into four non-overlapping multi-resolution sub bands. This results into four decomposed sub images L1, H1, V1, and D1. These sub images represent different frequency localizations of the original image which refer to Low-Low, Low-High, High-Low and High-High respectively. In each iterative step, only the sub image L1 is further decomposed.

The Discrete Wavelet Transform which is based on sub-band coding is discovered to produce a fast computation of Wavelet Transform (Murugan, Arumugam, Rajalakshmi & Manish, 2010). The one-dimensional wavelet decomposition is first applied along the rows of the images, and then their results are further decomposed along the columns.

2.3.2 Feature Based Approach (Analytic Method)

This approach is accomplished by the manipulation of distance, angles, and the area measurements of the visual features derived from an image (Sharif, Mohsin & Javed, 2012) . This method uses some simple measures to describe features of facial image and their various relationships. For instance, human face normally appears with two eyes that are symmetric to each other, a nose and mouth. Face images are detected by the application of these measures to locate features, these relationships which exist between features can be represented by their relative distances and positions.

Feature-based method extracts distinctive individual facial features, such as eyes, nose, lips, mouth etc as well as other fiducial marks to build up model based on the position and size of these characteristics to form a feature vector. The feature vector feature represents the face geometric relationships among facial points, thus reducing the input facial image to a vector of geometric features (Shan et al., 2009). Standard statistical pattern recognition methods are then used to match faces using these measurements. There are several approaches under the feature-based technique like Local Binary

Pattern (LBP), Elastic Bunch Graph Matching (EBGM) and Gabor-filter (Sufyanu, Mohamad, Yusuf, & Mustafa, 2016).

(a) **Local Binary Pattern (LBP)**

This is a very simple but efficient gray scale multi-resolution operator. It was introduced originally to describe the texture classification or analysis in image processing and computer vision (Chunjun, Junxing, Changhong & Yanhui, 2013) . LBP computes a value for each pixel in the image based on its relationship with neighborhood pixels so that it can capture the local texture of image by thresholding the 3×3 neighborhood pixel value with the center pixel value (Khryashchev, Priorov, Stepanova & Nikitin, 2015). The basic idea of the local binary patterns is to avoid image feature representation as a high-dimensional vector which contains a lot of redundant information. LBP operator works with the eight neighbors of a pixel, using the value of this center pixel as a threshold. If a neighbor pixel has a higher gray value than the center pixel (same gray value) then, one is assigned to that pixel, else it gets a zero. The LBP code for the center pixel is then produced by concatenating the eight ones or zeros to a binary code as shown in Figure 2.10.

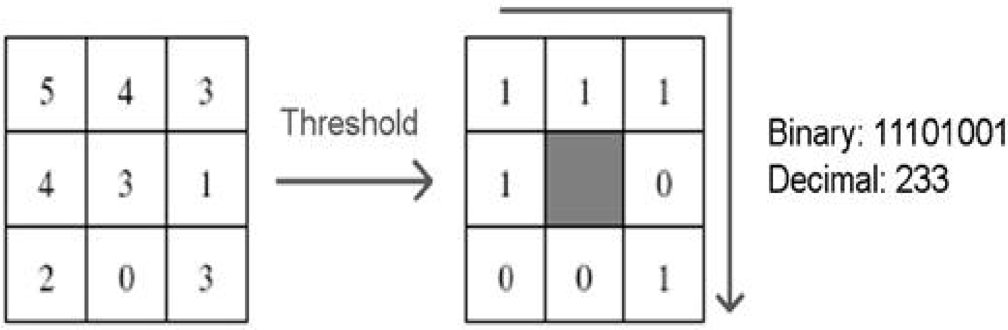


Figure 2.10: LBP Operator (Patinge & Deshmukh, 2015)

(b) Elastic Bunch Graph Matching (EBGM)

This technique is a feature-based face recognition algorithm, which takes into account the human facial features (López, Mora & Selva, 2016). EBGM obtains a bunch of jets for each training image and applies the jets to represent the graph node. Face recognition using elastic bunch graph matching is based on recognizing new faces by estimating a set of novel features using a data structure called a bunch graph (Jaiswal, Gwalior & Gwalior, 2011). Similarly for each test image, the landmarks are estimated and located using bunch graph. Then the features are extracted by convolution with the number of instances of Gabor filters followed by the creation of face graph. The matching score is calculated on the basis of similarity between face graphs of database and test image. When matching a bunch graph to an image, the jet extracted from the image is compared to all jets in the corresponding bunch attached to the bunch graph and the best matching one is selected.

(c) Two-Dimensional Gabor-filters Technique

Gabor technique is a method that involves the use of Gabor filters (wavelets) for facial feature extraction. The application of Gabor wavelets originally applied to facial recognition using Dynamic Link Architecture (DLA) framework (Lades et al., 1993). The DLA creates a flexible template comparison between Gabor wavelet representations of different face images. Wiskott, Fellous, Krüger, and von der Malsburg (1997) developed face recognition by expanding DLA using a Gabor wavelet-based elastic bunch graph matching (EBGM) algorithm to label and recognize human faces. The experimental test conducted on the FERET database revealed a high recognition rate for frontal face images. Gabor filters and Gabor wavelets are directly related since both can be designed for a number of dilations and rotations. The Gabor wavelet representation captures salient visual properties such as orientation selectivity, spatial localization and spatial frequency. It represents data at different scales and orientations. Gabor filters have been employed successfully and broadly in many applications such as handwritten numeral recognition and fingerprint recognition (Thangairulappan & Jeyasingh, 2012). The Gabor wavelets generally used in Facial recognition can be defined as follows (Liu & Wechsler, 2002; Bellakhdhar et al., 2013)

$$Gabor(x,y,\mu,v) = \theta(x,y,\mu,v)(\alpha - \beta) \tag{2.1}$$

Where

$$\theta(x,y,\mu,v) = \frac{\|k_{\mu,v}\|^2}{\sigma^2}\, e^{-\frac{\|k_{\mu,v}\|^2(x^2+y^2)}{2\sigma^2}} \tag{2.2}$$

$$\alpha = e^{ik_{\mu,v}z} \tag{2.3}$$

$$\beta = e^{-\frac{\sigma^2}{2}} \tag{2.4}$$

$$\varphi_{\mu,v}(z) = \frac{\|k_{\mu,v}\|^2}{\sigma^2}\, e^{-\frac{\|k_{\mu,v}\|^2\|z\|^2}{2\sigma^2}}\left[e^{ik_{\mu,v}z} - e^{-\frac{\sigma^2}{2}}\right] \tag{2.5}$$

Where z = (x, y) is the point with the horizontal coordinate x and the vertical coordinate y in the image plane (is the pixel location in the digital image). The θ represents projection angle. The parameters μ and v define the orientation and scale of the Gabor kernel. $\|.\|$ denotes the norm operator, and σ refers to standard deviations of the Gaussian window in the kernel. The wave vector $K_{\mu v}$ is defined as:

$$k_{\mu v} = k_v\, e^{i\varphi\mu} \tag{2.6}$$

Where $k_v = \frac{k_{max}}{f^v}$, (2.7)

$$\varphi_\mu = \frac{\pi\mu}{8} \tag{2.8}$$

K_{max} is the maximum frequency and f_v is the spatial frequency between kernels in the frequency domain. Gabor filters are selected relative to following parameters:

$$K_{max} = \frac{\pi}{2} \tag{2.9}$$

$$f = \sqrt{2}\ , \tag{2.10}$$

$$\sigma = \pi \tag{2.11}$$

The parameters ensure that frequencies are spaced in octave steps from 0 to π, typically each Gabor wavelet possess a frequency bandwidth of an octave that is sufficient to have less overlap and cover the entire spectrum. Two-dimensional Gabor filters correspond to a family of bi-dimensional Gaussian functions modulated by cosine function (real part) and sine function (imaginary part) representing orthogonal directions as shown in Figure 2.11. These two components may be formed into a complex number as indicated in Equation (2.12) or used separately as illustrated in Equation (2.13) and (2.14) respectively.

Complex Part:

$$g(x, y;\ \lambda, \theta, \sigma, \gamma) = \exp\left(-\ \frac{x'^2 + \gamma^2 y^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \varphi\right)\right) \tag{2.12}$$

Real Part (cosine function):

$$g(x, y;\ \lambda, \theta, \sigma, \gamma) = \exp\left(-\ \frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(i\left(2\pi \frac{x'}{\lambda} + \varphi\right)\right) \tag{2.13}$$

Imaginary part (sine function):

$$g(x, y;\ \lambda, \theta, \sigma, \gamma) = \exp\left(-\ \frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(i\left(2\pi \frac{x'}{\lambda} + \varphi\right)\right) \tag{2.14}$$

Where

$x' = x \cos\theta + y \sin\theta$

$y' = x \sin\theta + y \cos\theta$

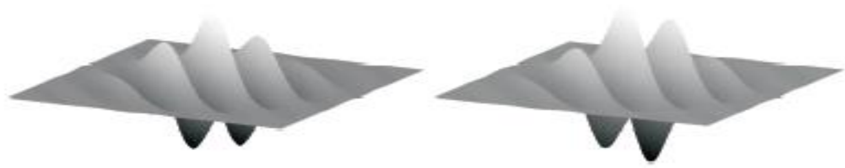
λ = wavelength of the cosine wave

θ = orientation of the filter (wavelet)

φ = phase of the sine wave

σ = Standard deviations of Gaussian envelope along x and y dimensions

γ = spatial aspect ratio of the Gaussian envelope



(a) Real part(cosine-type) (b) Imaginary part (sine-type)

Figure 2.11: Gabor filters: The real part and imaginary part (Struc, Gajsek & Pavesic, 2009).

2.4 Dimensionality Reduction (DR)

This is a process which involves the transformation of huge feature under considerable representation (Roweis & Lawrence, 2000). This is illustrated in Equation 2.15.

$$R^N \rightarrow R^N \ (M < N) \qquad (2.15)$$

Where N is the dimension of features, M is the reduced dimension of features in domain R. Preferably, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the feature. The intrinsic dimensionality of feature is the minimum number of parameters required to account for the observed properties of the feature. Dimensionality reduction (DR) is one of the most important techniques in pattern recognition and machine learning. It is not appropriate to directly use raw feature for recognition due to the fact that the significant features have not been extracted, but also for the purpose of extreme high dimensionality of these features. In facial recognition, each image to be recognized consists of thousands of pixels where each pixel is represented by multi-byte value. Recognizing this kind of image data will be a time consuming and computationally expensive task

Dimensionality reduction (DR) produces an approximation to original feature in fewer dimensions, while still maintaining same structure of original features. The important features employ for recognition functions only consider just a small portion of features and cannot be obtained by simple methods such as cropping and down-sampling. The objective of dimension reduction is to extract useful information and reduce the dimensionality of input data into classifiers in order to decrease the computational cost and resolve the problem of curse dimensionality. Feature selection and feature extraction are two major processes use effectively in feature dimensionality reduction (Pali & Bhaiya, 2013).

2.4.1 Feature Extraction (FE)

Feature extraction is a significant step in image processing and pattern recognition which major aim is to extract relevant information accurately for classification (Kumar & Bhatia, 2014). Feature extraction is a process through which a new set of features are produced (Kaur & Rajput, 2013). Facial feature extraction is an important component of pattern recognition system, it performs two crucial functions: transforms input vector into a feature vector and also reduces its dimensionality (Bakshi & Singhal, 2014). The goal of feature extraction is to represent data accurately in relatively low dimensional feature space (Raymer, Punch, Goodman, Kuhn, & Jain, 2000). The procedure involves the extraction of relevant features from a localized image. When performing analysis on complex data, one of the major problems stem from number of features or variables involved. It is necessary to extract a well-defined feature in order to make classification process more efficient. Analyzing large number of features generally require a large amount of memory and computation power. If features are carefully chosen, it is expected that the features set will remove the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

2.4.2 Feature Selection (FS)

Feature selection is a technique of obtaining subset of relevant features for use in pattern recognition and model construction (Patra & Dash, 2016). This technique is an extensive method that is applied to many fields such as data mining, biometrics, document/text classification, remote sensing and pattern recognition (Aghdam et al., 2009). Given a feature set of size n, the feature selection problem is actually on how to obtain a minimal feature subset set of size m$(m < n)$ while still maintaining a better higher accuracy in representing the original features (Nemati., Basiri , Ghasem-Aghaee, & Aghdam, 2009) Feature selection enables the simplicity of models by easier interpretation by researchers with shorter training time (Bermingham et al., 2015). Feature selection aids the removal of irrelevant, redundant and noisy features by selecting the subset of features that can result to best performance in term of accuracy and computational time (Jain, 2017).

The purpose for performing feature selection includes reducing computational cost, reducing data storage requirements, reducing the cost of future measurements and improving model interpretability (Oreski & Novosel, 2014). Feature selection is commonly employed in areas where there are many features and comparatively few samples. The archetypal case is the use of feature selection in analyzing DNA microarrays there are many thousands of features and a few tens to hundreds of samples. Feature selection algorithm can be seen as the combination of a search technique for getting new feature subsets, along with an evaluation measure, which scores the different feature subsets.

2.5 Feature Selection Algorithms

The feature subset selection algorithms can be categorized into three types: filter, wrapper and embedded technique (Kumari & Swarnkar, 2011).

2.5.1 Filter Techniques

These methods select features without depending of the type of classifier used. They remove unimportant, unnecessary and noisy features by applying preprocessing procedures before induction process (Yildirim, 2015). Filter method for feature selection rates features based on general properties such as interclass distance or statistical independence, without employing any mining algorithm. These techniques are usually less computationally expensive than wrapper (Sivakumar & Chandrasekar, 2014), but they produce a feature set which is not connected to a specific type of predictive model. The features are ranked by the score and either selected to be kept or remove from the database. Filter technique is illustrated in Figure 2.12.

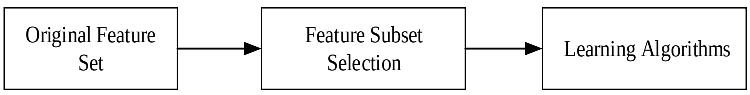


Figure 2.12: Filter method (Sheena, Krishan & Gulshan, 2016)

2.5.2 Wrapper Techniques

These techniques exploit a machine learning algorithm for assessing feature subsets through predictive accuracy (Wang & Liu, 2016). Wrapper techniques evaluate subset of features to discover the possible interactions between features (Ali Jan Ghasab, Khamis, Mohammad, & Jahani Fariman, 2015). These methods have been found to solve the problem of high dimensionality in bioinformatics (Purohit & Mehta, 2016). The complete wrapper process is shown in Figure 2.13. The wrapper techniques require more computation time and slower due to the repeating process of this method. Wrapper methods can be grouped into two major categories: greedy and random method such as Ant Colony Optimization, Simulated annealing, Randomized Hill climbing and Genetic algorithm (Saeys et al., 2007; Imani et al., 2012).

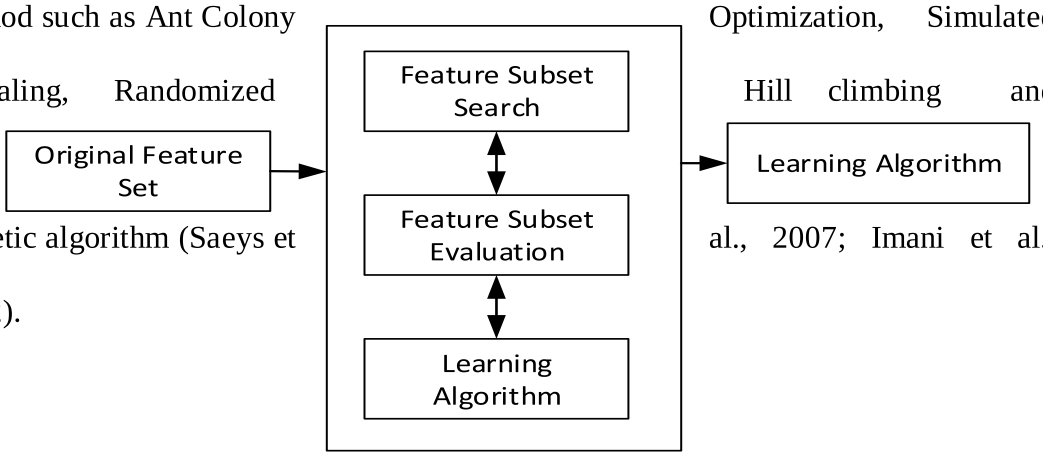


Figure 2.13: Wrapper Method (Senthil & Lopez , 2016)

2.5.3 Hybrid (Embedded) Techniques

These methods combine both the advantages of filter and wrapper methods so that efficiency is improved and feature selection become faster (Silva, Aguiar, Silva, & Inesctec, 2015). In embedded techniques, the filter methods are used to select a feature pool at first and later apply wrapper methods to find the optimal subsets of features from the selected feature pool. These methods have been designed in order to reduce the classification of machine learning (Purohit & Mehta, 2016). The embedded method is used to handle huge database, avoid the pre-specification of criterion and the quality of final results from a classifier algorithm provides a natural stopping measure. The most common type of embedded feature selection methods are regularization methods. Regularization methods are also known as penalization methods which introduce additional constraints into optimization of algorithm prediction. Figure 2.14 illustrates the hybrid technique.

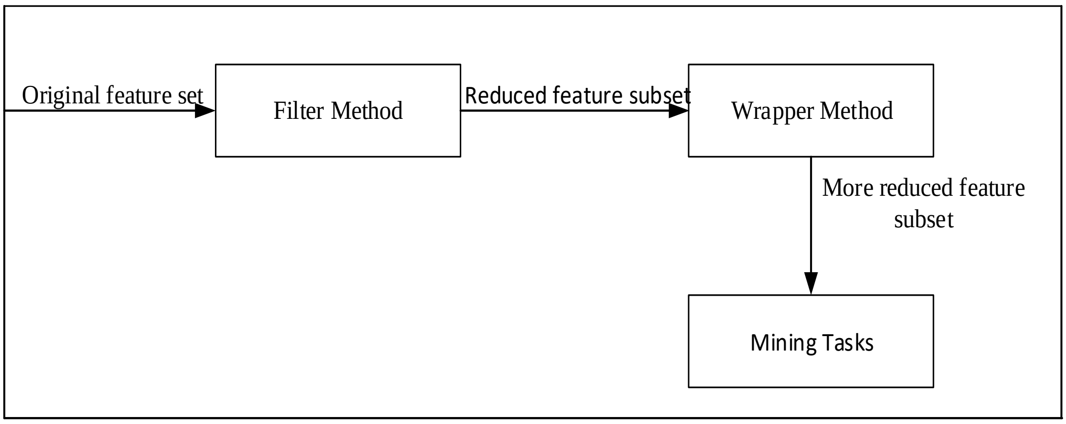


Figure 2.14: Hybrid Method (Sutha, 2015)

2.6 Procedures of Feature Selection Technique

A typical feature selection technique consists of four basic procedures, which are subset generation, subset evaluation, stopping criterion, and result validation (Kumar & Minz, 2014) as illustrated in Figure 2.15.

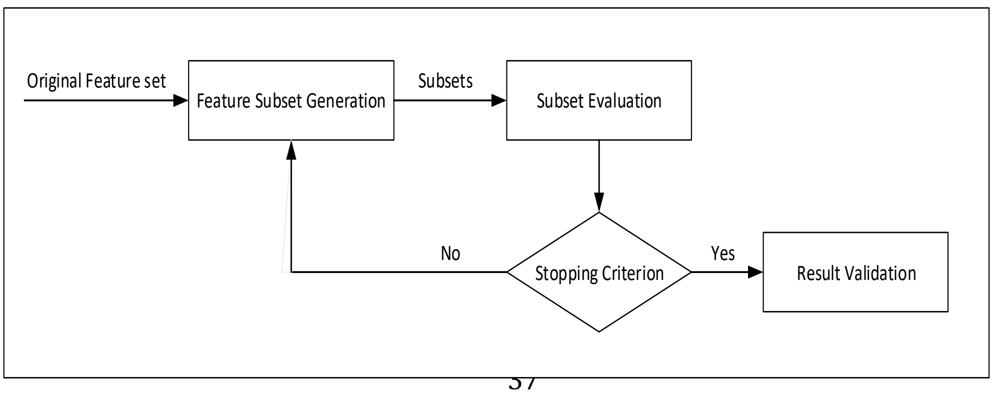


Figure 2.15: General Procedures of Feature Selection (Silva, Aguiar, Silva, & Inesctec, 2015)

2.6.1 Subset Generation

This is basically a search method that produces candidate feature subsets for evaluation based on a certain search technique (Ramaswami & Bhaskaran, 2009). The candidate subset is evaluated individually and compared with the previous best one according to certain evaluation. The subset generation is achieved by two basic steps. The first step is to find the search direction and also to identify the search strategy. The search direction can be forward, backward and random. If the search starts with empty set and adding sequentially, it is known as forward search. If the search starts with a complete set and removing features consecutively, it is known as backward search. Search may also begin by choosing randomly selected so that local optima be avoided (Zhang, Zhou & Chen, 2006). Second is to select a search strategy which generates different feature combinations to traverse through feature space. There are various search strategies include: complete, sequential and random search.

(a) Complete Search (Exhaustive)

This search technique returns the optimal subset during the search process. It guarantees the finding of the optimal result based on the evaluation criterion used. Different heuristic conditions can be employed to reduce the search space without disturbing the chances of obtaining optimal subset. The search space order is $O(2^N)$, a smaller number of subsets evaluated. An example is branch and bound (Gnana, Alias.

& Leavline, 2016). This is method has high accuracy but it is computationally expensive

(b) Sequential Search

This method adds or removes features one at a time. Sequential search approach can be divided into Sequential Forward Feature Selection (SFFS), Sequential Backward Feature Elimination (SBFE) and Bidirectional Selection Methods these are all Greedy Hill Climbing Algorithms which add or remove features one at step (Vidyavathi & Ravikumar, 2008). During search process, the SFFS does not remove redundant features generated while SBFE refuses to re-evaluate relevant features together with other features once a feature is eliminated. This technique has simple accuracy with less flexible backtracking. Generally, the sequential search algorithms are simple to implement and fast in producing subset as the order of the search space is usually $O(N^2)$ or less.

(c) Random Search

This technique randomly selects subset, which proceeds into two different methods. The first method is to follow sequential search, which injects randomness into the classical sequential approaches such as simulated annealing and random-start hill climbing (Jang, Han & Kim, 2004), while second method generates next subset in a complete random form. This search technique is applicable in area where feature space contains thousands of features such as image processing, bioinformatics and pattern recognition. In this situation, it is not feasible to search all the entire feature space. This technique trades-off optimality of solution for efficiency by searching only sampled portion of feature space. The use of randomness approach helps to select accuracy, speed and escape the local optima result in the search space with 0(N log N).

2.6.2 Subset Evaluation

There is a need for the evaluation of each generated subset after subset generation is completed. The goodness of a subset is constantly determined by a certain condition that is, an optimal subset selected using one measure may not be optimal according to another measure (Liu & Yu, 2005). The evaluation criteria can be categorized into two groups based on their dependency and independency on algorithms that will finally be applied on the selected feature subset (Kumar & Minz, 2014).

2.6.3 Stopping Criteria

A stopping criterion terminates the feature selection process; if not, the entire process may continue to run exhaustively through the space of subsets (Guyon & Elisseeff, 2003). Commonly employed stopping criteria for feature selection procedures are:

(a) When the search is completed

(b) When the particular bound is reached, here bound is given as least number of attributes or maximum number of iteration.

(c) When an estimated number of iterations are finished.

(d) When an optimum feature subset in agrees with evaluation criterion.

(e) When the change (adding up or elimination) of feature subsets will not create superior subsets.

2.6.4 Result Validation

The accurate method for result validation is obtained by measuring the outcome directly from previous information (Senthil & Lopez, 2016). As result, some indirect methods are depended on through the examination of changes in mining capability with the change of features. If the relevant features are known early as in the case of synthetic data, they can be compared with the selected features. Information on the irrelevant or redundant features can also help. However, the in real-world applications such as past knowledge is not available. For a selected feature subset, experiment can be conducted before and after to compare the error rate of the classifier learned on the full set of features and that learned on the selected subset.

2.7 Two-Dimensional Gabor Filter for Facial Features Extraction

A two-dimensional Gabor filter in spatial domain is a Gaussian kernel modulated by sinusoidal plane wave (Kyrki, Kamarainen, & Kälviäinen, 2004). The general procedure, when using Gabor filters for face recognition, is achieved by construction of a filter bank with filters of different scales and orientations in order to filter the given facial image with all the filters from bank (extraction of information is based on the use of bank of Gabor filters) (Kaur & Rajput, 2013). The bi-dimensional Gabor wavelet representation of a facial image is derived by the convolution of face with Gabor filters (Bellakhdhar et al., 2013). The convolution of image I and Gabor kernel $\varphi_{u,v}(z)$ is defined as follows:

$$G_{u,v}(z) = I(z) * \varphi_{u,v}(z) \quad (2.16)$$

Where z = (x, y) denotes the image position on coordinate x and y, the symbol $*$ denotes the convolution operator, $G_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at orientation u and scale v. The filtering process with Gabor filter with a face image can be rewritten as follows (Shen & Bai, 2006):

$$G_{u,v}(x, y) = I(x, y) * \varphi_{u,v}(x, y) \quad (2.17)$$

The Gabor wavelet coefficient is a complex function with a real and imaginary part, which can be rewritten as:

The convolution of an image with Gabor wavelet can be illustrated using the following mathematical procedures; if f (x, y) represents the intensity at the coordinate (x, y) in gray scale face image, its convolution with Gabor filter $\varphi_{f,\theta}$(x, y) is defined as:

$$g_{f,\theta}(x, y) = f(x, y) * \varphi_{f,\theta}(x, y) \tag{2.18}$$

Where $*$ is the convolution operator and $g_{f,\theta}(x, y)$ denotes the complex convolution result of a face image with a Gabor filter which can be decomposed into a real and an imaginary part; $\mathcal{R}\{g_{f,\theta}(x, y)\}$ and $\mathfrak{Z}\{g_{f,\theta}(x, y)\}$. The magnitude response $\|g_{f,\theta}(x, y)\|$ is expressed as:

$$\|g_{f,\theta}(x, y)\| = \sqrt{\mathcal{R}^2\{g_{f,\theta}(x, y)\} + \mathfrak{Z}^2\{g_{f,\theta}(x, y)\}} \tag{2.19}$$

This magnitude response $\|g_{f,\theta}(x, y)\|$ produces Gabor face features. A Gabor wavelet feature j is described by three key parameters. J(z, μ, v) = $\|g_{f,\theta}(x, y)\|$. Where z represents position, μ denotes orientation and v represents scale. Hence for a given image I(z) with N ×M pixels the number of Gabor wavelet feature representation is N × M × 40.

The convolution of an input image with 40 banks of Gabor filters with 5 different scales (v = 0,1…..4) and 8 orientations (μ = 0,1…. 7) is the 40 magnitude maps that produce the Gabor feature set as shown in Figure 2.16

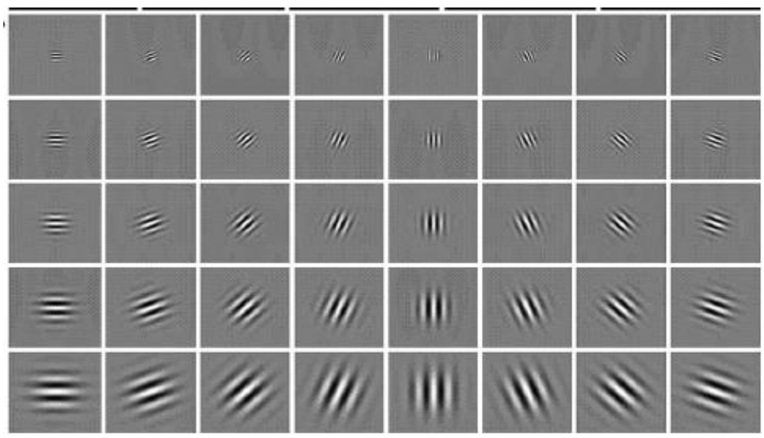


Figure 2.16: The set of 40 banks of Gabor filters for 5 scales and 8 orientations (Thangairulappan & Jeyasingh, 2012)

2.8 Swarm Intelligence Algorithms (SIA)

Swarm intelligence, which is also referred to as a computational intelligence, is newly introduced branch of artificial intelligence that is employed for representation of the collective behaviour of social insects or swarms such as honey bees, bird flocks, ant colonies, fish schooling, microbial intelligence and animal herding (Yang, 2014). A swarm contains a large number of homogenous, simple agents relating locally among themselves and their environments without any centralized control in order to allow a global intriguing behaviour to emerge. Swarm-based systems have appeared as a family of nature-inspired, population–based algorithms that have the ability of producing low computational cost, speed and robust solutions to complex problems (Davidovic, Tedorovic & Selmic, 2015).

The social swarms are simple with limited abilities on their own; they interact together with certain behavioural patterns to relatively achieve responsibilities important for their survival. The swarms' interaction in their environment can be direct or indirect. The direct interaction are through visual or sound contact such as the waggle dance of honey bees. Indirect communication occurs when individual swarm changes the environment and the other swarms respond to the new environment, like the pheromone trails of ants that they deposit on their way to search for food sources.

2.8.1 Artificial Bee Colony Optimization (ABCO)

Artificial Bee Colony Optimization is a swarm-based meta-heuristic algorithm, which is collaboratively applied in solving numerous complex combinatorial problems (Davidovic et al., 2015). This algorithm is inspired by the intelligent behaviour of real honey bees in finding food sources, known as nectar, and the sharing of information about that food source among other bees in the nest. In ABCO approach, the artificial agents are defined and categorized into three types, the employed bee, the onlooker bee, and the scout bee (Wahan, Nefti-Meziani & Atyabi, 2015) . Each of these bees has different tasks assigned to them in order to complete the algorithm's process.

The employed bees focus on a food source and retain the locality of that food source in their memories. The number of employed bees is equal to the number of food sources since each employed bee is associated with one and only one food source (Wong, Low & Chong, 2010). The onlooker bee receives the information of the food source from the employed bee in the hive. After that, one of the food sources is selected to gather the nectar. The scout bee is in charge of finding new food sources and the new nectar.

To apply ABC algorithm, the considered optimization problem is first converted to the problem of finding best parameter vector which minimizes an objective function. Then, the artificial bees randomly discover a population of initial solution vectors and then iteratively improve them by employing the strategies: moving towards better solutions by means of a neighbour search mechanism while abandoning poor solutions.

2.8.2 Bat Colony Optimization (BCO)

The bat algorithm is global optimization approach totally based on the echolocation behaviour of micro-bats with varying pulse rates of emission and loudness (Yang, 2011). Echolocation also referred to as bio sonar is the biological device used by several kinds of animals for navigation and foraging in their environments. Echo-locating animals emit calls out to the environment and listen to the echoes of those calls that return from various objects near them. They use these echoes to locate and detect the objects.

The method of the echolocation of micro-bats can be illustrated as follows: Each virtual bat flies randomly with a velocity V_i at position (solution) X_i with a varying frequency or wavelength and loudness A_i. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate r. Search is intensified by a local random walk. Selection of the best continues until certain stop criteria are met. This essentially uses a frequency-tuning technique to control the dynamic behaviour of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm.

2.8.3 Particle Swarm Optimization (PSO)

Particle swarm optimization is a population-based optimization or computational technique that is used for the optimization problem by iteratively improving a candidate solution with regard to a given measure of quality (Yuce, Packianather, Mastrocinque, & Pham, 2013). This algorithm is inspired by social behaviour such as animal herds, bird flocking and schooling of fish. Normally, a flock of birds that have no leaders will discover food by random movement. They only follow one of the members of the group that has the closest distance with a food source (potential solution) (Rini, Shamsuddin & Yuhaniz, 2011).

The flocks accomplish their best condition concurrently through direct communication among members who already have a better situation. Animal, which has a better condition, will inform it to its flocks and the others will move simultaneously to that place. This would happen repeatedly until the best conditions or a food source discovered. The process of PSO algorithm in finding optimal values follows the work of this animal society.

In the most common implementations of PSO, a set of software agent called particles move through the search space using a combination of an attraction to the best solution that they individually have found (Bratton & Kennedy, 2007) . Each particle is referred to as a solution of the considered problem and uses its own experience and the experience of neighbour particles to choose how to move around in the search space.

2.8.4 Ant Colony Optimization (ACO)

Ant Colony Optimization is a system based on behaviour of natural ants such as cooperation and adaptation mechanism that exist in their colon (Dorigo & Blum, 2005). This technique has been successfully applied to numerous hard combinatorial optimization problems in various domains in order to provide solutions in a reasonable computation time (Aghdam et al., 2009). ACO algorithm was developed based on the social behaviour of natural ants, ants have no vision and are capable of locating the food source by deposition of chemical substance (pheromone) which is used as a form of indirect communication method to mark the route (Bonabeau, Dorigo & Theraulaz, 2000).

The quantity of the laid pheromone is determined by the distance, quantity and quality of the food source. The probability with which an ant selects a route increases with the number of ants that previously selected the same route. This process is population-based method that allows exploitation of a positive feedback loop as search mechanism (Al-ani, 2007).

In Ant Colony Optimization (ACO), a set of software agents called artificial ants search for good solutions to a given optimization problem by transforming the problem to case of finding the minimum cost path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model which is a set of parameters associated with graph components (either nodes or edges) the values of which are modified at runtime by the ants.

Ant Colony Optimization was first introduced to solve the Travel Sales Problem, it was later extended to solve a wide-range of optimization problems such as sequential ordering problem, graph colouring, sheduling, routing in telecommunication network, data mining, quadratic assignment problem, vehicle routing problem and load balancing (Sharma & Rizvi, 2017). ACO algorithms can be applied to solve any optimization problem through implementation of following steps (Imani et al., 2012).

(i) Problem representation (Graph representation): - The problem must be represented as a graph with a set of nodes and edges between nodes. This enables the ants to incrementally design / modify solutions through the application of a probabilistic transition rule, based on the amount of pheromone in trail and on local problem dependent heuristic.

(ii) Heuristic desirability of edges: - A suitable heuristic measure of the goodness of path from one node to every other connected node in the graph.

(iii) Feasible solutions construction: - A technique to ensure only feasible and valid solutions are efficiently constructed.

(iv) Pheromone update rule: - A suitable approach of updating the pheromone levels on edges is required with corresponding evaporation rule. The typical method involves the selection of n best ants and updating the paths they chose.

(v) Probabilistic transition rule: The rule that shows the probability of an ant traversing from one node in the graph to the next.

2.8.5 Meta-heuristic of Ant Colony Optimization

The ACO meta-heuristic is described as a distributed stochastic search technique based on the indirect communication of a colony of artificial ants which is mediated by pheromone trails (Dorigo, Birattari & Stutzle, 2006) as illustrated in Figure 2.17. The ACO algorithm implementation on an image undergoes some modifications. The solution construction space for ants now is the Gabor feature dimensions, artificial ants are made to move over the image dimensions (Dorigo & Stützle, 2004). Thus, the artificial ants simulating the natural ants leave a pheromone on each node or image pixels. The edges of an image become the food for ants. Therefore, in this manner the ants develop pheromone matrix. The following phases take place in ACO meta-heuristic process:

(i) Probability Transition

Probability transition is a function of pheromone value and heuristic information of features. The decision of path taken is influenced by the local intensity value. Parameters considered here are total K ant and $\tau_{i,j}$. $\tau_{i,j}$ is the starting initial value of pheromone matrix, construction of ant solution is made possible through the local search on the solution space (image data matrix). Ants decide to move from node i to another j through probability rules (transition rule) as shown in Equation (2.20).

$$P_{i,j}^{\boldsymbol{k}}(t) = \frac{[\tau_{i,j}(t)]^{\propto}[\eta_{i,j}(t)]^{\beta}}{\sum_{j\in J_i^k}[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}\,(t)]^{\beta}} \text{ , if } j \in j_i^k \qquad (2.20)$$

Where $P_{i,j}^{k}(t)$ is the possibility of choosing node I by ant k while moving from current position node i to node j at iteration time (t). j_i^k is the set of feasible features (nodes) connected to node i that can be traversed by ant k. $\tau_{i,j}(t)$ represents value of pheromone on link i,j between node i and j during iteration (t). $\eta_{i,j}\,(t)$ represents heuristic desirability that gives information about attraction of movement during iteration (t). η heuristic information that denotes the priori information about the problem instance definition. $[\tau_{i,j}]^{\propto}$ is the pheromone amount on the arc connecting node i and node j weighed by α. α and β are two constant parameters that determine the importance of pheromone value and heuristic function. $[\eta_{i,j}]^{\beta}$is the heuristic value of the arc connecting node i and node j weighted by β. α shows the extent to which pheromone information is used as the ants develop their solution. β shows the extent to which heuristic information is used.

(ii) Pheromone Trail Update

The pheromone trails of all arcs on the construction graph are initialized to a small constant value (τ_0). After a trip (solution path) is constructed, the pheromone trails are updated in two manners as illustrated in Equation 2.21 and 2.24. Firstly, the pheromone trails of all arcs are decreased according to an evaporation rate (ρ) that follows ants to forget the suboptimal routes to which they previously converged. Pheromone evaporation rate is usually set to be sufficiently fast in order to favour the exploration of new areas of the search space and avoid a premature convergence of the algorithm toward a local optimum. Secondly, the pheromone trail values of the visited arcs are increased with amount inversely proportional to the cost their trip. During its return trip to the source the *k-th* ant deposits an amount $\Delta\tau^{k}$ of pheromone on arcs it has visited. The Update of pheromone after movement of each ant within each construction step is shown in Equation (2.21).

$$\tau_{i,j}^{k} \leftarrow (1-\rho).\tau_{ij} + \rho.\Delta(i,j)^{k} \quad (2.21)$$

where $\rho \in (0,1)$ is the pheromone update parameter and

$$\Delta(i,j) = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \quad (2.22)$$

$$\tau_{i,j}^{k} \leftarrow (1-\rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \quad (2.23)$$

$$\Delta\tau_{i,j}^{k}(t) = \{ {}^{Q}/_{C^{k}} \; if \; arc\,(i,j) \in \tau_{i,j}^{k} \quad (2.24)$$

Where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{i,j}^{k}$ is the quantity of pheromone laid on edge (i, j) by ant k

(iii) Pheromone trail evaporation

Pheromone trail evaporation makes ants to search for some new paths and in this manner opportunity to find a new shorter path in the unexplored area during the complete search process is opened. This mechanism is referred to as path exploration, which avoids the system quick convergence towards a suboptimal route. ACO exploits the advantage of exploitation and exploration to obtain solution through iteration of the optimal search part. The pheromone trail evaporation seems to be important in artificial ants probably due to the fact that the optimization problems tackled by artificial ants are much more complex than those real ants can solve. Evaporation of pheromone trails is accomplished after movement of all k ants within each construction step as indicated in Equation (2.9)

$$\tau^{(n)} = (1-\varphi).\tau^{k} + \varphi.\tau^{(0)} \quad (2.25)$$

where φ is the decay coefficient.

Step 1: Construction of graph to represent a solution space

Step 2: Set up ACO parameters

Step 3: Initialize ACO parameters

Step 4: Generate ant solutions from each ant's walk on the construction graph mediated by Pheromone.

Step 5: Update pheromone intensities

Step 6. Goto step 3 and repeat until convergence or termination conditions are met

Figure 2.17: Algorithm for ACO Meta-heuristic

2.8.6 Setting of ACO Parameters

The setting of parameters has a great effect on adaptation methods on the performance of the ACO algorithm (Shweta & Singh, 2013). Behaviour of ants is responsible and critical for a speedy convergence to near optimal solutions of the given prob**le**m. The application of adaptation methods is motivated by the fact that an instance-optimal parameter setting always obtains better results than any other setting. ACO applies a random input uniformly distributed in the interval [0, 1].

The most important parameters in ACO algorithm are the influence of the pheromone trails α, the heuristic information β, pheromone update factor ρ and pheromone decay coefficient φ. The strength of randomness in the search process is denoted using α.

The bigger in α value results into weaker random performance, making the algorithm to get to local optima. β represents the strength of apriority and certainty factors in the process of searching. The bigger the β value, the higher the possibility that the ants choose the local best result. High value of φ implies high pheromone decay rate which makes the algorithm fast in solution construction.

2.9 Image Preprocessing / Normalization

Image preprocessing involves a process of removing undesirable distortion due to deterioration in contrast, unwanted noise, inappropriate intensity permeation and blurring condition in a face image (Singh & Dixit, 2015). It is necessary in pattern recognition, 2-Dimensional face recognition, medical fields and computer vision to improve input images in order to bring out hidden information that are contained in a digital image. The pre-processing includes the cropping, scaling, rotating, contrast adjustment and translating of image ((Mounika, Reddy & Reddy, 2012). This technique involves the enhancement of face images just to deliver the most relevant and measurable features (Lalit & Ankur, 2015). The image preprocessing entails histogram equalization and fuzzy filtering.

2.9.1 Histogram Equalization (HE)

This is one of the methods in which an image contrast is adjusted or enhanced using image's histogram (Aarthy & Sumathy, 2014). The equalization of histogram is used to remap gray levels of image based on probability distribution function (PDF) of input image gray level (Gupta & Kaur, 2014). Histogram equalization (HE) flattens the histogram and stretches dynamic range of gray levels to perform overall contrast enhancement (Gonzalez & Wood, 2000). The image contrast enhancement is a classical problem in computer vision and image processing (Krishna, Rao & Sravya, 2013),

HE compensates and also improves the lighting conditions of an image. The Contrast of an image is element that can be defined as a ratio between the highest and lowest pixel intensities of a digital image. Images of different human face have the same contrast if only their root mean square (rms) contrast equal. The rms does not depend on spatial distribution of contrast in an image. Histogram equalization algorithm enables regions of lower local contrast to gain higher contrast. For image F(x, y) with discrete W gray values histogram is defined as the probability of occurrence of the gray level F

$$P(F) = \frac{n_F}{N} \qquad (2.26)$$

Where F ∈ 0, 1………. W-1 gray level and N represents the total number of pixels in the image. Transformation to a new intensity values is defined as:

$$F_{out} = \sum_{F=0}^{W-1} \frac{n_F}{W} = \sum_{F=0}^{W-1} P(F) \qquad (2.27)$$

Output values are from domain [0, 1]. To obtain pixel values in original in original domain, it must be rescaled by W-1 values. The histogram equalization is represented in Figure 2.18

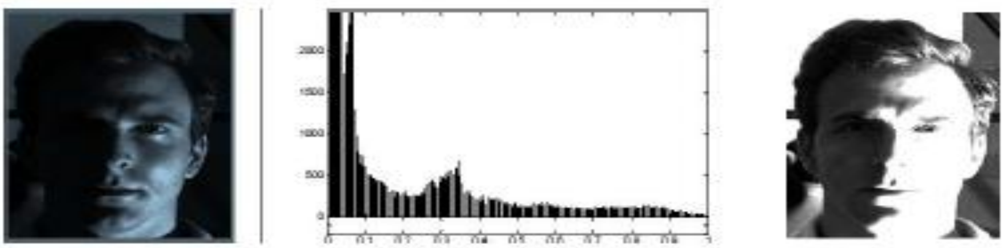


Figure 2.18: Linear Histogram Equalization from left to right (Anila & Devarajan, 2012).

2.9.2 Algorithm for Histogram Equalization

The histogram equalization process can be expressed as follows:

Given image X = {X(i,j)}, composed of L discrete gray levels denoted as {X_0, X_1, …X_{L-1}} where X(i,j) represents an intensity of image at the spatial location (i,j) and X(i,j) ∈ {X_0, X_1…. X_{L-1}}. For image X, probability density function P(X_k) is defined as:

$$P(X_k) = \frac{n^k}{n} \qquad (2.28)$$

For k = 0, 1, ……, L-1, where n^k represents number of times X_k appears in input image X and n is the total number of samples in input image. Here P(X_k) is associated with histogram of input image which represents number of pixels having specific intensity X_k. A plot of n^k against X_k is known as histogram of X. The cumulative density function (CDF).

$$C(x) = \sum_{j=0}^{k} P(X_k) \qquad (2.29)$$

Where X_k = x, for k = 0,1….. L-1, Here C(X_{L-1}) = 1 by definition. HE is scheme which maps input image into the entire dynamic range (X_0, X_{L-1}) by using CDF as transform function.

There are various histogram equalization such as Adaptive Histogram Equalization and Contrast Limited Adaptive Histogram Equalization.

2.9.3 Adaptive Histogram Equalization (AHE)

This is extension to traditional histogram equalization that enhances contrast of images by transforming values in the intensity image (Garg, Mittal & Garg, 2011). It is used for contrast enhancement in natural images, medical and non-visual images (Thamman & Bhatia, 2014). AHE is completely different from the traditional histogram equalization due to the fact that it computes several histograms, each corresponding to distinct section of the image and applies then to redistribute the lightness values of the images (generating the mapping of each pixel from the histogram in a surrounding window) (Stark, 2000). AHE is suitable for improving the local contrast (contextual region rather than entire image) and enhancing the definition of edges in region of an image.

2.10. Classification Techniques

This is a process of classifying any given input feature vector into predefined set of classes (Derakhshii & Ghaemi, 2014). Classification can be considered as a method with major objective of obtaining optimal patterns based on certain conditions in order to separate one class from the others (Deng et al., 2005). The classification involves the taking of extracted features from an image and them using to automatically classify images. An image can be classified based on three different concepts; similarity, probability and decision boundaries. The abstraction is to develop a metric that defines similarity measures or distance and them represent the same class samples (Sergyán, 2009).

In the probability concept, classifiers are generally designed based on the probabilistic approach. Bayes decision is commonly used method for probabilistic based classifier construction. This rule can be modified to take into consideration several factors that could lead to miss-classification. Decision boundaries is a concept that depends on chosen metric, this method minimizes a measurement of error between the candidate pattern and the testing pattern. There are various classification methods in face recognition such as Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) which is also known as distance measure classifiers (Khan, Ahmad, Nazir & Riaz, 2013).

2.10.1 Artificial Neural Network (ANN)

It is a popular, powerful and robust classification method which can be used for predicting not only known data but also unknown data. This technique for face recognition has better performance of more than 90% acceptance ratio (Kumar & Kaur, 2012). The ANNs have been applied to many areas like speech recognition, interpreting visual scenes, fingerprint recognition, face recognition and iris recognition (Sergyán, 2009). Neural Networks model the neurons system in biological organisms and are thus referred to as artificial neural networks. The mathematical models have been developed for these biological systems based on simple processing element called neuron. These neurons function by outputting a weighted sum of inputs and these weights are generated during the learning or training phase. The neural network consists of input, output and hidden layers (Deotale, Vaikole & Sawarkar, 2010) as shown in Figure 2.19.

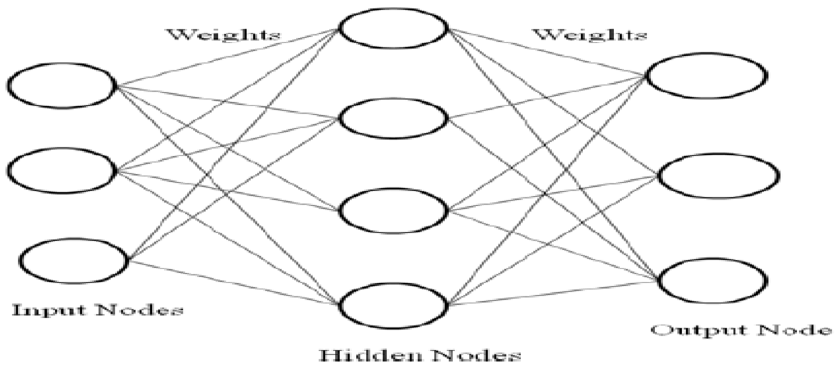


Figure 2.19: Structure of Neural Network (Kasar, Bhattacharyya, & Kim, 2016)

2.10.2 Support Vector Machines (SVMs)

These classification methods are formulated to find solution to a classical two class pattern recognition problem (Phillips, 1999). They achieve recognition between classes by finding a decision surface that has maximum distance to nearest point in the training set which are known as support vectors (Heisele, Ho & Poggio, 2001). SVMs have been used to solve many practical problems such as speaker identification, face detection, text categorization and isolated handwritten digit recognition (Munir, Gupta, Nemade & Alam, 2011). SVMs are based on an algorithm that finds a special kind of linear model called the maximum-margin hyper plane. The instances that are closest to the maximum-margin hyper plane, the ones with the minimum distance are called support vectors.

2.10.3 Distance Classifiers

This is a simplest techniques for identifying object based on closest training samples in the feature space (test set) (Imandoust & Bolandraftar, 2013). The object of interest is compared to every sample in training set by applying a distance measure, similarity measure or the combination of two measures (Gawande & Agrawal, 2014). The unknown object is recognized as fit in to the equivalent class as the closest sample in training set, this is identified by the smallest number if it is a distance measure or largest number it applies similarity measure. Many facial recognition systems perform their classification based on a distance measure (Rady, 2011). The distance measure methods have been applied in many applications such as text categorization, pattern recognition, data mining and object recognition (Alkasassbeh, Altarawneh, & Hassanat, 2015).

Various distance measures can be used as similarity measure to compare the feature vector of test image with that of trainee images. All the trainees as well as the test image are projected to the feature space of training dataset. Distances between the projected test image and the projection of all centred trainee images are calculated. Test image is supposed to have minimum distance with its corresponding equivalent image in the training dataset. There are various examples of distance measure methods of classification such as Euclidean, Manhattan, Chebyshev and Malanobis distance (Sodhi & Lal, 2013).

(i) Euclidean Distance

The Euclidean distance is the commonly used distance classifier in most applications. Euclidean distance classifier is used in testing for the calculation of minimum distance between test image and image to be classified from database. It is referred to l_2 distance (Sharma & Batra, 2014). The Euclidean distance can be illustrated as follows: If $u = (x_{1,} y_1)$ and $v = (x_{2,} y_2)$ are two points, then the Euclidean Distance between u and v is given by

$$EU(u, v) = \sqrt{(x_1 - x_2)^2 + (y_1 - x_2)^2} \quad (2.30)$$

If the dimensions is of n terms, such as $a = (x_{1,} x_2, \ldots\ldots x_n)$ and $b = (y_{1,} y_2, \ldots\ldots y_n)$ then Equation 2.30 can be generalized by defining the Euclidean distance between a and b (Thakur & Sahayam, 2013) as

$$EU = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \quad (2.31)$$

(ii) Chebyshev Distance

This is a metric defined on a vector space where the distance between vectors is the greatest of their differences along any coordinate dimension (Udovychenko, Popov, & Chaikovsky, 2015). The chebyshev distance between two vectors on point p and q, with standard co-ordinates p_i and q_i as shown by the following formula.

$$\text{Dchebyshev(p,q)} = max_i \left(|p_i - q_i| \right) \quad (2.32)$$

This equals the limit of the l_p metrics:

$$\lim_{k\to\infty} (\sum_{i=1}^{n} |p_{i-}\ q_i|^k)^{\frac{1}{k}}$$

In the two dimensions, if the points p and q have Cartesians coordinate $(x_{1,}y_{1,})$ and $(x_{2,}y_{2,})$ then the chebyshev distance is

$$\text{Dchebyshev} = \max\ (|x_{1\ -}\ x_2|),\ |y_{2\ -}\ y_1| \qquad (2.33)$$

(iii) City-Block (Manhattan Distance)

It examines the sum of absolute differences between the two feature vectors (Ponnmoli, 2014). It is real distance function because it follows the triangle inequality. As Manhattan or Taxicab distance is that passing from a point A to point B is achieved by walking around the block compared to straight line in Euclidean distance (Sodhi & Lal, 2013). The Manhattan Distance (MD) between a point u and v can be illustrated as follows:

If u = (x_1, y_1) and v = (x_2, y_2) are two points, then MD (u, v) is

$$\text{MH}\ (u, v) = |x_1 - x_2| + |y_1 - y_2| \qquad (2.34)$$

with points of u and v dimensions, as in a = $(x_1, x_2 \ldots\ldots, x_n)$ and b = $(y_1, y_2 \ldots\ldots., y_n)$ then equation (2.34) is generalized as:

$$\text{MH}\ (u, v) = |x_1 - y_1| + |x_1 - y_2| + \ldots\ldots\ldots.\ |x_n - y_n| = \sum |x_{i-}\ y_i| \qquad (2.35)$$

for i = 1,2, …………., n

Where i is the number of variables x_i and y_i are the values of the ith variables at point x and y respectively.

(iv) Malahanobis Distance

It is a descriptive statistic that gives a relative measure of data points distance from common point (Patil & Kiran, 2014). The Malanobis distance shows dissimilarity measure between two random vectors $\vec{x}, \vec{y}$ of same distribution with covariance matrix S (Sharma & Batra, 2014) as shown in Equation (2.36). Malahanobis distance of observation x = $(x_{1,} x_{2}, x_{3,} \ldots\ldots\ldots x_{N})^{T}$ from set of observation with mean μ = $(\mu_{1,} \mu_{2}, \mu_{3,} \ldots\ldots\ldots \mu_{N})^{T}$ and covariance matrix S is defined as:

$$D(x) = \sqrt{(x - \mu)^{T} S^{-1} (x - \mu)} \tag{2.36}$$

$$D(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^{T} S^{-1} (\vec{x} - \vec{y})} \tag{2.37}$$

2.11 Related work

Face representation remains a stern problem in facial biometric recognition system. Gabor filters as an effective facial representation still require that feature space should be reduced to feature subspace. Several studies have been done applying Gabor filter with different dimensionality reduction techniques. The followings are works achieved by researchers:-

Dora, Agrawal, Panda and Abraham (2017) made attempt to design a single optimized Gabor filter instead of a filter bank for feature extraction. This approach improved the filter performance by significantly reducing the computational complexity and response time. Hybrid particle swarm optimization-gravitational search algorithm (PSO-GSA) was employed for optimizing the parameters of the single Gabor filter. In this context, an evolutionary single Gabor kernel (ESGK) based filter approach was developed for face recognition. The proposed system extracted Gabor energy feature vectors from face images. The study used new eigenvalue based classification approach for face recognition. FERET, ORL, UMIST, GT and LFW databases were used to measure the efficiency of the proposed method. The experimental results were compared with a holistic Gabor filter bank based recognition methods. The proposed method outperformed the state-of-the-art methods.

Zhou, Zhang, Mei and Wang (2016) developed an algorithm based on the Gabor wavelet transformation and non-negative matrix factorization (G-NMF). The process included image pre-processing, feature extraction and classification. At first phase, the face region containing emotional information was obtained and normalized. Thereafter, expressional features are extracted by Gabor wavelet transformation and the high-dimensional data are reduced by non-negative matrix factorization (NMF). Finally, two-layer classifier (TLC) was applied for expression recognition. The system was evaluated using JAFFE facial expressions database. The results show that the method proposed has a better performance.

Ali Jan Ghasab et al., (2015) presented an automatic recognition system for different plant species through their leaf images by applying the ant colony optimization (ACO) as a feature decision-making algorithm. The ACO algorithm was used to investigate inside the feature search space in order to find the best discriminant features for the recognition of individual species. To obtain a feature search space, a set of feasible properties such as shape, morphology, texture and colour are extracted from the leaf images. The selected features were used by support vector machine (SVM) to classify the species. The performance evaluation was conducted using two datasets; FCA and

Flavia, of about 2050 leaf images. Experimental results of the study achieved an average accuracy of 95.53% from the ACO-based approach.

A face recognition was developed by applying Gabor-filter based feature extraction method with Anisotropic Diffusion as an unique pre-processing technique to improve the performance of face recognition algorithm (Abhishree, Latha, Manikantan, & Ramachandran, 2015). Gabor filter was employed to capture the features of face image aligned at specific angles. A binary particle swarm optimization based feature selection algorithm was applied to reduce the number of features used for classification decisions. This decreased the testing and improved the speed of face recognition system. The model was evaluated on four benchmark facial image datasets; ORL, Color FERET, Cropped Yale B and FEI datasets. The result showed outstanding performance compared with existing methods in the presence of pose, illumination and expression variation.

A new approach to improve 3D face recognition system performance was developed (Hafez et al., 2015). The model pre-processed and normalized all images in the database using 2D normalized cross correlation 2DNCC. The 3D face features were extracted by applying a set of selected orthogonal Gabor filters, which minimized the feature vectors extracted when compared to those ones that use complete Gabor filters bank. The study further employed linear discriminant analysis to compress dimensionality of the extracted features before classification. The model tested on CASIA and Gavab 3D face image database achieved 98.35% and 85% respectively. Experimental results showed the effectiveness of the system in term of dimensionality and recognition accuracy when compared with existing systems.

A comparative study on Face recognition Gabor wavelet features with PCA and KPCA (Vinay et al., 2015). The study used Gabor wavelet for feature extraction system. The face recognition algorithm applied Principal Component Analysis (PCA) and Kernel Principal Component Analysis (KPCA) for a post processing method after feature extraction to reduce dimensionality of filtered face images. Comparison analysis was carried out between the two methods. The model was evaluated using the publicly available ORL database. Results reviewed that GABOR-PCA outperformed the GABOR-KPCA method for facial recognition.

Khryashchev, Priorov, Stepanova and Nikitin (2015) proposed a model for accurate and robust face recognition by employing Gabor filters and local quantized patterns. The study mentioned that additional application of Gabor filters can significantly improve the robustness to changes in lighting conditions of face recognition system. The eye centers estimation was applied at the pre-processing phase. The evaluation of the algorithm on diverse samples from a standardized FERET database showed that the proposed technique was invariant to the general variations of lighting, expression, occlusion and aging. Recognition accuracy of 20% increase compared with the known face recognition algorithms from the OpenCV library.

Mohammed (2014) developed a face recognition system using principal Gabor-filter. The system extracted facial features through the use of Gabor-filter. The feature vectors from Gabor filter were used as the input of the classifier, which was a feed forward neural network on reduced feature subspace learned by Principal Component Analysis for recognition of face images. The effectiveness of the system was justified over a face database with faces captured under different illuminations and orientations.

Elgarrai et al., (2014) conducted a face recognition system using Gabor features and HIK Toolkit. Facial image features are extracted using Gabor filters. The dimensionality of those features was reduced using the linear discriminant analysis (LDA) method to keep only the most relevant information. The system injected the resulting feature vectors to the Hidden Markov Model Toolkit (HIKT). Experimental results on YALE and ORL database showed the efficiency of the proposed system.

Pal, Chourasia and Ahirwar (2013) proposed a face detection system based on wavelet transformation function for feature extraction. Ant Colony optimization technique was introduced for feature selection. The optimized features were passed into Support Vector Machine for classification. The proposed method was compared with PCA and SVM method for detection of group image. Experimental result showed better performance in compression of PCA and SVM.

Bellakhdhar et al (2013) came up with a face recognition system using Gabor wavelets, Principal Component Analysis (PCA) and Support Vector Machine (SVM). The model combined the magnitude and phase of Gabor filter. PCA was applied to extract the eigenvectors of the face images with SVM for classification. The performance of the model tested on the public and largely used datasets of FRGCV2 and ORL. Experimental results showed that the combination of the magnitude with the phase of Gabor features can achieve better results.

Imani et al. (2012) developed a new hybrid of Ant Colony Optimization (ACO) and Genetic Algorithms (GAs) as a feature selector and Support Vector Machine as a classifier are integrated effectively. Based on the combination of the fast global search ability of GA and the positive feedback mechanism of ACO, a novel algorithm was proposed in the domain of feature selection. The system tested on the extracted features of ten common Persian fonts. Experimental result showed that proposed feature selection achieved better performance than normal GA and ACO.

Mounika et al (2012) proposed a neural network based face detection system using Gabor filter response. The study convolved a face image with series of Gabor filter coefficients at different scales and orientations. Gabor features were fed to feed forward neural network based on Bi-directional Memory (BAM) for dimensional reduction and multilayer perception with back propagation algorithm for training the Gabor features. The recognition performance was improved by contrast equalization using the root mean square value of image pixel. The application of hybrid network (Bi-directional Association Memory (BAM) and Back Propagation Neural Network (BPNN)) takes less iteration to train and less time to recognize. The effectiveness of their recognition model was evaluated using Yale database with different illumination conditions. An efficient face recognition using Gabor filters to extract facial features presented (Thangairulappan, Jeyasingh, Beulah, Jeyasingh, & Jeyasingh, 2012). In the study, the large dimensional Gabor features were reduced by wavelet transformation. Discriminative common vectors were obtained using within-class matrix method to get a feature representation of face images with enhanced discrimination and also classification was done using radial. Radial basis function network. The proposed system was validated with three face datasets: ORL, JAFFE and Essex face database. Experimental results showed that the recognition model reduces the number of features, minimizes the computational complexity and also yielded better recognition accuracy.

An intelligent neural network system for facial recognition proposed by (Bouzalmat et al., 2011). Gabor filters used for the feature extraction as these filters present desirable characteristics of spatial locality and orientation selectivity. The study applied Sparse

Random projection (RP) technique to reduce large feature dimensions into feature subspace. Back Propagation Neural Network (BPNN) was used on the feature vectors for classification. The face recognition model was evaluated using AR database with a collection of twenty people from database. Each person represented by twenty samples, ten used for training and ten for testing. The recognition system achieved higher recognition rate and better classification efficiency when feature vectors have low dimensions.

Karnan, Akila and Kalamani (2009) developed a feature subset selection in keystroke dynamics for identity verification. The results of experimenting Ant Colony Optimization technique on keystroke duration, latency and digraph for feature selection. ACO was used to reduce the redundant feature values and also minimized the search space. Optimal feature subset was obtained using keystroke duration values when compared with other two feature values.

Principal Gabor-filters for face recognition (Štruc, Gajšek, & Pavešić, 2009). A novel orthogonal Gabor filters constructed from the linear combination of the original Gabor filters to reduce the problem of high dimensionality of Gabor features. The novel filters were derived from correlation matrices of the original filters by means of principal component analysis. XM2VTS and YaleB datasets used to evaluate their model. Experimental results obtained in a series of verification and identification analyses reviewed that the new filters result to better performance with a significantly reduced computational complexity.

Daoud (2009) proposed enhancement of face recognition using modified Fourier Gabor filter. They performed experiments to verify the effectiveness of the proposed model using five popularly used techniques in face recognition applied to four datasets

(AT&T, IFD, Faces 95 and Yale datasets); the methods are implemented without and with suggested filters. The experimental results showed that using suggested Fourier-Gabor filter enhances the classification rates for all methods, datasets, training and testing percentage. The highest classification rates obtained when Fourier Gabor filter with batch linear discriminant analysis (FG-Batch-ILDA).

A Gabor filter coefficient-based neural network method for face recognition (Bhuiyan & Liu, 2007). The study convolved a face image with a series of Gabor filter coefficients at different scales and orientations. Contrast equalization and fuzzily skewed filter technique were introduced at image pre-processing phase which contributed to the performance of the face recognition model. Fifteen Gabor filters (three for scaling and five for orientations) were used in order to reduce the huge dimension of Gabor features. A neural network based on multi-layer perceptron (MLP) architecture with back propagation algorithm was applied for classification.

CHAPTER THREE

RESEARCH METHODOLOGY

This chapter discusses the approach used in this study. It presents the steps in image processing which involves the extraction and selection of facial features using Gabor-filters and ACO respectively. The data and their collection are presented. The parameters (metrics) for evaluating the performance of the ACOFG system are also defined in this chapter.

3.1 ACO Gabor Feature-Based Facial Recognition System (ACOFG)

The development of the system commenced with the creation of face image databases. This study employed two face image datasets for training and testing phase; Olivetti Research Laboratory (ORL) database and locally acquired image database (Locally Acquired Images of Students from University of Ilorin). All face images were taken against a dark homogenous background with the subjects (images) in uprights and frontal positions. Preprocessing / normalization technique was performed on each facial image of the two datasets by cropping, resizing of facial image and the contrast adjustment (illumination normalization was done using Adaptive Histogram Equalization). Gabor filters were used to obtain facial features representation which represents Gabor features. The meta-heuristic Ant Colony Optimization Algorithm was applied to select optimal features from high Gabor feature dimensions. The generated feature subsets were used as template for matching with different distance measures. The block diagram of system is shown Figure 3.1.

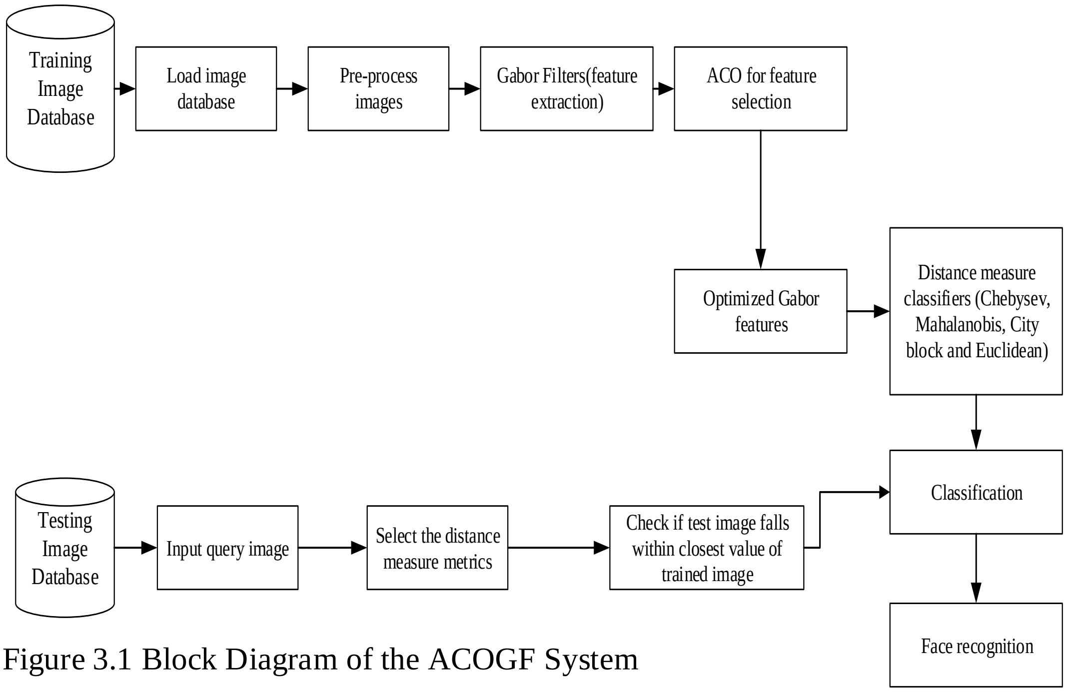


Figure 3.1 Block Diagram of the ACOGF System

3.2 Acquisition of Face Images

The evaluation of face recognition system is necessary and should be carried out with image database. There is a need to create an image database of different people. Although, certain provision has been made available for several standard online face datasets, general features of these commonly available datasets include face image taking in a well-controlled environment and image tailoring towards a specific requirement of an algorithm. On the other hand, the performance of face recognition system is affected when datasets used to benchmark the algorithms changes due to differences in facial features from race to race. These conditions result to the need for a new database. Facial Images used in this work was acquired from Olivetti Research laboratory (ORL) Database and locally acquired black faces were captured from student of the University of Ilorin.

3.2.1 Locally Acquired Facial Image (LAFI) Database

The original frontal face images of various students of the University of Ilorin were captured using static HP Digital Camera. The images were taken under different lighting conditions and backgrounds to form a facial database. It has been duly considered that there can be a change in the expressions of the people. So this database was designed with different expressions for all subjects for better results. The LAFI face database consist of images of 60 person with 5 images per person, which gave the total number of 300 face images captured from the students of University of Ilorin**.** The images in database were split into two (Jemma & Khanfir, 2009). 240 **(**80%) of the total number of images were used for training while the remaining 60 (20%) images

were used for testing.

Figure 3.2. Sample of Faces (LAFI Database)

3.2.2 Olivetti Research Laboratory (ORL) Database of Faces

The ORL face database was developed at the Olivetti Research Laboratory in Cambridge, United Kingdom between April 1992 and April 1994. The face image database is composed of 10 different face images from 40 people. Images were taken at different times, varying the lighting (illumination conditions not consistent from image to image), expressions (open, closed eyes and smiling) and occlusions (glasses, no glasses) as illustrated in figure 3.3. Only 300 face images were considered from the database, 240 (80%) images were employed for training and 60 (20%) images for

Figure 3.3. Sample of Faces (ORL Database)

testing.

3.3 Image Pre-processing

The images in two face image datasets were preprocessed for normalization. The preprocessing phase involves the improvement with deference to the quality of the image, but not to head position (tilt) or emotion. Locally acquired face images were properly preprocessed since they were not captured under a controlled condition compared to what is obtainable in ORL database. The pre-process methods involve Geometrical Normalization, Image Grayscale conversion and Photometric normalization.

3.3.1 Geometrical Normalization

The region of face images from two databases were cropped with the removal of parts like ear and fore-head with no distortions from the original size of the image picture using Adobe Photoshop Software. Each image was then resized into following sizes 75 x 75, 100 x 100, 125 x 125, 150 x 150 pixels, resulting into 5625, 10,000, 15625, 22500 dimensions.

3.3.2 Conversion of Face Image from Colour to Gray-scale

The colour face image is converted to gray-scale in order to retain much information contained in face images and reduce processing requirement (Suganya & Menaka, 2014). A RGB image is referred to as image in which each pixel is specified by three channels depicting red, green and blue components of the pixel scalar, thus this makes the RGB image a 3-dimensional information image. A gray scale image is a digital image containing only single channel which is the intensity information, this makes it 1-dimensional image with processing requirements. When converting RGB image to gray scale, the R, G and B value must be picked for each pixel and also make a single value reflecting the brightness of that particular pixel. The original colour images from the LAFI database were converted into grayscale images. The pseudocode for grayscale of an image as shown Figure 3.4.

```
For Each Pixel in Image {
    Red = Pixel. Red
    Green = Pixel.Green
    Blue = Pixel.Blue
    Gray = (Red + Green + Blue) / 3
    Pixel.Red = Gray
    Pixel.Green = Gray
    Pixel.Blue = Gray
}
```

Figure 3.4: Pseudocode for Grayscale of an Image (Saravanan, 2010)

3.3.3 Illumination Normalization

Illumination variation is one of the major challenges in face recognition which can be solved by illumination normalization. The function of illumination normalization is to transform images with arbitrary illumination condition to standard illumination invariants ones. Images obtained using cameras over longer distances or under poor visibility conditions are often not suitable for investigation and observation. Only marginal improvement in the images captured from the camera can be captured using standard camera settings such as contrast, diaphragm, brightness and shutter time.

The images gathered are subject to some illumination variations. So therefore, it is necessary to conduct normalization process on face images of the datasets to eliminate illumination variations, improve contrast and standard visual quality in order to have uniform histogram equalization for all images. Adaptive Histogram Equalization with high information retaining capability compared to conventional histogram equalization was used to normalize illumination effects. In adaptive histogram algorithm each pixel is modified based on the pixels in a region (contextual region) surrounding pixels. The pseudocode for Adaptive Histogram Equalization is shown in Figure 3.5

```
for each (x, y) in image do
    {
            rank = 0
            for each (i, j) in contextual region (local area) of (x, y) do
        {
            If image [x, y] > image [i, j] them
            rank = rank + 1
        }
            Output [x, y] = rank × max_intensity / No of pixels in contextual region (local area)
        }
```

$$\text{Output } [x, y] = rank \times \frac{\max\_intensity}{No\ of\ pixels\ in\ contextual\ region\ (local\ area)}$$

Figure 3.5 Algorithm of Adaptive Histogram Equalization (Zhu & Huang, 2012)

3.4 Gabor-filters for Facial Features Extraction

This module finds the key features in the face region that will be employed in classification phase. It is responsible for composing a feature vector that is well enough to represent the image. Gabor filters was applied for extraction of features in this module. The first stage involved the design of Gabor filters parameters. The parameters were set as follows: 8 orientations $\theta_{\mu} = \frac{\mu\pi}{8}$, $radius$ $0 \leq \mu \leq 7$. the wave vector $\overrightarrow{K} = K_V\,(cos\theta_{\mu}\overrightarrow{V_x} + sin\theta_{\mu}\overrightarrow{U_y})$ has angle independent magnitude. $K_v = \frac{K_{max}}{f^v}$, with $K_{max} = \frac{\pi}{2}$, frequency ratio (5 scales) f = $\sqrt{2}$, $0 \leq V \leq 4$. The response of an image I to a wavelet φ is calculated as the convolution G = I $* \varphi$. The face image of size 75 x 75, 100 x100, 125 x 125 and 150 x 150 image pixel from the two datasets (ORL and LAFI Database) were convoluted separately by applying designed Gabor-filters on each face image as shown Figure 3.6. The convolution result was further decomposed into real and imaginary part. For this two parts, magnitude of filter responses calculated and then concatenate to produce Gabor features in this study.

Step 1: Input face image from the image database

Step 2: Check if the face image is gray scale, if the image is not, thus gray scale

Step 3: Preprocess the gray-scale face image

Step 4: Design filter-banks (setup parameters for Gabor-filters)

Step 5: Apply the created filter on preprocessed face image by convolution of face image I (x, y) with a filter bank containing filters of different 5 scales (u) and 8 orientations (v).

Step 6: Decompose convolution output $G_{u,v}(x, y)$ into complex values of real and imaginary part

$$G_{u,v}(x, y) = I(x, y) * g_{u,v}(x, y)$$
$$E_{u,v}(x, y) = Re[G_{u,v}(x, y)]$$
$$O_{u,v}(x, y) = Im[G_{u,v}(x, y)]$$

Step 7: Compute the magnitude $A_{u,v}(x, y)$ of filter responses and $\emptyset_{u,v}(x, y)$ of phase

$$A_{u,v}(x, y) = \sqrt{E^2u, v(x, y) + O^2u, v(x, y)}$$
$$\emptyset_{u,v}(x, y) = \arctan(\frac{o_{u,v}(x, y)}{E_{u,v}(x, y)})$$

Step 8: Discard the Gabor phase features

Step 9: Concatenate magnitude of the Gabor responses of convoluted image into face image

Step 10: End

Figure 3.6: Algorithm for 2-Dimensional Gabor-filters for Feature Extraction

3.5 Feature Selection of Gabor Feature Using ACO

The image data matrix obtained from the feature extraction phase was passed to ACO meta-heuristic Algorithm. A Region of Interest from the output of Gabor feature dimensions were defined from the features based on the information of ROI that do not change over period of time. The ROI was passed unto ACO for optimal feature selection. Based on feature correlation, a subset of the Gabor feature ROI was selected, then was used as the permissible range of ant movement. The ants move randomly over the Gabor features to construct a pheromone matrix.

The size of the pheromone matrix for this study varies with different Gabor feature matrix of image of the cropped image, the heuristic desirability which is the measure of attractiveness of a feature image based on the local statistics was obtained using equation (2.24). The heuristic desirability is obtained by computing the correlation between pairs of pixels. The construction process of solution by the ant was carried out by adopting the probabilistic transition rule in equation (2.20). Global and Local Pheromone update were performed by employing equation (2.21) and equation (2.23) respectively. This resulted to a subset of pixels which represents the optimal features of Gabor feature dimensions

Step 1: Create ACO parameters α, β, ρ, τo, η, φ, K, N,

Step 2: Initialize pheromone matrix on Gabor-filtered image data matrix

Step 3: Construct solution

$$P_{i,j}^{\boldsymbol{k}}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{\boldsymbol{j}\in \boldsymbol{J}^{k}}[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}$$

Step 4: Find the first update of the pheromone matrix (Global update of pheromone matrix)

$$\tau_{\mathrm{i,j}}^{\mathrm{k}} \leftarrow (1-\rho).\tau_{ij} + \rho.\Delta(\mathrm{i,j})^{\mathrm{k}}$$

where $\rho(0 < \rho \leq 1)$ is the parameter for pheromone update

Step 5: All K ants move over Gabor features **Goto STEP 6** or If All K ants do not move else **Goto STEP 2.**

Step 6: Find the second update of pheromone matrix (Local update of pheromone matrix).

$$\tau^{(n)} = (1-\varphi).\tau^{k} + \varphi.\tau^{(0)}$$

Step 7: If all iterations performed **Goto STEP 8** or if not else **Goto STEP 2.**

Step 8: Return optimal feature subsets.

Step 9: End

Figure 3.7: Algorithm of ACO for Optimal Feature Selection

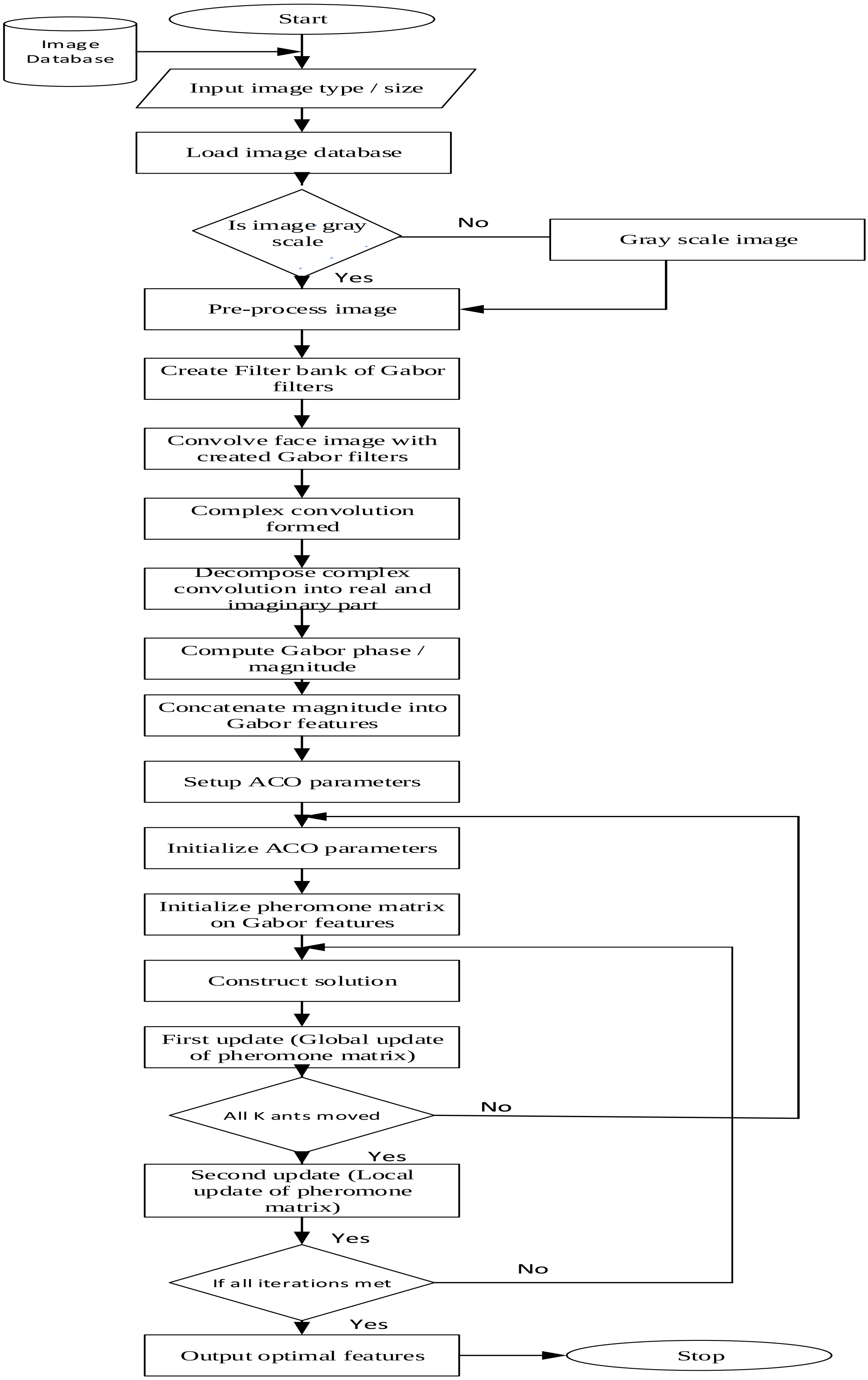


Figure 3.8: Algorithm of the proposed Feature Extraction and Feature Selection

Step 1: Select face image database type and size

Step 2**:** Load face image database

Step 3: Preprocessing / Normalization of face image

Step 4: Apply Gabor filter on preprocessed face image, convolution of face image I (x, y) with a filter bank containing filters of different 5 scales (u) and 8 orientations (v).

Step 5: Apply Ant Colony Optimization Algorithm on the extracted Gabor features

Step 6: Compute optimized Gabor features (Gabor feature subsets)

Step 7**:** Select distance measure metrics, City block, Malanobis, Euclidean and Chebysev to be applied on optimized Gabor features for classification.

Step 8: Test image within training template of distance measure classifiers for matching,

Step 9: If test image falls within the closest distance measures (Threshold value) image matched or mismatched.

Step 10: End.

Figure 3.9 Algorithm of ACOGF System

3.6 Training Phase for ACOGF System

In this phase, face images were loaded from one of the two datasets (ORL and LAFI) into system. The images were preprocessed and thereafter normalized using adaptive histogram equalization. The features were extracted from each facial image loaded using a filter bank of different orientations and scales of Gabor-filters. The Gabor features converted into image data matrix were passed into ACO, where optimal features were selected. The optimal features representation obtained were used for training and template for matching with four distance measure classifiers; City-block (Manhattan), Euclidean, Mahalanobis and Chebyshev. All the training face images were used by distance classifiers to compare with the test face image as indicated in Figure 3.10.

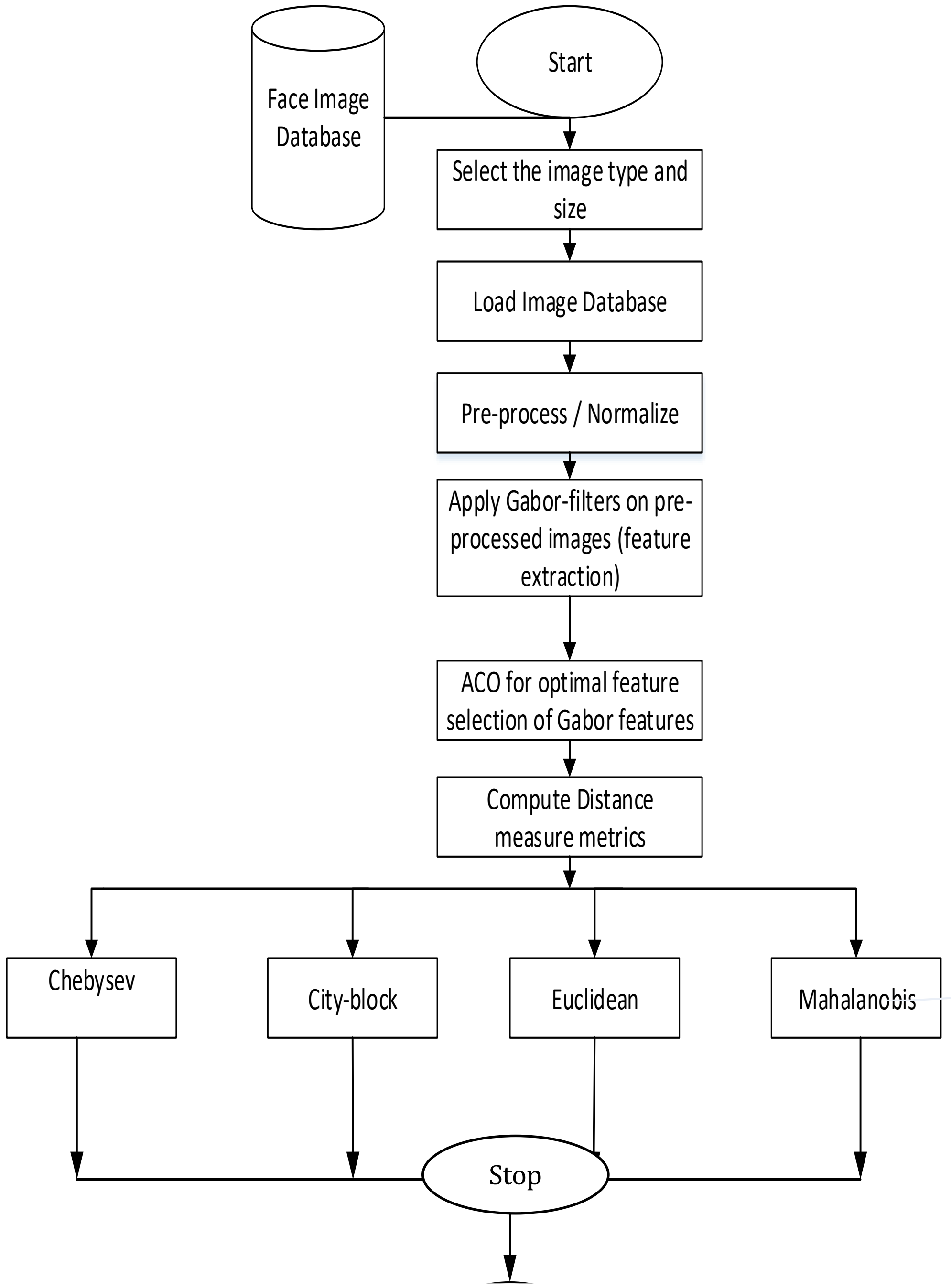


Figure 3.10: Training Phase of the ACOGF System

3.7 Classification Phase of ACOGF System

The most discriminating features from Gabor features were selected by ACO from the face region and then encoded so that comparison between patterns can be obtained. The feature vectors corresponding to this subset of pixels were used for the recognition process as shown Figure 3.11. Four distance measures for these feature vectors and the test image vector is determined by comparing the vectors of the test image and each of the trained images as shown in Figure 3.6. The distance measures of a projected test image from all the trained images were computed and the minimum value is chosen in order to get the train image which is most similar to the test image. The test image is expected to fall in the same class that the closest train image belongs.

Given an input query face to be recognized, the first phase is to preprocess face image. Gabor filters apply to obtain equivalent Gabor features which represent the entire original face. The feature subset selection is achieved by applying ACO in order to reduce and also obtain optimized Gabor feature dimensions as illustrated in Figure 3.7. A threshold is heuristically set based on distance measure metrics for distance classifiers (City-block, Euclidean, Mahalanobis and Chebyshev) of the input vector for all the training vectors based on the computations. The distance between the input vector and the weight vector of the trained images is compared and the weight vectors with the minimum distance measure metrics from the input vector is selected as the closest match (value) indicating a matched image. A distance value from the threshold results into unrecognized face.

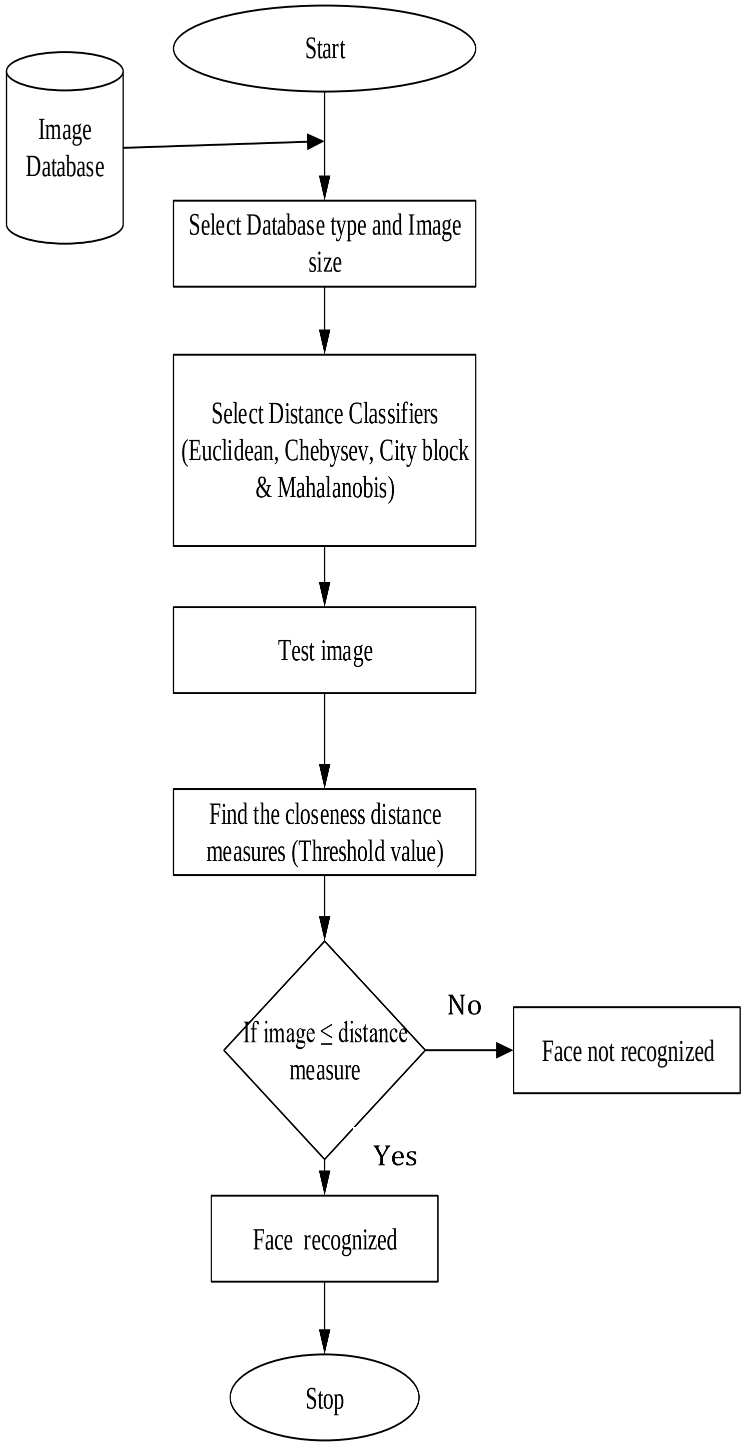


Figure 3.11: Testing Phase for ACOGF System

3.8 Performance Evaluation

Different metrics can be used to conduct efficient rating of performance of biometric recognition solution in which face recognition is an example. To evaluate the performance of face recognition system, the most popular performance measurement standard which has been applied by many researchers is the recognition rate (classification accuracy). Accuracy is a mere proportion of correct guesses, which is one of the commonly used metrics for the evaluation of the performance of a classifier. Few researchers also consider the use of confusion matrix which is referred to as contingency table that allows more detailed analysis of the results of classification than accuracy.

In the confusion matrix, instances of the predicted class are shown on the column while instances of the actual class are represented on the rows. The indices in a confusion matrix are true positive (TP) that is the total number of authorized face images that are correctly recognized by the system, false positive (FP) that is the total number of unauthorized face images wrongly recognized by the system, true negative (TN) that is the total number of authorized images correctly unrecognized by the system and false negative (FN) that is the total number of unauthorized images that are wrongly unrecognized by the system.

These results can be represented in terms of receiver operating characteristics (ROC) curve, which is, plotting correct classification versus false positive. However, the training time, classification time and influence of Image sizes are significant computational cost evaluation metrics. Furthermore, sensitivity, specificity and error rate can be used to evaluate the performance of classifiers if the system is considering multiple classifiers in face recognition system

$$\text{Recognition Accuracy (\%)} = \frac{No\ of\ correctly\ detected\ images}{Total\ number\ of\ validation\ set} \text{ x } 100 \tag{3.1}$$

$$= \frac{TP+TN}{TP+TN+FP+FN} \text{ x } 100$$

$$\text{Sensitivity (\%)} = \frac{No\ of\ true\ positive}{No\ of\ true\ positive + No\ of\ false\ negative} \text{ x } 100 \tag{3.2}$$

$$= \frac{TP}{TP+FN} \text{ x } 100$$

$$\text{Specificity (\%)} = \frac{No\ of\ true\ negative}{No\ of\ true\ negative + No\ of\ false\ positive} \text{ x } 100 \tag{3.3}$$

$$= \frac{TN}{TN+FP} \text{ x} 100$$

$$\text{Error Rate (\%)} = \frac{No\ of\ misclassification}{No\ of\ face\ \ image\ samples\ in\ validation\ set} \text{ x } 100 \tag{3.4}$$

$$= \frac{FP+FN}{TP+TN+FP+FN} \text{ x } 100$$

CHAPTER FOUR

RESULTS AND DISCUSSION

This chapter presents the detail results of optimal Gabor features for facial recognition system. It discusses the result in order to state clearly how the objectives listed for this study and as well as workability of the approach employed to actualize these objectives

4.1 Stepwise Results

The following sections presents the results of each of the phases of image processing defined in the proposed system. It was discovered during experimental results that all of the face images performed well to the preprocessing method used as shown in subsections 4.1.1, 4.1.2 and 4.1.3.

4.1.1 Results of Gray-Scale Conversion

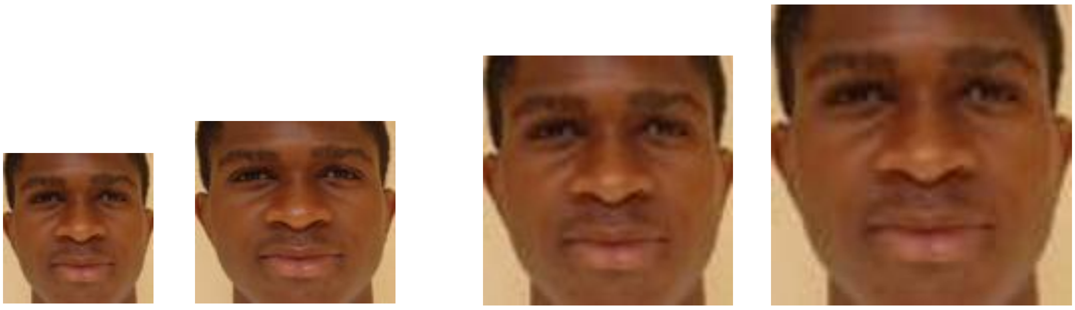
The face images from LAFI database were passed through gray-scale process. The

sample of result is shown in Figure 4.1.

Figure 4.1: Sample of Gray-Scale Images (LAFI Database)

4.1.2 Results of Geometric Normalization

The face images from LAFI and ORL database were normalized geometrically by cropping and resizing. The sample results of the geometric normalization are shown in Figures 4.2 and Figure 4.3 for LAFI and ORL database respectively. These various sizes show varying number of fundamental facial features and the sizes were chosen arbitrarily to test for the effect of the variation in image size on the developed face recognition system. In each category of the pixel resolution, there are total number of 300 images each for LAFI and ORL face image Database. The image size used in this study are 75x75, 100x100, 125x125 and 150x150 for the two database.



75x75 100x100 125x125 150x150

Figure 4.2: Sample of Geometric Normalization (LAFI Database)

75x75 100x100 125x125 150x150

Figure 4.3 Sample of Geometric Normalization (ORL Database)

4.1.3 Results of Illumination Normalization

The face images were loaded from the two databases with loading time as shown in Table 4.1, then later passed for normalization by applying Adaptive Histogram

Equalization (AHE) technique as shown in Figure 4.4, with Table 4.2 showing normalization time. The AHE is used to stretch the contrast on all parts of the face image and adjust the intensity of image pixels. It was observed based on visual quality of the original image from database, original grayscale image and the enhanced image that the face images responded well to the enhancement technique adopted for illumination normalization.

In LAFI Database, the normalization of histogram for grayscale image and that of the enhanced image by AHE is shown in Figure 4.5 and Figure 4.6. From Figure 4.5, it was observed that the histogram (frequency distribution of pixels on the face image) of the original image is not equalized. This implies that the frequencies of occurrence of the pixel values are not evenly distributed. The histogram of the enhanced image as shown in Figure 4.6 is well balanced as the histogram value for the entire bin which represents the frequency of occurrence of the pixels is evenly distributed. The similar process happened in Figure 4.7, Figure 4.8 and Figure 4.9 for ORL Database

Table 4.1: Loading Time of Face Image Datasets

Image Size	Loading Time for Database (secs)	
(pixel)	LAFI(secs)	ORL(secs)
75x75	23.8136	23.8976
100x100	24.8287	24.9223
125x125	24.8345	24.9783
150x150	25.8455	25.9716

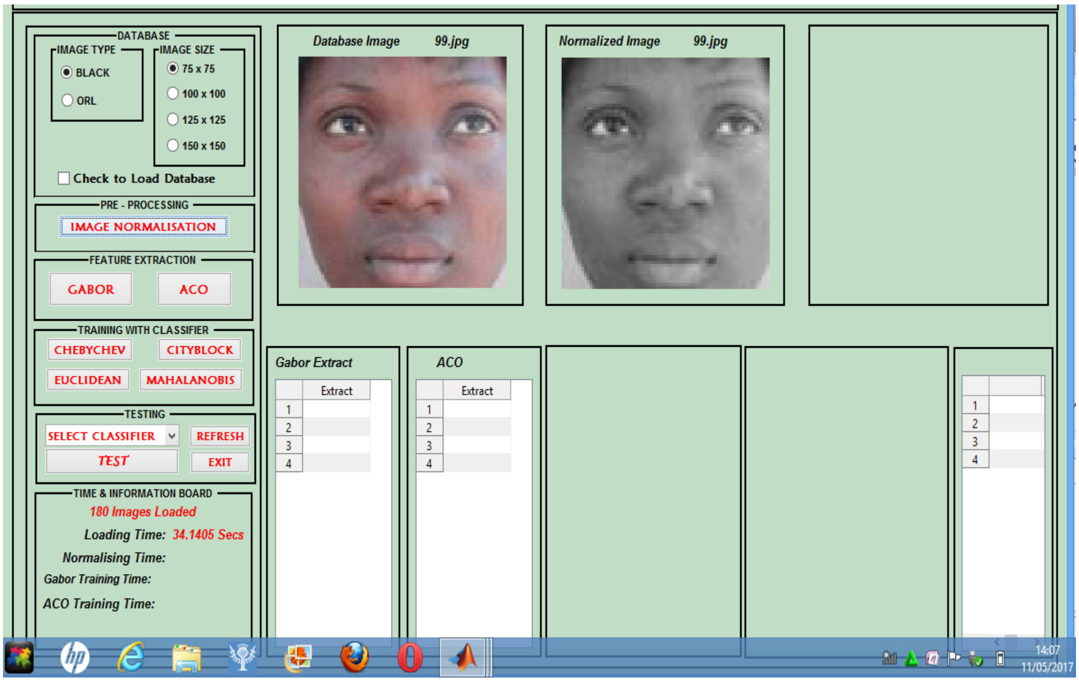


Figure 4.4: GUI Sample of Normalization of Original Faces (LAFI Database)

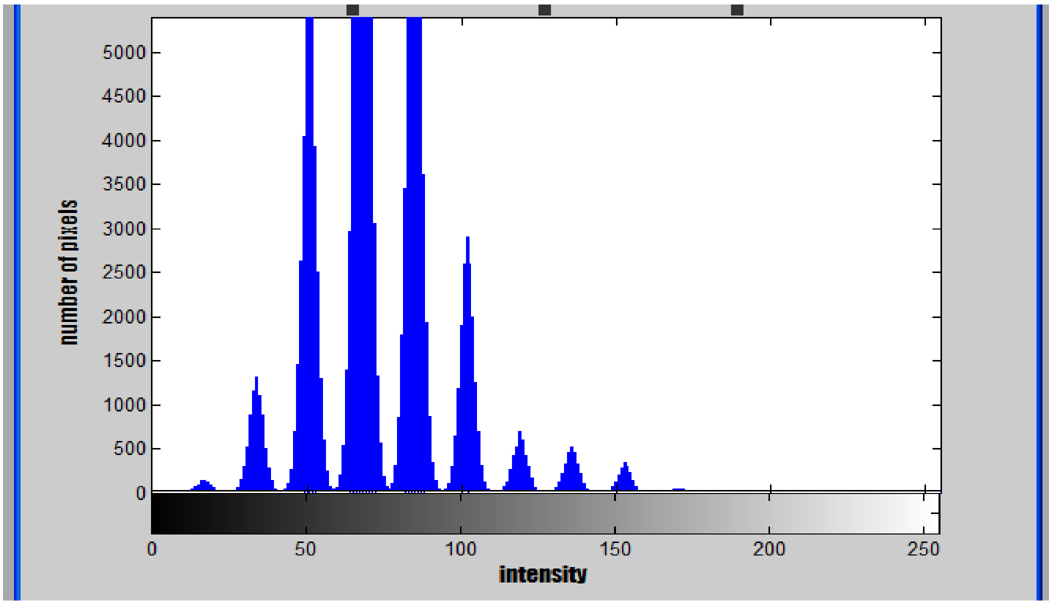


Figure 4.5: Histogram Sample of the Original Grayscale Image (LAFI Database)

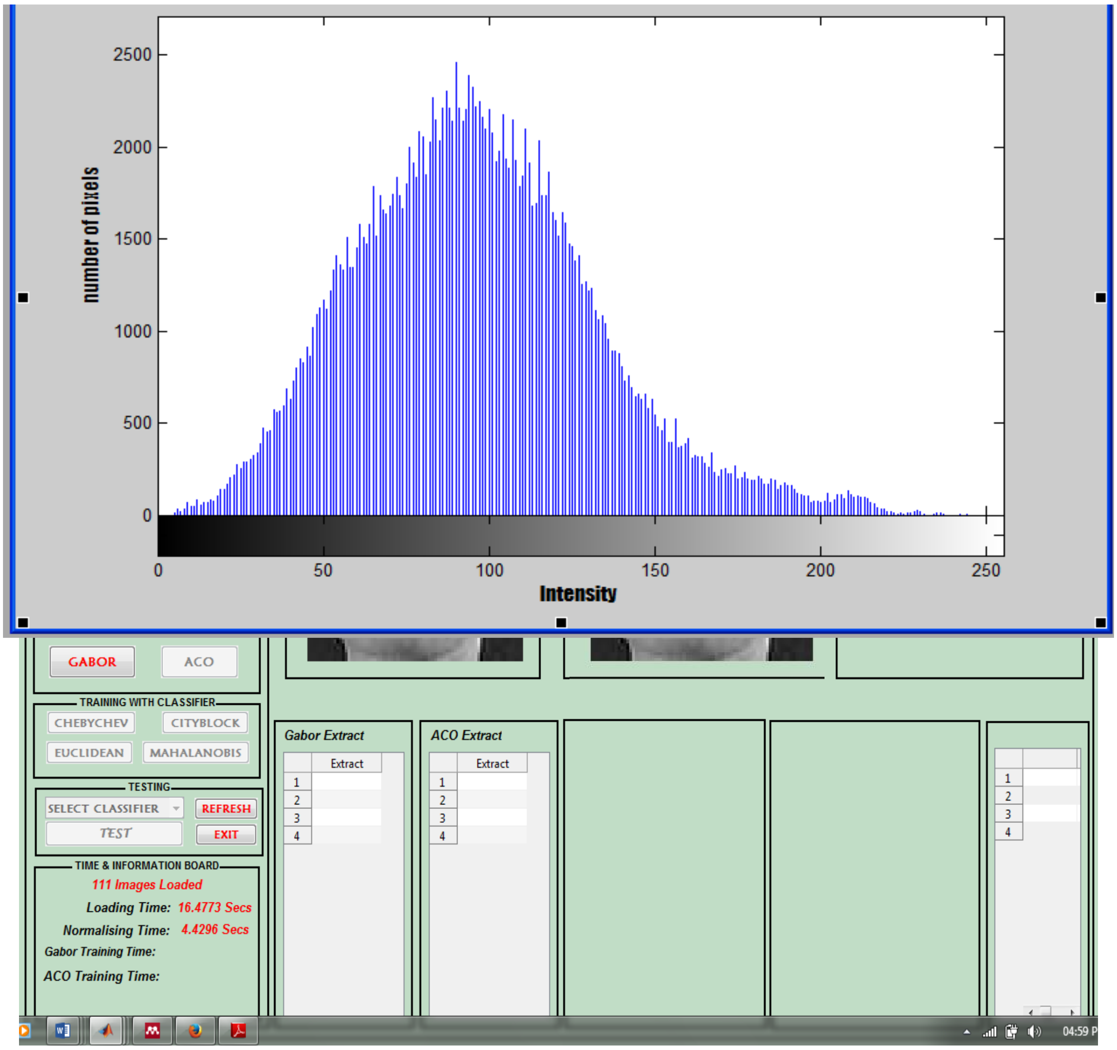


Figure 4.7: GUI Sample of Normalization of Original Faces (ORL Database)

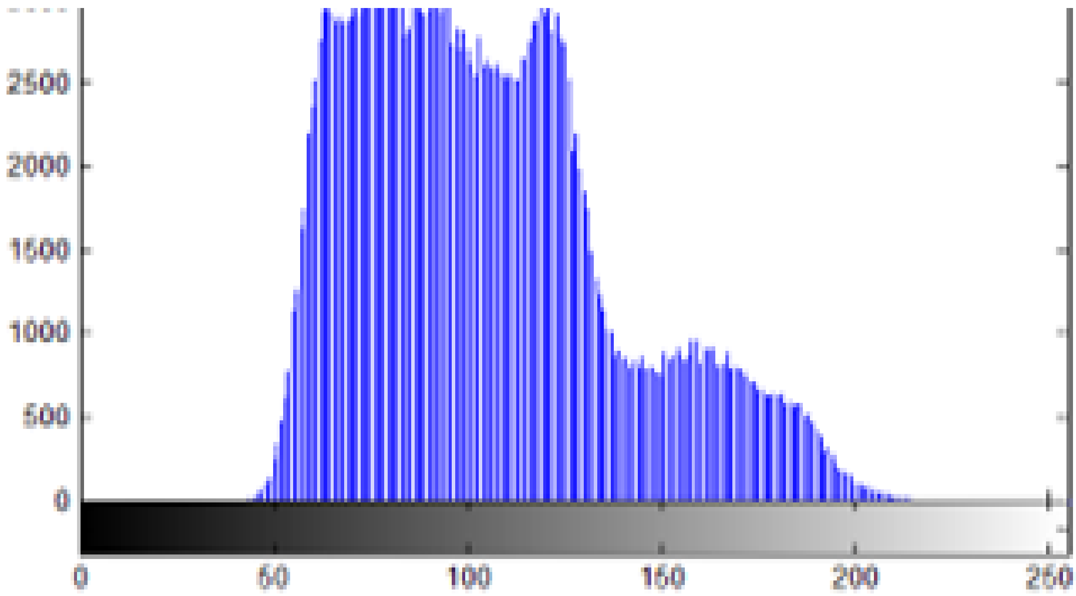


Figure 4.8: Sample of Histogram of the Original Grayscale Image (ORL Database)

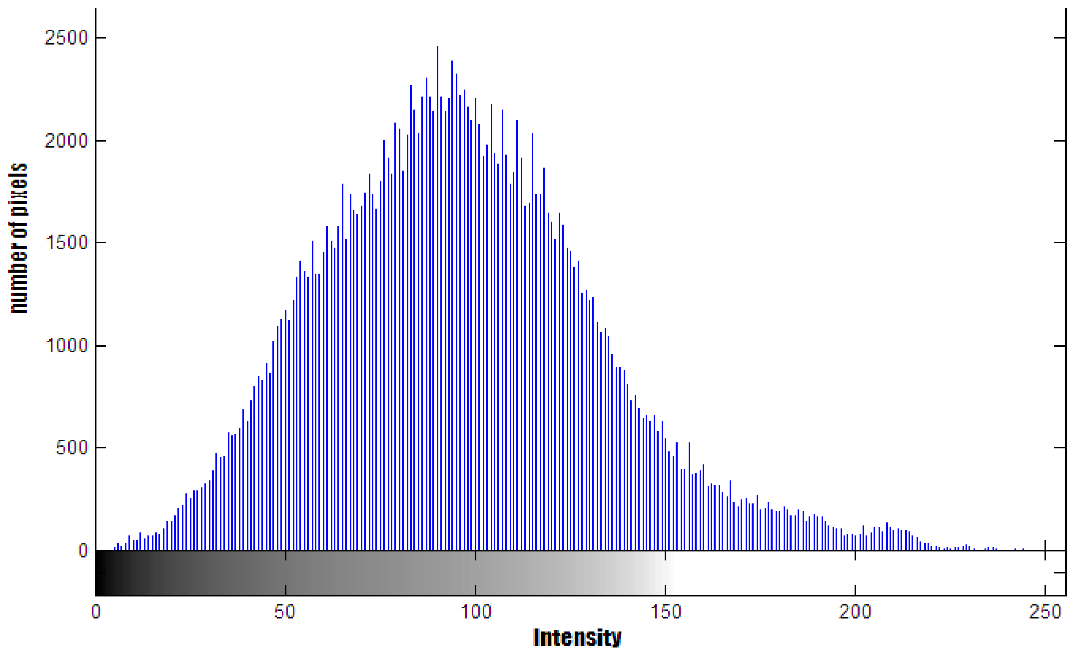


Figure 4.9: Sample of Adaptive Histogram Equalization (ORL Database)

Table 4.2: Normalization Time of Images in the Database

Image Size (pixel)	Normalizing Time for Database (secs)	
	LAFI(secs)	ORL(secs)
75x75	10.8382	10.7632
100x100	11.1853	11.4848
125x125	11.3437	11.5865
150x150	11.4691	11.7616

4.2 Experimental Results of ACOGF System

In this study, several experiments were carried to show the efficiency of the optimal Gabor feature subset selection for facial recognition system. All experiments were conducted on hp ProBook 4330s laptop with the following configuration: 2.40 GHZ CPU, with window 7 operating system, 64-bit Operating System and 4 GB RAM. The face recognition system varied with feature vectors based on the image size 75x75, 100x100, 125x125 and 150x150 pixels of cropped face images were tested. The obtained results of the feature extraction and feature selection process are discussed in subsections 4.2.1 and 4.2.2.

4.2.1 Feature Extraction Results

As shown in Table 4.3, the total time taken to extract Gabor features from each image pixel for LAFI Database are 75x75 pixel size for 118.2510secs, 121.1642secs for 100x100, 156.4176secs for 125x125 and 197.1528secs for 150x150 pixel size. Also Table 4.4 for the ORL, the total time taken to extract Gabor features per image size are, 75x75 image size is 117.1351secs, 100x100 is 120.5418secs, 125x125 is 153.6782secs and 150x150 is 195.3781secs. The extraction of features phase is shown by GUI in Figure 4.10 for LAFI Database. Figure 4.11 represents the GUI of extracted Gabor features for ORL. The samples of magnitude and real part of the extraction for the two datasets is shown in Appendix A. It was observed that the time taken to perform feature extraction for two datasets increase with increase in image pixel size (the higher image pixel size the higher time taken to extract features).

Table 4.3: Features Extraction Time (LAFI)

Image size	Time (secs)
75x75	118.2510
100x100	121.1642
125x125	156.4176
150x150	197.1525

Table 4.4: Features Extraction Time (ORL)

Image size	Time (secs)
75x75	117.1351

100x100	120.5418
125x125	153.6782
150x150	195.3781

Figure 4.10: GUI Sample of Extracted Gabor features (LAFI Database)

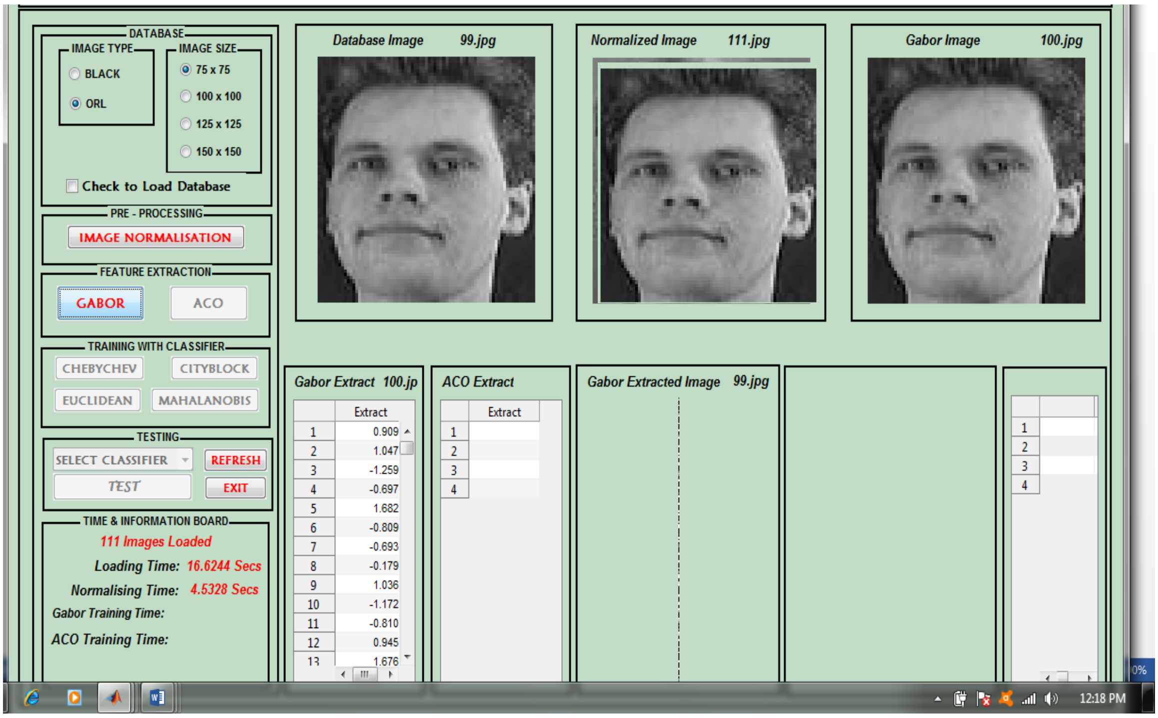


Figure 4.11: GUI Sample of Extracted Gabor Features (ORL Database)

4.2.2 Results of Optimized Gabor Features Using ACO

The Gabor features obtained from the face images in the feature extraction phase serves as a platform which provides ACO a suitable representation for encoding the procedure for optimal feature subsets selection. The input to the ACO feature subset selection process was the product of Gabor feature dimensions obtained from the feature extraction phase. Only subset from feature vectors of Gabor features was retained by the feature selection algorithm as indicated in Figure 4.12 for LAFI Database and Figure 4.13 optimal feature subsets of Gabor features for ORL Database.

The dimension of face image used in this experiment varies; 5625, 10000, 15625 and 22500 pixels for 75x75, 100x100, 125x125 and 150x150 size respectively. The time taken during feature subset selection by ACO for LAFI database are; 63.6105secs for 75x75, 95.5845 for 100x100, 133.1755secs for 125x125 and 144.2482secs for 150x150 as shown in Table 4.5. Also for ORL database; 68.2341secs for 75x75, 101.5673secs for 100x100, 141.9745secs for 125x125 and 153.4327secs for 150x150 as illustrated in Table 4.6. Finally, it was observed that the higher the image size the higher time taken by ACO for feature subset selection of the two datasets.

Figure 4.12: GUI Sample of ACO Feature Selection of Gabor Features (LAFI Database)

Figure 4.13: GUI Sample of ACO Feature Selection of Gabor Features (ORL Database)

Table 4.5: Training Time of Gabor Feature Selection Using ACO (LAFI)

Image size	Time (secs)
75x75	63.6105
100x100	95.5845
125x125	133.1755
150x150	144.2482

Table 4.6: Training Time of Gabor Feature Selection Using ACO (ORL)

Image size	Time (secs)
75x75	68.2341
100x100	101.5673
125x125	141.9745
150x150	153.4323

4.3 Training Time of Distance Classifiers for ACOGF System

The study applied some selected distance measures metrics; Chebysev, Mahalanobis, City block and Euclidean. The training time taken by each distance measure classifier varies as shown in Table 4.7 with Figure 4.14 illustrating the graphical representation. Chebysev distance classifier of image 75x75 recorded a very prolong training time of 13.0515secs compared with other classifiers for LAFI Database. While the City-block of 75x75 gave the best lowest training time of 7.2424secs of all distance classifiers. Also for the ORL Database of images, the training time for each distance classifier is shown in Table 4.8 with Figure 4.15 depicting the graphical representation. It is shown that Chebysev of image size 75x75 recorded the prolong training time of 13.4309secs and best lowest training time of 7.6854 was obtained in City-block classifier of 75x75 image pixel size compared with other classifiers. The performance analysis of the distance classifiers on the two image datasets used in this study indicated that image size has direct influence on Chebyshev prolong training time, the lower the image size the higher training time. While the image size does not really show complete effect on other three classifiers; city-block, euclidean and Mahalanobis.

Table 4.7: Training Time of the Distance Classifiers (LAFI Database)

Image Size (pixel)	Chebysev (secs)	City-Block (secs)	Euclidean (secs)	Mahalanobis (secs)
75x75	13.0515	7.2424	7.6164	7.3195
100x100	12.7830	7.4493	7.7043	7.5502
125x125	12.5741	7.6757	7.7419	7.7744

150x150	12.2892	8.6039	7.9238	7.9567

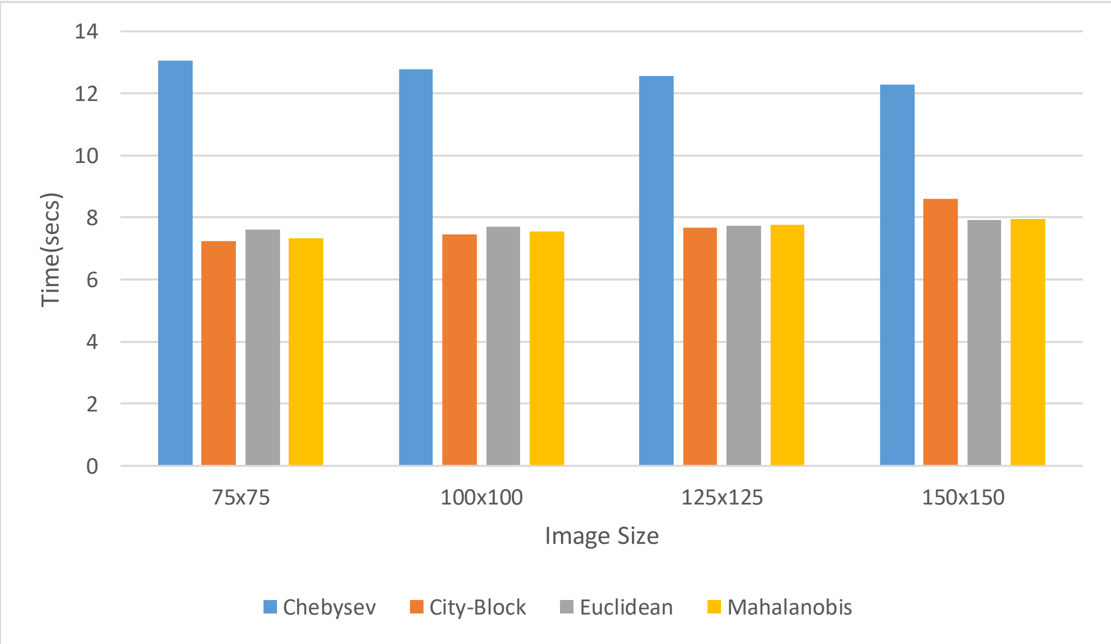
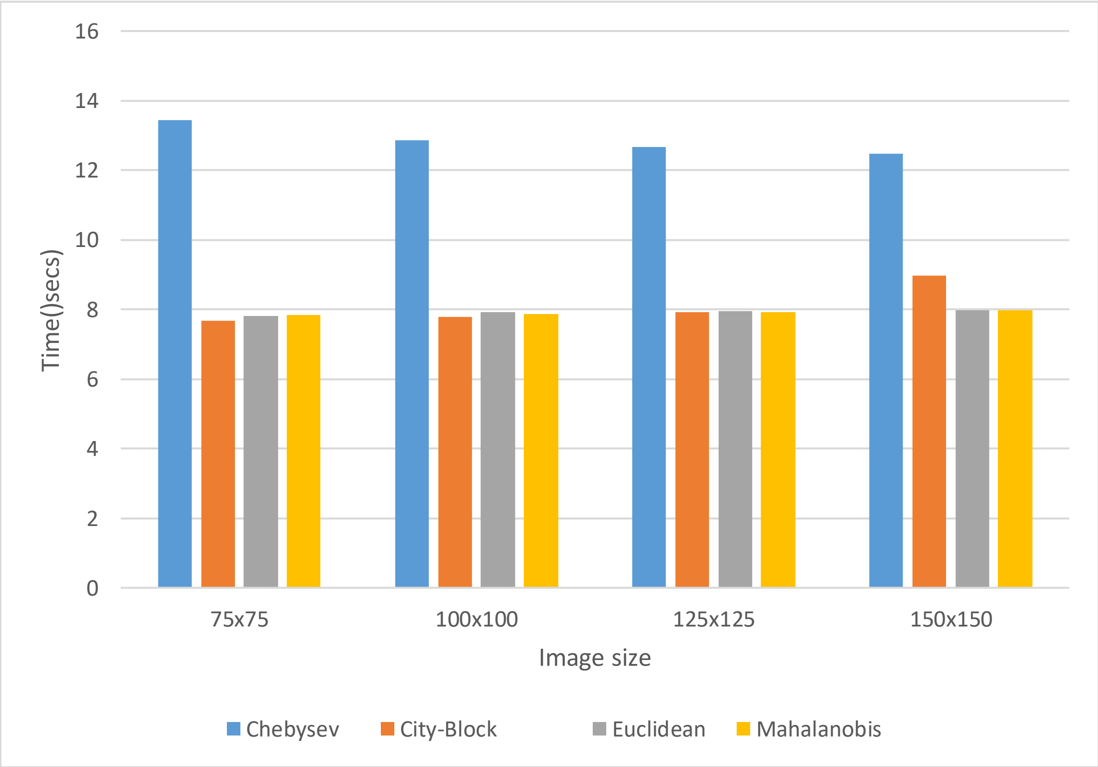


Figure 4.14: Distance Classifiers Training Time (LAFI Database)

Table 4.8: Training Time of Distance Classifiers (ORL Database of images)

Image Size (pixel)	**Chebysev (secs)**	**City-Block (secs)**	**Euclidean (secs)**	**Mahalanobis (secs)**
75x75	13.4309	7.6854	7.8122	7.8342
100x100	12.8622	7.7726	7.9340	7.8534
125x125	12.6743	7.9218	7.9572	7.9123

150x150	12.4863	8.9830	7.9722	7.9689



4.4 Setting of Threshold Values for Distance Classifiers

The recognition of face image was carried out using four different distance measure classifiers; Chebysev, City-block, Euclidean and Mahalanobis. For each distance classifier used, threshold values were set heuristically by considering the generally common values of the calculated of distance classifier during training time. The

Figure 4.15: Distance Classifiers Training Time (ORL Database)

threshold values were set according to the image pixel size and dataset as discussed in Table 4.9 and Table 4.10.

Image Size	Threshold Value			
	Mahalanobis	Euclidean	Chebyshev	City Block
75x75	≤ 24.70	≤ 21.30	≤ 27.70	≤ 23.12
100x100	≤ 27.95	≤ 24.12	≤ 25.45	≤ 29.25
125x125	≤ 38.27	≤ 28.85	≤ 31.56	≤ 36.81
150 x 150	≤ 39.20	≤ 30.61	≤ 32.86	≤ 38.88

Table 4.9: Setting of Threshold Values (LAFI Database)

Table 4.10: Setting of Threshold Values (ORL Database)

Image Size	Threshold Value			
	Mahalanobis	Euclidean	Chebyshev	City Block
75x75	≤ 20.54	≤ 18.29	≤ 15.16	≤ 18.75

100x100	≤ 29.45	≤ 21.70	≤ 18.75	≤ 19.97
125x125	≤ 35.56	≤ 25.05	≤ 29.75	≤ 26.56
150x150	≤ 37.78	≤ 34.37	≤ 33.59	≤ 33.59

4.5 Classification Time of the Distance Classifiers for ACOGF System

The distance classifiers classified face image at different time intervals with respect to image pixel size for the two face image datasets. This is shown in Table 4.11 with graphical representation in Figure 4.16 for LAFI database and Table 4.12 with graphical representation in Figure 4.17 for ORL Database. The total time required by each classifier was drastically reduced, this reflects the ability of ACO to reduce and select relevant features before the classification of face image into either matched or mismatched. Considering the LAFI database, it was observed that Chebyshev classifier of image size 100x100 presented the most prolong classification time of 0.53045secs compared to other classifiers for all image pixel size, the Mahalanobis of image size of 125x125 gave the most reduced classification time of 0.42507secs.

In ORL database, it was observed that Chebysev classifier of image size 150x150 produced the most prolong classification time of 0.56846secs compared with other distance classifiers. The shortest classification time of 0.40422secs was obtained in Mahalanobis classifier for image size 125x125. Finally, it was discovered from the two datasets that chebyshev distance classifier produced the most prolong classification time with image size not really affecting the classification time compared with other classifiers.

Table 4.11: Classification Time of the Distance Classifiers (LAFI Database)

Image Size (pixel)	Chebysev (secs)	City-Block (secs)	Euclidean (secs)	Mahalanobis (secs)
75x75	0.52173	0.45392	0.45587	0.44391
100x100	0.53045	0.43615	0.47108	0.43864
125x125	0.46210	0.44232	0.43426	0.42507
150x150	0.47792	0.42599	0.43206	0.43414

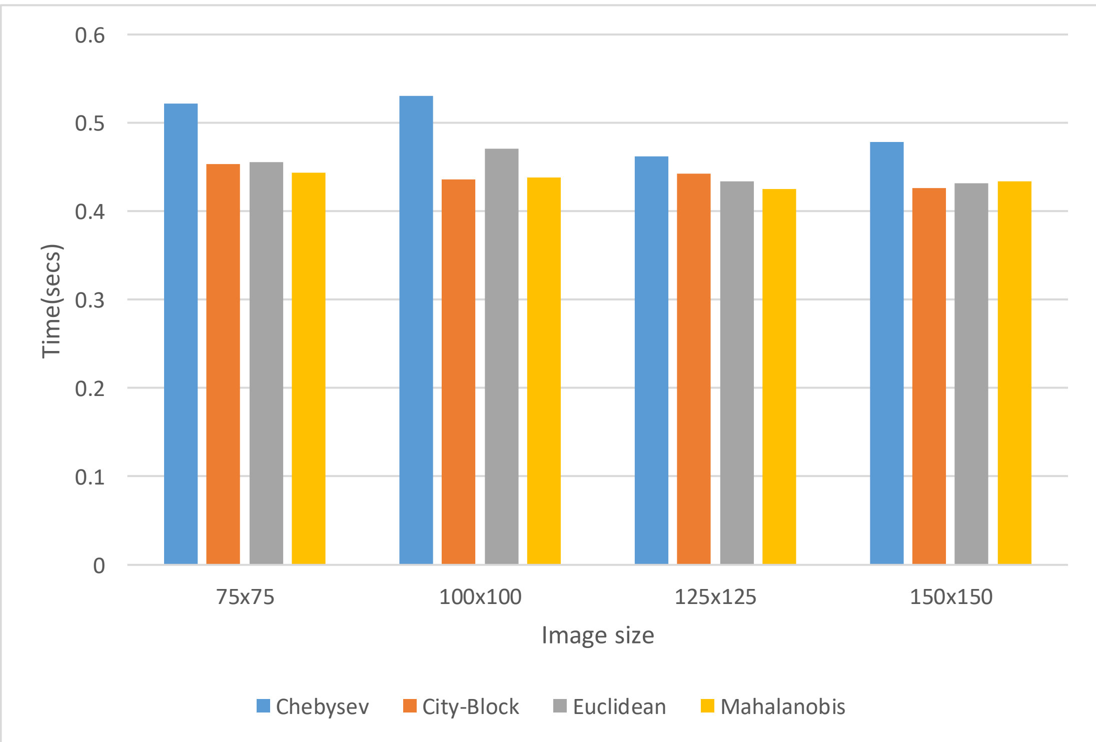


Table 4.12: Classification Time of Distance Classifiers (ORL Database)

Image Size (pixel)	Chebysev (secs)	City-Block (secs)	Euclidean (secs)	Mahalanobis (secs)
75x75	0.53147	0.49849	0.42027	0.49157
100x100	0.49332	0.42079	0.41824	0.42006
125x125	0.53616	0.43647	0.42954	0.40422
150x150	0.56846	0.53880	0.42266	0.43292

Figure 4.16: Classification Time of Distance Classifers (LAFI Database)

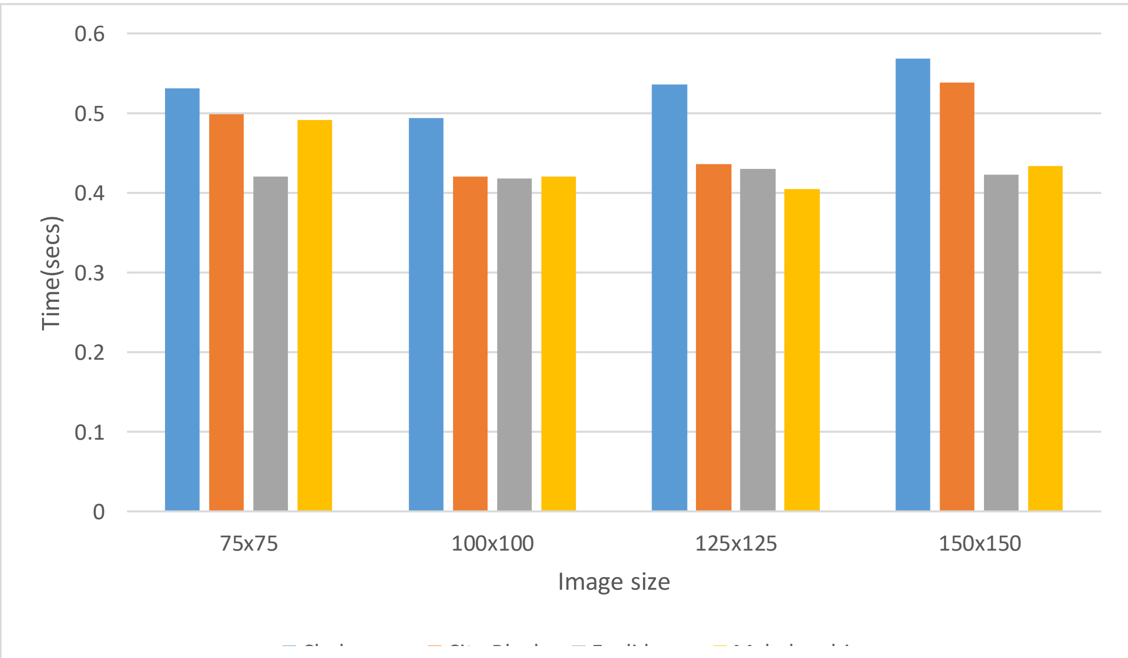


Figure 4.17: Classification Time of Distance Classifers (ORL Database)

4.6 Performance Evaluation of Distance Classifiers for ACOGF System

The experimental results for classification performance measurement of ACOGF for each distance classifier used (Malahanobis Chebysev, Euclidean and City-block) for the two datasets were revolved around using the number of matched and mismatched face images in term of True Negative, True Positive, False Negative and False Positive as illustrated in Subsection 4.6.1 and 4.6.2.

4.6.1 Performance Evaluation of ACOGF System (LAFI Database)

In LAFI Database, 10 images were used as impostors (images not trained and also not contained in the database) to test the ACOGF system for true negative (images that were not seen in the database), the highest number of true negative of 9 images was obtained in Mahalanobis of image size 150x150, Euclidean of image size 75 x 75, 100x100, 125x125, Chebyshev image size of 75x75, 125x125, City-block of image size of 75x75. Also the most reduced number of true negative of 7 images obtained Mahalanobis of image size 125x125, City-block of image size 100x100, 150x150 as shown Table 4.13. For true positive (correctly matched images), 60 images were used to test the ACOGF system, the highest number of 59 images was obtained in Mahalanobis of image pixel of 150x150 and City-block of image size 125x125 as shown in Table 4.14.

For false positive (mismatched images), 10 images were considered as impostors to test the ACOFG system, where the number of mismatched images with distance classifiers as follows: the highest number of 3 images was obtained in Mahalanobis of image size 125x125, City-block of image size 100x100 and 150x150. While lowest number of 1 image was recorded in Mahalanobis image 150x150, Euclidean of image size 75x75, 100x100, 125x125, Chebyshev of image 75x75, 125x125, City-block of image size 75x75 as shown in Table 4.15. For false negative (mismatched), 60 images were used to test the system, where the following mismatched images were achieved; the highest number of 5 images obtained in City-bock of image size of 75x75, while the lowest number of 1 person was obtained in Mahalanobis of image size 150x150 and City-block of image size 125x125 as shown in Table 4.16 for LAFI database

Table 4.13: True Negative (LAFI Database)

Image size	**Distance Classifier**			
	Mahanalobis	**Euclidean**	**Chebyshev**	**City-Block**
75 x 75	8	9	9	9
100 x 100	8	9	8	7
125 x125	7	9	9	8
150 x150	9	8	8	7

Total number of tested images (Impostors) = 10

Image size	Distance Classifier			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City-Block**
75 x75	57	56	56	55
100x100	58	57	56	56
125x125	56	58	57	59
150x150	59	58	58	58

Table 4.14: True Positive (LAFI Database)

Total number of tested images = 60

Image size	Distance Classifier			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City-Block**
75 x75	2	1	1	1
100x100	2	1	2	3
125x125	3	1	1	2
150x150	1	2	2	3

Table 4.15: False Positive (LAFI Database)

Total number of tested images (impostors) = 10

Image size	Distance Classifier			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City-Block**
75x75	3	4	4	5
100x100	2	3	4	4
125x125	4	2	3	1
150x150	1	2	2	2

Table 4.16: False Negative (LAFI Database)

Total number of tested images = 60

4.6.2 Performance Evaluation of ACOGF System (ORL Database)

In ORL Database, for true negative, total number of 10 images were used as impostors to test the system, the highest number of true negative of 9 images was obtained in Mahalanobis of image size 150x150, Euclidean of image size 100x100, City-block of image size 100x100, 150x150, while the lowest true negative of 7 images achieved in Mahalanobis of image size 75x75, Chebyshev of image size 125x125 as shown in table 4.17. For the true positive (correctly matched images), 60 persons were used to test the system, the highest number of 59 persons was obtained for Euclidean of image size 150x150 and City-block of image size 100x100, while lowest number of 56 images obtained in Mahalanobis of image size 125x125, Chebyshev of image 75x75 and City-block of image size 125x125, as shown in Table 4.18.

For the false positive (images wrongly matched with the database), 10 images were used as impostors to test the system, with highest number of 3 images were obtained in Mahalanobis of image size 75x75 and Chebysev of image size 125x125, while the lowest number of 1 person was obtained Mahalanobis of image size 150x 150, Euclidean of image size 100x100 and City-block of image size 75x75, 125x125 as shown in Table 4.19. For false negative (mismatched images), 60 images were used to test the system, highest number of 5 persons were detected for Mahalanobis of image size 75x75, Euclidean of image size 100x100, while the lowest number of 1 person was detected in City-block of image size 100x100 and Euclidean of image size 150x150 as shown in Table 4.20.

Image size	**Distance Classifier**			
	Mahanolobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	7	8	8	9
100x100	8	9	8	8
125x125	8	8	7	8
150x150	9	8	8	9

Table 4.17: True Negative (ORL Database)

Total number of tested images (impostors) = 10

Table 4.18: True Positive (ORL Database)

Image size	Distance Classifier			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City Block**
75 x75	55	58	56	58
100x100	57	55	57	59
125x125	56	57	58	56
150x150	58	59	58	57

Total number of tested images = 60

Image	Distance Classifier

Size	Mahalanobis	Euclidean	Chebyshev	City Block
75x75	3	2	2	1
100x100	2	1	2	2
125x125	2	2	3	2
150x150	1	2	2	1

Table 4.19: False Positive (ORL Database)

Total number of tested images (impostors) = 10

Image size	Distance Classifier			
	Mahalanobis	Euclidean	Chebyshev	City Block
75x75	5	2	4	2
100x100	3	5	3	1
125x125	4	3	2	4
150x150	2	1	2	3

Table 4.20: False Negative (ORL Database)

Total number of tested images = 60

4.7 Classification Accuracy of ACOGF System with Distance Classifiers

The best classification accuracy percentage of 97.14% was obtained in Mahalanobis classifier of image size 150x150, while the lowest classification accuracy percentage of 91.43% obtained in Chebyshev classifier of image size 100x100 and City-block of image size 75x75 for LAFI Database as shown in Table 4.21. Also for ORL Database, the best classification accuracy of 95.71% was achieved in Mahalanobis of image size 150x150, Euclidean of image size 150x150, City-block of image size 75x75, 100x100. While the lowest classification accuracy of 88.57% was achieved in Mahalanobis of image size 75x75 as shown in Table 4.22. It was observed from the classification phase that ACOGF gave outstanding results in classification accuracy with the image size showing complete effect on classification accuracy.

Table 4.21: Classification Accuracy of Distance Classifiers (LAFI Database)

Image size	Distance Classifier (%)			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	92.86	92.86	92.86	91.43
100x100	94.29	94.29	91.43	90.00
125x125	90.00	95.71	94.29	95.71
150x150	97.14	94.29	94.29	92.86

Table 4.22: Classification Accuracy of Distance Classifiers (ORL Database)

Image size	Distance Classifier (%)			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City Block**
75 x75	88.57	94.29	91.43	95.71
100x100	92.86	91.86	92.86	95.71
125 x125	91.43	92.86	92.86	91.43
150x150	95.71	95.71	94.29	95.71

4.8 Sensitivity of Distance Classifiers for ACOGF System

The sensitivity is the true positive rate (TPR), this is an applied mathematical measure that shows how properly each distance classifier positively recognized images. The best percentage of sensitivity of 98.33% was achieved in Mahalanobis of image size 150x150, City-block of image size 125x125, while the lowest sensitivity percentage of 91.67% was obtained in City-block of image size 75x75 as shown in Table 4.23 for LAFI Database. Also for ORL Database, the best sensitivity percentage of 98.33% was achieved in Euclidean of image size 150x150, City-block of image size 100x100 as shown in Table 4.24. The results of sensitivity in percentage showed that face image size has effect on the sensitivity of the classifiers.

Table 4.23: Sensitivity of Distance Classifiers (LAFI Database)

Image size	Distance Classifier (%)			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City**

				Block
75 x75	95.00	93.33	93.33	91.67
100x100	96.67	95.00	93.33	93.33
125x125	93.33	96.67	95.00	98.33
150x150	98.33	96.67	96.67	96.67

Table 4.24: Sensitivity of Distance Classifiers (ORL Database)

Image Size	**Distance Classifier (%)**			
	Mahalanobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	91.67	96.67	93.33	96.67
100x100	95.00	91.67	95.00	98.33
125x125	93.33	95.00	96.67	93.33
150x150	96.67	98.33	96.67	95.00

4.9 Specificity of Distance Classifiers for ACOGF System

This is true negative rate of face images, it is an applied mathematical measure of how well a distance classifier in the proposed system properly identifies the negative cases (mismatched face). The best specificity percentage of 90% was obtained in Mahalanobis of image size 150x150, Euclidean of image size 75x75, 100x100, 125x125, Chebyshev of image size 75x75, 125x125, City-block of image size 90x90. While the lowest specificity percentage of 70% was achieved in Mahalanobis of image size 125x125, City-block of image size 100x100, 150x150 for LAFI Database as shown in Table 4.25.

Also for ORL Database, the best percentage specificity of 90% was achieved in Mahalanobis of image size 150x150, Euclidean of image size 100x100, City-block of image size 75x75, 150x150, while worst percentage of 70% was obtained in Mahalanobis of image size 75x75, Chebyshev of image size 125x125 for ORL Database as shown Table 4.26. The results from the two datasets, shows that the best specificity percentage of selected distance classifiers deduced that ACOGF properly classified with the image size some effects on the specificity percentage.

Table 4.25: Specificity of Distance Classifiers (LAFI Database)

Image Size	Distance Classifier (%)			
	Mahanolobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	80.00	90.00	90.00	90.00
100x100	80.00	90.00	80.00	70.00
125x125	70.00	90.00	90.00	80.00
150x150	90.00	80.00	80.00	70.00

Image size	Distance Classifier

	Mahanolobis	Euclidean	Chebyshev	City Block
75x 75	70.00	80.00	80.00	90.00
100x100	80.00	90.00	80.00	80.00
125x125	80.00	80.00	70.00	80.00
150x150	90.00	80.00	80.00	90.00

Table 4.26: Specificity of Distance Classifiers (ORL Database)

4.10 Error Rate of Distance Classifiers for ACOGF System

The error rate is an acceptable performance measure for the comparison of different classifiers used in the ACOGF system given balanced datasets. The best percentage of error rate of 2.86% was obtained in Mahanolobis of image size 150x150, while the worst error rate of 10% was achieved in Mahanolobis of image size 125x125, city-block of image size 100x100 for LASFI Database as shown in Table 4.27

For ORL Database, the best error rate percentage of 4.29%% was obtained in Mahanolobis of image size 75x75, while the best error rate percentage of 4.29% was obtained in Mahanolobis of image size 150x150, Euclidean of image size 150x150 and City-block of image size 75x75, 100x100 and the worst error rate of 11.43% in Mahanolobis of image size 75x75 as shown in Table 4.28. It was observed that the best error rate obtained in LAFI database compared to ORL database. Experimental results also revealed that image size has some effects of the error rate recorded.

Table 4.27: Error Rate of Distance Classifier (LAFI Database)

Image size	**Distance Classifier (%)**			
	Mahanolobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	7.69	7.69	7.69	8.57
100x100	5.71	5.71	8.57	10.00
125x125	10.00	4.29	5.71	4.29
150x150	2.86	5.71	5.71	7.14

Table 4.28: Error Rate of Distance Classifiers (ORL Database)

Image Size	**Distance Classifier (%)**			
	Mahanolobis	**Euclidean**	**Chebyshev**	**City Block**
75x75	11.43	5.71	8.57	4.29
100x100	7.14	8.57	7.14	4.29
125x125	8.57	7.14	7.14	8.57

150x150	4.29	4.29	5.71	5.71

4.11 Performance Analysis of Existing Systems with ACOGF System

The performance analysis of the existing systems and the ACOGF system is briefly summarized in Table 4.29.

References	Method	Classification Accuracy (%)	Classification Time (secs)	Sensitivity (%)	Specificity %	Error Rate (%)
Abhishree, T. M., Latha, K., Manikantan and Ramachandran, (2015)	Gabor-filters + Binary Particle Swarm Optimization	94.41	0.56345	91.76	89.15	4.12
Dora et al (2017)	Gabor-filters + Hybridize PSO & Gravitational Search Algorithm	96.00	0.51234	94.58	85.89	3.59

Developed ACOGF System (2018)	Gabor-filters + Ant Colony Optimization	97.14	0.42507	98.33	90.00	2.86

Table 4.29: Performance Analysis of ACOGF System with Existing Systems

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

The chapter discusses the conclusion made from the study carried out, contribution to knowledge and recommendations for future studies.

5.1 Summary

The recognition of face image applying Gabor-filters has been identified by so many researchers in image processing, computer vision and pattern recognition system to be one of the robust algorithms for feature extraction. Despite the promising performance of Gabor-filters, they are computationally expensive for both processing time and memory usage due to very high dimensional space. Several dimensionality reduction techniques such as down-sampling, feature selection through subspace approaches such as Principal Component Analysis, Linear Discriminant Analysis and reduction of filter number have been applied to reduce Gabor features. But all these techniques have not really given optimal features.

To obtain features from Gabor features require a proper feature selection approach to be employed. The selection of optimal features has been considered to be a non-deterministic polynomial hard problem. A problem like this requires an efficient optimization algorithm. This study addressed the challenge by introducing a nature inspired optimization approach known as meta-heuristic Ant Colony Optimization

Algorithm. ACO meta-heuristics formulates its optimization problem by means of a graph in which nodes indicate features and edges indicate path connecting features.

The Gabor features obtained from the feature extraction stage gave ACO a suitable representation for selecting optimal features. The features obtained from ACO algorithm provided discriminating features that carried the most relevant information that was passed to some selected distance classifiers; Mahanolobis, Chebysev, Euclidean and City-block for classification of face image. The performance of the system was evaluated using the two datasets; real-time image database (LAFI Database) and Olivetti Research Laboratory (ORL) publicly available database. The best classification accuracy of 97.14% was obtained in Mahanolobis of image size 150x150 for LAFI Database, while classification accuracy of 95.71% was achieved in Mahanolobis of image size 150x150, Euclidean of image size 150x150, City-block of image size of 75x75, 100x100 and 150x150 for ORL database.

The lowest classification time of 0.42507secs was obtained in Mahanolobis of image size 125x125 for LAFI Database and 0.40422secs was obtained in Mahalanobis of image size 125x125 for ORL Database. The sensitivity of 98.33% was obtained in Mahanolobis of image size 150x150, City-block of image size 125x125 for LAFI, while the same percentage of 98.33% in Euclidean of image size 150x150 for ORL. The best specificity of 70% was achieved in Mahanolobis of image size 125x125, City-block of image (100x100, 150x150). The best error rate of 2.86% was achieved in Mahanolobis of image size 150x150 for LAFI Database and 4.29% was obtained in Mahanolobis of image 150x150, Euclidean of image size 150x150 and City-block of image size of 75x75, 100x100 for ORL database.

5.2 Conclusion

The feature extraction remains a significant process in facial recognition algorithm. Among several feature extraction techniques such as Fisher Linear Discriminant Analysis (FLDA), Principal Component Analysis (PCA), Elastic Bunch Graph Matching (EBGM) and Local Binary Pattern (LBP), Gabor-filters possess the ability of obtaining multi-orientation features from a facial image at several scales with the derived information being of local nature. Its optimal functionality in facial recognition is linked to its biological importance (similarity to the receptive fields of simple cells in primary visual cortex) and computational properties (optimal for calculating local spatial frequencies). Despite all the outstanding properties of Gabor-filters, this technique suffers high feature dimensionality. This study addressed the problem of high feature dimensionality by application of Ant Colony Optimization meta-heuristic algorithm for feature selection of relevant and optimal features. Some selected distance measure classifiers are employed to classify face image. Two face image databases; Olivetti Research Laboratory (ORL) Database and Locally Acquired Face Image Database (LAFI) were used to evaluate the performance of the proposed facial recognition model. The final experimental results showed better performance in term of classification time, accuracy, sensitivity specificity and error rate.

5.3 Contribution to Knowledge

The major contributions of this study to body of knowledge are as follows:

(i) This study has demonstrated that the population-based meta-heuristic ACO optimization algorithm has effectively aided the reduction of curse of dimensionality which has been identified as a predominant challenge to Gabor features-based approach for facial recognition. The reduction effect will encourage researchers in image processing to come up with a real-time facial biometric security systems.

(ii) The level of classification accuracy was increased by the introduction of ACO meta-heuristic algorithm as a feature selection technique for obtaining the most relevant, discriminant features and also reducing redundant features which contribute no further useful information during classification stage.

(iii) The investigation of the effect of image size on optimized Gabor features using some selected distance measure classifiers help to know the influence of variation in image in classification phase.

5.3 Recommendations

The following recommendations are listed in this section to draw the attention of future studies in facial recognition system using Gabor-based approach:

(i) Future studies could incorporate other classifiers such as Neural Network and Support Vector Machine a part from distance classifiers to investigate the effect of optimized Gabor features.

(ii) Also datasets can be increased from two to more datasets in the future researches in order to further validate experimental results.

(iii) Another aspect to consider is the combination of distance classifiers into a single classifier just to further improve classification accuracy and also reduce the level of error rate.

REFERENCES

Aarthy, M. & Sumathy, P. (2014). Comparison of Histogram Equalization Method and Histogram Expansion. *International Journal of Computer Science and Mobile Application*, *2*(3), 25–34.

Abhishree, T. M., Latha, K., Manikantan, & Ramachandran, S. (2015). Face Recognition Using Gabor Filter Based Feature Extraction with Anisotropic Diffusion as a Pre-processing Technique ☆. *International Conference on Advanced Computing Technologies and Applications*, *45*, 312–321.

Abikoye, O., Sadiku, J. S., Adewole, K., & Jimoh, R. (2014). Iris Feature Extraction for Personal Identification using Fast Wavelet Transform (FWT). *International Journal of Applied Information Systems*, *6*(9), 1–6.

Afolabi, A. O., & Akintaro, O. M. (2017). Design of DNA Based Biometric Security System for Examination Conduct. *Journal of Adavances in Mathematics and Computer Science*, *23*(6), 1–7. http://doi.org/10.9734/JAMCS/2017/27251

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications*, *36*, 6843–6853. http://doi.org/10.1016/j.eswa.2008.08.022

Al-ani, A. (2007). Ant Colony Optimization for Feature Subset Selection. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *1*(4), 999–1002.

Ali Jan Ghasab, M., Khamis, S., Mohammad, F., & Jahani Fariman, H. (2015). Feature decision-making ant colony optimization system for an automated recognition of plant species. *Expert Systems with Applications*, *42*(5), 2361–2370. http://doi.org/10.1016/j.eswa.2014.11.011

Alkasassbeh, M., Altarawneh, G. A., & Hassanat, A. B. (2015). On Enhancing the Performance of Nearest Neighbour. *Canadian Journal of Pure and Applied Sciences*, *9*(1), 1–6.

Alyasseri, Z. A. A. (2015). Face Recognition using Independent Component Analysis Algorithm. *International Journal of Computer Applications*, *126*(3), 975–8887.

Anila, S., & Devarajan, N. (2012). Preprocessing Technique for Face Recognition. *Global Journal of Computer Science and Technology Graphics & Vision*, *12*(11), 13–18.

Babatunde, R. S., Olabiyisi, S. O., Omidiora, E. O., & Ganiyu, R. A. (2015). Local Binary Pattern and Ant Colony Optimization Based Feature Dimensionality Reduction Technique for Face Recognition Systems. *British Journal of Mathematics & ComputerScience*, *11*(5), 1–11. http://doi.org/10.9734/BJMCS/2015/19490

Bahurupi, S. P., & Chaudhari, D. S. (2012). Principal Component Analysis for Face Recognition. *International Journal of Engineering and Adavnced Technology*, *1*(5), 91–94.

Bakshe, R. C., & Patil, A. M. (2015). Hand Geometry as a Biometric for Human Identification. *International Journal of Science and Research*, *4*(1), 2744–2748.

Bakshi, U., & Singhal, R. (2014). A survey on face detection methods and feature extraction techniques of face recognition. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *3*(3), 233–237.

Bala, M., Singh, P., & Meena, M. S. (2016). Face Recognition Using Linear Discriminant Analysis. *International Journal of Electrical and Electronics Research*, *4*(2), 96–103.

Barbu, T. (2010). Gabor filter-based face recognition technique. *Proceedings of the Romanian Academy Series A - Mathematics Physics Technical Sciences Information Science*, *11*(3), 277–283.

Beham, M. P., & Roomi, S. M. M. (2012). Face Recognition Using Appearance Based Approach : A Literature Survey. In *International Journal of Computer Applications® (IJCA)* (pp. 16–21).

Bellakhdhar, F., Loukil, K., & Abid, M. (2013). Face Recognition Approach sing Gabor Wavelets, PCA and SVM. *International Journal of Computer Science Issues*, *10*(2), 201–207.

Bermingham, M. L., Pong-Wong, R., Spiliopoulou, a, Hayward, C., Rudan, I., Campbell, H., … Haley, C. S. (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific Reports*, *5*, 10312. http://doi.org/10.1038/srep10312

Bhable, S. G., Kayte, S., Maher, R., Kayte, J., & Kayte, C. (2015). DNA Biometric. *IOSR Journal of VLSI and Signal Processing*, *5*(5), 82–84. http://doi.org/10.9790/4200-05518284

Bhattacharyya, D., Ranjan, R., A, F. A., & Choi, M. (2009). Biometric Authentication : A Review. *International Journal of U- and E- Source and Technology*, *2*(3), 13–28.

Bhele, S. G. & Mankar, V. H. (2012). A Review Paper on Face Recognition Techniques. *International Journal of Advanced Research in Computer Engineering & Technology*, *1*(8), 2278–1323. http://doi.org/10.1117/12.2012238

Bhuiyan, A., & Liu, C. H. (2007). On Face Recognition using Gabor Filters. *Engineering and Technology*, *2*(1), 51–56. http://doi.org/10.1109/ICCITECHN.2008.4803030

Bonabeau, E. Dorigo, M., & Theraulaz, G. (2000). Ant Algorithms and Stimergy. *Future Gener, Computer System*, *16*(8), 851–871.

Bouzalmat, A., Belghini, N., Zarghili, A., Kharroubi, J., & Majda, A. (2011). Face Recognition Using Neural Network Based Fourier Gabor Filters & Random Projection. *International Journal of Computer Science and Security (IJCSS)*, *5*(3), 376.

Bratton, D., & Kennedy, J. (2007). Defining a Standard for Particle Swarm Optimization. In *Proceedings of the 2007 IEEE Swarm Intelligencem symposium* (pp. 120–127).

Cappelli, R., Maio, D., Maltoni, D., Wayman, J. L., & Jain, A. K. (2006). Performance evaluation of fingerprint verification systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 3–17. http://doi.org/10.1109/TPAMI.2006.20

Chitra, A & Balakrishnan, G. (2012). A Survey of Face Recognition on Feature Extraction Process of Dimensionality Reduction Techniques. *Journal of*

Theoretical and Applied Information Technology, *36*(1), 92–100.

Chunjun, B., Junxing, Z., Changhong, L. & Yanhui, S. (2013). Gender Classification Based on Relaxed Pixel -Pattern-Based Texture Feature. *Information Technology Journal*, *12*(7), 1387–1393.

Dai, G., & Qian, Y. (2004). A Gabor Direct Fractional-Step LDA Algorithm for Face Recognition. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)* (Vol. 1, pp. 61–64). http://doi.org/10.1109/ICME.2004.1394125

Dangi, P., & Ahmad, S. (2009). Survey of face recognition. *International Journal on Recent and Innovation Trends in Computing and Communication*, *3*(10), 6021–6023.

Daoud, E. Al. (2009). Enhancement of the face recognition using a modified Fourier-Gabor filter. *International Journal of Advances in Soft Computing and Its Applications*, *1*(2), 119–131.

Das, P., & Meshram, S. (2013). An Efficient Hand-Geometry System for Biometric Identifications. *Journal of Electronic and Communication Engineering*, *4*(4), 17–19.

Davidovic, T., Tedorovic, D., & Selmic, M. (2015). Bee Colony Optimization Part I : The Algorithm Overview. *Yugoslav Journal of Operational Research*, *25*(1), 33–56.

Deng, H., Jin, L., Zhen, L., & Huang, J. (2005). Local Gabor Filter Bank and PCA plus LDA. *Journal of Information Technology*, *11*(11), 86–96.

Deniz, O., Castrillon, M., & Hernandez, M. (2003). Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, *24*(13), 2153–2157. http://doi.org/10.1016/S0167-8655(03)00081-3

Déniz, O., Castrillón, M., & Hernández, M. (2003). Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, *24*(13), 2153–2157. http://doi.org/10.1016/S0167-8655(03)00081-3

Deotale, N., Vaikole, S. L., & Sawarkar, S. D. (2010). Face recognition using artificial neural networks. *2nd Int Conf on Computer and Automation Engineering 2010 (ICCAE)*, *7*(2), 446–450. http://doi.org/10.1109/ICCAE.2010.5451595

Derakhshii, M. R. & Ghaemi, M. (2014). Classifying Different Feature Selection Algorithms Based on the Search Strategies. *International Conference on Machine Learning, Electrical and Mechanical Engineering*, 17–21.

Dora, L., Agrawal, S., Panda, R., & Abraham, A. (2017). An Evolutionary Single Gabor kernel-based filter Approach to Face Recognition. *Engineering Applications of Artificial Intelligence*, *62*, 286–301. http://doi.org/10.1016/j.engappai.2017.04.011

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence*, *1*(4), 28–39. http://doi.org/10.1109/MCI.2006.329691

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, *344*(2–3), 243–278. http://doi.org/10.1016/j.tcs.2005.05.020

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge.

Elgarrai, Z., Meslouhi, O. E. L., & Allali, H. (2014). Face Recognition System using Gabor Features and HTK Toolkit. In *Tenth International Conference on Signal - Image Technology & Internet-Based Systems* (pp. 2–6). http://doi.org/10.1109/SITIS.2014.71

Flynn, P. (2006). A survey of approaches and challenges in 3D and multi - modal 3D + 2D face recognition. *Computer Vision and Image Understanding*, *101*(1), 1–15.

Gandhe, S. T., Talele, K. T., & Keskar, A. G. (2007). Intelligent Face Recognition Techniques : A Comparative Study. *International Journal on Graphics, Vision and Image Processing*, *7*(2), 59–66.

Garg, R., Mittal, B., & Garg, S. (2011). Histogram Equalization Techniques For Image Enhancement. *Iject*, *2*(1), 107–111.

Gawande, M. P., &, & Agrawal, D. G. (2014). Face Recognition Using PCA and Different Distance Classifiers. *Journal of Electronic and Communication Engineering*, *9*(1), 1–5.

Gnana, D. A., Alias, S. A., & Leavline, E. J. (2016). Literature Review on Feature Selection Methods for High-Dimensional Data. *International Journal of Computer Applications*, *136*(1), 9–17. http://doi.org/10.5120/ijca2016908317

Goldstein, A. J, Harmon, L. D., & Lesk, A. B. (1971). Identification of Human Faces. *Proceedings of IEEE*, *59*(5), 748–760. http://doi.org/10.1109/HiPC.2012.6507515

Gonzalez, R & Wood, R. (2000). *Digital Image Processing*.

Grigorescu, S. E. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, *11*(10), 1160–1167. http://doi.org/10.1109/TIP.2002.804262

Gupta, E. S., & Kaur, E. Y. (2014). Review of Different Histogram Equalization Based Contrast Enhancement Techniques. *InternationalJournal of Advanced Research in Computer and Communication Engineering*, *3*(7), 7585–7589.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection 1 Introduction. *Journal of Machine Learning Rsearch*, *3*(1), 1157–1182.

Hafez, S. F., Selim, M. M., & Zayed, H. H. (2015). 3D Face Recognition Based on Normal Map Features Using Selected Gabor filters and Linear Discriminant Analysis. *International Journal of Biometrics*, *7*(4), 373–390. http://doi.org/10.1504/IJBM.2015.076138

Haider, W., Bashir, H., Sharif, A., Sharif, I. & Wahab, A. (2014). A Survey on Face Detection and Recognition Techniques. *Research Journal of Recent Sciences*, *3*(4), 56–62. http://doi.org/10.5815/ijmecs.2014.08.05

Han, K & Kim, J. (2005). Quantum-Inspired Evoluntionary Algorithm for a Class of Combinatorial Optimization. *IEEE Transactions on Evolutionary Computation*, *6*(6), 580–593. http://doi.org/10.1109/TEM.2011.2128873

Heisele, B., Ho, P., & Poggio, T. (2001). Face recognition with support vector machines: global versus component-based approach. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, *2*, 688–694. http://doi.org/10.1109/ICCV.2001.937693

Hira, Z. M., & Gillies, D. F. (2015). A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Advances in Bioinformatics*, *2015*, 198363. http://doi.org/10.1155/2015/198363

Homayon, S., & Cruz, S. (2015). Iris Recognition for Personal Identifcation Using Lamstar Neural Network. *International Journal of Computer Science & Information Technology*, *7*(1), 1–8. http://doi.org/10.5121/ijcsit.2015.7101

Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events : Theoretical Background. *Int. Journal of Engineering Research and Applications*, *3*(5), 605–610.

Imani, M. B., Pourhabibi, T., Keyvanpour, M. R., & Azmi, R. (2012). A New Feature Selection Method Based on Ant Colony and Genetic Algorithm on Persian Font Recognition. *International Journal of Machine Learning and Computing*, *2*(3), 278–282.

Jafri, R., & Arabnia, H. R. (2009). A Survey of Face Recognition Techniques. *Journal of Information Processing Systems*, *5*(2), 41–68.

Jain, A. K., Ross, A., & Pankanti, S. (2006). Biometrics : A Tool for Information Security. *IEEE Transactions on Information Forensics and Secuirty*, *1*(2), 125–143. http://doi.org/10.1109/TIFS.2006.873653

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An Introduction to Biometric Recognition. *IEEE Transactions of Circuits and Systems for Video Technology, Special Issue on Image - and Video-Based Biometrics.*, *14*(1), 1–29.

Jain, K. (2017). A survey on feature selection techniques. *International Journal of Innovations in Engineering Research and Technology, 4*(5), 5–8.

Jain, R. (2004). Application Fields for Biometrics Technology, 1–10.

Jaiswal, S., Gwalior, M. I. T. S., & Gwalior, M. I. T. S. (2011). Comparison Between Face Recognition Algorithm-Eigenfaces, Fisherfaces and Elastic Bunch Graph Matching. *Journal of Global Research in Computer Science*, *2*(7), 3–9.

Jang, J., Han, K., & Kim, J. (2004). Face Detection using Quantum- inspired Evolutionary Algorithm". IEEE transaction, Evolutionary Computation. *IEEE Transactions on Evolutionary Computation*, *9*, 2100–2106.

Javed, A. (2013). Face Recognition Based on Principal Component Analysis. *International Journal of Image, Graphics and Signal Processing*, *5*(2), 38–44. http://doi.org/10.5815/ijigsp.2013.02.06

Jemma, Y., & Khanfir, S. (2009). Automatic Local Gabor Features Extraction for Face Recognition. *International Journal of Computer Science and Information Security*, *3*(1), 1–7.

Jin, Y., & Ruan, Q. (2009). Face Recognition Using Gabor-Based Improved Supervised Locality Preserving Projections. *Computing and Informatics*, *28*, 81–95.

Joshi, A. G. & Deshpande, A. S. (2015). Review of Face Recognition Techniques. *International Journal of Advanced Research in Computer Science and Software Engineeering*, *5*(1), 71–75.

Kanade, T. (1973). *Computer recognition of human faces*.University of Kyoto, Japan

Karamizadeh, S., Abdullah, S. M., Manaf, A. A., Zamani, M., & Hooman, A. (2013). An Overview of Principal Component Analysis. *Journal of Signal and Information Processing*, *4*(August), 173–175. http://doi.org/10.4236/jsip.2013.43B031

Karnan, M., Akila, M. & Kalamani, A. (2009). Feature Subset Selection Using Ant Colony Optimization. *Journal of Engineering and Technology Research*, *1*(5), 72–80.

Kasar, M. M., Bhattacharyya, D., & Kim, T. (2016). Face Recognition Using Neural Network : A Review. *International Journal of Security and Its Applications*, *10*(3), 81–100.

Kashem, M. A., Akhter, N., Ahmed, S., & Alam, M. (2011). Face Recognition System Based on Principal Component Analysis (PCA) with Back Propagation Neural Networks (BPNN). *International Journal of Scientific & Engineering Research*, *2*(6), 1–10.

Kaur, S., Kaur R., & Randhawa, Y. (2014). An Approach to Detect and Recognize Face using Swarm Intelligence and Gabor Filter. *International Journal of Adavnaced Research in Computer Science and Software Engineering*, *4*(6), 1389–1393.

Kaur, R., & Rajput, R. (2013). Face recognition and its various techniques : a review. *Internationa Journal of Science, Engineering and Technology Research*, *2*(3), 670–675.

Khan, S. A., Ahmad, M., Nazir, M., & Riaz, N. (2013). A Comparative Analysis of Gender Classification Techniques. *International Journal of Bio-Science and Bio-Technology*, *5*(4), 223–244. http://doi.org/10.5829/idosi.mejsr.2014.20.01.11434

Khryashchev, V., Priorov, A., Stepanova, O. & Nikitin, A. (2015). Face Recognition Using Local Quantized Patterns and Gabor Filters. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*, 25–27. http://doi.org/10.5194/isprsarchives-XL-5-W6-59-2015

Krishna, A. S., Rao, G. S., & Sravya, M. (2013). Constrast Enhancement Technique Using Histogram Equalization Methods on Colour Images with Poor Ligthening. *International Journal of Computer, Engineering and Applications*, *3*(4), 2013.

Kumar, G., & Bhatia, P. K. (2014). A Detailed Review of Feature Extraction in Image Processing Systems. In *Internationa Conference on Advanced Computing & Communication* (pp. 5–12). http://doi.org/10.1109/ACCT.2014.74

Kumar, V. & Minz, S. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3), 211–229. http://doi.org/10.6029/smartcr.2014.03.007

Kumar, P., & A., Shaikh A. A, Katwate D. P., & Jamdar, R. N. (2014). Face Recognition System Using PCA. *International Journal of Research in Advent Technology*, *2*(4), 139–143.

Kumar, S., & Kaur, H. (2012). Face Recognition Techniques : Classification and Comparisons. *International Journal of Information Technology and Knowledge Management*, *5*(2), 361–363.

Kumar, V., & Minz, S. (2014). Feature Selection : A literature Review. *Smart Computing Review*, *4*(3). http://doi.org/10.6029/smartcr.2014.03.007

Kumari, B., & Swarnkar, T. (2011). Feature Subset Selection in Large Dimensionality Micro array using Wrapper Method : A Review. *International Journal of Computer Science and Information Technologies*, *2*(2), 792–798.

Kyrki, V., Kamarainen, J. K., & Kälviäinen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters*, *25*(3), 311–318. http://doi.org/10.1016/j.patrec.2003.10.008

Lades, M., Member, S., Vorbruggen, J. C., Buhmann, J., Lange, J., Wurtz, R. P., & Konen, W. (1993). Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Transactions on Computers*, *42*(3), 300–310.

Lalit & Ankur, G. (2015). A Review : Image enhancement approaches using Histogram Equalization. *International Journal of Enigeering and Computer Science*, *4*(8), 14036–14038. http://doi.org/10.18535/ijecs/v4i8.73

Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Image Processing, IEEE Transactions on*, *11*(4), 467–476. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=999679

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. In *Knowledge and Data Engineering, IEEE Transactions on*, (Vol. 17, pp. 491–502).

López, F. A. P., Mora, H. M., & Selva, J. A. G. (2016). A neural network framework for face recognition by elastic bunch graph matching. *Recent Adavances on Electroscience and Computers*, *3*(2), 75–81.

Malik, D., & Bansal, S. (2016). Face Recognition Based on Principal Component Analysis and Linear Discriminant Analysis. *International Journal of Research in Electronics and Communication Technology*, *3*(2), 7–10.

Miche, Y., Bas, P., Lendasse, A., Jutten, C., & Simula, O. (2007). Advantages of Using Feature Selection Techniques on Steganalysis Schemes. *9th International Work-Conference on Artificial Neural Networks, IWANN'2007*, 606–613. http://doi.org/10.1007/978-3-540-73007-1_73

Mohammed, J. (2014). Face Recognition using Principle Gabor Filter. *International Journal of Advanced Research in ComputerScience and Software Engineering*, *4*(2), 176–180.

Moon, H., & Phillips, P. J. (2001). Computational and Performance Aspects of PCA-Based Face-Recognition Algorithm. *Perception*, *30*, 303–321. http://doi.org/10.1068/p2896

Mounika, B. R, Reddy N. J, & R. V. B. . (2012). A Neural Network Based Face Detection Using Gabor Filter Response. *International Journal of Neuaral Networks*, *2*(1), 6–9.

Munir, S., Gupta, V., Nemade, S., & Alam, M. Z. (2011). Face recognition using support vector machines. *International Journal of Computer Trends and Technology*, *4*(8), 2556–2560.

Murugan, D., Arumugam, S., Rajalakshmi & Manish, T. (2010). Performance Evaluation of Face Recognition Using Gabor Filter, Log Gabor filter and Discrete Wavelet transform. *International Journal of Computer & Information Technology*, *2*(1), 125–133.

Nemati, S., Basiri M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A Novel ACO-GA Hybrid Algorithm for Feature Selection in Protein Function Prediction. *Expert Systems with Applications*, *36*, 12086–12094.

Oreski, D., & Novosel, T. (2014). Comparison of Feature Selection Techniques in Knowledge Discovery Process. *TEM Journal*, *3*(4), 285–290.

Pal, S. K., Chourasia, U., & Ahirwar, M. (2013). A Method for Face Detection based on Wavelet Transform and optimised feature selection using Ant Colony

Optimisation in Support Vector Machine. *International Journal of Innovative Research in Computer and Communication Engineering*, *1*(2), 358–363.

Pali, V & Bhaiya, L. K. (2013). Genetic Algorithm Based Feature Selection and BPNN Based Classification for Face Recognition. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(5), 269–273.

Parmar, D. N., & Mehta, B. B. (2013). Face Recognition Methods & Applications. *International Journal Computer Technology & Applications*, *4*(1), 84–86.

Patil, S. S., & Kiran, S. K. (2014). Rice Genomes Classification based on Efficient Distance Measures Classifiers. *International Journal of Engineering Sciences & Research Technology*, *3*(12), 781–784.

Patinge, P. B. & Deshmukh, C. N. (2015). Local Binary Pattern Base Face Recognition System. *International Journal of Science, Engineering and Technology Research*, *4*(5), 1356–1361.

Patra, B., & Dash, S. (2016). A FRGSNN Hybrid Feature Selection Combining FRGS filter and GSNN wrapper. *International Journal of Latest Trends in Engineering and Technology*, *7*(2), 8–15.

Paul, L. C., & Sumam, A. Al. (2012). Face Recognition Using Principal Component Analysis Method. *International Journal of Advanced Research in Computer Engineering & Technology*, *1*(9), 135–139.

Phillips, P. J. (1999). Support Vector Machines applied to Face Recognition. *Advances in Neural Information Processing Systems 11*, 803–809.

Ponnmoli, K. M. (2014). Analysis of Face Recognition using Manhattan Distance Algorithm with Image Segmentation. *International Journal of Computer Science and Mobile Computing*, *3*(7), 18–27.

Prasanthi, B. V., Hussain, S. M., Kanakam, P., Chakravarthy, A. S. (2015). Palm Vein Biometric Technology : An Approach to Upgrade Security in ATM Transactions. *International Journal of Computer Applications*, *112*(9), 1–5.

Purohit, M & Mehta, P. (2016). An Evaluation of Feature Selection Methods for Multiclass Learning in Bio Informatics. *Interantional Journal of Computer Applications*, *138*(6), 24–27.

Rady, H. (2011). Face Recognition using Principle Component Analysis with D ifferent D istance Classifiers. *International Journal of Computer Science and Network Security*, *11*(10), 134–144.

Ramaswami, M., & Bhaskaran, R. (2009). A Study on Feature Selection Techniques in Educational Data Mining. *Journal of Computing*, *1*(1), 7–11. Retrieved from http://arxiv.org/abs/0912.3924

Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *4*(2), 164–171. http://doi.org/10.1109/4235.850656

Rini, D. P., Shamsuddin, S. M., & Yuhaniz, S. S. (2011). Particle Swarm Optimization : Technique , System and Challenges. *International Journal of Computer Applications*, *14*(1), 19–27.

Roweis, S. T., & Lawrence, K. S. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, *290*, 2323–2326. http://doi.org/10.1126/science.290.5500.2323

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–2517. http://doi.org/10.1093/bioinformatics/btm344

Sanchez-avila, C., & Gonzalez-marcos, A. (2000). Biometric identification through hand geometry measurements - Pattern Analysis and Machine Intelligence, IEEE Transactions on. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(10), 1168–1171.

Saraç, E., Ay, S., & Özel, G. (2014). An Ant Colony Optimization Based Feature Selection for Web Page Classification. *The Scientific World Journal*, *2*(3), 1–17.

Saravanan, C. (2010). Color Image to Grayscale Image Conversion (pp. 1–4). http://doi.org/10.1109/ICCEA.2010.192

Senthil, K. P., & Lopez P. D., A. (2016). A Review on Feature Selection MethodsforHigh Dimensional Data. *International Journal of Engineering and Technology*, *8*(2), 669–672.

Sergyán, S. (2009). A new approach of face detection-based classification of image databases. *Acta Polytechnica Hungarica*, *6*(1), 175–184.

Shah, J. H., Sharif, M., Raza, M., & Azeem, A. (2013). A Survey : Linear and Nonlinear PCA Based Face Recognition Techniques. *International Arab Journal of Information Technology*, *10*(6), 536–545.

Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing*, *27*(6), 803–816. http://doi.org/10.1016/j.imavis.2008.08.005

Sharif, M., Mohsin, S., & Javed, M. Y. (2012). A Survey: Face Recognition Techniques. *Research Journal of Applied Sciences, Engineering and Technology*, *4*(23), 41–50.

Sharma, H. & Rizvi, M. (2017). Prediction of Heart Disease using Machine Learning Algorithms : A Survey. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(8), 99–104.

Sharma, M., & Batra, A. (2014). Analysis of Distance Measures in Content Based Image Retrieval. *Global Journal of Computer Science and Technology*, *14*(2), 11–16.

Shebani, Q. Al. (2015). The feasability of implementing a face recognition system based on Gabor filter and nearst neighbour techniques in an FPGA device for door control systems. *Journal of Computer*, *10*(2), 115–129.

Sheena, Krishan, K., & Gulshan, K. (2016). Analysis of Feature Selection Techniques : A Data Mining Approach. In *International Conference on Engineering & Technology* (pp. 17–21).

Shen, L., & Bai, L. (2006). A Review on Gabor Wavelets for Face Recognition. *Pattern Analysis and Applications*, *9*(2–3), 273–292. http://doi.org/10.1007/s10044-006-0033-y

Shen, L., & Ji, Z. (2009). Gabor Wavelet Selection and SVM Classification for Object Recognition. *Acta Automatica Sinica*, *35*(4), 350–355. http://doi.org/10.1016/S1874-1029(08)60082-8

Shende, P. M., & Sarode, M. V. (2016). Multiple Biometric System Application : Iris And Fingerprint Recognition System. *International Journal of Application or*

Innovation in Engineering & Management, *5*(3), 34–38.

Shweta, K., & Singh, A. (2013). An Effect and Analysis of Parameter on Ant Colony Optimization for Solving Travelling Salesman Problem. *International Journal of Computer Science and Mobile Computing*, *2*(11), 222–229.

Silva, J., Aguiar, A., Silva, F., & Inesctec, C. (2015). A Parallel Computing Hybrid Approach for Feature Selection. In *IEEE 18th International Conference on Computational Science and Engineering* (pp. 97–104). http://doi.org/10.1109/CSE.2015.34

Singh, R. P., & Dixit, M. (2015). Histogram Equalization : A Strong Technique for Image Enhancement. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(8), 345–352.

Singh, S., Sharma, M., & Rao, D. (2011). Accurate face recognition using pca and lda. *… on Emerging Trends in Computer and …*, 1–7. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.374.8351&rep=rep1&type=pdf

Singhal, Z., Gupta, P., & Garg, K. (2012). Biometric Recognition : Personal Identification Technique. *International Journal Computational Engineering & Management*, *15*(3), 6–10.

Sirovich, L., & Kirby, M. (1987). Low Dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America A*, *4*(3), 519–524.

Sivakumar, S. & Chandrasekar, C. (2014). Feature selection based on filter approach using mutual information with Genetic algorithm. *International Journal of Advance Studies in Computer Science and Engineering*, *3*(4), 5–9.

Sodhi, K. S., & Lal, M. (2013). Comparative Analysis of PCA-based Face Classifiers. *Interantional Journal of Application or Innovation in Engineering & Mnanagement*, *2*(7), 341–348.

Srivastva, A., & Singh, V. (2011a). Biometrics Based Identification Techniques (BIT). *Journal of Global Research in Computer Science*, *2*(11), 11–15.

Srivastva, A., & Singh, V. (2011b). Biometrics BASED Identification Techniques (BIT). *Journal of Global Research in Computer Science*, *2*(11), 11–15.

Stark, J. A. (2000). Generalizations of Histogram Equalization. *IEEE Transaction on Image Processing*, *9*(5), 889–896.

Struc, V., Gajsek, R., & Pavesic, N. (2009). Principal Gabor filters for Face Recognition. In *IEEE 3rd International Conference on Biometric: Theory, Application, and Systems, 2009. BTAS '09.* (pp. 11–16). http://doi.org/10.1109/BTAS.2009.5339020

Struc, V. & Pavesic, N. (2009). A Case Study on Appearance Based Feature Extraction Techniques and Their Susceptibility to Image Degradations for the Task of Face Recognition. *International Journal of Electrical, Computer Energetic, Electronic and Communication Engineering*, *3*(6), 648–656.

Štruc, V., Gajšek, R., & Pavešić, N. (2009). Principal gabor filters for face recognition. *IEEE 3rd International Conference on Biometrics: Theory, Applications and Systems, BTAS 2009*, (October 2016). http://doi.org/10.1109/BTAS.2009.5339020

Sufyanu, Z., Mohamad, F. S., Yusuf, A. A., & Mustafa, B. (2016). Enhanced Face Recognition Using Discrete Cosine Transform. *Engineering Letters*, *24*(1), 1–10.

Suganya, S., & Menaka, D. (2014). Performance Evaluation of Face Recognition Algorithms. *International Journal in Recent and Innovation Trends in Computing and Communication*, *1*(2), 135–140.

Sutha, K. (2015). A Review of Feature Selection Algorithms for Data Mining Techniques. *International Journal on Computer Science and Engineering*, *7*(6), 63–67.

Tan, X., & Triggs, W. (2007). Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition. *Analysis and Modeling of Faces and Gestures*, *4778*, 235–249. http://doi.org/10.1007/978-3-540-75690-3_18

Thakur, A. S., & Sahayam, N. (2013). Speech Recognition Using Euclidean Distance. *International Journal of Emerging Technology and Advanced Engineering*, *3*(3), 587–590.

Thamman, P. & Bhatia, R. (2014). A New Methodology for Improvement of Contrast to Show Fractures in X-ray Images. *International Journal of Computer Science and Information Technologies*, *5*(5), 6400–6403.

Thangairulappan, K., & Jeyasingh, J. (2012). Face Representation Using Combined Method of Gabor Filters, Wavelet Transformation and DCV and Recognition Using RBF. *Journal of Intelligent Learning System and Applications*, *4*(1), 266–273. http://doi.org/10.4236/jilsa.2012.44027

Thangairulappan, K., Jeyasingh, J., Beulah, J., Jeyasingh, V., & Jeyasingh, J. (2012). Face Representation Using Combined Method of Gabor Filters, Wavelet Transformation and DCV and Recognition Using RBF. *Journal of Intelligent Learning ...*, *2012*(November), 266–273. http://doi.org/10.4236/jilsa.2012.44027

Turk, M., & Pentland, a. (1991). Eigenfaces for Face Detection / Recognition. *Journal of Cognitive Neuroscience*, *3*(1), 1–11. http://doi.org/10.1162/jocn.1991.3.1.71

Uddin, M. N., Sharmin, S., Hasnat, A., Ahmed, S., & Hasan, E. (2011). A Survey of Biometrics Security System. *International Journal of Computer and Network Secuirty*, *11*(10), 16–23.

Udovychenko, Y., Popov, A., & Chaikovsky, I. (2015). Binary Classification of Heart Failures Using k-NN with Various Distance Metrics. *International Journal of Electronics and Telecommunications*, *61*(4), 339–344. http://doi.org/10.2478/eletel-2015-0044

Vidyavathi, B. M & Ravikumar, C. N. (2008). A Novel Hybrid Filter Feature Selection Method for Data Mining. *Ubiquitous Computing and Communication Journal*, *3*(3), 118–121.

Vinay, A., Shekhar, V. S., Murthy, K. N. B., & Natarajan, S. (2015). Face Recognition using Gabor Wavelet Features with PCA and KPCA - A Comparative Study. *Procedia - Procedia Computer Science*, *57*, 650–659. http://doi.org/10.1016/j.procs.2015.07.434

Wadkar, P. D., & Wankhade, M. (2012). Face Recognition Using Discrete Wavelet Transforms. *International Journal of Advanced Engineering Technology*, *3*(1), 1–4.

Wahan, M. N., Nefti-Meziani, S. & Atyabi, A. (2015). A Comprehensive Review of Swarm Optimization Algorithms. *PloS One*, *10*(5), 112–123.

Wahid, T. (2013). Face Recognition using Local Binary Patterns (LBP). *Global Journal of Computer Science and Technology Graphics & Vision*, *13*(4), 1–8.

Wang, H., & Liu, S. (2016). An Effective Feature Selection Approach Using the Hybrid Filter Wrapper. *International Journal of Hybrid Information Technology*, *9*(1), 119–128.

Wiskott, L., Fellous, J. M., Krüger, N., & von der Malsburg, C. (1997). Face recognition using by Elastic Bunch Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775–779. http://doi.org/10.1109/34.598235

Wong, l., Low M. Y., & Chong, C. C. (2010). International Journal on Artificial. *International Journal on Artificial Intelligence Tools*, *19*(3), 305–334.

Wongchoosuk, C., Lutz, M., & Kerdcharoen, T. (2009). Detection and Classification of Human Body Odor Using an Electronic Nose. *Sensors*, *9*, 7234–7249. http://doi.org/10.3390/s90907234

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(1), 131–137. http://doi.org/10.1109/TPAMI.2004.1261097

Yang, X. (2014). Swarm Intelligence Based Algorithms : A Critical Analysis. *Evoluntionary Intelligence*, *7*(1), 1–16.

Yang, X. S. (2011). Bat Algorithm for Multi-objective Optimization. *Internationa Journal of Bio-Inspired Computation*, *3*(5), 267–274.

Yildirim, P. (2015). Filter Based Feature Selection Methods for Prediction of Risks in Hepatitis Disease. *International Journal of Machine Learning and Computing*, *5*(4), 258–263. http://doi.org/10.7763/IJMLC.2015.V5.517

Yu, L., & Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, *5*(1), 1205–1224. http://doi.org/10.1145/1014052.1014149

Yuce, B., Packianather, M. S., Mastrocinque, E., & Pham, D. T. (2013). Honey Bees Inspired Optimization Method : The Bees Algorithm. *Insects*, *4*(4), 646–662. http://doi.org/10.3390/insects4040646

Zhang, D. Zhou, Z,. & Chen, S. (2006). Diagonal Principal Component Analysis for Face Recognition. *Pattern Recognition*, *39*(1), 140–142.

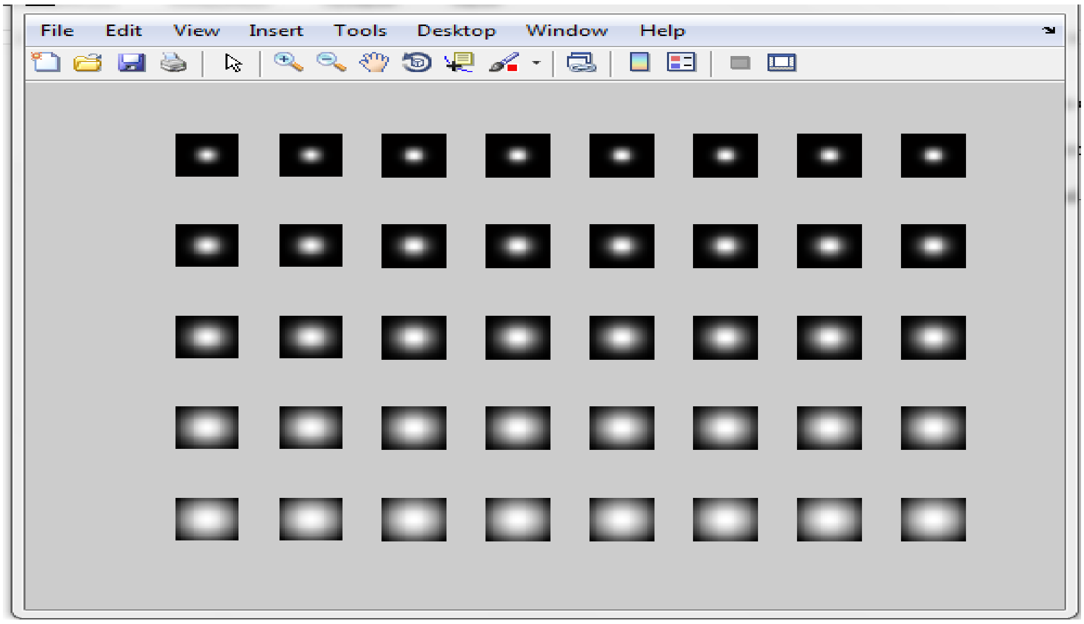
Zhen, X. (2013). Feature Extraction and Representation for Human Action Recognition (pp. 1–21). Retrieved from http://www.intechopen.com/books/face-recognition/feature-extraction-and-representation-for-recognition

Zhou, J., Zhang, S., Mei, H., & Wang, D. (2016). A method of facial expression recognition based on Gabor and NMF. *Pattern Recognition and Image Analysis*, *26*(1), 119–124.

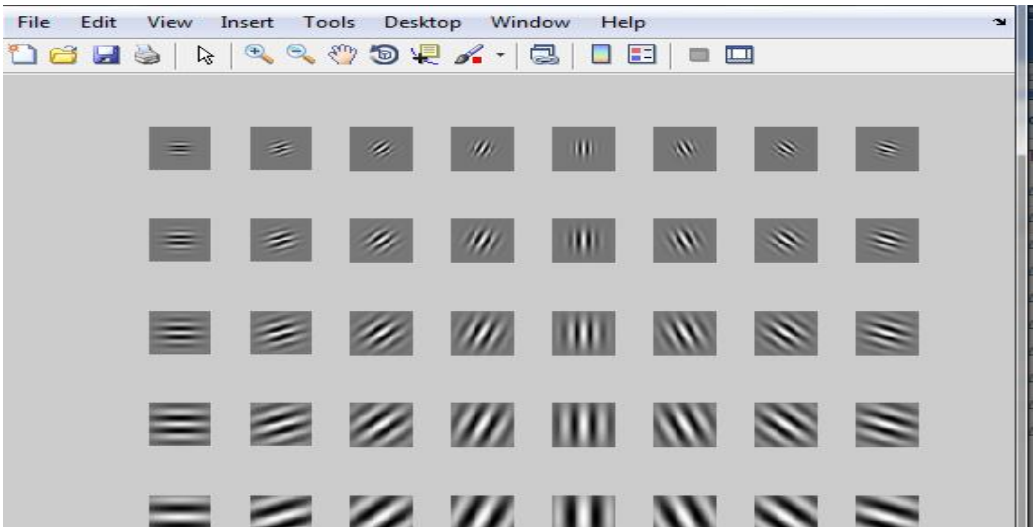
Zhu, Y., & Huang, C. (2012). An Adaptive Histogram Equalization Algorithm on the Image. *Physics Procedia*, *25*, 601–608. http://doi.org/10.1016/j.phpro.2012.03.132

Zou, J., Ji, Q., & Nagy, G. (2007). A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing : A Publication of the IEEE Signal Processing Society*, *16*(10), 2617–2628. http://doi.org/10.1109/TIP.2007.904421

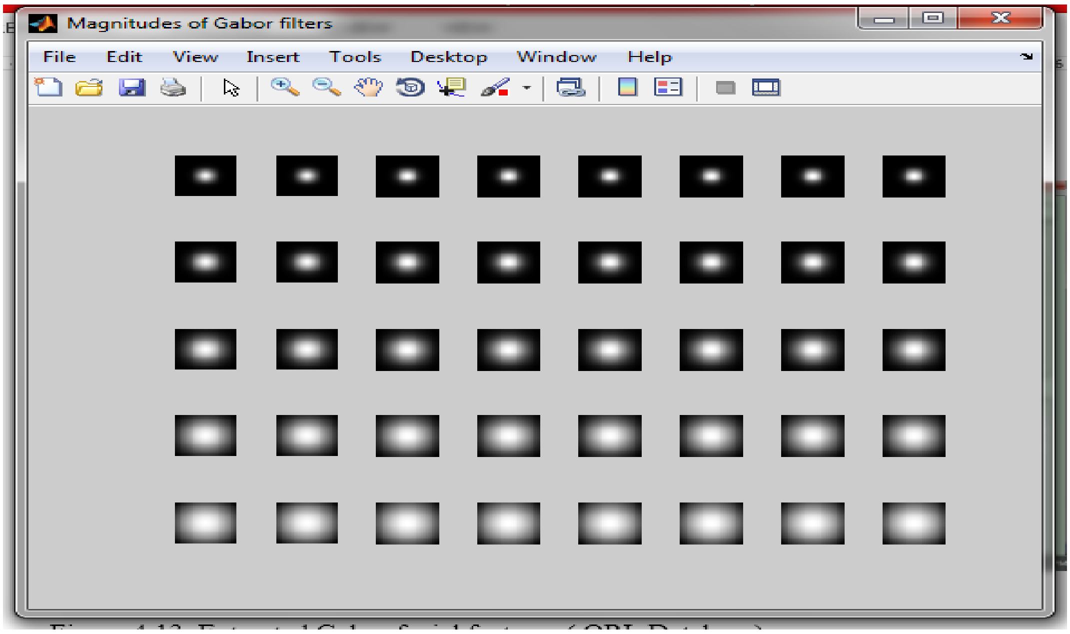
APPENDIX A: SAMPLE OF REAL AND MAGNITUDE PART OF EXTRACTED GABOR FEATURES



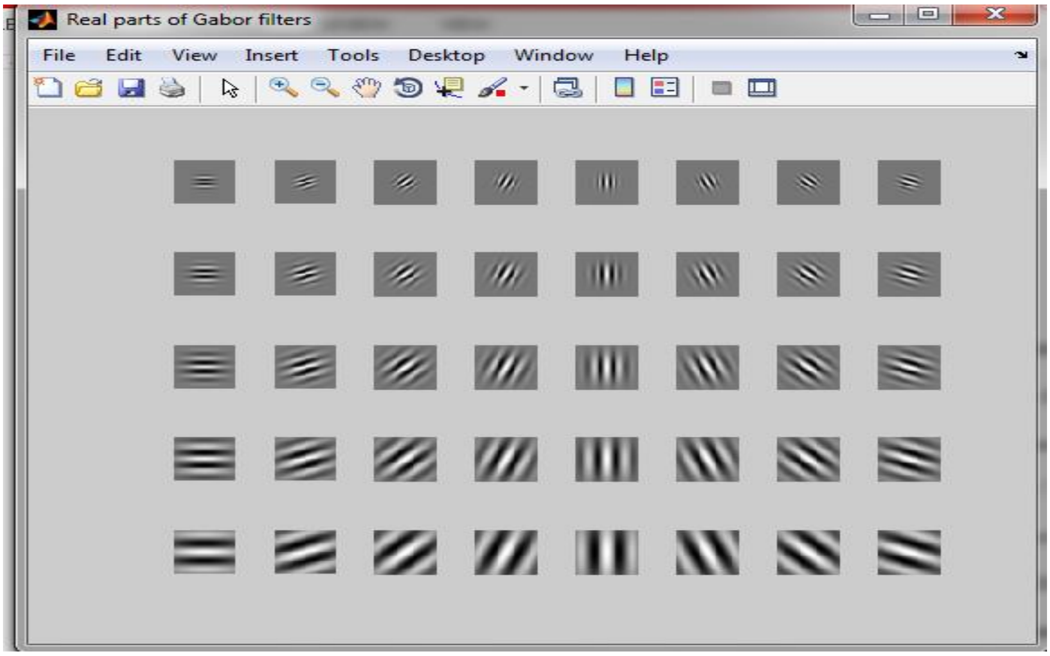
Sample of Magnitude Part with a bank of 40 Gabor-filters (LAFI Database)



Sample of Real Part with a bank of 40 Gabor-filters (LAFI Database)



Sample of Magnitude Part with a bank of 40 Gabor-filters (ORL Database)



APPENDIX B: SAMPLE CODE

```
function varargout = Gabor_Filter_Face_Recognition_Technique(varargin)
% GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE MATLAB code for
Gabor_Filter_Face_Recognition_Technique.fig
%      GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE, by itself, creates a
new GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE or raises the existing
%      singleton*.
%
%      H = GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE returns the
handle to a new GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE or the
handle to
%      the existing singleton*.
%
%
GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE('CALLBACK',hObject,ev
entData,handles,...) calls the local
%      function named CALLBACK in
GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE.M with the given input
arguments.
%
%      GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE('Property','Value',...)
creates a new GABOR_FILTER_FACE_RECOGNITION_TECHNIQUE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before
Gabor_Filter_Face_Recognition_Technique_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to
Gabor_Filter_Face_Recognition_Technique_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Gabor_Filter_Face_Recognition_Technique

% Last Modified by GUIDE v2.5 13-May-2017 13:49:42

% Script Written by Software Developer Drey from delsmy.com

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@Gabor_Filter_Face_Recognition_Technique_OpeningFcn, ...
                   'gui_OutputFcn',
@Gabor_Filter_Face_Recognition_Technique_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Gabor_Filter_Face_Recognition_Technique is made visible.
function Gabor_Filter_Face_Recognition_Technique_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Gabor_Filter_Face_Recognition_Technique (see VARARGIN)

% Choose default command line output for Gabor_Filter_Face_Recognition_Technique
handles.output = hObject;
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%------------------------------------------------------------------------
set(handles.text23,'string','');
set(handles.text24,'string','');
set(handles.text25,'string','');
set(handles.text26,'string','');
set(handles.radiobutton13,'enable','off');
set(handles.radiobutton14,'enable','off');
set(handles.radiobutton1,'enable','off');
set(handles.radiobutton2,'enable','off');
set(handles.radiobutton15,'enable','off');
set(handles.radiobutton16,'enable','off');
set(handles.Load_Database,'enable','off');
set(handles.Chebyshev,'enable','off');
set(handles.pushbutton6,'enable','off');
set(handles.pushbutton12,'enable','off');
set(handles.pushbutton14,'enable','off');
set(handles.Image_Normalization,'enable','off');
set(handles.Gabor,'enable','off');
set(handles.ACO,'enable','off');
set(handles.popupmenu1,'enable','off');
set(handles.pushbutton19,'enable','off');
distance_classifier;
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%------------------------------------------------------------------------
```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Gabor_Filter_Face_Recognition_Technique wait for user response
(see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Gabor_Filter_Face_Recognition_Technique_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
pause(0.5)

choice=questdlg('PLEASE SELECT IMAGE TYPE',...
   'SELECT IMAGE TYPE',...
   'OK','OK');
   % Handles response
switch choice
   case 'OK'
set(handles.radiobutton13,'BackgroundColor','red')
set(handles.radiobutton14,'BackgroundColor','red')
pause(0.2)
set(handles.radiobutton13,'BackgroundColor','white')
set(handles.radiobutton14,'BackgroundColor','white')
pause(0.2)
set(handles.radiobutton13,'BackgroundColor','red')
set(handles.radiobutton14,'BackgroundColor','red')
pause(0.2)
set(handles.radiobutton13,'BackgroundColor','white')
set(handles.radiobutton14,'BackgroundColor','white')
pause(0.2)
set(handles.radiobutton13,'BackgroundColor','red')
set(handles.radiobutton14,'BackgroundColor','red')
pause(0.2)
set(handles.radiobutton13,'BackgroundColor',[0.7568 0.8666 0.7764])
set(handles.radiobutton14,'BackgroundColor',[0.7568 0.8666 0.7764])
set(handles.radiobutton13,'enable','on')
set(handles.radiobutton14,'enable','on')
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in Image_Normalization.
function Image_Normalization_Callback(hObject, eventdata, handles)
% hObject    handle to Image_Normalization (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
set(handles.text24,'string','Normalized Image');
global variable size type
Image_size=size;
h = waitbar(0,['Normalising ' , Image_size , ' Image size in Database Pls Wait...']);
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
close(h);
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Normalise Image
if (type == 1 & variable == 1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\75 x
75');
elseif (type == 1 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\100
x 100');
elseif (type == 1 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\125
x 125');
elseif (type == 1 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\150
x 150');
elseif (type == 2 & variable ==1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\75 x 75');
elseif (type == 2 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\100 x 100');
elseif (type == 2 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\125 x 125');
elseif (type == 2 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\150 x 150');
end
normaliseddata=dir('*.jpg');
```

```
fileNames={normaliseddata.name};
nume=numel(fileNames);
for ifile=1:numel(fileNames)
    newName=sprintf('%1d.jpg',ifile);
    %movefile(fileNames{ifile},newName);
end
q=1;
p=1;
  set(handles.text24,'string','Normalized Image');
for i=1:nume
oq=imread(strcat(num2str(q),'.jpg'));
timing=tic;
axes(handles.axes16);
    imshow(oq);
    imagetiming=toc(timing);
    avgt=imagetiming*nume;
    name=strcat(num2str(q),'.jpg');
    set(handles.text5,'string',name);

q=q+1;
p=p+1;

pause(0.1)

end
Normalisedtime=num2str(avgt);
ldt=[Normalisedtime , ' Secs'];
set(handles.text10,'string',ldt);
tot=nume;
to=num2str(tot);
msgbox(['Image Normalising Complete!!!  ' , to ,'  Images Normalised'], 'Image
Database', 'warn');
% meso=[to , ' Images Normalised'];
% set(handles.text28,'string',meso);
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
set(handles.Gabor,'enable','on');
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
% --- Executes on button press in Gabor.
function Gabor_Callback(hObject, eventdata, handles)
% hObject    handle to Gabor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Info
h = waitbar(0,'Extracting Image features with Gabor Please wait...');
steps = 1000;
for step = 1:steps
```

```
    % computations take place here
    waitbar(step / steps)
end
close(h);
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%% Gabor Filter
j=gaborFilterBank(5,8,39,39);
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%% Normalise Image
global variable type
if (type == 1 & variable == 1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\75 x 75');
elseif (type == 1 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\100 x 100');
elseif (type == 1 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\125 x 125');
elseif (type == 1 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\150 x 150');
elseif (type == 2 & variable ==1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\75 x 75');
elseif (type == 2 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\100 x 100');
elseif (type == 2 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\125 x 125');
elseif (type == 2 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\150 x 150');
end
extractimage=dir('*.jpg');
fileNames={extractimage.name};
nume=numel(fileNames);
for ifile=1:numel(fileNames)
    newName=sprintf('%1d.jpg',ifile);
    %movefile(fileNames{ifile},newName);
    %movefile('gaboree.mat','gaboree.jpg')
end
q=1;
p=1;
  set(handles.text39,'string','Gabor Image');
  set(handles.text26,'string','Gabor Extracted Image');
```

```
for i=1:nume
oq=imread(strcat(num2str(q),'.jpg'));
timing=tic;
axes(handles.axes18);
    imshow(oq);
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
%% Gabor Extraction
k=gaborFeatures(oq,j,40,40);
set(handles.uitable1,'Data',k);
name=strcat(num2str(q),'.jpg');
set(handles.text32,'string',name);
set(handles.text38,'string',name);
axes(handles.axes13);
   imshow(k);
save k
e=getimage(handles.axes13);

if (type == 1 & variable == 1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\75 x 75\Gabor_Extract_Image');
elseif (type == 1 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\100 x 100\Gabor_Extract_Image');
elseif (type == 1 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\125 x 125\Gabor_Extract_Image');
elseif (type == 1 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\150 x 150\Gabor_Extract_Image');
elseif (type == 2 & variable ==1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\75 x 75\Gabor_Extract_Image');
elseif (type == 2 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\100 x 100\Gabor_Extract_Image');
elseif (type == 2 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\125 x 125\Gabor_Extract_Image');
elseif (type == 2 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\150 x 150\Gabor_Extract_Image');
end

imwrite(e,[num2str(q),'.jpg']);

if (type == 1 & variable == 1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\75 x 75');
elseif (type == 1 & variable ==2);
```

```
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\100
x 100');
elseif (type == 1 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\125
x 125');
elseif (type == 1 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\N_Image\Train\150
x 150');
elseif (type == 2 & variable ==1);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\75 x 75');
elseif (type == 2 & variable ==2);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\100 x 100');
elseif (type == 2 & variable ==3);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\125 x 125');
elseif (type == 2 & variable ==4);
cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Train\150 x 150');
end
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
name=strcat(num2str(q),'.jpg');
%set(handles.text19,'string',name);
set(handles.text21,'string',name);
    imagetiming=toc(timing);
    avgt=imagetiming*nume;
q=q+1;
p=p+1;

pause(0.1)

end
gabortime=num2str(avgt);
ldt=[gabortime , ' Secs'];
set(handles.text13,'string',ldt);
msgbox('Features Extracted');
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
set(handles.ACO,'enable','on');
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
% --- Executes on button press in ACO.
function ACO_Callback(hObject, eventdata, handles)
% hObject    handle to ACO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
```

```
%-----------------------------------------------------------------------
%% Info
h = waitbar(0,'Extracting Image features with ACO Please wait...');
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
close(h);
acofeature;
msgbox('Features Extracted');
set(handles.Chebyshev,'enable','on');
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
h = waitbar(0,['Searching for the Nearest Neighbour Data between'...
    ' Testing and Training Image using Mahalanobis Distances Metrics Pls Wait...']);
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
timing=tic;
global type
if type==1
matrixg;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
mahatrain;
elseif type==2
matrixgORL
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
mahatrainORL;
end
close(h)
timing2=toc(timing);
timing2=num2str(timing2);
Mahatime=[timing2 , ' Secs'];
Maha='Maha Training Time';
set(handles.text28,'string',Mahatime)
set(handles.text51,'string',Maha)
msgbox('Mahalanobis Distance Created');
set(handles.popupmenu1,'enable','on');

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% andles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in Load_Database.
function Load_Database_Callback(hObject, eventdata, handles)
% hObject    handle to Load_Database (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
set(handles.text23,'string','Database Image');
%% Image Size
global variable size type

target=get(handles.radiobutton13,'Value')
targetg=get(handles.radiobutton14,'Value')

if (type == 1 & variable == 1);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\O_Image\Train\75 x 75');
elseif (type == 1 & variable ==2);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\O_Image\Train\100 x 100');
elseif (type == 1 & variable ==3);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\O_Image\Train\125 x 125');
elseif (type == 1 & variable ==4);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\O_Image\Train\150 x 150');
elseif (type == 2 & variable ==1);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\75 x 75');
elseif (type == 2 & variable ==2);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\100 x 100');
elseif (type == 2 & variable ==3);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\125 x 125');
elseif (type == 2 & variable ==4);
folder_name = uigetdir('C:\Users\ARO\Documents\MATLAB\Gabor\Image Database\ORL_Image\Train\150 x 150');
end
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Info
Image_size=size;
h = waitbar(0,['Loading ' , Image_size , ' Image size to Database Pls Wait...']);
steps = 1000;
```

```
for step = 1:steps
   % computations take place here
   waitbar(step / steps)
end
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%% select image directory
if ( folder_name ~= 0 )
   handles.folder_name = folder_name;
   guidata(hObject, handles);
else
   return;
end

if (~isfield(handles, 'folder_name'))
   errordlg('Please select an image directory first!');
   return;
end

% construct folder name foreach image type
pgmImagesDir = fullfile(handles.folder_name, '*.pgm');
pngImagesDir = fullfile(handles.folder_name, '*.png');
jpgImagesDir = fullfile(handles.folder_name, '*.jpg');
bmpImagesDir = fullfile(handles.folder_name, '*.bmp');

% calculate total number of images
num_of_png_images = numel( dir(pngImagesDir) );
num_of_jpg_images = numel( dir(jpgImagesDir) );
num_of_pgm_images = numel( dir(pgmImagesDir) );
num_of_bmp_images = numel( dir(bmpImagesDir) );

totalImages = num_of_png_images + num_of_jpg_images + num_of_bmp_images +
num_of_pgm_images;

pgm_files = dir(pgmImagesDir);
jpg_files = dir(jpgImagesDir);
png_files = dir(pngImagesDir);
bmp_files = dir(bmpImagesDir);

if ( ~isempty( pgm_files ) || ~isempty( jpg_files ) || ~isempty( png_files ) || ~isempty(
bmp_files ) )
   % read jpg images from stored folder name
   % directory and construct the feature dataset
   pgm_counter = 0;
   jpg_counter = 0;
   png_counter = 0;
   bmp_counter = 0;

   for k = 1:totalImages
```

```
    if ( (num_of_pgm_images - pgm_counter) > 0)
        imgInfoPGM = imfinfo( fullfile( handles.folder_name,
pgm_files(pgm_counter+1).name ) );
        if ( strcmp( lower(imgInfoPGM.Format), 'pgm') == 1 )
            % read images
            sprintf('%s \n', pgm_files(pgm_counter+1).name)
            % extract features
            image = imread( fullfile( handles.folder_name,
pgm_files(pgm_counter+1).name ) );
            [pathstr, name, ext] = fileparts( fullfile( handles.folder_name,
pgm_files(pgm_counter+1).name ) );

        end
name1=name;
ful=ext;
g=strcat(name1,ful);
name11=cellstr(g);
set(handles.text4,'string',name11);

        pgm_counter = pgm_counter + 1;

    elseif ( (num_of_jpg_images - jpg_counter) > 0)
        imgInfoJPG = imfinfo( fullfile( handles.folder_name,
jpg_files(jpg_counter+1).name ) );
        if ( strcmp( lower(imgInfoJPG.Format), 'jpg') == 1 )
            % read images
            sprintf('%s \n', jpg_files(jpg_counter+1).name)
            % extract features
            image = imread( fullfile( handles.folder_name,
jpg_files(jpg_counter+1).name ) );
            [pathstr, name, ext] = fileparts( fullfile( handles.folder_name,
jpg_files(jpg_counter+1).name ) );

        end
name1=name;
ful=ext;
g=strcat(name1,ful);
name11=cellstr(g);
set(handles.text4,'string',name11);

        jpg_counter = jpg_counter + 1;

    elseif ( (num_of_png_images - png_counter) > 0)
        imgInfoPNG = imfinfo( fullfile( handles.folder_name,
png_files(png_counter+1).name ) );
        if ( strcmp( lower(imgInfoPNG.Format), 'png') == 1 )
            % read images
            sprintf('%s \n', png_files(png_counter+1).name)
            % extract features
            image = imread( fullfile( handles.folder_name,
png_files(png_counter+1).name ) );
```

```
            [pathstr, name, ext] = fileparts( fullfile( handles.folder_name,
png_files(png_counter+1).name ) );

         end
name1=name;
ful=ext;
g=strcat(name1,ful);
name11=cellstr(g);
set(handles.text4,'string',name11);

         png_counter = png_counter + 1;

      elseif ( (num_of_bmp_images - bmp_counter) > 0)
         imgInfoBMP = imfinfo( fullfile( handles.folder_name,
bmp_files(bmp_counter+1).name ) );
         if ( strcmp( lower(imgInfoBMP.Format), 'bmp') == 1 )
            % read images
            sprintf('%s \n', bmp_files(bmp_counter+1).name)
            % extract features
            image = imread( fullfile( handles.folder_name,
bmp_files(bmp_counter+1).name ) );
            [pathstr, name, ext] = fileparts( fullfile( handles.folder_name,
bmp_files(bmp_counter+1).name ) );

         end
name1=name;
ful=ext;
g=strcat(name1,ful);
name11=cellstr(g);
set(handles.text4,'string',name11);

         bmp_counter = bmp_counter + 1;

      end
      timing=tic;
      axes(handles.axes1);
      imshow(image);
      pause(0.1);
   end
end
timing2=toc(timing);
Loadingtim=timing2*totalImages;
Loadingtime=num2str(Loadingtim);
ldt=[Loadingtime , ' Secs'];
set(handles.text7,'string',ldt);
close(h);
tot=totalImages;
to=num2str(tot);
msgbox(['Image Loading Complete!!!  ' , to ,'  Images Loaded ', 'Proceed to Normalise
Images'], 'Image Database', 'help');
% Hint: get(hObject,'Value') returns toggle state of Load_Database
```

```
meso=[to , ' Images Loaded'];
set(handles.text18,'string',meso);
set(handles.Image_Normalization,'enable','on');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% --- Executes when selected object is changed in uibuttongroup3.
function uibuttongroup3_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup3
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------global variable size
global variable size
value=get(eventdata.NewValue,'Tag');
switch value
    case 'radiobutton1';
      variable=1
      size='75*75';
    case  'radiobutton2';
       variable=2
       size='100*100';
    case 'radiobutton15';
      variable=3
      size='125*125';
    case  'radiobutton16';
       variable=4
       size='150*150';
end
set(handles.Load_Database,'enable','on');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% --- Executes during object creation, after setting all properties.
% --------------------------------------------------------------------
function exit_Callback(hObject, eventdata, handles)
% hObject    handle to exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global type
h = waitbar(0,['Searching for the Nearest Neighbour Data between'...
```

```
    ' Testing and Training Image using Euclidean Distances Metrics Pls Wait...']);
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
timing=tic;
if type==1
matrixg;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
euclitrain;
elseif type==2
matrixgORL
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
euclitrainORL;
end
close(h)
timing2=toc(timing);
timing2=num2str(timing2);
Euclitime=[timing2 , ' Secs'];
Eucli='Eucidean Training Time';
set(handles.text28,'string',Euclitime)
set(handles.text51,'string',Eucli)
msgbox('Euclidean Distance Metrics Created');
set(handles.pushbutton6,'enable','on');
 % --- Executes on button press in Chebyshev.
function Chebyshev_Callback(hObject, eventdata, handles)
% hObject    handle to Chebyshev (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
h = waitbar(0,['Searching for the Nearest Neighbour Data between'...
    ' Testing and Training Image using Chebychev Distances Metrics Pls Wait...']);
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
timing=tic;
global type
if type==1;
matrixg;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
chebytrain;
elseif type==2;
matrixgORL;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
chebytrainORL;
end
close(h);
timing2=toc(timing);
timing2=num2str(timing2);
```

```
Chebytime=[timing2 , ' Secs'];
Cheby='Chebychev Training Time';
set(handles.text28,'string',Chebytime)
set(handles.text51,'string',Cheby)
msgbox('Chebychev Distance Metrics Created');
set(handles.pushbutton14,'enable','on');

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
h = waitbar(0,['Searching for the Nearest Neighbour Data between'...
    ' Testing and Training Image using Cityblock Distances Metrics Pls Wait...']);
steps = 1000;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
timing=tic;
global type
if type==1;
matrixg;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
citytrain;
elseif type==2;
matrixgORL;
cd('C:\Users\ARO\Documents\MATLAB\Gabor')
citytrainORL;
end
close(h);
timing2=toc(timing);
timing2=num2str(timing2);
Citytime=[timing2 , ' Secs'];
City='Cityblock Training Time';
set(handles.text28,'string',Citytime)
set(handles.text51,'string',City)
msgbox('Cityblock Distance Metrics Created');
set(handles.pushbutton12,'enable','on');
 % --- Executes during object creation, after setting all properties.
function uibuttongroup3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uibuttongroup3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global variable index Recognized_index exact type distance_cover
```

```
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Matching 75 x 75 Black Face
if (type == 1 & variable == 1);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\N_Image\Test\75 x 75');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
         newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)
         newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matching75;
%  fot='Equvalent Matched Image';
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Matching 100 x 100 Black Face
if (type == 1 & variable == 2);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\N_Image\Test\100 x 100');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
```

```
   return
else
        newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)
        newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matching100;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];

  set(handles.text47,'string',exact);
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Matching 125 x 125 Black Face
if (type == 1 & variable == 3);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\N_Image\Test\125 x 125');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
        newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)

     newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
```

```
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matching125;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%% Matching 150 x 150 Black Face
if (type == 1 & variable == 4);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\N_Image\Test\150 x 150');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
         newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)
         newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matching150;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Image');
```

```
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Matching 75 x 75 ORL Face
if (type == 2 & variable == 1);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Test\75 x 75');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
         newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)
         newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matchingORL75;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Matched Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Recog='Recognised Index in Database:';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text44,'string',Recog);
```

```
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
  set(handles.text45,'string',imgfound);
  set(handles.text38,'string',imgfound);
  set(handles.text19,'string',imgfound);
  set(handles.text21,'string',imgfound);
  set(handles.text26,'string','Gabor Extracted Image');
  set(handles.text25,'string','ACO Extracted Image');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%% Matching 100 x 100 ORL Face
if (type == 2 & variable == 2);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Test\100 x 100');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
         newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)

     newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matchingORL100;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Matched Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Recog='Recognised Index in Database:';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text44,'string',Recog);
  set(handles.text42,'string',Tes);
```

```
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
  set(handles.text45,'string',imgfound);
  set(handles.text38,'string',imgfound);
  set(handles.text19,'string',imgfound);
  set(handles.text21,'string',imgfound);
  set(handles.text26,'string','Gabor Extracted Image');
  set(handles.text25,'string','ACO Extracted Image');
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
%% Matching 125 x 125 ORL Face
if (type == 2 & variable == 3);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Test\125 x 125');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else

     newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)

     newread=newread1;

axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matchingORL125;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Matched Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Recog='Recognised Index in Database:';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text44,'string',Recog);
```

```
  set(handles.text42,'string',Tes);
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
  set(handles.text45,'string',imgfound);
  set(handles.text38,'string',imgfound);
  set(handles.text19,'string',imgfound);
  set(handles.text21,'string',imgfound);
  set(handles.text26,'string','Gabor Extracted Image');
  set(handles.text25,'string','ACO Extracted Image');
%----------------------------------------------------------------------
%----------------------------------------------------------------------
%----------------------------------------------------------------------
%% Matching 150 x 150 ORL Face
if (type == 2 & variable == 4);
pathname=cd('C:\Users\ARO\Documents\MATLAB\Gabor\Image
Database\ORL_Image\Test\150 x 150');
 [filename,pathname,mini]= uigetfile('*.jpg','Select Image')
if isequal(filename,0)
    msgbox('Please Select an Image')
   disp('User selected Cancel')
   return
else
         newread1 = imread([pathname filename]);
     [pathstr,nam,ext]=fileparts(filename)

     newread=newread1;
 axes(handles.axes16);
imshow(newread);
g=([nam,ext]);
set(handles.text24, 'string','Test Image');
set(handles.text5, 'string',g);
end
cd('C:\Users\ARO\Documents\MATLAB\Gabor\pca');
nam=str2num(nam);
timing=tic;
index=nam;
matchingORL150;
cd('C:\Users\ARO\Documents\MATLAB\Gabor');
 set(handles.text39, 'string','Equvalent Matched Image');
 timing2=toc(timing);
 set(handles.text43,'string', timing2);
end
Units='M';
Recog='Recognised Index in Database:';
Tes='Testing Time:';
Test='Secs';
imgfound=[num2str(Recognized_index),'.jpg'];
   set(handles.text47,'string',exact);
  set(handles.text44,'string',Recog);
  set(handles.text42,'string',Tes);
```

```
  set(handles.text49,'string',Test);
  set(handles.text48,'string',Units);
  set(handles.text46,'string',distance_cover);
  set(handles.text45,'string',imgfound);
  set(handles.text38,'string',imgfound);
  set(handles.text19,'string',imgfound);
  set(handles.text21,'string',imgfound);
  set(handles.text26,'string','Gabor Extracted Image');
  set(handles.text25,'string','ACO Extracted Image');
%----------------------------------------------------------------------
%----------------------------------------------------------------------
%----------------------------------------------------------------------
% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup1
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global type
value=get(eventdata.NewValue,'Tag');
switch value
    case 'radiobutton13';
      type=1
      img_type='BLACK';
    case  'radiobutton14';
      type=2
      img_type='ORL';
end
 % --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global dista;
val=get(hObject,'Value');
switch val
    case 1
   dista=0
   distan=''
    case 2
   dista=2
   distan='Chebychev'
    case 3
   dista=3
   distan='Cityblock'
   case 4
   dista=4
   distan='Euclidean'
   case 5
   dista=5
   distan='Mahalanobis'
end
```

```
set(handles.pushbutton19,'enable','on');
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1

 % --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
  % --- Executes when uibuttongroup1 is resized.
function uibuttongroup1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to uibuttongroup1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup1
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
global type
value=get(eventdata.NewValue,'Tag');

switch value

    case 'radiobutton13';

      type=1;

      img_type='BLACK FACE'

    case  'radiobutton14';

      type=2;

      img_type='ORL FACE'

end

set(handles.radiobutton1,'enable','on');

set(handles.radiobutton2,'enable','on');

set(handles.radiobutton15,'enable','on');

set(handles.radiobutton16,'enable','on');
```