

**DEVELOPMENT OF A DISCRETE-FIREFLY ALGORITHM BASED FEATURE
SELECTION SCHEME FOR IMPROVED FACE RECOGNITION**

By

Shittu Shehu DANRAKA

**Department of Computer Engineering,
Faculty of Engineering,
Ahmadu Bello University,
Zaria, Nigeria**

DECEMBER, 2017

**DEVELOPMENT OF A DISCRETE-FIREFLY ALGORITHM BASED FEATURE
SELECTION SCHEME FOR IMPROVED FACE RECOGNITION**

By

**Shittu Shehu, DANRAKA, B.Eng. (ABU, 2010)
P13EGCM8005
ssdanraka@gmail.com**

**A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE
STUDIES, AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE (M.Sc)
DEGREE IN COMPUTER ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING,
FACULTY OF ENGINEERING,
AHMADU BELLO UNIVERSITY,
ZARIA, NIGERIA**

December, 2017

DECLARATION

I Shittu Shehu DANRAKA, hereby declare that the work in this dissertation Development of a Discrete Firefly Algorithm (DFA) based Feature Selection Scheme for improved face recognition has been carried out by me in the Department of Electrical and Computer Engineering. The information derived from literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

_____ **Shittu Shehu DANRAKA** _____ **Signature**
Date

CERTIFICATION

This Dissertation entitled DEVELOPMENT OF A DISCRETE FIREFLY ALGORITHM (DFA) BASED FEATURE SELECTION SCHEME FOR IMPROVED FACE RECOGNITION by Shittu Shehu DANRAKA meets the regulations governing the award of degree of Master of Science (MSc) in Computer Engineering of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

_____	_____	

Chairman, Supervisory Committee (Dr. Sani Man-Yahya)	Signature	Date

_____	_____	

Member, Supervisory Committee (Dr. I.J. Umoh)	Signature	Date

_____	_____	

Head of Department (Dr. Y. Jibril)	Signature	Date

_____	_____	

Dean. School of Postgraduate Studies (Professor. S. Z. Abubakar)	Signature	Date

DEDICATION

This research work is dedicated to the Almighty ALLAH.

ACKNOWLEDGEMENT

All praise is due to Almighty ALLAH who knows everything, for His infinite Blessings and Guidance towards the successful completion of this work.

I wish to express my deepest gratitude to my supervisor Dr Sani Man-Yahya, for his tireless efforts, valuable guidance and constant supervision towards the success of this work. The completion of this work could not have been possible without your constant participation and assistance. Thank you very much sir. My thanks also goes to my co-supervisor Dr. I. J. Umoh for his constant encouragement and expertise.

I sincerely wish to express my immense gratitude and appreciation to Prof. M.B. Muazu for his invaluable significant, contributions, guidance throughout the research work which makes it more qualitative. Once again I am very thankful for your continues patience, sir you are never to be forgotten and only ALLAH can reward you, thank you sir. My deep appreciation goes to the Computer & Control Research Group for their suggestions and constructive criticisms during the discussion stages of the work. I'm glad and privileged to be part of the group.

I acknowledge and appreciate the contribution of all the lecturers of Electrical and Computer Engineering, Ahmadu Bello University, namely: Prof. B. G. Bajoga, Prof. U.O Aliyu, Prof. B. Jimoh, Dr. A. M. S. Tekanyi, Dr. S. M. Sani, Dr. S. Garba, Dr. Y. Jibril, Dr.Y.A. Shaban, Engr. M. J. Musa, Engr. A. I. Abdullahi, Engr. E. A. Gbenga, and most especially, those

whose names could not be mentioned. My sincere appreciation goes to Dr. K.A. Abu-Bilal for continuous advice sir your reward is with ALLAH, not forgetting Dr. E.A Adedokun sir thank you for your effort, my appreciation also goes to Malam Abdullahi Tukur for his administrative support towards the completion of the work.

I am extremely thankful to my special friends Abubakar Umar, Salawudeen A. Tijjani, Zaharudeen Haruna, Mukthar Abubakar, Abdullahi Bala Kunya, Yusuf Ibrahim, Abdul aziz Ango, Sanusi Audee, Oyibo Prosper, Atuman G. Joel, Oga Ajayi, Their continuous support and contributions toward the success of this work would never be forgotten, may God reward them and strengthen our friendship. I am very much thankful to all my course mates Zainab, Jagila, Mal. Tukur, Hafiz, Idris and bukky may God reward you all. I am also very grateful to all friends most especially, those whose names could not be mentioned thank you for all your prayers.

Above all, I am everly indebted to my parents Alhaji Shehu Mustapha Danraka & Hajiya Maryam Shehu Danraka, your endless love and support, kind and understanding will always be appreciated and only ALLAH can pay you all. Thank you very much. To my siblings Halima (Yabi), Sani (babati), Suwaidu, Afiya, Aisha, Fatima, Zaliha, Rahma, Jamila, Muhammadu, I owe you all a big thank you. Last but not the least, a big thank you goes to Abduljalil, Aminu, Shafiu, Isiyaku, hayatu (barde), Suwaiba, Mubarakatu, all Danraka's family am highly grateful for your constant prayer and all those that have directly and indirectly contributed to the success of this work. May God bless you real good.

Shittu Shehu DANRAKA
November

2017

ABSTRACT

This research presents the development of a Discrete Firefly Algorithm (DFA) based feature selection scheme for improved face recognition. Discrete Cosine Transform (DCT) and Haar wavelet based Discrete Wavelet Transform (DWT) were used for feature extraction, and Nearest Neighbour Classifier (NNC) was used as classifier. Extracted features are mostly discrete in nature and most of the optimization techniques used in feature selection are continuous so DFA is employed for feature selection. The developed DFA based feature selection scheme was tested on Olivetti Research Labs (ORL) and Yale databases, and was compared with Firefly Algorithm (FA), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) respectively. The simulation was carried out in MATLAB R2013b simulation environment, and the result obtained from ORL database for DFA showed that the recognition accuracy (R.A) was found to be 97.75 % and recognition time (R.T) was 42.27 seconds while for FA, the R.A was found to be 95.53% and R.T was 49.71 seconds. For the Yale database, the DFA had a R.A of 89.30% and a R.T of 40.33 seconds, for FA, the R.A was 85.33% and R.T was 43.65 seconds. On applying DFA on local images the R.A and R.T was 72.02% and 25.89 seconds respectively. The improvements in terms of R.A and R.T of this system when comparing DFA with FA on ORL database were 2.27% and 14.97%, while the improvements on Yale database were 4.45% and 7.61% respectively. Also, when compared with PCA, it gave an improvement of 25.48% in R.A and 23.82% in R.T, while for LDA it gave an improvement of 38.84% in R.A and 27.81% in R.T for ORL database. Also for the Yale database, when compared with PCA, it gave an improvement of 23.11% in R.A and 16.21% in R.T, and for LDA, it gave an improvement of 26.61% in R.A and 20.01% in R.T respectively.

TABLE OF CONTENTS

DECLARATION	I
CERTIFICATION	II
DEDICATION	III
ACKNOWLEDGEMENT	IV
ABSTRACT	VI
TABLE OF CONTENTS	VII
LIST OF FIGURES	XII
LIST OF TABLES	XI
LIST OF APPENDICES	
XERROR! BOOKMARK NOT DEFINED.	
LIST OF ABBREVIATIONS	11

CHAPTER ONE

INTRODUCTION

12

1.1 Background of Research

12

1.2 Motivation

3

1.3 Significant of Research

4

1.4 Statement of Problem	4
1.5 Aim and Objectives	5
1.6 Methodology	5
1.7 Dissertation Organisation	7

CHAPTER TWO

LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Review of Fundamental Concepts	9
2.2.1 Face Recognition	9
<i>2.2.1.1 Face Recognition Design</i>	<i>11</i>
2.2.2 Feature Extraction	13
2.2.2.1 <i>Local binary pattern (LBP)</i>	<i>13</i>
2.2.2.2 <i>Gabor Filter</i>	<i>14</i>
2.2.2.3 <i>Histogram of oriented gradients (HOG)</i>	<i>14</i>
2.2.2.4 <i>Discrete cosine Transform (DCT)</i>	<i>15</i>
2.2.2.5 <i>Discrete Wavelet Transform (DWT)</i>	<i>16</i>
2.2.3 Feature Selection.	19
2.2.3.1 <i>Genetic Algorithm (GA)</i>	<i>21</i>
2.2.3.2 <i>Particle Swarm Optimization (PSO)</i>	<i>21</i>
2.2.3.3 <i>Ant Colony Optimization (ACO)</i>	<i>22</i>

2.2.3.4	<i>Cuckoo Search (CS)</i>	22
2.2.3.5	<i>Firefly algorithm (FA)</i>	22
2.2.3.6	<i>Discrete Firefly algorithm (DFA)</i>	25
2.2.4	Classification or recognition	27
2.2.4.1	<i>Support Vector Machine (SVM).</i>	28
2.2.4.2	<i>Hidden Markov Model (HMM)</i>	28
2.2.4.3	<i>Back propagation neural network (BPNN).</i>	29
2.2.4.4	<i>Self Organizing Map (SOM).</i>	29
2.2.4.5	<i>Nearest Neighbor Classifier (NNC)</i>	29
2.2.4.5.1	<i>Recognition accuracy</i>	30
2.3	Review of Similar Works	32

CHAPTER THREE

MATERIALS AND METHODS	41
3.1 Introduction	41
3.2 Implementation of Firefly Algorithm for Feature Selection for Face Recognition	41
3.2.1 Image acquisition	41
3.2.2 ORL face database	42
3.2.3 Yale face database	42
3.2.4 Local Images Face Database.	43

3.2.5 Importing of the Images into MATLAB ('imread')	44
3.2.6 Converting RGB into Grayscale Images	44
3.3 Extraction using DCT and DWT	45
3.4 Feature Selection with FA.	46
3.4.1 Initialization of the FA Parameters	46
3.4.2 Initialization of Light Intensity.	46
3.4.3 Generating New Solution by Updating the Position of the Firefly.	47
3.5 Feature selection with DFA	47
3.5.1 Generating new solution by updating the position of the firefly	48
3.6 Feature selection with LDA	49
3.7 Feature Selection with PCA	50
3.8 Classification using Nearest Neighbour Classifier (NNC)	51
3.9 Performance Evaluation.	52
3.9.1 Recognition Accuracy	52
3.9.2 Recognition Time	52
3.9.3 Percentage Improvement	53

CHAPTER FOUR

RESULTS AND DISCUSSIONS	54
4.1 Introduction	54
4.2 Result of Gray Scale Conversion of Images on ORL, Yale and Local Images	54

4.3 Result of FA on ORL Database	56
4.4 Result of DFA on ORL Database	56
4.5 Result of FA on Yale Database	57
4.6 Result of DFA on Yale Database	57
4.7 Comparison of Results	58
4.7.1 Comparison of the Recognition Accuracy for DFA, FA, PCA and LDA on ORL Database	58
4.7.2 Comparison of the Recognition Time of DFA, FA, PCA and LDA on ORL Database	59
4.7.3 Comparison of the Recognition Accuracy for DFA, FA, PCA and LDA on Yale Database	59
4.7.4 Comparison of the Recognition Time of DFA, FA, PCA and LDA on Yale Database	60
4.8 Result of DFA on local Images	61
4.9 Performance Evaluation	62
4.9.1 Percentage Improvement on ORL Database	62
4.9.2 Percentage Improvement on Yale Database	62

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION	64
5.1 Summary	64
5.2 Conclusion	64

5.3 Significant Contributions	64
5.4 limitation	65
5.5 Recommendations for Further Work	65
REFERENCES	66

LIST OF FIGURES

Figure 2.1: Steps in Face Recognition System	13
Figure: 2.2: Discrete Wavelet Transform	18
Figure 2.3: Face Recognition System	20
Figure 2.4: Flowchart of Firefly Algorithm	24
Figure 2.5: Flowchart of Discrete Firefly Algorithm	27
Figure 2.6: Steps of Face Recognition Approach based on Firefly Algorithm for Feature Selection	31
Figure 2.7: Steps of the Proposed Approach	31
Figure 2.8: Proposed Flow Chart of the Face Recognition System	32
Figure 3.1: Obtained Images from ORL Face Database	42
Figure 3.2: Obtained Images from Yale Face Database	43
Figure 3.3: Obtained Images from local Images Database	44
Figure 3.4: Snippet of the Conversion to Grey Scale Image	45
Figure 3.5: Snippet of the Discrete Cosine Transform	45
Figure 3.6: Snippet of Discrete Wavelet Transform	46

Figure 3.7: Snippet of FA for Feature Selection	47
Figure 3.8: Snippet of DFA for Feature Selection	48
Figure 3.9: Snippet of the LDA for Face Recognition	50
Figure 3.10: Snippet of the PCA for Face Recognition	51
Figure 3.11: Snippet of the NNC for Classification in Face Recognition	52
Figure 4.1: Grayscale Images Obtained from ORL Face Database	55
Figure 4.2: Grayscale Images Obtained from Yale face Database	55
Figure 4.3: Grayscale Images Obtained from Local Images	55
Figure 4.4: Comparison of Recognition Accuracy for DFA, FA, PCA and LDA on ORL Database	58
Figure 4.5: Comparison of Recognition Time for DFA, FA, PCA and LDA on ORL Database	59
Figure 4.6: Comparison of Recognition Accuracy for DFA, FA, PCA and LDA on Yale Database	60
Figure 4.7: Comparison of Recognition Time for DFA, FA, PCA and LDA on Yale Database	61

LIST OF TABLES

Table 4.1: Average Recognition Accuracy and Recognition Time on ORL using DFA, FA, PCA and LDA Techniques	55
Table 4.2: Average Recognition Accuracy and Recognition Time on YALE database using DFA, FA, PCA and LDA Techniques	57
Table 4.3: Average Recognition Accuracy and Recognition Time on Local Images	62

LIST OF APPENDICES

APPENDIX A1

MATLAB CODE OF FIREFLY ALGORITHM (FA) ON ORL DATABASE 73

APPENDIX A2

MATLAB CODE OF FIREFLY ALGORITHM (FA) ON ORL DATABASE 76

APPENDIX A3

MATLAB CODE OF FIREFLY ALGORITHM (FA) ON YALE DATABASE 79

APPENDIX A4

MATLAB CODE OF DISCRETE FIREFLY ALGORITHM (DFA) ON YALE DATABASE
82

APPENDIX A5

MATLAB CODE OF DISCRETE FIREFLY ALGORITHM (DFA) ON LOCAL IMAGES
85

APPENDIX B1

PERCENTAGE IMPROVEMENT OF RECOGNITION ACCURACY AND RECONITION
TIME OF ORL AND YALE DATABASES 88

LIST OF ABBREVIATIONS

Acronyms	Definition
FA	Firefly Algorithm
DFA	Discrete Firefly Algorithm
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
NNC	Nearest Neighbour Classifier
ORL	Olivetti Research Labouratory
GA	Genetic Algorithm
ACO	Ant colony Optimization
CSA	Cuckoo Search Algorithm
PSO	Particle Swarm Optimization
MATLAB	Matrix Laboratory
M-FILE	MATLAB file
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
FS	Feature Selection
FRS	Face Recognition System
R.A	Recognition Accuracy
R.T	Recognition Time

LBP	Local Binary Pattern
SVM	Support Vector Machine
SOM	Self Organizing Map

CHAPTER ONE

INTRODUCTION

1.1 Background of Research

In the field of pattern recognition, face recognition has attracted interest from researchers due to its numerous applications (public security, law enforcement and commerce, credit card verification, criminal identification, access control, human-computer intelligent interaction, digital libraries and information security) (Bakshi & Singhal, 2014). Face recognition is a process of identifying or verifying a person's identity by matching input face biometrics as against pre-defined faces in a database (Zhou *et al.*, 2014).

In day to day social activities and interactions, the face seems to be an important factor for easy identification (Shivdas., 2014.). Face recognition has advantages over the traditional methods of identification, which involved the use of passwords and personal identification number that provides accuracy and its case sensitiveness (Angle *et al.*, 2005; Kaur & Singh, 2015), and it also offers non-contact process, captured or videoed easily, and provides reliable face matching, and offers a wide range of applications (Bakshi & Singhal, 2014).

The face acts as a key factor of consideration in the public domain, playing a foremost function in conveying uniqueness and emotion (Maini & Aggarwal, 2009). Face recognition works basically in three stages which comprises of detection, feature extraction and classification or recognition (Maini & Aggarwal, 2009). And the choice of approach to each

of these stages is vital to attaining better recognition accuracy. The face recognition problem is complicated by age, skin, colour, gender, differing image qualities, facial expressions, background, and illumination condition (Bakshi & Singhal, 2014).

In face detection, the goal is to find an object in an image as a face candidate that its shape resembles the shape of a face (Saleh, 2009). In other words, it can be regarded as a process of automatically detecting a face from a complex background to which the face recognition algorithm can be applied. Researchers use pre-processing at this stage (Agarwal & Bhanot, 2015).

In feature extraction, high level information about individual patterns (like eyes, nose, lips, etc..) to facilitate recognition is extracted (Hemalatha & Govindan, 2015). Selection of feature extraction method is probably the single most important factor in achieving high recognition performance (Saleh, 2009). Approaches used for face extraction include discrete cosine transform (DCT) (Hemalatha Gayatri & Govindan, 2015; Jadon *et al.*, 2015), gabor filter (Keche *et al.*, 2015; Ruan *et al.*, 2010), principal component analysis (PCA) (Bakshi & Singhal, 2014; Satone & Kharate, 2014; Sawalha & Doush, 2012), local binary pattern (LBP) (Babatunde *et al.*, 2015), and discrete wavelet transform (DWT) (Kallianpur *et al.*, 2016; Manikantan *et al.*, 2012).

In the classification or recognition stage, face samples are compared or matched with existing known faces in the database (Richa & Josan, 2013). Some methods reported at this stage are support vector machine (SVM) (Satone & Kharate, 2014; Xu & Lee, 2014), Hidden Markov Model (HMM) (Jameel, 2015), Nearest Neighbour Classifier (NNC) (Agarwal & Bhanot, 2015) Back Propagation Neural Network (BPNN) (Shivdas., 2014.) and self-organizing map (SOM) (Bakshi & Singhal, 2014).

The feature selection process involves determination of a feature subset which can best represent a given feature set (Manikantan *et al.*, 2012). Feature selection problem, is

challenging due to its combinatorial nature. Feature selection phase of the face recognition process attempts to obtain the most discriminative features between two or more individual's faces to produce the best accuracy in databases capturing variations in illumination, pose, expression or occlusion (Agarwal & Bhanot, 2015). Many among the large features are not vital features and as such it causes over fitting of the face data and consequently reduce performance of the system (Agarwal & Bhanot, 2015). Some of the optimization technique used in feature selection are particle swarm optimization (PSO) (Hemalatha & Govindan, 2015; Ramadan & Abdel-Kader, 2009; Unler & Murat, 2010; Xue *et al.*, 2014), firefly algorithm (Agarwal & Bhanot, 2015), genetic algorithm (GA) (Boubenna & Lee, 2016; Satone & Kharate, 2014), ant colony optimization (ACO) (Babatunde *et al.*, 2015; Kanan & Faez, 2008) and artificial bee colony (ABC) (Kallianpur *et al.*, 2016; Khan & Gupta, 2016). However, these algorithms are continuous and requires continuous problem thus the face recognition which is discrete requires to be converted to continuous or the algorithm is converted to discrete which is time consuming hence this research hopes to employ a discrete firefly algorithm (DFA) for feature selection to address this challenge in order to improve recognition accuracy.

1.2 Motivation

Face recognition is a process of identifying or verifying a person's identity by matching input face biometrics as against pre-defined faces in a database. The steps in face recognition are feature extraction, feature selection and classification. Feature selection process involve determination of a feature subset which can best represent a given feature set. Previous, researchers have proposed many metaheuristic search algorithm and hybrid for feature selection, which are continuous as such require more Recognition time and reduces recognition accuracy, thus feature selection which consist of extracted feature which are

discrete require discrete algorithms which lead to an efficient Recognition time and recognition accuracy. Thus, this research work offers the use of a metaheuristic algorithm known as discrete firefly algorithm (DFA) for feature selection to address this problem.

1.3 Significance of Research

The significance of this research is the development of a discrete firefly algorithm based feature selection scheme for improved face recognition, which has added capabilities to the standard feature selectors in face recognition through the minimization of the number of features that improve performance prediction of recognition rate. This has not been considered by the previous researchers.

1.4 Statement of Problem

Face recognition is a process of identifying or verifying a person's identity by matching input face biometrics as against pre-defined faces in database. It finds application in areas such as public security, law enforcement, credit card verification, criminal identification, access control, human-computer intelligent interaction and information security. The steps in face recognition are detection, feature extraction, feature selection and classification. Feature selection problem, is challenging due to its combinatorial nature. Feature selection phase of the face recognition process attempts to obtain the most discriminative features between two or more individual's faces to produce the best accuracy in databases capturing variations in illumination, pose, expression or occlusion. However, the problem of over fitting of the face data results in reduced performance of the system. Some of the optimization technique used in feature selection are PSO, FA, GA, ACO and CSA. However, these algorithms are continuous and requires continuous problem thus the face recognition which is discrete requires to be converted to continuous or the algorithm is converted to discrete which is time

consuming hence this research hopes to employ a Discrete Firefly Algorithm to address this challenge.

1.5 Aim and Objectives

The aim of this research work is to develop an improved face recognition based on DFA for feature selection. To achieve this aim, the objectives are as follows:

1. To implement and evaluate the performance of the face recognition system (FRS) using firefly algorithm (FA) and discrete firefly algorithm (DFA) respectively for feature selection in terms of Recognition time and recognition accuracy using the Olivetti Research Labs (ORL) and Yale facial images databases.
2. Comparison of the performance of DFA based system with, linear discriminant analysis (LDA) and principal component analysis (PCA) based system on (1).
3. To apply the DFA based system on faces of 10 people with 5 different facial expressions collected at Zaria city of Kaduna state.

1.6 Methodology

The methodology to be adopted for this research are as follows:

1. Obtaining images that are used in the research as follows:
 - a. Benchmark face database of Olivetti Research Labs (ORL) and Yale from <http://www.uk.research.att.com/facedatabase.html> and <http://cvc.yale.edu/projects/yalefaces/yalefaces.html> respectively.
 - b. Local images in Zaria city of Kaduna state to create a database for application using the developed method.
2. Implementation of the face recognition system based on Firefly Algorithm (FA) for feature selection in MATLAB 2013b as follows:

- a. Importing of the images into MATLAB 2013b simulation environment and converting them into grayscale images.
 - b. Extraction of all the images using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT).
 - c. Carrying out feature selection with FA as follows:
 - i. Initializing the FA parameters population X and light absorption coefficient (γ) .
 - ii. Initializing the light intensity (I_1) .
 - iii. Generating new solution by updating the position of the firefly $(I_j > I_1)$.
 - iv. Evaluating new solution and updating the light intensity.
 - d. Classification using Nearest Neighbour Classifier (NNC).
3. Implementing the face recognition system using Discrete Firefly Algorithm (DFA) for feature selection and comparing the performance in (2) on the benchmark face database of Olivetti Research Labs (ORL) and Yale.
- a. Pre-processing and importing of the images into MATLAB 2013b simulation environment and converting them into grayscale images.
 - b. Extraction of all the images using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT).
 - c. Carrying out feature selection with DFA as follows:
 - i. Initializing the DFA parameters population X and light absorption coefficient (γ) .
 - ii. Initializing the light intensity (I_1) .
 - iii. Generating new solution by updating the position of the firefly $(I_j > I_1)$ using hamming distance, lengths of the firefly and movement function.

- iv. Evaluating new solution and updating the light intensity.
- d. Classification using Nearest Neighbour Classifier (NNC).
- 4. Carrying out feature selection with LDA as follows:
 - i. Create a D-dimensional samples $\{X^{(1)}, X^{(2)}, \dots, X^{(N)}\}$
 - ii. Find a measure of separation of the two classes, within class scatter and between class scatter.
 - iii. Find the optimum vector W.
- 5. Carrying out feature selection with PCA as follows:
 - i. Create A matrix from training images and Compute B matrix from A.
 - ii. Compute eigenvectors of C from eigenvectors of B.
 - iii. Select few most significant eigenvectors of C for face recognition.
 - iv. Compute coefficient vectors corresponding to each training images.
 - v. For each person, coefficient will form a cluster, compute mean of mean cluster.
- 6. Comparison of the DFA, FA, LDA and PCA in terms of Recognition Time and Recognition Accuracy
- 7. Application of the DFA based approach on the faces of 10 people with 5 different facial expressions each, collected at Zaria city of Kaduna state as implemented in (3).

1.7 Dissertation Organization

The general introduction has been presented in Chapter One. The rest of the chapters are structured as follows: First, detailed review of related literature and relevant fundamental concepts about face recognition, discrete cosine transform (DCT), discrete wavelet transform (DWT), firefly algorithm (FA), discrete firefly algorithm (DFA) for feature selection and Nearest Neighbour Classifier (NNC) are carried out in chapter two. Second,

an in depth approach and relevant mathematical models describing the development of an improved face recognition base on discrete fire fly for feature selection. Third the analysis, performance and discussion of the result are shown in chapter four. Finally, conclusion and recommendations of further work makes up the chapter five. Finally list of cited references and MATLAB codes in the appendices are provided at the end of this dissertation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The literature review comprises of the overview of fundamental concept and the review of similar works. In the review of fundamental concepts, most of the pertinent works and the fundamental theories that will be used for the success of this research will be reviewed, after which similar works are reviewed.

2.2 Review of Fundamental Concepts

In this section, concepts fundamental to the research such as face recognition, discrete cosine transform (DCT), discrete wavelet transform (DWT), firefly algorithm, discrete firefly algorithm (DFA) and Nearest Neighbour Classifier (NNC) are reviewed.

2.2.1 Face Recognition

Face recognition is a process of identifying or verifying a person's identity by matching input face biometrics as against pre-defined faces in a database (Zhou *et al.*, 2014). In face recognition, verification (authentication) refers to the determination of a person's claimed identity while identification crosschecks a complete pre-defined database for verification matching without user collaboration (Zhou *et al.*, 2014).

Over the years, face recognition has attracted the scientific and industrial society due to its numerous importance and applications in access control, entertainment, information security, law enforcement, smart cards and video surveillance (Li, 2014). To conduct a face recognition analysis, face recognition can be roughly divided into image acquisition, face location registration, pre-processing, feature extraction and recognition (Zhang *et al.*, 2012).

Holistic and feature based are the two categories of face recognition study in research (Fattah *et al.*, 2011). The holistic or global approaches to face recognition assumes that all faces are constrained to a particular positions, orientations and scales (Fattah *et al.*, 2011), while the feature based approaches rely on detection and characterization of individual facial features and their geometric relationships (Fattah *et al.*, 2011).

The essential large amount of data carried by images requires efficient techniques to reduce the total number of bits necessary to represent the images without information loss or, more practically, with an acceptable and controlled level of information loss (image distortion). In order to reduce the amount of data on images, i.e., the bits representation of the image, without information loss or with an acceptable level of control of the information loss (image distortion), an efficient technique is required (Cappellini & Del Re, 1985).

Data reduction is called data compression, and it finds application in communications, remote sensing, bioengineering, robotics, artificial intelligence and so on (Cappellini & Del Re, 1985).

Data compression can be carried out before storing the image data to reduce memory size or before transmitting the image data to reduce transmission rate and conserve bandwidth (Cappellini & Del Re, 1985). Direct operation on sample images seems difficult due to certain features (non-linearity, high dimensionality, large quantity, etc.) (Shenglin & Shan-an, 2011). To reduce dimensionality while maintaining the original features, so many researches used various algorithms such as principal component analysis (PCA), multi-dimensional scaling (MDS), artificial neural network (ANN), local linear embedding (LLE) (Shenglin & Shan-an, 2011). Both PCA and MDS are linear methods which finds it difficult to solved nonlinear structures (Shenglin & Shan-an, 2011). LLE attempts in discovering non-linear structure in a data, as most images are non-linear in nature (Shenglin & Shan-an, 2011).

Feature extraction or statement of human face is an important part in face recognition which has an effect in the recognition accuracy (Zhang *et al.*, 2012).

Recognition becomes a difficult task due to variations in illumination, pose (Gross *et al.*, 2011), position scale, environment, accessories and age difference among others (Fattah *et al.*, 2011). The speed of recognition posed a serious challenge, as the reliability and time response of object detection and recognition in a database has influence in the performance and utility of the recognition system (Dai *et al.*, 2011). Various methods used in face recognition achieved quite remarkable performance in a controlled environment over large database, however, in an uncontrolled environment, the issues of expressions, illumination, partial occlusions, pose, etc., still persists (Li, 2014).

Facial expressions refer to facial changes in response to person's internal emotional state, intentions or social communications. And the use of computer systems in an attempt to automatically analyse and recognize facial motions and facial feature changes from visual information is referred to as facial expression analysis (Tian *et al.*, 2005). Automatic facial expressions finds application in areas such as emotions and paralinguistic communication, clinical psychology, psychiatry, neurology, pain assessment, lie detection, intelligent environments, and multimodal human-computer interface (Tian *et al.*, 2005).

2.2.1.1 Face recognition design

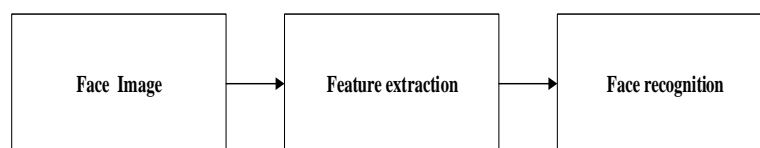
The face recognition system has been designed to perform recognition on images, it involves three major tasks (Saleh, 2009):

1. **Face detection:** It is the first step of face recognition where it automatically detects a face from a complex background to which the face recognition algorithm can be applied. It is also a way of detecting the pixels in the image which represent the face (Saleh, 2009). Being it a stage before feature extraction so the feature extraction

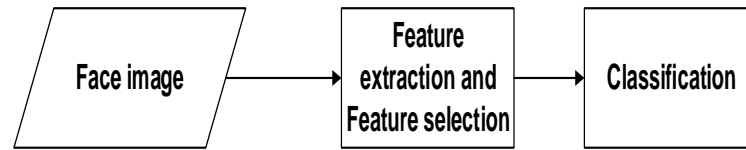
depends on detection, so before a face is recognized, one must reliably find a face and its land marks. Majority of the face detection systems extract a fraction of the whole face and eliminate most of the background and other areas of individual head such as hair that are not necessary for face recognition (Saleh, 2009).

2. **Feature extraction:** It is the most essential step in face recognition, it extract the main important information about individual patterns for recognition. It helps in achieving high recognition performance. Certain features from the detected and extracted images are derived by the feature selection in pattern recognition to reduce the amount of data used for classification and provide discriminating power. For measurement cost and classification accuracy, the number of features should be kept as small as possible hereby making the system faster and uses less memory. On the other hand using a wide feature set may cause “curse of dimensionality”. Feature extraction methods aims at reducing the feature dimensions used in the classification stage (Saleh, 2009). Some of researchers use one level dimensional reduction as in figure 2.1a and some used two level dimensional reduction as figure 2.1b. The two level dimensional technique indicate an efficient performance for dimensionality reduction.

Figure 2.1a and 2.1b presents a block diagram of the one and two level face recognition technique (Saleh, 2009).



a) One-level Face Recognition System



b) Two-level Face Recognition System

Figure 2.1: Steps in Face Recognition System (Saleh, 2009)

3. **Classification or recognition stage:** face tested samples are compared or matched with existing known face samples in the database (Saleh, 2009). The accuracy of classifier is determine by feature selection whereby the feature selection reduces the number of features used in the classification while maintaining acceptable classification accuracy (Babatunde *et al.*, 2015).

2.2.2 Feature Extraction

It is the most essential step in face recognition, it extract the main important information about individual patterns for recognition. As discussed in section 2.2.2.1

Some of approaches used for feature extraction are:

1. Local binary pattern (LBP).
2. Gabor Filter
3. Histogram of oriented gradients (HOG)
4. Discrete cosine Transform (DCT)
5. Discrete Wavelet Transform (DWT)

2.2.2.1 Local Binary Pattern (LBP)

The LBP is texture descriptor which gives a facial representation that is independent of expression and posed artifacts. It is non-parametric texture classification method which summarize the local structures of an image efficiently (Babatunde *et al.*, 2015).the most important properties of LBP feature are tolerance against the monotonic illumination changes and computational simplicity (Babatunde *et al.*, 2015).The LBP are operators are easy to

compute, hence they are suitable for real time applications. The face image can be seen as a composition of micro-pattern which are describe by LBP (Ojala *et al.*, 2000). To obtain the local information of face, face images were equally divided into small sub-regions to extract LBP histogram. The LBP features extracted from each sub-region are concatenated into a single, spatially enhanced feature histogram (Huang *et al.*, 2011). The extracted feature histogram represents the local texture and global shape of face images. These features are the statistics of gray differential features representing the tonal variation of the gray value of pixels.

2.2.2.2 Gabor Filter

The gabor filter is a linear filter used for texture analysis, which means that it basically analyses whether there are any specific frequency content in specific frequency content in image in specific directions in a localized region around the point or region of analysis. Frequency and orientation representations of gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave (Kaur, 2012). A Gabor filter-based feature extraction is proposed. Some feature vectors which provide optimal characterization of the visual content of facial images. For this reason 2D Gabor filtering is chosen which is a widely used image processing tool, for feature extraction (Yang & Zhang, 2010).A Gabor filter (Gabor Wavelet) represents a band-pass linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Thus, a bidimensional Gabor filter constitutes a complex sinusoidal plane of particular frequency and orientation modulated by a Gaussian envelope (Kaur, 2012).

2.2.2.3 Histogram of oriented gradients (HOG)

Histogram of oriented gradients (HOG) is a shape descriptor that counts occurrences of gradient orientations in localized portions of an image. Local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge

directions, even without precise knowledge of the corresponding gradient or edge positions (Déniz *et al.*, 2011). This statement leads to the definition of the HOG technique that has been used in its mature form in Scale Invariant Features Transformation. HOG descriptor is based on the accumulation of gradient directions over the pixel of a small spatial region referred as “cell” and in the subsequent construction of a 1D histogram whose concatenation supplies the features vector to be considered for further purposes (Boubenna & Lee, 2016) Pyramid histogram of oriented gradient (PHOG) is a type of histogram of oriented gradient is used to extract feature. PHOG uses a similar process as in histogram of oriented gradient (HOG). However PHOG used different structure called pyramid. This structure consists of extracting features in a hierarchy way (Boubenna & Lee, 2016). Thus after image normalization, the input image is divided into smaller cells. In which the gradient histogram is calculated for each pixel. Finally the feature descriptor is built by the concatenation of these histograms (Déniz *et al.*, 2011).

2.2.2.4 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) was first introduced by Ahmed *et al.*, (1974). It is a technique that converts a spatial domain waveform into its constituent components represented by a set of coefficients (El Aroussi *et al.*, 2008). In other words, it expresses a sequence of data points in terms of a sum of cosine functions oscillating at different frequencies (Richa & Josan, 2013). Discrete cosine transform is important to numerous applications in science and engineering, from Lossy compression of audio and images (where small high frequency components can be discarded), to spectral methods for numerical solution of partial differential equations (Richa & Josan, 2013).

The DCT has good performance in compaction efficiency, computationally efficient due to its relation to discrete Fourier transform, its energy is concentrated on low frequencies for

natural images due to the fact that natural images mostly possess low frequency feature (El Aroussi *et al.*, 2008). The Discrete Cosine Transform (DCT) has been used in feature extraction for face images to reduce dimensionality or data compression (Richa & Josan, 2013). The DCT is used in both holistic and feature appearance based approach (El Aroussi *et al.*, 2008).

The common variant of DCT is the type-II DCT (DCT) and its inverse, the type-III DCT (IDCT). The DCT feature extractions consists of two stages, where the DCT is applied on the entire image to obtained the coefficient and then the usage of some of these coefficient for construction of feature vectors (Dabbaghchian *et al.*, 2010). The DCT coefficients for an $M \times N$ image corresponds to a 2D matrix are calculated as follow (Dabbaghchian *et al.*, 2010):

$$F(u, v) = \frac{1}{\sqrt{MN}} \alpha(u) \alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \times \cos\left(\frac{(2x+1)u\pi}{2M}\right) \times \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (2.1)$$

$$u = 0, 1, \dots, M \quad v = 0, 1, \dots, N$$

where

$$\alpha(\omega) = \begin{cases} \frac{1}{\sqrt{2}} & \omega = 0 \\ 1 & \text{otherwise} \end{cases} \quad (2.2)$$

$F(x, y)$ is the image intensity function and $F(U, V)$ is a 2D matrix of DCT coefficients. Large values for discriminant coefficients with an image size of $M \times N$ matrix can be estimated as follows (Dabbaghchian *et al.*, 2010):

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ a_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix}_{M \times N} \quad (2.3)$$

2.2.2.5 Discrete Wavelet Transforms

Wavelet Transform is a popular tool in computer vision and image processing, for its ability to capture localized time-frequency information of image extraction (Xu & Lee, 2014). The term wavelet as used in discrete wavelet transform comes from the fact that they integrate to zero, wave up and down across the axis. This property ensures that data is not over represented. Wavelets are signals which are local in time and scale and generally have an irregular shape (Narendra *et al.*, 2014).

The DWT converts the image starting from the spatial domain to frequency domain. Where the image is divided by vertical and horizontal lines which could be into four parts to represents the first-order of DWT (Kaur & Singh, 2015). Wavelets have been used to decompose the face image using multi-resolution analysis. DWT are high in their information packing ability. The face images are signals in space and time, and they can be approximated using orthogonal basis functions. Wavelets are used for multi-resolution analysis of face images and provide a means for localization of signals in both frequency and time (Narendra *et al.*, 2014).

The filters in DWT decomposition are design in such a way that successive layers of the pyramid include only details not considered by the proceeding level (Kaur & Singh, 2015). The decomposition of the data into different frequency ranges allows to isolate the frequency components introduced by intrinsic deformations due to expression variations into certain sub-bands, and considered only sub-bands with most relevant information to better represent the data (Xu & Lee, 2014).

Discrete wavelet transform is often used in signal processing to represent a signal. Two dimensional DWT (2D-DWT) of an image is defined by the expression as follows (Xu & Lee, 2014):

$$DWT(i, j, k, h) = \frac{1}{\sqrt{z}^i} \sum_{x=0}^{m-1} f(x) \phi\left(\frac{x}{2} - k\right) + \frac{1}{\sqrt{z}^j} \sum_{y=0}^{n-1} f(y) \varphi\left(\frac{y}{z} - h\right) \quad (2.4)$$

where i and j are the power of binary scaling, k and h are constant of the filters.

Similar to one dimensional wavelet transform of signal, in image processing, the approximation of images at various resolutions with orthogonal projections can also be computed by multi-resolution which characterized by the two-channel filter bank that governs the loss of information across resolutions. The one-dimensional wavelet decomposition is first applied along the rows of the images, and then their results are further decomposed along the columns (Keche *et al.*, 2015). This results in four decomposed sub images. These sub images are localized in frequency and orientation by Low-Low (LL), Low-High (LH), High-Low (HL) and High-High (HH) as shown in Figure 2.2 corresponding to approximate, horizontal, vertical and diagonal features respectively (Keche *et al.*, 2015).

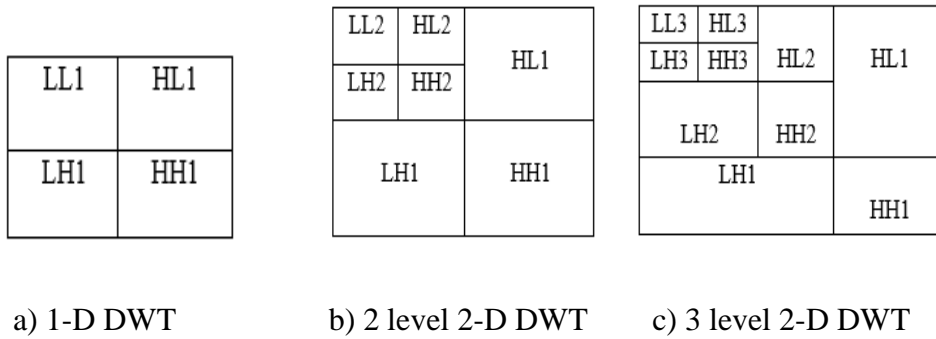


Figure 2.2: Discrete Wavelet Transform: (Xu & Lee, 2014).

Each of these sub-bands can be thought of as a smaller version of the image representing different image properties. The band LL is a closer approximation to the original image. The bands LH and HL record the changes of the image along horizontal and vertical directions, respectively. The HH band shows the high frequency component of the image. Second level decomposition can then be conducted on the LL sub-band (Keche & Dhore, 2015; Xu & Lee, 2014).

Haar wavelet is discontinuous, and resembles a step function. It represents the same wavelet as Daubechies db1. Haar used these functions to give an example of an orthonormal system for the space of square-integrable function on the unit interval $[0, 1]$ (Gupta & Choubey, 2015).

For an input represented by a list of numbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale, finally resulting in differences and one final sum (Gupta & Choubey, 2015). The Haar Wavelet Transformation is a simple form of compression which involves averaging and differencing terms, storing detail coefficients, eliminating data, and reconstructing the matrix such that the resulting matrix is similar to the initial matrix (Gupta & Choubey, 2015).

A Haar wavelet is the simplest type of wavelet. In discrete form, Haar wavelets are related to a mathematical operation called the Haar transform. The Haar transform serves as a prototype for all other wavelet transforms. Like all wavelet transforms, the Haar transform decomposes a discrete signal into two sub-signals of half its length. One sub-signal is a running average or trend; the other sub signal is a running difference or fluctuation (Gupta & Choubey, 2015).

2.2.3 Feature Selection.

Feature selection is a way of finding a subset of features from the original set of features according to the feature selection criterion. Finding a subset of features with predictive performance compared to the full set of features is the main goal of feature selection (Abd-Alsabour, 2014). Feature selection does not generate unwanted artifacts i.e. it is carried out in the original feature space. It is also achieved without losing the original concept of data by removing the redundant and/or irrelevant features. Feature selection is one aspect of dimension reduction which is the main idea for simplifying the data (Abd-Alsabour, 2014).

Taking data in a high dimensional space and mapping it into a new space whose dimensionality is much smaller is called dimensional reduction. The main purpose of dimensionality reduction is to present features in such a way of preserving the useful information or feature most and eliminating the extra data or components that are redundant to classification process. Feature extraction and feature selection are due to perform efficient dimensionality reduction. Feature extraction is a process that involves transformation of data (Babatunde *et al.*, 2015). Feature selection problem, is challenging due to its combinatorial nature. Feature selection phase of the face recognition process attempts to obtain the most discriminative features between two or more individual's faces to produce the best accuracy in databases capturing variations in illumination, pose, expression or occlusion (Agarwal & Bhanot, 2015). The reason of reducing the dimensionality of data is that high dimensional data impose computational efficiency challenge. Moreover, high dimensional data may lead to poor generalization abilities of the learning algorithm (Abd-Alsabour, 2014). Some of the reasons for performing feature selection are improving performance prediction, reducing computational requirements, reducing data storage requirements, reducing the cost of future measurements and improving data or model understanding (Babatunde *et al.*, 2015). The facerecognition system with feature selection is shown in Figure 2.3.

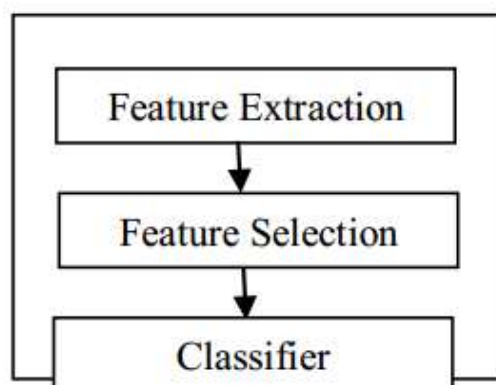


Figure: 2.3: Face Recognition System (Jakhar *et al.*, 2011)

Some metaheuristic-based methods for feature selection are:

1. Genetic Algorithm (GA)
2. Particle Swarm Optimization (PSO)
3. Ant Colony Optimization (ACO)
4. Cuckoo Search (CS)
5. Firefly algorithm (FA)
6. Discrete Firefly algorithm (DFA)

2.2.3.1 Particle Swarm Optimization Algorithm

Eberhart and Kennedy (1995) introduced and implemented PSO in 1995 which is also a nature based algorithm where a group of birds or particles indicates swarm behavior. The following individuals are said to be particles and the group of population is said to be a swarm. This type of swarm behavior of a flock of birds in nature can be viewed at the point of searching food or finding a new location to stay. During this process every particle of swarm bequests by fragmenting the information and move together globally with each step knowing each other's position. As compared to other algorithms it is known to be more useful as it has small population area; and also it converges fast as well as it has a very few parameters (Abd-Alsabour, 2014).

2.2.3.2 Genetic Algorithm

Holland (1975) introduced genetic algorithm (GA) as an evolutionary method that takes advantage from operators such as natural selection, crossover, and mutation (Eberhart & Kennedy, 1995). Selection is the operation of ranking chromosomes based on their fitness (here, fitness refers to chromosomes distances from goal) and sub selecting high ranked chromosomes (those that represent shorter or smoother paths) for generating new population. Crossover is the operation used for generating new population from the sub-selected chromosomes by selection operator. Typically, the crossover operator divides the

sub-selected chromosomes into two parts and exchanges their parts with each other in order to generate the new population. Mutation is the operation in which one or a group of chromosomes are chosen to be fully or partially randomized. It is typical to use mutation operator whenever the population is converged toward local optimum, trapped between obstacles, or the performance (Abd-Alsabour, 2014).

2.2.3.3 *Ant Colony Optimization*

Proposed by Marco Dorigo in 1992 as ant system, ACO is a population-based approach for solving combinatorial optimization problems and it is inspired by the foraging behavior of ants and their inherent ability to find the shortest path from a food source to their nest (Dorigo & Blum, 2005). Naturally, ants are blind and they tend to wander randomly, upon finding food, they return to their nest laying a chemical pheromone trail that help them find the shortest path between their nest and the food source thereby increasing the probability of that path being followed by other ants (Tian *et al.*, 2008).

2.2.3.4 *Cuckoo Search*

Yang and Deb (2009) Developed Cuckoo Search (CS) by idealizing the features of brood parasitism of some cuckoo species as three rules. The first rule is the process in which each cuckoo lays an egg and dumps it in a randomly chosen nest. For the search rule, the best nest with high quality eggs will be carried over the next generations. The number of host nests is fixed. The eggs of a cuckoo may be discovered by the host bird with a probability between 0 and 1 for the last rule (Yang & Deb, 2009).

2.2.3.5 *Firefly Algorithm*

The firefly algorithm (FA) is a metaheuristics algorithm inspired by the flashing behavior of the natural fireflies. The fireflies move in the search space to obtain the best position in

the space so as to acquire the maximum brightness (Yang, 2009). A firefly moves to a brighter firefly due to the attractiveness of the later in the eyes of the former firefly. The attractiveness of a firefly depends not only on its own light, but also depends on its distance from the firefly which is looking at it (Yang, 2009). It is important to highlight the following three idealized rules for proper understanding of fire fly algorithm (Yang, 2009):

1. All the fireflies of the swarm are unisex, and one firefly will be attracted to other ones regardless of their sex.
2. Attractiveness is proportional to the brightness, which means that, for any two fireflies, the brighter one will attract the less bright one. The attractiveness decreases as the distance between the fireflies increases. Furthermore, if one firefly is the brightest one of the swarm, it moves randomly.
3. The brightness of a firefly is directly determined by the objective function of the problem under consideration. In this manner, for a maximization problem, the brightness can be proportional to the objective function value (Yang, 2009). On the other hand, for a minimization problem, it can be the reciprocal of the objective function value (Yang, 2009).

The attractiveness of a firefly at a distance 'r' is given by (Yang, 2009):

$$\beta = \beta_0 e^{-\gamma r^2} \tag{2.5}$$

where β_0 is the attractiveness at distance $r = 0$, and γ (Gamma) is the light absorption coefficient.

The distance between any two fireflies i and j at x_i and x_j is the Cartesian distance (Yang, 2009).

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$$

(2.6)

where $X_{i,k}$ is the k^{th} component of spatial coordinate X_i of the i^{th} firefly and $X_{j,k}$ is the k^{th} component of spatial coordinate X_j of the j^{th} firefly.

The brightness of a firefly is defined by its fitness value. The fireflies keep moving in the search space and keep acquiring new positions in the search space. If X_i is the new position in the ' d ' dimensional space acquired by a firefly F_i , then (Yang, 2009):

$$X_i = X_i + \beta_o \exp^{-\gamma r^2} \|X_j - X_i\| + \alpha \varepsilon_i \quad (2.7)$$

Where, first term in equation (2.3) is the position of the firefly reached so far. Second term is the total distance moved by the less bright firefly F_i towards brighter firefly F_j . Third term refers to random movement of a firefly. The parameters α , known as the randomization parameter, and ε_i , a vector of random values drawn from a uniform distribution or Gaussian distribution, are selected randomly in the interval [0 1] (Agarwal & Bhanot, 2015).

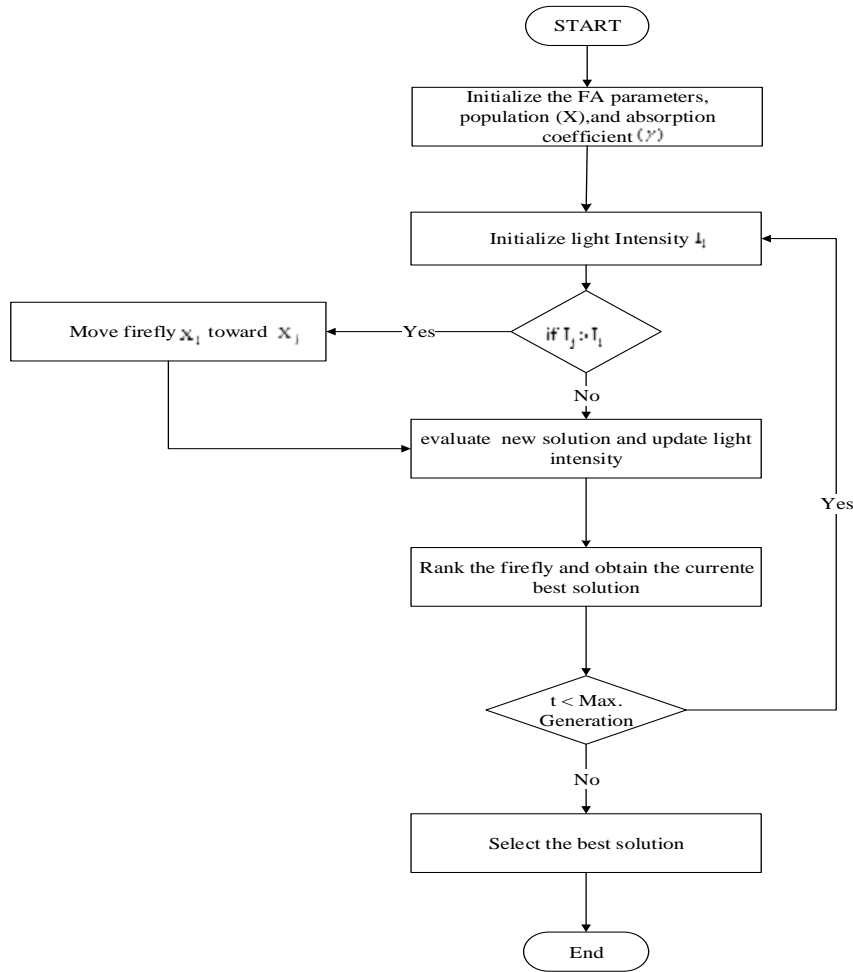


Figure 2.4: Flowchart of Firefly Algorithm. (Osaba *et al.*, 2016)

However, these algorithms are continuous and requires continuous problem thus the face recognition which is discrete requires to be converted to continuous or the algorithm is converted to discrete which is time consuming hence this research hopes to employ a discrete firefly algorithm (DFA) to address this challenge.

2.2.3.6 Discrete Fire Fly Algorithm

In the classical version of the FA which was developed primarily for solving continuous optimization problems, requires conversion to solve a discrete problem. For this reason, the classical FA cannot be applied directly to solve the proposed feature selection problem.

Therefore, discretization of the continuous FA are needed in order to prepare it for addressing such problem (Osaba *et al.*, 2016).

In the proposed DFA, each firefly in the swarm represents a possible and feasible solution for feature selection. All the fireflies are initialized randomly. In addition, the concept of light absorption is also represented in this version of the FA. In this case, $\gamma = 0.95$, and this parameter is used in the way as can be seen in Equation (2.3) besides that, the distance between two different fireflies is represented by the Hamming Distance (Osaba *et al.*, 2016).

Finally, the movement of a firefly i attracted to another brighter firefly j is determined by (Osaba *et al.*, 2016):

$$n = \text{Random}(2, r_{ij} \cdot \gamma^g) \quad (2.8)$$

$$x_i = \text{InsertionFunction}(x_i, n) \quad (2.9)$$

where r_{ij} is the Hamming Distance between firefly i and firefly j , and g is the iteration number. In this case, the length of the movement of a firefly will be a random number

between 2 and $r_{ij} \cdot \gamma^g$, and x_i is the movement function.

As for the movement function, the Insertion function has been used. This function selects and extracts feature subset randomly n from feature set. This function takes into account the capacity constraint, in order not to create infeasible solutions (Osaba *et al.*, 2016).

In discrete firefly algorithm variant of the work of Osaba *et al.*, (2016), the fireflies do not have directions to move. Instead, fireflies move using evolution strategies. In this way, each

firefly moves using n times the Insertion Function, generating n potential successors. After these n movements, the best one is performed, generating the new firefly (Osaba *et al.*, 2016). Finally, the flow chart of the discrete firefly algorithm (DFA) variant of Osaba *et al.*, (2016) is depicted in Figure 2.5.

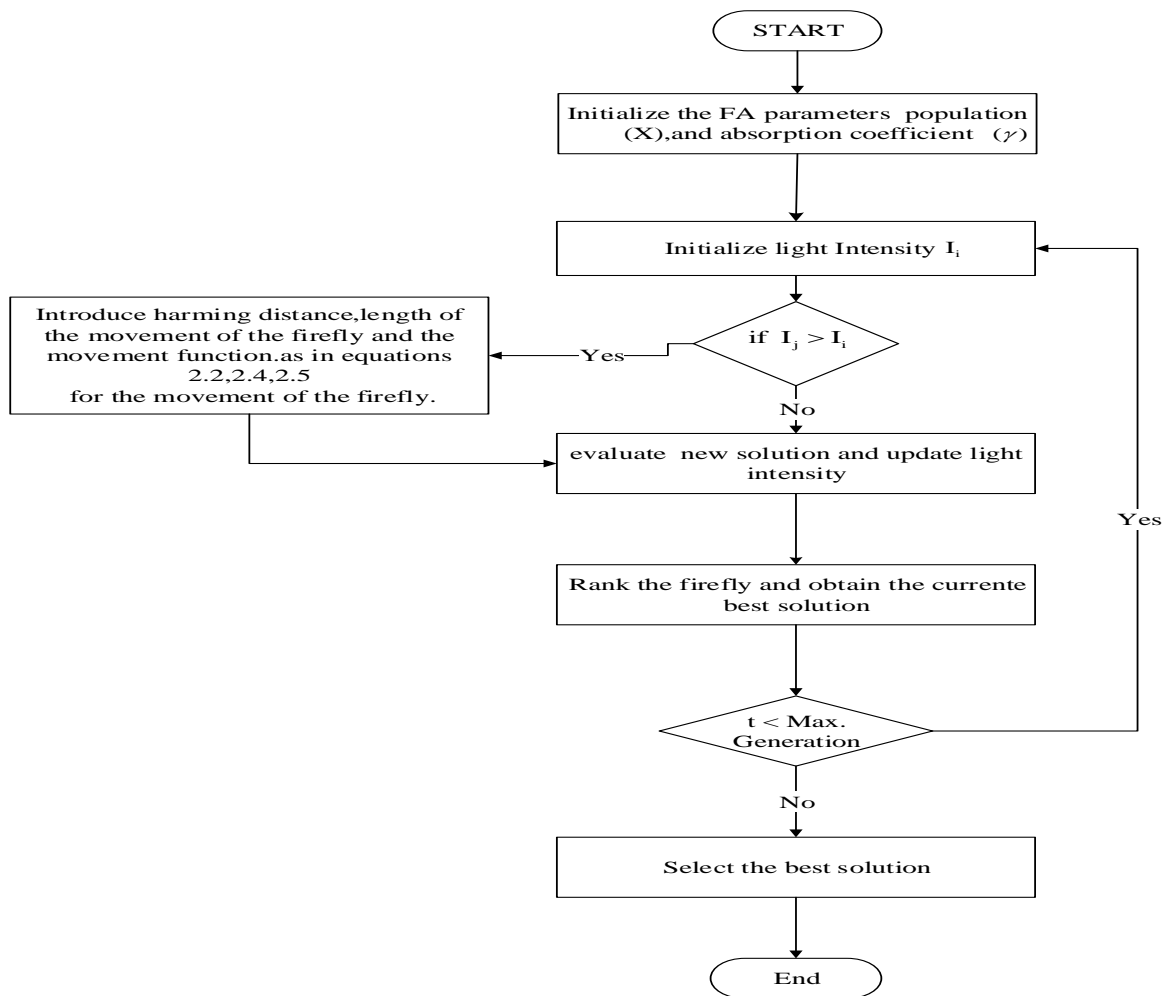


Figure 2.5: Flowchart of Discrete Firefly Algorithm (Osaba *et al.*, 2016)

2.2.4 Classification or recognition

At this stage face tested samples are compared or matched with existing known face samples in the database (Saleh, 2009). It is also the last stage in face recognition. As discussed in section 2.2.2.1

Some of approaches used for classification are:

1. Support Vector Machine (SVM).
2. Hidden Markov Model (HMM)
3. Back propagation neural network (BPNN).
4. Self Organizing Map (SOM).
5. Nearest Neighbor Classifier (NNC)

2.2.4.1 Support Vector Machine (SVM)

Support Vector Machines (SVM) are one of the most popular and talked about machine learning algorithms. The SVM is a fast and dependable classification algorithm that performs very well with a limited amount of data (Xu & Lee, 2014). SVM method is a classification method which separate the two data sets by searching for an optimal separation hyperplane (OSH) between them (Satone & Kharate, 2014). If data not linearly separable then it is transformed into new space using kernel and then finds the OSH. If data is not separated then it search the OSH which maximizes the misclassification. The SVM allows you to classify data that's linearly separable and if it's not linearly separable kernel tricks can be used. Compared to some classifiers like neural networks, they have higher speed and better performance with a limited number of sample (Xu & Lee, 2014).

2.2.4.2 Hidden Markov Model (HMM)

The Hidden Markov Model (HMM) are a technique applied in practical pattern recognition applications. Recently it has been used in vision: texture segmentation, face finding, object recognition and face recognition (Chen & Kundu, 1995) . In simpler Markov_models the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters, while in the hidden Markov model, the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens (Chen & Kundu, 1995). Therefore, the sequence of tokens generated

by an HMM gives some information about the sequence of states. face image represented as a sequence of states produced when the face is scanned from top to bottom, and HMM is made of states, where the probability to move from one state to another depends only on those two states and not any further history (Jameel, 2015) . HMMs generally work on sequences of symbols called observation vectors, such that the face image is divided into regions which each is assigned to a state in a left to right one dimensional HMM (Jameel, 2015).

2.2.4.3 Back propagation neural network (BPNN)

Back propagation is a multi-layer feed forward, supervised learning network based on gradient. The Back Propagation Network (BPN) is the best known and widely used learning algorithm in training multilayer perceptrons (MLP). The MLP refer to the network consisting of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes (Kumar & Kaur., 2012) . This BPNN provides a computationally efficient method for changing the weights in feed forward network, with differentiable activation function units, to learn a training set of input-output data. Being a gradient descent method it minimizes the total squared error of the output computed by the net (Liu *et al.*, 2013)

2.2.4.4 Self Organizing Map (SOM)

The principal goal of self-organizing maps is to transform an incoming signal pattern of arbitrary dimension into a one or two dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion (Bakshi & Singhal, 2014). Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors. (Bakshi & Singhal, 2014).The SOM neural network classify data's and compare with the database to check whether it is found in the database or not. The architecture of SOM is concern with feed-forward structure with a single computational layer of neurons arrange in rows and

columns each neuron is fully connected to all the source units in the input layer (Bakshi & Singhal, 2014).

2.2.4.5 Nearest Neighbour Classifier

The nearest neighbour classifier (NNC) is an important classifier for face recognition. It is also one of the simplest classifiers used in recent years by researchers to achieve better rate of classification. The goal of the NNC is to determine how close the samples are to the test sample (Kaur, 2012). An image in the test set is recognized by assigning to it the label of the closest point in the learning set, where distance are measured in image space (Zhu *et al.*, 2015). The closeness between the data points in the kernel nearest neighbor KNN is chosen by the Euclidean distance. A distance is assigned between all pixels in a dataset. Distance is defined as the Euclidean distance between two pixels. The Euclidean distance is given by (Zhu *et al.*, 2015):

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}$$

(2.10)

This Euclidean distance is by default in a KNN classifier. But the distance between two features can be measured based on one of the distance cosine and correlation (Kaur, 2012).

The KNN have a lesser execution time and better accuracy compared to other commonly used methods like hidden markov model and kernel method suite it best for classifying persons based on their images.

The NNC does not require learning (term: memory-base), It can be used even with few examples, It works very well in low dimensions for complex decision surfaces. It is slow in classification, suffers from the curse of dimensionality (Zhu *et al.*, 2015).

2.2.6.1 Recognition Accuracy

Recognition accuracy is the ratio of the total number of correctly identify probe images to the total number of probe images. It is also the number images correctly matching with the training images (Agarwal & Bhanot, 2015).

$$\text{Recognition Accuracy} = \frac{\text{no. of correctly recognised images}}{\text{total no. of test images}} \times 100$$

(2.11)

Figure 2.6 shows the block diagram of the work of Agarwal and Bhanot (2015), the first stages was detection of the face by preprocessing followed by feature extraction using discrete cosine transform (DCT) and haar wavelet based discrete wavelet transform (DWT), feature selection using firefly algorithm and finally classification using nearest neighbor classification.

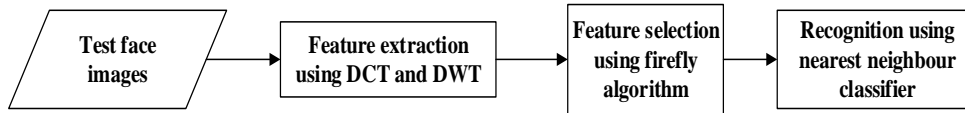


Figure 2.6: Steps of Face Recognition Approach Based on Firefly Algorithm for Feature Selection (Agarwal & Bhanot, 2015).

Figure 2.7 shows the block diagram of the work of the proposed approach where the images will be trained for detection and feature extraction using discrete cosine transform (DCT) and Haar wavelet based discrete wavelet transform (DWT), Feature selection using discrete firefly algorithm and finally classification using nearest neighbor classification.

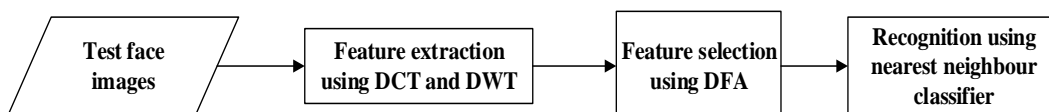


Figure 2.7: Steps of the Developed Approach

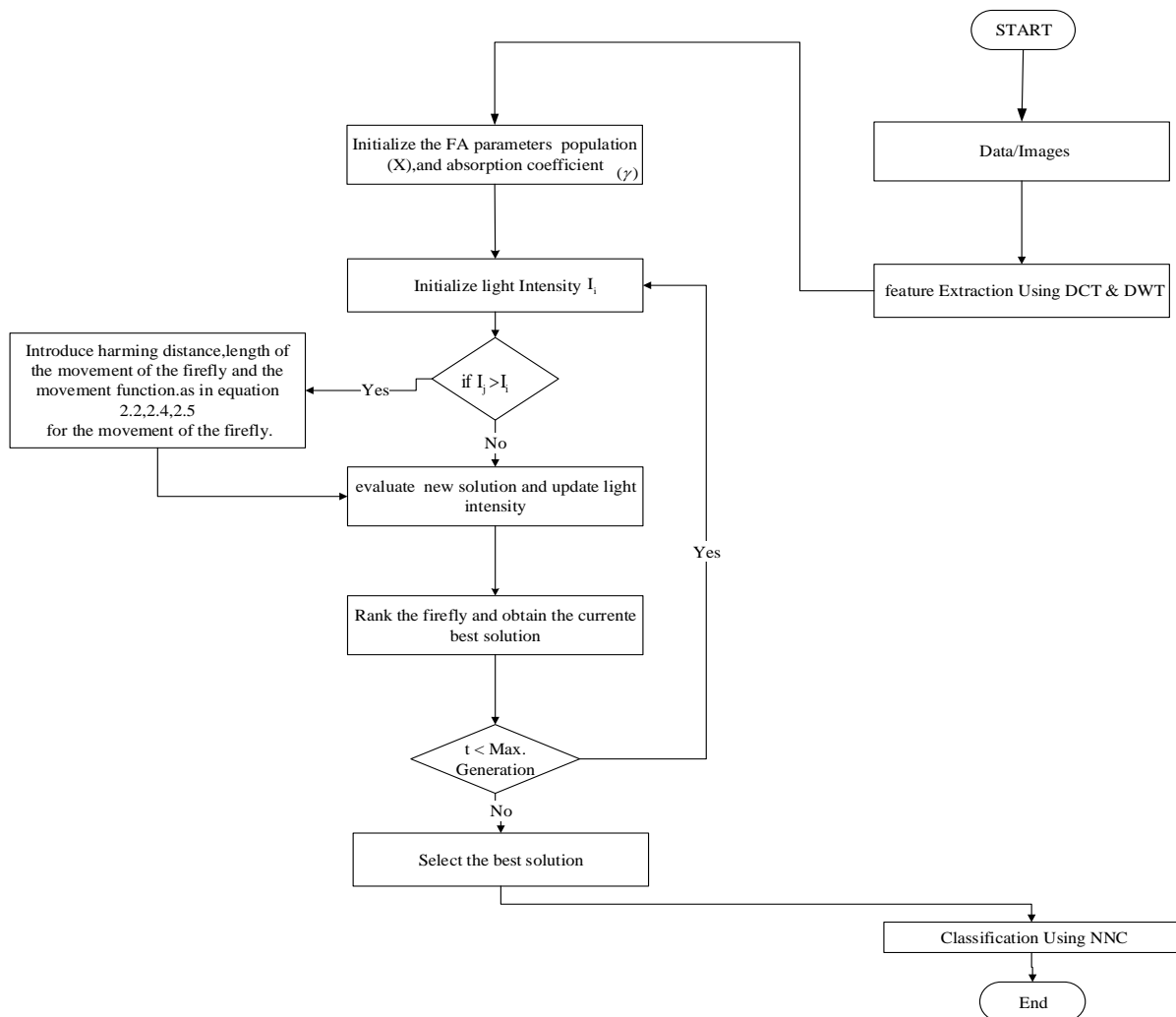


Figure 2.8: Flow chart of the Developed Face Recognition System

2.3 Review of Similar Works

Some literature relevant to the subject area of feature selection in face recognition approaches are described in this section

Kanan and Faez (2008) proposed an improved feature selection method based on Ant Colony Optimization (ACO) for face recognition system. In feature extraction, discrete

wavelet transform was used, after which pyramid algorithm was applied to each pre-processed image for three level resolution decomposition and the approximation of the images at level three was converted into vector by concatenating the columns. The ACO algorithm inspired ant's social behaviour in their search for the shortest paths for food sources was used for optimal selection of feature subset in terms of shortest feature length and best performance of classifier. The experimental results using ORL database showed that the proposed approach was easily implemented and without any priori information of features and its total performance was better than that of GA-based and other ACO-based feature selection methods. However, their algorithm converged very slowly which reduced the recognition rate.

Ramadan and Abdel-Kader (2009) presented a feature selection algorithm using particle swarm optimization (PSO) for face recognition. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) were used in extracting coefficients (features) and the PSO was used to select the optimum features of image. The optimum subset of features are selected by PSO based on the fitness function. ORL database was used for testing the performance. Face recognition was carried out based on Euclidean distance between features of unknown face and the features of faces in the database. The proposed algorithm has better performance when compared to GA-based feature selection algorithms in terms of number of features selected and classification accuracy rate. However, the performance cannot be guaranteed if the number of features was increased and as such, recognition accuracy will be reduced.

Ruan *et al.*, (2010) presented real adaboost feature selection for face recognition. In their work, adaboost was used for feature selection, Gabor for feature extraction, principal component analysis with fisher discriminant analysis for classification. A feature selection method based on Real Adaboost for Face Recognition was proposed based on intra-person

and extra-person which performs the multi-class-to binary transformation. Experimental results on the Face Recognition Grand Challenge version 2.0 with comparison to Joint Boosting and Discrete Adaboost confirm the effectiveness of Real Adaboost for Face Recognition using FRGC 2.0 large scale database. They also conclude that classifier based on feature selection outperforms classifiers based on original features. However their algorithm converged very slowly which reduced recognition rate.

Sawalha and Doush (2012) presented a face recognition approach with Harmony Search Algorithm (HSA) Based selected features. Principal component analysis (PCA) was used for feature extraction to extract eigenvectors which are a set of features that together characterized the variation between face images. HSA was used for selection of optimal combination of features that gives an optimal matching for ease of classification, Euclidean distance and cosine similarity were used as measure for the similarity between the test vector and the reference vectors in the training set. The performance of the proposed face recognition system was evaluated using the standard Cambridge ORL face database, which contains 400 images (of 40 subject), with size of 92x112 pixels, with 256 grey levels per pixel and each subject has 10 different images taken in various sessions varying the lighting, facial expressions (open/closed eyes, smiling/ not smiling) and facial details (glasses/ no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. The first 20 image classes were used. Five images per person were used in the training set, and the remaining five images were used for testing. The experimental results showed an efficient of the HS-based feature selection algorithm in generating excellent recognition accuracy with a minimal set of selected features. The work concluded that in future studies, the performance of the proposed face recognition algorithm can be improved in terms of recognition rate, time and memory space by using improved

feature selection and the use of another fitness function. Hence, increase in feature set decreases the performance of the approach, consequently, reduce recognition accuracy.

Tiwari and Vipinkumar (2012) presented a face recognition approach with cuckoo search algorithm (CSA) Based selected features. . Discrete Cosine Transform (DCT) was used for features extracting and Euclidean distance for classification. Their algorithm for feature selection work was implemented in MATLAB where each of 2-dimensional subset DCT array was converted to 1-dimensional array using raster scan. This is achieved by processing the images row by row concatenating the consecutive rows into a column vector which was input into the cuckoo feature selection algorithm. The experimental result shows that the CSA was more efficient than PSO for feature selection in face recognition. However their algorithm converged very slowly which reduced recognition rate.

Vignolo et al., (2013) presented a multi-objective wrapper based on genetic algorithms (GA) for selection of features for face recognition tasks. The technique explores the space of multiple feasible selections in order to minimize the cardinality of the feature subset, and also to maximize its discriminative capacity. In the study, they presented three different wrappers for feature selection in face recognition applications. The first wrapper was a classical GA, in which each individual represents a particular selection of the set of facial features extracted from an input image by means of active shape model (ASM), The second wrapper is a multi-objective GA with an aggregative fitness function, which chains classification accuracy and the number of features in a single equation and finally, the third wrapper consists of a multi-objective GA (MOGA), with the same objective functions considered for the second alternative. The technique was applied on face images of Essex face database, which contains a significant diversity of individuals and expression changes. For comparative evaluation of the experimental results with respect to other approaches available in the literature, 100 face classes were used. Five face images per class were randomly selected for training and other

fifteen face images per class were separated for the test set. Experiments results shows that the optimized feature sets gave improved classification accuracy when compared with other state of the art approaches. The work concluded with the suggestion that experiments with a larger data set, with increased variability of pose and illumination, and other options in terms of feature set optimization should be considered. However, their algorithm converged very slowly which reduced the recognition rate.

Satone and Kharate (2014) proposed a novel algorithm for face recognition in which a low frequency component of the wavelet was used for principal component analysis (PCA) representation. The discrete wavelet transform derives the multi-resolution features, where PCA transform the features into low dimensional space. Only the best features of PCA or best subset of eigen features were selected using genetic algorithm (GA) in order to improve face recognition performance. Support vector machine (SVM) and nearest neighbour classifier were used for classification. Classification accuracy was examined by changing number of training images, number of features and kernel function. Results were evaluated on ORL, FERET, Yale and YaleB databases. Experiments showed that the proposed method gives a better recognition rate than other popular methods. However their algorithm converged very slowly to solution which reduced the recognition rate.

Hemalatha Gayatri and Govindan (2015) presented an optimal feature selection using modified particle swarm optimization for face recognition. In order to improve the quality of the feature set by eliminating insignificant features which have very less or no contribution to outcome, thereby, reducing the number of features and decreasing computational cost and/or increasing accuracy of classification rate. In the stage of feature extraction, discrete cosine transform was used, where few coefficient are used and a 10*10 matrix was the resulting input to the modified PSO. Euclidean distance was used for identifying the images and face image corresponding to lowest Euclidean distance was considered as result in the approach.

The approach was tested on ORL database and outperformed some previously reported work in terms of recognition accuracy. However, the algorithm does not guarantee that the technique will record a better recognition accuracy which reduced recognition rate.

Babatunde *et al.*, (2015) developed a feature dimensionality reduction technique that employed Local Binary Pattern (LBP) for feature extraction and Ant Colony Optimization algorithm for selection of optimal feature subsets in face recognition systems. Experiments were conducted on Locally Acquired Face Database (LAFDAB) of 720 images (120 subjects), with a resolution of 1080*1920, with variations in facial expression (closed eyes, glasses worn, presence of moustache), and with illumination variations (was achieved by capturing at different times of the day, also with indoor and outdoor capture). The results of the experiment revealed that the developed technique has a very high performance in terms of average training time, recognition time and recognition rate that produced a robust and reliable face recognition system. Further evaluation of the technique on standard face databases was not considered and it converged slowly which reduced the recognition rate.

Agarwal and Bhanot (2015) proposed an adaptive technique for feature selection in face recognition using firefly algorithm. In feature selection, the challenge is evolving a technique that selects features closest to the best features and the number of these features is very small. As such, the firefly algorithm which is inspired by the flashing behaviour of natural fireflies was designed to obtain the best features extracted using Discrete Cosine Transform (DCT) and Haar wavelets based Discrete Wavelet Transform (DWT). The technique was validated using benchmark face database of Olivetti Research Lab (ORL) and Yale, which outperformed previously existing techniques in terms of recognition accuracy for five randomly training samples from the ORL database and six training samples for Yale face database. However, the algorithm is continuous which converged slowly, hence reduced the recognition rate.

Kallianpur et al., (2016) presented a novel and optimized Artificial Bee Colony to perform facial recognition in order to improve recognition accuracy. The variant of the ABC used contains certain elements of Particle Swarm Optimization (PSO), yielding a hybrid algorithm uniting the best of the two algorithms. In feature extraction, Discrete Wavelet Transform was employed, the hybrid discrete artificial bee colony algorithm for feature selection and classification or recognition was achieved using Euclidean classifier. The technique was tested on databases of Labeled Faces in the Wild (LFW) and Carnegie Mellon, which performed well in terms of recognition accuracy when compared with existing techniques. However, the algorithm was time consuming and works only for certain wavelets under very specific conditions. Another reason is that it takes a lot of training and testing time and demands a conclusive database. Also, database that contain images that are visually similar but have minor differences in them cannot be distinguished from each other as in the case of twins. Consequently, reduced recognition accuracy.

Khan and Gupta (2016) developed an improved artificial bee colony algorithm for face recognition system. In the work, face images were divided into several sub-windows using sub-window extraction algorithm. After that, with the help of artificial bee colony algorithm, these different sub windows are reduced or optimized for improving recognition rate. Finally principal component analysis (PCA) algorithm is applied on these reduced sub-windows for recognition of face images. The experiment was performed on two different datasets collected from FACE_94 and VITM College Gwalior. And both datasets contains 200 face images, image size of 320*240 and 50*35 sub-window size. The developed technique shows an improvement in recognition rate over existing methods. However, face facial expressions (e.g. pose, illumination) were not taken into consideration. As such, this does not guarantee that the technique will record a better recognition accuracy which reduced the recognition rate.

Boubenna and Lee (2016) introduced a new feature selection technique for facial emotion recognition system based on genetic algorithm (GA) and linear discriminant analysis (LDA) for optimizing the feature selection. For feature extraction, pyramid histogram of oriented gradient (PHOG) was used, GA for feature selection which find and select informative features from high dimensional data that explicitly maximizes the classification accuracy from linear discriminant analysis (LDA) classifier. Experiments were performed on the radboud faces database (RaFD), which consists of 67 samples: male, female and children, with a resolution of 681*1024 pixels, with 8 facial emotions (neutral, anger, fear, disgust, happiness, contempt, surprise and sadness) with three gaze directions. The approach was further applied on new samples of 465 images (76 subjects) of size 100*100, 306 of the images where used for training and 150 for testing with six basic emotions (fear, anger, happiness, surprise, disgust and sadness). Experimental results, shows that the approach is effective in decreasing the number of features and can classify efficiently than existing approaches. The work concluded with the suggestion that other methods that can deal with facial features can be incorporated. Hence, the technique if used for facial features will reduced recognition accuracy.

Mistry *et al.*, (2017) proposed a facial expression recognition system using evolutionary particle swarm optimization (PSO)-based feature optimization. In their work, they employed modified local binary patterns (mLBP) for feature extraction, which conduct horizontal and vertical neighbourhood pixel comparison, to generate a discriminative initial facial representation. The PSO variant embedded with the concept of a micro genetic algorithm (mGA), called mGA embedded PSO, is proposed to perform feature optimization or feature selection. It integrates a non-replaceable memory, a small-population secondary swarm, a new velocity updating strategy, a sub dimension based in-depth local facial feature search, and a cooperation of local exploitation and global exploration search mechanism to lessen the

premature convergence problem of conventional PSO. Multiple classifiers such as neural network and backpropagation, SVM and ensemble classifiers were used for recognizing seven facial expressions. Based on a comprehensive study using within- and cross-domain images from the extended Cohn Kanade and MMI benchmark databases, respectively, the empirical results indicates that the proposed system outperforms other state-of-the-art PSO variants, conventional PSO, classical GA, and other related facial expression recognition models reported in the literature by a significant margin. However their algorithm converged very slowly which reduced recognition rate.

Mistry *et al.* (2017) proposed a novel facial expression recognition using firefly based feature optimization. In their work, modified Local Gabor Binary Patterns (mLGBP) for feature extraction and firefly algorithm (FA) variant for feature selection. An extended overlap LGBP to extract initial discriminative facial features was employed in so as to deal with illumination changes, rotation variations and scaling difference. The modified FA was introduced to reduce the dimensionality of the extracted facial features. This FA variant employs Gaussian, Cauchy and Levy distributions to further mutate the best solution identified by the FA to increase exploration in the search space to avoid premature convergence. Multiple classifiers such as artificial neural network (NN), support vector machine (SVM), neural network (NN)-based ensemble and SVM-based ensemble were employed to recognize seven emotions including anger, sadness, disgust ,surprise, happiness, fear and neutral .The overall system was evaluated using three facial expression databases; cohn kanade (CK+), MMI, and JAFFE. Experimental result showed that the proposed system outperformed other heuristic search algorithms such as Genetic Algorithm and Particle Swarm Optimization and other existing state-of-the-art facial expression recognition research significantly. However their algorithm converged very slowly which reduced recognition rate.

It is evident from these literatures that metaheuristics search algorithm have been given a significant research attention leading to their application on feature selection for face recognitions. However, these literatures considered are continuous and need to be converted to discrete which is time consuming and complex, which tend to reduce the recognition accuracy.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

In this chapter, the methods, materials and procedures employed for the successful completion of this research are discussed and the feature selection based on discrete firefly algorithm was developed. The steps of the methodology adopted for this research, towards developing an improved face recognition based on discrete firefly for feature selection algorithm are as highlighted in section 1.6.

The experiment was performed on 64-bit OS system 64-based processor, intel (R) core (TM) 15-3470 CPU@3.20GHZ with RAM 8.00GB.

3.2 Implementation of Firefly Algorithm for Feature Selection for Face Recognition

The processes involved in the implementation of the FA for feature selection are discussed in details in the following sub-sections

3.2.1 Image acquisition

Some of the images used for this report were obtained as follows:

- a) Benchmark face database of Olivetti Research Labs (ORL) from <http://cvc.yale.edu/projects/yalefaces/yalefaces.html> .
- b) Benchmark face database of Yale <http://www.uk.research.att.com/facedatabase.html>.
- c) Some local images were gotten from Zaria city of Kaduna State

3.2.2 ORL face database

The ORL face database was developed by the Olivetti Research Labs. It consists of frontal images of 40 persons, capturing variations in pose, illumination and accessories like spectacles. There are 10 sample images of each person taken against dark homogeneous background. The database captures facial expressions with open/closed eye, smiling/not smiling, spectacles etc. The size of the face images is 92×112 pixels with 256 (8-bit) grey levels per pixel. It consist of five samples for testing and five samples for training from each person.

Figure 3.1 shows some images of ORL face database.



Figure 3.1: Obtained Images from ORL Face Database

3.2.3 Yale face database

The Yale face database consists of 165 images of 15 persons, 11 samples. These images capture variations in expression (normal, sad, surprise, sleepy, wink and happy) and

illumination (left, right, centre). The size of the face images is 92×112 pixels with 256 (8-bit) grey levels per pixel. It consist of five samples for testing and six samples for training from each person .Figure 3.2 shows some images of Yale face database



Figure 3.2: Obtained Images from Yale Face Database

3.2.4 Local Images Face Database

The local images face database obtained from Zaria city of Kaduna state which consist of 50 images of 10 persons, 5 samples (facial expressions). Most of these local images obtained are visible with cameras that retain the image quality as much as possible. The local images were captured using digital camera SAMSUNG ES63 (12 megapixel lens, 2.5inch LD screen, 3X optical zoom, 6.3-18.9mm). The mages were captured with different facial expressions (normal, happy, sad, with illumination and with glasses). The size of the face images was 92×112 pixels with 256 (8-bit) grey levels per pixel. The local images were used for application in this research work. It consist of one sample for testing and four samples for training from each person.

As a pre-processing step, background pixels that do not form part of face image were removed and only the face image will be considered. Since the local face image captured using digital camera SAMSUNG ES63 were prone to noise, we employed the use of a simple averaging filter were used to remove the noise in the local images. Figure 3.3 shows some images of local face database



Figure 3.3: Obtained Images from local Images Database

3.2.5 Importing of the Images into MATLAB ('imread')

'imread' is the command used in MATLAB to read images from the graphics file. It reads a grayscale or colour images from the file specified by the string (FILENAME). MATLAB supports several graphics format such as JPG, BMP, TIFF, PNG, HDT etc. In this research, each of the data is divided into testing images and training images in form of folders. As soon as the image to be processed was selected from the image folders, it was read into the MATLAB using the 'imread' command and subsequently, assigned to a MATLAB variable A.

3.2.6 Converting RGB into Grayscale Images

In converting RGB into grayscale images, 'rgb2gray' is the command used in MATLAB to convert coloured images to gray scale images. MATLAB eliminate the hue and saturation information in image while retaining its luminance. The coloured images of figure 3.3 was converted to gray image and assigned the ima in folder name capture14. It was then plotted in a 2x2 figure window, shown and titled using the 'imshow' and 'title' command respectively. Figure 3.4 shows the snippet of the conversion into grey scale image.

```

clc
ima=imread('C:\USERS\OMOSOFE\Desktop\Capture14.jpg');
imagray=rgb2gray(ima);
imshow(imagray)

```

Figure 3.4: Snippet of the Conversion to Grey Scale Image

3.3 Extraction using DCT and DWT

The DCT possess high information packing ability and can represent the face image in a very less number of coefficients. Wavelet have been used to decompose the face image using multi-resolution analysis. Discrete Wavelet Transform (DWT) are high in their information packing ability.

In this research work the features from a 10×10 square window from the upper left corner of the DCT transformed face image are first used. Second, all coefficients of the level 3, DWT approximation coefficients of Haar Wavelets are used in the formation of initial feature vector. Size of the level 3 DWT coefficient matrix is 12×14 (i.e. 168). The total feature vector size is 268. Figure 3.5 and 3.6 shows the snippet of DCT and DWT transformation.

```

% % DCT
f1 = @(block_struct) dct2(block_struct.data);
f2 = @(block_struct) idct2(block_struct.data);

J = blockproc(Im, [8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

```

Figure 3.5: Snippet of the Discrete Cosine Transform

```

% DWT

wname = 'haar';

[CA_LL1, CH_LL1, CV_LL1, CD_LL1] = dwt2 (Im, wname);
[CA_LL2, CH_LL2, CV_LL2, CD_LL2] = dwt2 (CA_LL1, wname);
[CA_LL3, CH_LL3, CV_LL3, CD_LL3] = dwt2 (CA_LL2, wname);

CCA_LL3=CA_LL3/max (max (CA_LL3));
dwt_fv=reshape (CCA_LL3, numel (CCA_LL3), 1);

feature_vector = vertcat (dct_fv, dwt_fv);

FV=horzcat (FV, feature_vector);

```

Figure 3.6: Snippet of Discrete Wavelet Transform.

3.4 Feature Selection with FA

In order to carry feature selection using firefly algorithm, the following steps was followed.

3.4.1 Initialization of the FA Parameters

The performance of FA depends on the appropriate selection of its control parameters (population and absorption coefficient). For the purpose of replication and implementation of FA algorithm, the FA parameters used in the work of Agarwal & Bhanot, (2015) were used. The appropriate values selected for these parameters are 0.95 as in equation 2.3, presented in subsection 2.2.2.

3.4.2 Initialization of Light Intensity

The performance of FA also depends on the light intensity of FA since the light intensity is a value that represents the brightness of a firefly, also the attractiveness of the firefly is determined by its brightness i.e. light intensity, which in turn is associated with the encoded objective function. The attractiveness can be calculated using equation (2.1).

3.4.3 Generating New Solution by Updating the Position of the Firefly

The movement of a firefly i attracted to another brighter (more attractive) firefly j that is moving firefly x_i towards x_j . The movement of the firefly is always from less brighter firefly to brighter firefly. The movement of firefly to a new position is determine by equation (2.3). Figure 3.7 shows the snippet of the FA algorithm used for the feature selection.

```
function FAFS()
clc
tic      % start timer

[training_data, testing_data] = split_data();
d=268;
T = feature_vector(testing_data);
V = feature_vector(training_data);
Q = 20;          %number of fireflies
gamma = 0.00001;
N = 20;         %maximum iteration
beta_0 = 1;
alpha = ones(1,d);
R = [1, d];
F = [];
fitness=[];
for i=1:Q
    F(i,:) = R(1) + (R(2)-R(1))* rand(1, d);
    fitness = [fitness; fitness_function(F(i,:), V, T)];
end
% GF = fitness_function(F(1,:), V, T)
result=[];
for iter = 1:N
    disp(['Iteration ', num2str(iter)])
    for u = 1:Q
        % GFu = fitness_function(F(u,:), V, T);
        GFu = fitness(u);
        for v = 1:Q
            % GFv = fitness_function(F(v,:), V, T);
            GFv = fitness(v);
```

Figure 3.7: Snippet of FA for Feature Selection

3.5 Feature selection with DFA

The steps of carrying out feature selection is as followed

3.5.1 Generating new solution by updating the position of the firefly

The movement of a firefly i attracted to another brighter (more attractive) firefly j is determined by equation (2.4) and (2.5). The movement of the firefly is i towards j and the length of movement of firefly will be randomly selected from equation (2.4). In this research, the movement of the firefly is based on hamming distance, length of movement of firefly and the movement function which are represented in equations (2.2), (2.4), and (2.5) respectively. For the movement function, the insertion function was used in this research work. Figure 3.8 shows the snippet of the DFA algorithm used for the feature selection.

```
function DFS()
clc
tic    % start timer

[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);

n=10;    % Population size, typically 10 to 25
d=268;   % Dimension of the search variables

gamma=0.95;

% Iteration parameters
maxIter=20;    % maximum number of iterations

% Initialize the population/solutions
for i=1:n,
    Sol(i,:) = randi([1, 268], [1,d]);
    Lightn(i) = fitness_function(Sol(i,:), V, T);
end
[val, ind] = sort(Lightn, 2, 'descend');
Sol = Sol(ind,:);
Lightn = val;

result = [];

for g=1:maxIter
```

Figure 3.8: Snippet of DFA for Feature Selection

3.6 Feature selection with LDA

The steps of carrying out feature selection is as follows:

1. Create a D-dimensional samples

Two samples with each having D-dimension are considered then each are projected on a line for easy separation.

2. Find a Measure of Separation of the Two Classes, Within Class Scatter and Between Class Scatter

Measure of separation is determine by the distance between the projected means, in measuring the measure of separation, standard deviation is not considered in within class scatter since it is not a good measure.

3. Find the optimum vector W

W is determine by maximizing between class scatter and minimize within class scatter.

Where:

$$W = \operatorname{argmax} \left| \frac{W' S_b W}{W' S_w W} \right|$$

(3.1)

Figure 3.9 shows the snippet of the LDA used for the face recognition.

```

function lda()
addpath('/LDA (FDA)');

tic          %start timer

[training_data, testing_data] = split_data();

% I1 = int8(reshape(testing_data(:,6), 112, 92));
% % I2 = imread('./fr/s1/1.pgm');
% % V = [training_data(:,1), reshape(I2, prod(size(I2)), 1)]
% figure; imshow(I1)

trainFV = feature_vector(training_data);
testFV = feature_vector(testing_data);

L = reshape(repmat(1:40, 5, 1), prod(size(repmat(1:40, 5,1))), 1);
A = trainFV;
X = A'*A;

[Y, W, lambda] = LDA(X, L);

```

Figure 3.9: Snippet of the LDA for Face Recognition

3.7 Feature Selection with PCA

The steps of carrying out feature selection is as followed

1. Creating matrix

Create A matrix from training images, example is a person with different variations and Compute B matrix from A. Compute eigenvectors of C from eigenvectors of B.

2. Reducing dimension

The PCA reduces the dimension by selecting few most significant eigenvectors of C for face recognition. Compute coefficient vectors corresponding to each training images. For each person, coefficient will form a cluster, compute the mean of the mean cluster. Figure 3.10 shows the snippet of the PCA for the face recognition.

```

function pca()
clc

tic          %start timer

[training_data, testing_data] = split_data();

trainFV = feature_vector(training_data);
testFV = feature_vector(testing_data);

L = reshape(repmat(1:40, 5, 1), prod(size(repmat(1:40, 5,1))), 1);
A = trainFV;

% % dimension reduction using PCA
C = A'*A;

[vectorC, valueC] = eig(C);
ss = diag(valueC);
[ss, iii] = sort(-ss);
vectorC = vectorC(:, iii);

vectorL = A*vectorC(:, :);

coeff = A'*vectorL;

```

Figure 3.10: Snippet of the PCA for Face Recognition

3.8 Classification using Nearest Neighbour Classifier (NNC)

In face-based classification, the objective of classifier is to compare the representation of a probe (or query) face image with those of the training set templates, and determine the category to which the probe image belongs. However, in this work we employed the nearest neighbour classifier (NNC) as discussed in chapter two, subsection 2.2.7. The goal of the NNC is to determine how close the samples are to the test sample. Figure 3.11 shows the snippet of the NNC for classification in the face recognition system.

```

%% % classification using nearest neighbor
Nc=0;
for i=1:200
    dist=[];
    for j=1:200
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/5) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = 200;
GF = (Nc/Nt) * 100;

```

Figure 3.11: Snippet of the NNC for Classification in Face Recognition

The complete flowchart of the work is shown in Figure 2.8.

3.9 Performance Evaluation

The performance of the developed feature selection for an improved face recognition was evaluated using the Recognition Accuracy and the Recognition Time.

3.9.1 Recognition Accuracy

Recognition accuracy or recognition rate is the performance metrics used in face recognition in order to estimate the effect of recognition rate between FA and DFA for feature selection which is determined using equation (2.11)

3.9.2 Recognition Time

This is time taken by the image to be recognised which was recorded by the difference between the value of the MATLAB command *tic* which record the beginning time of a running program and the value of the MATLAB command *toc*, which record the end time of

a running program. The elapsed time was then displayed in the MATLAB command line interface

3.9.3 Percentage Improvement

To show that DFA significantly improves the results of the FA, the percentage improvement was calculated using equation (3.2) and recorded.

$$\text{percentage improvement} = \frac{\text{DFA} - \text{FA}}{\text{DFA}} \times 100 \quad (3.2)$$

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, result and discussion of the theoretical concepts from the previous two chapters are presented. The performance of firefly algorithm (FA) and that of discrete firefly algorithm (DFA) all for feature selection in face recognition are discussed and relevant results reported. Furthermore the effectiveness of discrete firefly algorithm is compared with linear discriminant analysis (LDA) and principal component analysis (PCA) on ORL, Yale and local images databased and are demonstrated through the various simulation and comparison of the result.

4.2 Result of Gray Scale Conversion of Images on ORL, Yale and Local Images

The obtained images on ORL, Yale and local images were converted to gray scale images. This stage is important because the image is made up of wrinkles and speckles are both removed when converted to gray scale. Hence, in order to reduce the processing time of the algorithm the images must be converted from coloured to gray scale. The images obtained from ORL and Yale data bases are already processed and converted to grayscale but the local images collected require to be converted to gray scale. The result of the gray conversion of the obtained images of Figure 3.3 are presented in Figure 4.1.



Figure 4.1: Grayscale Images Obtained from ORL Face Database



Figure 4.2: Grayscale Images Obtained from Yale Face Database



Figure 4.3: Grayscale Images Obtained from Local Images Database

Table 4.1 shows the average recognition accuracy and the recognition time on ORL database using DFA, FA, PCA and LDA for fifty (50) runs.

Table 4.1: Average Recognition Accuracy and Recognition Time on ORL using DFA, FA, PCA and LDA Techniques

TECHNIQUES	RECONITION ACCURACY (%)	RECONITION TIME (SECS)	RECONITION TIME/ IMAGE (SECS)
DFA	97.75	42.27	0.21
FA	95.53	49.71	0.25
PCA	72.84	55.49	0.28
LDA	63.69	58.55	0.29

From Table 4.1. The recognition time per image was obtained by dividing the recognition time by the number of test images.

4.3 Result of FA on ORL Database

From Table 4.1 the recognition accuracy was obtained from MATLAB 2013b, it showed that the efficiency of the average recognition accuracy was 95.53% and its average recognition time was found to be 49.71 seconds for fifty runs. The recognition time/image was 0.25 sec/image on ORL database. The complete MATLAB code of the FA on ORL database is given in Appendix A1.

4.4 Result of DFA on ORL Database

From Table 4.1 the recognition accuracy obtained also showed that DFA achieved an average recognition rate of 97.75% and its average recognition time was 42.27 seconds for fifty runs.

The recognition time/image was 0.21 sec/image on ORL database the complete MATLAB code is given in Appendix A2.

Table 4.2 shows the average recognition accuracy and the recognition time on YALE database using DFA, FA, PCA and LDA for fifty (50) runs.

Table 4.2: Average Recognition Accuracy and Recognition Time on YALE database using DFA, FA, PCA and LDA Techniques

TECHNIQUES	RECOGNITION ACCURACY (%)	RECONITION TIME (SECS)	RECONITION TIME/IMAGE (SECS)
DFA	89.30	40.33	0.54
FA	85.33	43.65	0.58
PCA	68.66	48.13	0.64
LDA	65.54	50.42	0.67

From Table 4.2. The recognition time per image was obtained by dividing the recognition time by the number of test images.

4.5 Result of FA on Yale Database

From Table 4.2 the average recognition accuracy obtained was found to be 85.55% and its respective recognition time was also found to be 43.65 seconds for fifty runs. The recognition time/image was 0.58 sec/image on Yale database. The complete MATLAB code is given in Appendix A3.

4.6 Result of DFA on Yale Database

From Table 4.2 the performance of DFA on Yale database in terms of recognition accuracy and recognition time was also found to be 89.30% and 40.33 seconds respectively. The

recognition time/image was 0.54 sec/image on Yale database the MATLAB code is given in Appendix A4.

4.7 Comparison of Results

The comparison was carried out between DFA, FA, PCA and LDA in terms of Recognition accuracy and recognition time on the database of ORL and Yale.

4.7.1 Comparison of the Recognition Accuracy for DFA, FA, PCA and LDA on ORL Database

The comparison of DFA, FA, PCA and LDA in terms of Recognition Accuracy on the Database of ORL is shown in Figure 4.4.

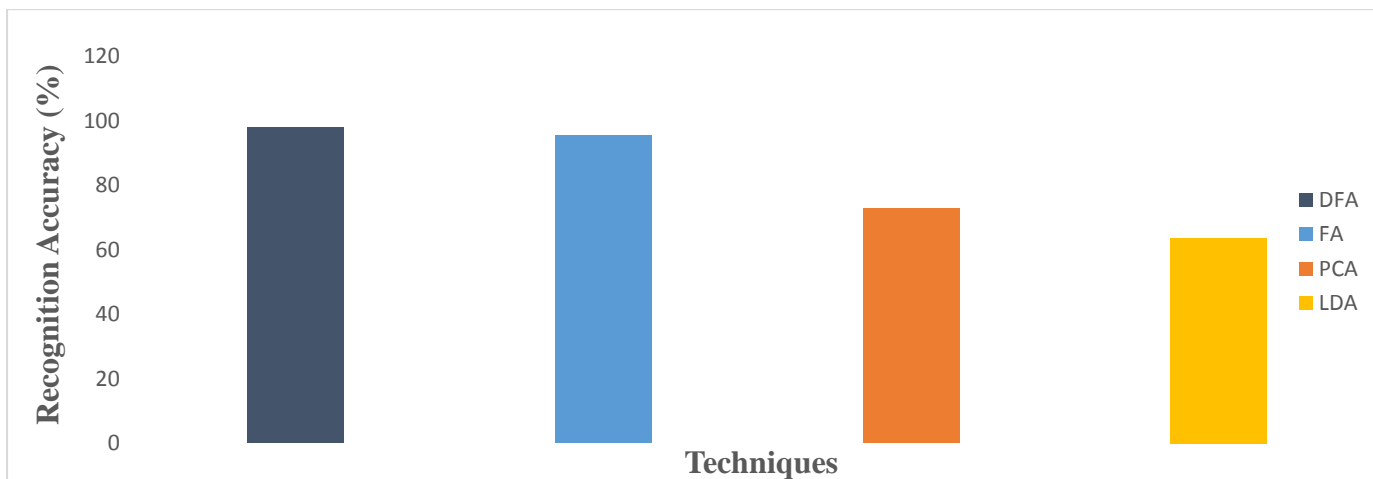


Figure 4.4: Comparison of Recognition Accuracy for DFA, FA, PCA and LDA on ORL Database

From Figure 4.4, it can be observed that the result obtained on DFA was better than that of FA, PCA and LDA in terms of Recognition Accuracy. The Recognition Accuracy of DFA was 97.75% while that of FA was gotten to be 95.53%, that of PCA was 72.84% and for LDA was 63.69%. The DFA shows an improvement in terms of the recognition accuracy and proved to be more efficient, as shown in Table 4.1.

4.7.2 Comparison of the Recognition Time of DFA, FA, PCA and LDA on ORL

Database

The comparison of DFA, FA, PCA and LDA in terms of Recognition Time on the Database of ORL is shown in Figure 4.5.

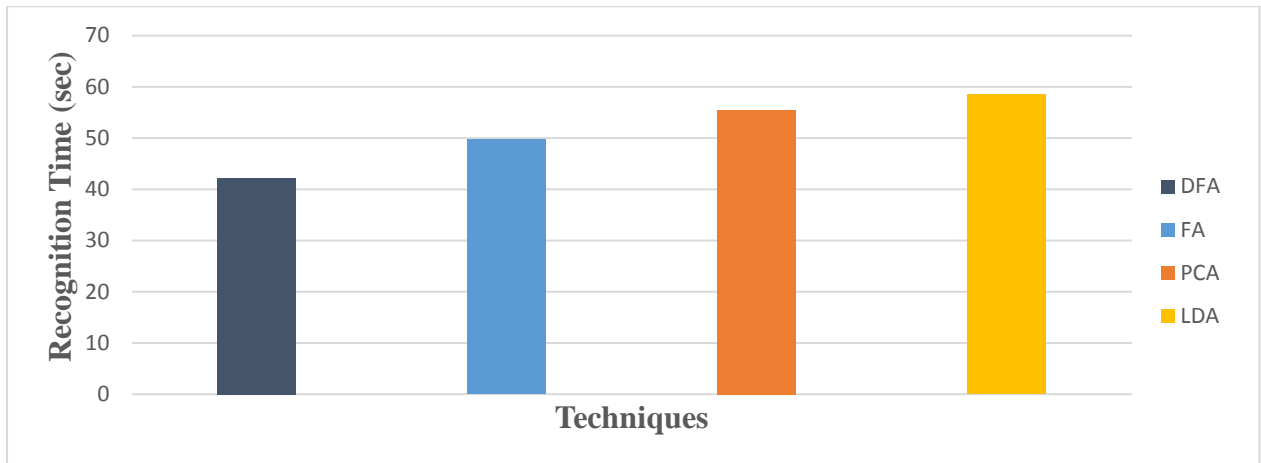


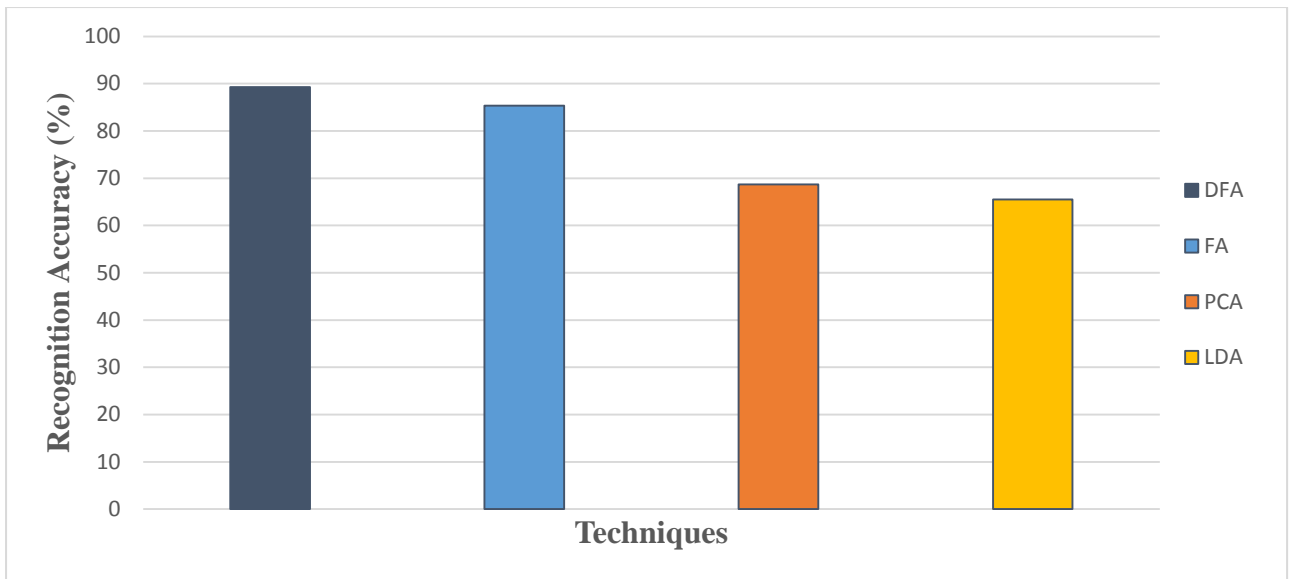
Figure 4.5: Comparison of Recognition Time for DFA, FA, PCA and LDA on ORL Database

Figure 4.5, shows the relationship in terms of Recognition time between DFA, FA, PCA and LDA respectively. It shows a minimization in Recognition time of DFA when compared to FA, PCA and LDA. However the Recognition time of DFA, FA, PCA and LDA was averagely found to be 42.27 seconds, 49.71 seconds, 55.49 seconds and 58.55 seconds respectively. From the result which shows that DFA is a more effective technique than FA, PCA and LDA. This is shown in Table 4.1.

4.7.3 Comparison of the Recognition Accuracy for DFA, FA, PCA and LDA on Yale

Database

The obtained result of the comparison of DFA, FA, PCA and LDA in terms of Recognition Accuracy on the database of Yale is shown in Figure 4.6.



From Figure 4.6, it can be observed that the result obtained on DFA was better than that of FA, PCA and LDA in terms of recognition accuracy. The recognition accuracy of DFA was averagely found to be 89.30% while that of FA was also averagely found to be 85.33% and that of PCA and LDA was 68.66% and 65.54% respectively. Which shows an improvement in terms of the recognition accuracy. This is shown in Table 4.2.

4.7.4 Comparison of the Recognition Time of DFA, FA, PCA and LDA on Yale

Database

The comparison of DFA, FA, PCA and LDA in terms of Recognition time on the database of Yale is shown in Figure 4.7.

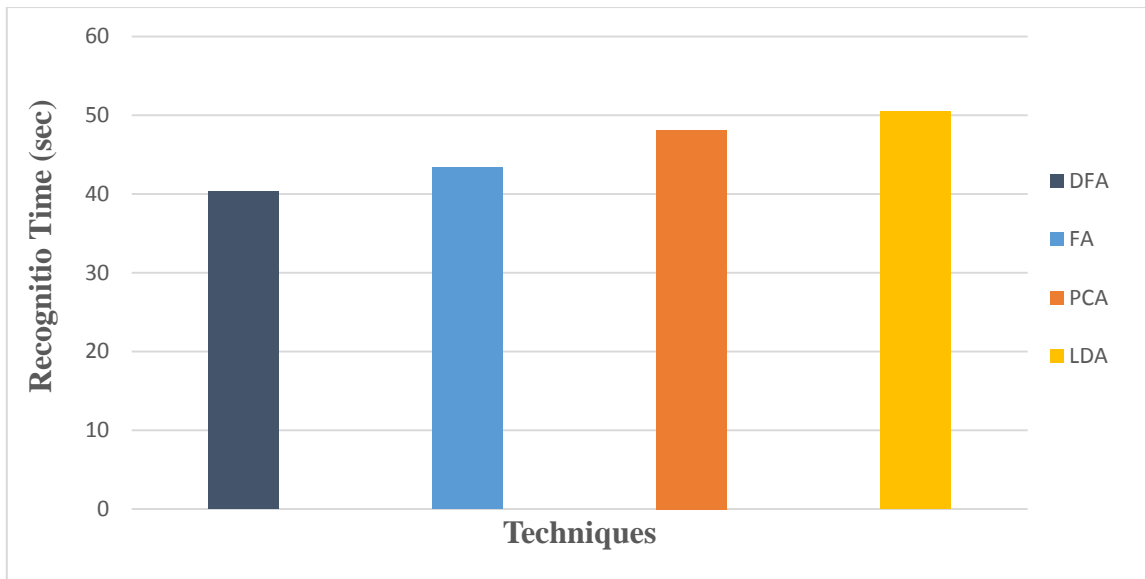


Figure 4.7: Comparison of Recognition Time for DFA, FA, PCA and LDA on Yale Database

Figure 4.7, shows the bar charts which represent the recognition time of each DFA, FA, PCA, and LDA respectively. It shows a minimization in recognition time of DFA when compared with FA, PCA and LDA. The obtained result of the DFA, FA, PCA and LDA in terms of recognition time was averagely found to be 40.33 seconds, 43.65 seconds, 48.13 seconds and 50.42 seconds respectively. This invariably shows that DFA is a more effective technique than FA, PCA and LDA. This is shown in Table 4.2

4.8 Result of DFA on local Images

The obtained recognition accuracy and recognition time on the local images was found to be 72.02% and 25.89 seconds respectively for fifty runs. The recognition time/Image was found to be 0.86 secs/image. This is shown in Table 4.3, and the complete MATLAB code is given in Appendix A5.

Table 4.3: Average Recognition Accuracy and Recognition Time on Local Images

TECHNIQUES	RECOGNITION ACCURACY (%)	RECONITION TIME (SECS)	RECONITION TIME/ IMAGE (SECS)
DFA	72.02	25.89	0.86

From Table 4.3. The recognition time per image was obtained by dividing the recognition time by the number of test images.

4.9 Performance Evaluation

The performance of the developed feature selection for an improved face recognition was evaluated using recognition accuracy and recognition time, using the percentage improvement formula as discussed in section 3.8.

4.9.1 Percentage Improvement on ORL Database

The percentage improvement of DFA with a recognition accuracy of 97.75% on ORL database when compared with FA with respect to the recognition accuracy was 2.27%, while with respect to the recognition time was 14.97%. Also when compared with PCA, it had a percentage improvement of 25.48% in terms of recognition accuracy and 23.82% in terms of recognition time. Also when compared with LDA, it had an improvement of 38.84% in terms of recognition accuracy, and 27.81% in terms of recognition time. This is shown in Appendix B1.

4.9.2 Percentage Improvement on Yale Database

The percentage improvement of DFA on Yale database when compared with FA in terms of recognition accuracy was 4.45 %, and in terms of recognition time was 7.61%. When compared with PCA and LDA, it gave an improvement in terms of recognition accuracy of

23.11% and 26.61%, and recognition time of 16.21% and 20.01% respectively. This is shown in Appendix B1.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary

This is aimed at the development of an improved face recognition based on Discrete Firefly Algorithm (DFA) for feature selection. DCT and DWT were used for feature extraction and NNC was used as classifier. The performance of DFA feature selection shows that it outperformed FA for feature selection. The developed DFA showed better recognition accuracy and recognition time.

5.2 Conclusion

The DFA based feature selection scheme was developed to improve face recognition. The DCT and DWT were used for feature extraction and NNC was used as classifier. In order to overcome the problem (time consumption) associated with feature selection, DFA has been developed. The work was developed in MATLAB R2013b, it was observed that the recognition accuracy and the recognition time was better when compared with the work that used firefly for feature selection. The algorithm was tested on benchmark of ORL and Yale face databases. The recognition accuracy, recognition time improvement was 2.27% and 14.97% respectively for ORL database, the recognition accuracy and recognition time improvement was 4.45% and 7.61% respectively were obtained for Yale database.

5.3 Significant Contributions

The significant contributions of this research work are as follows:

1. The development of DFA for feature selection.
2. The developed DFA showed an improvement of 2.27% and 14.97% for recognition accuracy and recognition time respectively on ORL database. The improvement of

4.45% and 7.61% for recognition accuracy and recognition time respectively on Yale database when compared with FA.

5.4 limitation

The limitation of this research work is as follow:

1. Acquiring more quality images for application.

5.5 Recommendations for Further Work

The following possible areas of further work are recommended for consideration for future research:

1. Employing different discrete algorithms for feature selection
2. Modification of the algorithm can be made by hybridizing the algorithm with other meta-heuristics heuristic algorithms.
3. Modification can be done on the discrete algorithm (Parametric study can be done on the parameters of the algorithm for better accuracy).
4. Different feature extractor and classifier should be introduced

REFERENCES

- Abd-Alsabour, N. (2014). *A Review on Evolutionary Feature Selection*. Paper presented at the European Modelling Symposium (EMS), 2014.
- Agarwal, V., & Bhanot, S. (2015). *Firefly inspired feature selection for face recognition*. Paper presented at the Eighth International Conference on Contemporary Computing (ICCC), 2015.
- Algorithms, F. R. (2010). Proyecto Fin de Carrera, June 16, 2010. *Ion Marques, Supervisor: Manuel Grana*.
- Angle, S., Bhagtani, R., & Chheda, H. (2005). *Biometrics: A further echelon of security*. Paper presented at the UAE International Conference on Biological and Medical Physics.
- Babatunde, R., Olabiyisi, S., Omidiora, E., & Ganiyu, R. (2015). Local Binary Pattern and Ant Colony Optimization Based Feature Dimensionality Reduction Technique for Face Recognition Systems. *British Journal of Mathematics & Computer Science*, 11(5), 1-11. doi: 10.9734/BJMCS/2015/19490.
- Bakshi, U., & Singhal, R. (2014). A New Approach of Face Recognition Using DCT, PCA, and Neural Network in Matlab. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 3(3).
- Boubenna, H., & Lee, D. (2016). *Feature selection for facial emotion recognition based on genetic algorithm*. Paper presented at the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016.
- Cappellini, V., & Del Re, E. (1985). Image data compression by the discrete cosine transform. *Mathematics and computers in simulation*, 27(5-6), 599-608.

- Chen, J.-L., & Kundu, A. (1995). Unsupervised texture segmentation using multichannel decomposition and hidden Markov models. *IEEE Transactions on Image Processing*, 4(5), 603-619.
- Dabbaghchian, S., Ghaemmaghami, M. P., & Aghagolzadeh, A. (2010). Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition*, 43(4), 1431-1440.
- Dai, F., Kushida, N., Shang, L., & Sugisaka, M. (2011). A survey of genetic algorithm-based face recognition. *Artificial Life and Robotics*, 16(2), 271.
- Déniz, O., Bueno, G., Salido, J., & De la Torre, F. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12), 1598-1603.
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2), 243-278.
- Eberhart, R. C., & Kennedy, J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the Proceedings of the sixth international symposium on micro machine and human science.
- El Aroussi, M., Amine, A., Ghouzali, S., Rziza, M., & Aboutajdine, D. (2008). *Combining DCT and LBP feature sets for efficient face recognition*. Paper presented at the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. .
- Fattah, S. A., Khan, M. R., Sharin, A., & Imtiaz, H. (2011). *A face recognition scheme based on spectral domain cross-correlation function*. Paper presented at the IEEE Region 10 Conference TENCON 2011-2011.
- Gross, R., Baker, S., Matthews, I., & Kanade, T. (2011). Face recognition across pose and illumination *Handbook of Face Recognition* (pp. 197-221): Springer.

- Gupta, D., & Choubey, S. (2015). Discrete wavelet transform for image processing. *International Journal of Emerging Technology and Advanced Engineering*, 4(3), 598-602.
- Hemalatha Gayatri, L., & Govindan, V. (2015). Feature Selection using Modified Particle Swarm Optimisation for Face Recognition. *International Journal of Research in Engineering and Technology*, 4(2), 679-683.
- Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*.
- Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6), 765-781.
- Jadon, S., Kumar, M., & Rathi, Y. (2015). *Face recognition using Som Neural Network with Ddct facial feature extraction techniques*. Paper presented at the International Conference on Communications and Signal Processing (ICCSP), 2015.
- Jakhar, R., Kaur, N., & Singh, R. (2011). Face recognition using bacteria foraging optimization-based selected features. *International Journal of Advanced Computer Science and Applications*, 1(3).
- Jameel, S. K. (2015). Face Recognition System Using PCA and DCTin HMM. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(1), 13-18. doi: 10.17148/IJARCCCE.2015.4103
- Kallianpur, A. K., Kalyani, A., Naligay, S. E., & Kallimani, J. S. (2016). *Facial recognition using discrete ABC*. Paper presented at the International Conference on Advances in Computing, Communications and Informatics (ICACCI).

- Kanan, H. R., & Faez, K. (2008). An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Applied Mathematics and Computation*, 205(2), 716-725.
- Kaur, M. (2012). K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion. *International Journal of Computer Applications*, 60(14).
- Kaur, R., & Singh, E. h. (2015). Review on Image Fusion and Its Techniques. *International Journal of Computer Networks and Wireless Communications (IJCNWC)*, 5(2), 508-515.
- Keche, JK, Dhore, & P, D. M. (2015). Human Face Recognition System–A Framework. *International Journal of Research in Computer Engineering & Electronics*, 4(2).
- Keche, J., & Dhore, D. M. P. (2015). Human Face Recognition System–A Framework. *International Journal of Research in Computer Engineering & Electronics*, 4(2).
- Khan, N., & Gupta, M. (2016). *Face Recognition System using Improved Artificial Bee Colony algorithm*. Paper presented at the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.
- Li, Z. K. (2014). *A Robust Face Representation Method for Face Recognition*. Paper presented at the Advanced Materials Research.
- Liu, M., Wang, M., Wang, J., & Li, D. (2013). Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. *Sensors and Actuators B: Chemical*, 177, 970-980.
- Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1), 1-11.

- Manikantan, K., Shet, M. S., Patel, M., & Ramachandran, S. (2012). DWT-based illumination normalization and feature extraction for enhanced face recognition. *International Journal of Engineering & Technology*, 1(4), 483-504.
- Mistry, K., Zhang, L., Sexton, G., Zeng, Y., & He, M. (2017). *Facial Expression Recognition using Firefly-based Feature Optimization*. Paper presented at the IEEE Congress on Evolutionary Computation (CEC), 2017 Donostia, San Sebastián, Spain.
- Mistry, K., Zhang, L., Neoh, S. C., Lim, C. P., & Fielding, B. (2017). A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. *IEEE Transactions on Cybernetics*, 47(6), 1496-1509. doi: 10.1109/TCYB.2016.2549639
- Narendra R., Dhoriya Devang U., & Shah. (2014). Comparative Study of Compression Techniques and Compressed Image Face Recognition System. *International Journal of Engineering Science Invention*.
- Ojala, T., Pietikäinen, M., & Mäenpää, T. (2000). Gray scale and rotation invariant texture classification with local binary patterns. Paper presented at the European Conference on Computer Vision.
- Osaba, E., Yang, X.-S., Diaz, F., Onieva, E., Masegosa, A. D., & Perillos, A. (2016). A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing*, 1-14.
- Ramadan, R. M., & Abdel-Kader, R. F. (2009). Face recognition using particle swarm optimization-based selected features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(2), 51-65.
- Jagroop, R., & Josan, K. (2013). Face Recognition System-A Survey. *International Journal of Science and Research (IJSR)*, 4(1), 2021-2024. doi: SUB15737

- Ruan, C., Ruan, Q., & Li, X. (2010). *Real adaboost feature selection for face recognition*. Paper presented at the IEEE 10th International Conference on Signal Processing (ICSP), 2010.
- Saleh, I. A. (2009). Principle Component Selection for Face Recognition Using Neural Network. *Reference Journal of Computation and Mathematics*, 6(1), 179-191.
- Satone, M., & Kharate, G. (2014). Face recognition technique using PCA, wavelet and SVM. *International Journal on Computer Science and Engineering*, 6(1), 58.
- Sawalha, R., & Doush, I. (2012). Face recognition using harmony search-based selected features. *International Journal of Hybrid Information Technology*, 5(2), 1-16.
- Shenglin, Z., & Shan-an, Z. (2011). *Face Recognition by LLE Dimensionality Reduction*. Paper presented at the Proceedings of the 2011 Fourth International Conference on Intelligent Computation Technology and Automation-Volume 01.
- Shivdas., A. E. (2014.). Face Recognition Using Artificial Neural Network *International Journal of Research in Management, Science & Technology*.
- Tian, J., Yu, W., & Xie, S. (2008). *An ant colony optimization algorithm for image edge detection*. Paper presented at the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence).
- Tian, Y.-L., Kanade, T., & Cohn, J. F. (2005). Facial expression analysis *Handbook of face recognition* (pp. 247-275): Springer.
- Tiwari, V. (2012). Face Recognition Based on Cuckoo Search Algorithm. *Indian Journal of Computer Science and Engineering (IJCSE)* 7(3), 401-405.
- Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, 206(3), 528-539.

- Vignolo, L. D., Milone, D. H., & Scharcanski, J. (2013). Feature selection for face recognition based on multi-objective evolutionary wrappers. *Expert Systems with Applications*, 40(13), 5077-5084.
- Xu, W., & Lee, E.-J. (2014). Face Recognition Using Wavelets Transform and 2D PCA by SVM Classifier. *International Journal of Multimedia and Ubiquitous Engineering*, 9(3), 281-290.
- Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18, 261-276.
- Yang, X.-S. (2009). *Firefly algorithms for multimodal optimization*. Paper presented at the International symposium on stochastic algorithms.
- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.
- Yang, X.-S., & Deb, S. (2009). *Cuckoo search via Lévy flights*. Paper presented at the World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009.
- Yang, M., & Zhang, L. (2010). Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. *Computer Vision–ECCV 2010*, 448-461.
- Zhang, X., Ren, D., Yan, L., Jing, Z., & Zhang, J. (2012). *A survey of approaches to feature extraction in face recognition*. Paper presented at the 2nd International Conference on Computer Science and Network Technology (ICCSNT), 2012.
- Zhou, H., Mian, A., Wei, L., Creighton, D., Hossny, M., & Nahavandi, S. (2014). Recent advances on singlemodal and multimodal face recognition: A survey. *IEEE Transactions on Human-Machine Systems*, 44(6), 701-716.

Zhu, N., Xu, T., Wei, L., & Tang, T. (2015). A nearest neighbor classifier based on virtual test samples for face recognition. *Optik-International Journal for Light and Electron Optics*, 126(21), 2799-2803.

APPENDIX A1

MATLAB CODE OF FIREFLY ALGORITHM (FA) ON ORL DATABASE

```
function FAFS()
clc
[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);
d=size(T,1);

Q = 20; %number of fireflies
gamma = 0.00001;
N = 20; %maximum iteration
beta_0 = 1;
alpha_bar = ones(1,d);
R = [1, d];
F = [];
fitness=[];
for i=1:Q
    F(i,:) = R(1) + (R(2)-R(1))* rand(1, d);
    fitness = [fitness; fitness_function(F(i,:), V, T)];
end
% GF = fitness_function(F(1,:), V, T)
result=[];
for iter = 1:N
    disp(['Iteration ', num2str(iter)])
    for u = 1:Q
        % GFu = fitness_function(F(u,:), V, T);
        GFu = fitness(u);
        for v = 1:Q
            % GFv = fitness_function(F(v,:), V, T);
            GFv = fitness(v);
            if GFu < GFv
                posVec = abs(F(v,:)-F(u,:));
                r = norm(posVec);
                beta = beta_0 * exp(-gamma * r^2);
```

```

        D=zeros(1,d);
        for i=1:d
            random_shift = alpha(i)*(rand-0.5);
            heuristic_shift = beta * posVec(i);
            D(i) = random_shift + heuristic_shift;
        end

        newPos = F(u,:) + D;
        for k=1:d
            if(newPos(k) > R(2))
                newPos(k) = R(2);
            elseif (newPos(k) < R(1))
                newPos(k) = R(1);
            end
        end
        F(u,:)=newPos;
        fitness(u) = fitness_function(F(u,:), V, T);
    end
end
end
tic    % start timer

    [fmax ind]= max(fitness);
    result = [result fmax];
end
toc    % end timer

result;
fmax
F(ind,:);
figure; plot(result)

function [training_data, testing_data] = split_data()
data=load('ORL_dataset');

training_data = [];
testing_data = [];

for i=0:10:390
    perm = randperm(10,10);
    training_data = horzcat(training_data, i+perm(1:5));
    testing_data = horzcat(testing_data, i+perm(6:10));
end

training_data = double(data.out(:, training_data));
testing_data = double(data.out(:, testing_data));

function FV = feature_vector(data)
FV = [];
for i=1:200
    Im=reshape(data(:,i), 112, 92);
    size(Im);

    %% DCT
    f1 = @(block_struct) dct2(block_struct.data);
    f2 = @(block_struct) idct2(block_struct.data);

```

```

J = blockproc(Im, [8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

%% %% DWT

wname = 'haar';

[CA_LL1,CH_LL1,CV_LL1,CD_LL1] = dwt2(Im,wname);
[CA_LL2,CH_LL2,CV_LL2,CD_LL2] = dwt2(CA_LL1,wname);
[CA_LL3,CH_LL3,CV_LL3,CD_LL3] = dwt2(CA_LL2,wname);

CCA_LL3=CA_LL3/max(max(CA_LL3));
dwt_fv=reshape(CCA_LL3, numel(CCA_LL3), 1);

feature_vector = vertcat(dct_fv, dwt_fv);

FV=horzcat(FV, feature_vector);

end

function GF = fitness_function(Fi, V, T)
Sf = unique(round(Fi));
V = V(Sf,:);
T = T(Sf,:);

%% %% classification using nearest neighbor
Nc=0;
for i=1:200
    dist=[];
    for j=1:200
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/5) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = 200;
GF = (Nc/Nt) * 100;

```

APPENDIX A2

MATLAB CODE OF DISCRETE FIREFLY ALGORITHM (DFA) ON ORL DATABASE

```
function DFS()

[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);

n=20;      % Population size, typically 10 to 25
d=268;     % Dimension of the search variables

gamma=0.95;

% Iteration parameters
maxIter=20; % maximum number of iterations

% Initialize the population/solutions
for i=1:n,
    Sol(i,:)= randi([1, 268], [1,d]);
    Lightn(i) = fitness_function(Sol(i,:), V, T);
end
[val, ind] = sort(Lightn, 2, 'descend');
Sol = Sol(ind,:);
Lightn = val;

result = [];

for g=1:maxIter
    disp(['Iteration ', num2str(g)])
    Lighto = Lightn;
```

```

for i=1:n
    for j=1:n,
        if Lightn(i)<Lightn(j), % Brighter and more attractive
            rij = hammingDistance(Sol(i,:), Sol(j,:));
            a= 2;
            b = rij * gamma^g;
            v = round(a + (b-a) * rand);
            [S, F] = insertionFunction(Sol(j,:),Lightn(j), v, V, T);

            newSol(i,:) = S;
            newLightn(i) = F;

        end
    end % end for j
end % end for i
[Lightn, ind] = sort(newLightn, 2, 'descend');
Sol = newSol(ind,:);

[fmax ind]= max(Lightn);

tic % start timer
result = [result fmax];
end
toc % end timer

result;
fmax
Sol(ind,:);
figure; plot(result)

% Hamming distance
function hd = hammingDistance(x, best)
hd=0;
d=length(x);
for i=1:d
    if x(i)~=best(i)
        hd=hd+1;
    end
end

function [S, F] = insertionFunction(x,l, v, V, T)
S = x;
F = l;

for i=1:v

    x_temp = x;
    pos = randperm(268,1);
    x_temp(pos) = randperm(268,1);
    fNew = fitness_function(x_temp, V, T);

    if fNew > F
        S = x_temp;
        F = fNew;
    end
end

function [training_data, testing_data] = split_data()

```

```

data=load('ORL_dataset');

training_data = [];
testing_data = [];

for i=0:10:390
    perm = randperm(10,10);
    training_data = horzcat(training_data, i+perm(1:5));
    testing_data = horzcat(testing_data,i+perm(6:10));
end

training_data = double(data.out(:, training_data));
testing_data = double(data.out(:, testing_data));

function FV = feature_vector(data)
FV = [];
for i=1:200
Im=reshape(data(:,i), 112, 92);
size(Im);

% % DCT
f1 = @(block_struct) dct2(block_struct.data);
f2 = @(block_struct) idct2(block_struct.data);

J = blockproc(Im,[8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

% % DWT

wname = 'haar';

[CA_LL1,CH_LL1,CV_LL1,CD_LL1] = dwt2(Im,wname);
[CA_LL2,CH_LL2,CV_LL2,CD_LL2] = dwt2(CA_LL1,wname);
[CA_LL3,CH_LL3,CV_LL3,CD_LL3] = dwt2(CA_LL2,wname);

CCA_LL3=CA_LL3/max(max(CA_LL3));
dwt_fv=reshape(CCA_LL3, numel(CCA_LL3), 1);

feature_vector = vertcat(dct_fv, dwt_fv);

FV=horzcat(FV, feature_vector);

end

function GF = fitness_function(Fi, V, T)
Sf = unique(Fi);

in = find(Sf == 0);
Sf(in) = [];

```



```

% Sf=find(Fi==1);
V = V(Sf,:);
T = T(Sf,:);

% % classification using nearest neighbor
Nc=0;
for i=1:200
    dist=[];
    for j=1:200
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/5) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = 200;
GF = (Nc/Nt) * 100;

```

APPENDIX A3

MATLAB CODE OF FIREFLY ALGORITHM (FA) ON YALE DATABASE

```

function FAFS()

[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);
d=size(T,1);

Q = 20; %number of fireflies
gamma = 0.0000001;
N = 20; %maximum iteration
beta_0 = 1;
alpha = ones(1,d);
R = [1, d];
F = [];
fitness=[];
for i=1:Q
    F(i,:)= R(1) + (R(2)-R(1))* rand(1, d);
    fitness = [fitness; fitness_function(F(i,:), V, T)];
end
% GF = fitness_function(F(1,:), V, T)
result=[];
for iter = 1:N
    disp(['Iteration ', num2str(iter)])
    for u = 1:Q
        % GFu = fitness_function(F(u,:), V, T);
        GFu = fitness(u);
        for v = 1:Q
            % GFv = fitness_function(F(v,:), V, T);
            GFv = fitness(v);
            if GFu < GFv
                posVec = abs(F(v,:)-F(u,:));
                r = norm(posVec);
                beta = beta_0 * exp(-gamma * r^2);
            end
        end
    end
end

```

```

        D=zeros(1,d);
        for i=1:d
            random_shift = alpha(i)*(rand-0.5);
            heuristic_shift = beta * posVec(i);
            D(i) = random_shift + heuristic_shift;
        end

        newPos = F(u,:) + D;
        for k=1:d
            if(newPos(k) > R(2))
                newPos(k) = R(2);
            elseif (newPos(k) < R(1))
                newPos(k) = R(1);
            end
        end
        F(u,:)=newPos;
        fitness(u) = fitness_function(F(u,:), V, T);
    end
end
end

tic      % start timer

    [fmax ind]= max(fitness);
    result = [result fmax];
end
toc      % end timer

result;
fmax
F(ind,:);
figure; plot(result)

function [training_data, testing_data] = split_data()
data=load('YALE_dataset');

training_data = [];
testing_data = [];

for i=0:11:154
    perm = randperm(11,11);
    training_data = horzcat(training_data, i+perm(1:6));
    testing_data = horzcat(testing_data, i+perm(7:11));
end

training_data = double(data.out(:, training_data));
testing_data = double(data.out(:, testing_data));

function FV = feature_vector(data)
FV = [];
for i=1:size(data,2)
    Im=reshape(data(:,i), 112, 92);
    size(Im);

% % DCT

```

```

f1 = @(block_struct) dct2(block_struct.data);
f2 = @(block_struct) idct2(block_struct.data);

J = blockproc(Im, [8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

% DWT

wname = 'haar';

[CA_LL1,CH_LL1,CV_LL1,CD_LL1] = dwt2(Im,wname);
[CA_LL2,CH_LL2,CV_LL2,CD_LL2] = dwt2(CA_LL1,wname);
[CA_LL3,CH_LL3,CV_LL3,CD_LL3] = dwt2(CA_LL2,wname);

CCA_LL3=CA_LL3/max(max(CA_LL3));
dwt_fv=reshape(CCA_LL3, numel(CCA_LL3), 1);

feature_vector = vertcat(dct_fv, dwt_fv);

FV=horzcat(FV, feature_vector);

end

function GF = fitness_function(Fi, V, T)
Sf = unique(round(Fi));
V = V(Sf,:);
T = T(Sf,:);

% % classification using nearest neighbor
Nc=0;
for i=1:size(T, 2)
    dist=[];
    for j=1:size(V, 2)
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/6) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = size(T, 2);
GF = (Nc/Nt) * 100;

```

APPENDIX A4

MATLAB CODE OF DISCRETE FIREFLY ALGORITHM (DFA) ON YALE DATABASE

```
function DFS()

[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);

n=20;      % Population size, typically 10 to 25
d=268;     % Dimension of the search variables

gamma=0.95;

% Iteration parameters
maxIter=20; % maximum number of iterations

% Initialize the population/solutions
for i=1:n,
    Sol(i,:)= randi([1, 268], [1,d]);
    Lightn(i) = fitness_function(Sol(i,:), V, T);
end
[val, ind] = sort(Lightn, 2, 'descend');
Sol = Sol(ind,:);
Lightn = val;

result = [];

for g=1:maxIter
    disp(['Iteration ', num2str(g)])
```

```

Lighto = Lightn;
for i=1:n
    for j=1:n,
        if Lightn(i)<Lighto(j), % Brighter and more attractive
            rij = hammingDistance(Sol(i,:), Sol(j,:));
            a= 2;
            b = rij * gamma^g;
            v = round(a + (b-a) * rand);
            [S, F] = insertionFunction(Sol(j,:),Lightn(j), v, V, T);

            newSol(i,:) = S;
            newLightn(i) = F;

        end
    end % end for j
end % end for i
[Lightn, ind] = sort(newLightn, 2, 'descend');
Sol = newSol(ind,:);

tic      % start timer

    [fmax ind]= max(Lightn);
    result = [result fmax];
end
toc      % end timer

result;
fmax
Sol(ind,:);
figure; plot(result)

% Hamming distance
function hd = hammingDistance(x, best)
hd=0;
d=length(x);
for i=1:d
    if x(i)~=best(i)
        hd=hd+1;
    end
end

function [S, F] = insertionFunction(x,l, v, V, T)
S = x;
F = l;

for i=1:v

    x_temp = x;
    pos = randperm(268,1);
    x_temp(pos) = randperm(268,1);
    fNew = fitness_function(x_temp, V, T);

    if fNew > F
        S = x_temp;
        F = fNew;
    end
end

```

```

end

function [training_data, testing_data] = split_data()
data=load('YALE_dataset');

training_data = [];
testing_data = [];

for i=0:11:154
    perm = randperm(11,11);
    training_data = horzcat(training_data, i+perm(1:6));
    testing_data = horzcat(testing_data,i+perm(7:11));
end

training_data = double(data.out(:, training_data));
testing_data = double(data.out(:, testing_data));

function FV = feature_vector(data)
FV = [];
for i=1:size(data,2)
Im=reshape(data(:,i), 112, 92);
size(Im);

% % DCT
f1 = @(block_struct) dct2(block_struct.data);
f2 = @(block_struct) idct2(block_struct.data);

J = blockproc(Im,[8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

% DWT

wname = 'haar';

[CA_LL1,CH_LL1,CV_LL1,CD_LL1] = dwt2(Im,wname);
[CA_LL2,CH_LL2,CV_LL2,CD_LL2] = dwt2(CA_LL1,wname);
[CA_LL3,CH_LL3,CV_LL3,CD_LL3] = dwt2(CA_LL2,wname);

CCA_LL3=CA_LL3/max(max(CA_LL3));
dwt_fv=reshape(CCA_LL3, numel(CCA_LL3), 1);

feature_vector = vertcat(dct_fv, dwt_fv);

FV=horzcat(FV, feature_vector);

end

function GF = fitness_function(Fi, V, T)
Sf = unique(round(Fi));
V = V(Sf,:);

```

```

T = T(Sf, :);

% % classification using nearest neighbor
Nc=0;
for i=1:size(T, 2)
    dist=[];
    for j=1:size(V, 2)
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/6) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = size(T, 2);
GF = (Nc/Nt) * 100;

```

APPENDIX A5

MATLAB CODE OF DISCRETE FIREFLY ALGORITHM (DFA) ON LOCAL IMAGES

```

function DFS_local()

[training_data, testing_data] = split_data();

T = feature_vector(testing_data);
V = feature_vector(training_data);

n=20;      % Population size, typically 10 to 25
d=268;     % Dimension of the search variables

gamma=0.95;

% Iteration parameters
maxIter=100; % maximum number of iterations

% Initialize the population/solutions
for i=1:n,
    Sol(i,:)= randi([1, 268], [1,d]);
    Lightn(i) = fitness_function(Sol(i,:), V, T);
end
[val, ind] = sort(Lightn, 2, 'descend');
Sol = Sol(ind,:);
Lightn = val;

result = [];

for g=1:maxIter
    disp(['Iteration ', num2str(g)])

```

```

Lighto = Lightn;
for i=1:n
    for j=1:n,
        if Lightn(i)<Lighto(j), % Brighter and more attractive
            rij = hammingDistance(Sol(i,:), Sol(j,:));
            a= 2;
            b = rij * gamma^g;
            v = round(a + (b-a) * rand);
            [S, F] = insertionFunction(Sol(j,:),Lightn(j), v, V, T);

            newSol(i,:) = S;
            newLightn(i) = F;

        end
    end % end for j
end % end for i
[Lightn, ind] = sort(newLightn, 2, 'descend');
Sol = newSol(ind,:);

[fmax ind]= max(Lightn);
result = [result fmax];
end

tic    % start timer
fmax = fitness_function(Sol(ind,:), V, T);
toc    % end timer

result;
fmax
Sol(ind,:);
figure; plot(result)

% Hamming distance
function hd = hammingDistance(x, best)
hd=0;
d=length(x);
for i=1:d
    if x(i)~=best(i)
        hd=hd+1;
    end
end

function [S, F] = insertionFunction(x,l, v, V, T)
S = x;
F = l;

for i=1:v

    x_temp = x;
    pos = randperm(268,1);
    x_temp(pos) = randperm(268,1);
    fNew = fitness_function(x_temp, V, T);

    if fNew > F
        S = x_temp;
        F = fNew;
    end
end
end

```



```

function [training_data, testing_data] = split_data()
data=load('local_dataset');

training_data = [];
testing_data = [];

for i=0:5:46
    perm = randperm(5,5);
    training_data = horzcat(training_data, i+perm(1:4));
    testing_data = horzcat(testing_data, i+perm(5:5));
end

training_data = double(data.out(:, training_data));
testing_data = double(data.out(:, testing_data));

function FV = feature_vector(data)
FV = [];
for i=1:size(data,2)
Im=reshape(data(:,i), 112, 92);
size(Im);

% % DCT
f1 = @(block_struct) dct2(block_struct.data);
f2 = @(block_struct) idct2(block_struct.data);

J = blockproc(Im, [8 8], f1);
dept = find(abs(J) < 50);
J(dept) = zeros(size(dept));
K = blockproc(J, [8 8], f2) / 255;

dct_fv = reshape(K(1:10,1:10), 100, 1);

% DWT
wname = 'haar';

[CA_LL1,CH_LL1,CV_LL1,CD_LL1] = dwt2(Im,wname);
[CA_LL2,CH_LL2,CV_LL2,CD_LL2] = dwt2(CA_LL1,wname);
[CA_LL3,CH_LL3,CV_LL3,CD_LL3] = dwt2(CA_LL2,wname);

CCA_LL3=CA_LL3/max(max(CA_LL3));
dwt_fv=reshape(CCA_LL3, numel(CCA_LL3), 1);

feature_vector = vertcat(dct_fv, dwt_fv);

FV=horzcat(FV, feature_vector);

end

function GF = fitness_function(Fi, V, T)
Sf = unique(round(Fi));

```

```

V = V(Sf, :);
T = T(Sf, :);

% % classification using nearest neighbor
Nc=0;
for i=1:size(T, 2)
    dist=[];
    for j=1:size(V, 2)
        dist(j) = norm(T(:,i)-V(:,j));
    end

    [dmin, ind] = min(dist);
    if ceil(ind/6) == ceil(i/5)
        Nc=Nc+1;
    end
end

Nt = size(T, 2);
GF = (Nc/Nt) * 100;

```

APPENDIX B1

PERCENTAGE IMPROVEMENT OF RECOGNITION ACCURACY AND RECONITION TIME OF ORL AND YALE DATABASES

ORL DATABASE

TECHNIQUES	RECOGNITION ACCURACY (%)	PERCENTAGE IMPROVEMENT (%)
FA	95.53	2.27
PCA	72.84	25.48
LDA	63.69	38.84

TECHNIQUES	RECONITION TIME (SECS)	PERCENTAGE IMPROVEMENT (%)
FA	49.71	14.97
PCA	55.49	23.82

LDA	58.55	27.81
------------	--------------	--------------

YALE DATABASE

TECHNIQUES	RECOGNITION ACCURACY (%)	PERCENTAGE IMPROVEMENT (%)
FA	85.33	4.45
PCA	68.66	23.11
LDA	65.54	26.61

TECHNIQUES	RECONITION TIME (SECS)	PERCENTAGE IMPROVEMENT (%)
FA	43.65	7.61
PCA	48.13	16.21
LDA	50.42	20.01