KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY,
KUMASI, GHANA

Statistical Evaluation of Enhanced Face Recognition Techniques under Variable Constraints

by

Louis Asiedu (BSc. MPhil Statistics)

A Thesis submitted to the Department of Mathematics, College of Science in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

AUGUST, 2016

CERTIFICATION

I hereby declare that this submission is my own work towards the PhD and that, to the best of my knowledge, it contains no material previously published by another person, nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

Louis Asiedu (Student) ID:
PG2522314

Signature
....
Date
....

Certified by: Dr. Atinuke O. Adebanji (Principal Supervisor)

Signature
....
Date
....

Certified by: Dr. Francis T. Oduro (Co-Supervisor)

Signature
....
Date
....

Certified by: Dr. R. K. Avuglah (Head of Department)

Signature
....
Date
....

ABSTRACT

Face recognition is a dedicated process in the brain which starts from 24 hours of birth. The idea of mimicking this essential skill in human beings by machines can be worthwhile though developing an intelligent and self-learning system may require adequate supply of information to the machine. This study modified Principal Component Analysis and Singular Value Decomposition (PCA/SVD) face recognition algorithm by incorporating Fast Fourier Transform (FFT) and Whitening at preprocessing stages. This study proposed multivariate statistical evaluation of the recognition performance of these face recognition algorithms under varying constraints. In particular, the Repeated Measures Design, Paired Comparison test and Profile Analysis were used for performance evaluation of the algorithms on the merit of efficiency and consistency in recognizing face images with variable facial expressions. The results indicated that, FFT-PCA/SVD is more consistent and computationally efficient when compared to PCA/SVD and Whitened PCA/SVD. Fast Fourier Transform is recommended as a viable noise removal mechanism, which should be adopted during preprocessing stages in image or pattern recognition.

TABLE OF CONTENTS

CHAPTER THREE

CHAPTER FOUR

LIST OF TABLES

LIST OF PLATES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CKFE	Cohn Kanade Facial Expressions
DARPA	Defense Advanced Research Products Agency
DFT	Discrete Fourier Transform
EBGM	Elastic Bunch Graph Matching
EVD	Eigen Value Decomposition
FBG	Face Bunch Graph
FERET	Face Recognition Technology Evaluation
FFT	Fast Fourier Transform
FRVT	Face Recognition Vendor Test
FRGC	Face Recognition Grand Challenge
GFE	Ghanaian Face Expressions
IC	Independent Component
ICA	Independent Component Analysis
IDFT	Inverse Discrete Fourier Transform
JAFFE	Japanese Female Facial Expressions
LDA	Linear Discriminant Analysis
LFA	Local Feature Analysis
ML	Maximum Likelihood
NMF	Non-Negative Matrix Factorization
PC	Principal Component
PCA	Principal Component Analysis
SSS	Small Sample Size
SVD	Singular Value Decomposition
RP	Random Projections

ACKNOWLEDGEMENT

I thank Almighty God for His guidance throughout this work. I take this opportunity to express my profound gratitude and deep regards to my supervisors, Dr. Atinuke O. Adebanji and Dr. Francis T. Oduro for their exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. From the formative stages of this thesis, to the final draft, I owe an immense debt of gratitude to my supervisors. Their sound advice and careful guidance were invaluable. I am grateful to Dr. Felix O. Mettle, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am obliged to Dr. Edward Prempeh, Dr. Samuel Iddi and Mr. Abeku AsareKumi, for the contributions provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Finally, I thank my parents, siblings and colleagues for their constant encouragement without which this assignment would not have been possible.

Thank you all for this immeasurable support. May the Almighty bless you all.

CHAPTER ONE

Introduction

1.0: Background of Study

The intricacy of a face's features originate from continuous changes in the facial features that take place over time (Rahman, 2013). Regardless of these changes, we are able to recognize a person very easily.

In human interactions, the articulation and perception of constraints; like, facial expressions form a communication channel that is additional to voice and that carries crucial information about mental, emotional and even physical states of a conversation (Bartlett, Donato, Ekman, Hager & Sejnowski, 1999).

Conventional methods for facial expressions extract features of facial organs such as eyes, mouth and recognize the expressions from changes in their shapes or their geometrical relationships by different facial expression (Bartlett *et al*., 1999).

It was shown by David Hubel and Torsten Wiesel as cited by Wagner (2012) that, our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we do not see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them (Wagner, 2012).

According to Starovoitov, Samal, Votsis and Kollias (1999), there are two main types of face recognition systems. One is a system that checks to see if a person belongs to a restricted group. Such systems are usually used in access control. The other is a system that identifies a person from a large database by photo search.

These systems can be created by different approaches such as, eigenface analysis (Chellapa & Wilson, 1995), template matching (Brunelli & Poggio, 1993), fiducial point based approach (McKenna & Gong, 1997) and others (Samal & Starovoitov, 1998).

In recent years, face recognition techniques have gained significant attention from researchers partly because face recognition is non-invasive with a sense of primary identification. This trend can be attributed to at least two other reasons; the first is the commercial and law enforcement applications, and the second is the availability of feasible technologies after years of research. Face recognition is an easy task for humans. Although the ability to infer the intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable (Turk & Pentland, 1991).

Currently, all face recognition techniques work in either of two ways. One is the local face recognition system which uses facial features (nose, mouth, eyes) of a face. That is to consider fiducial points in the face in order to associate the face uniquely with a person. The local-feature method computes the descriptor from parts of the face and gathers information into one descriptor. Some local-feature methods are,

Local Feature Analysis (LFA), Garbor Features, Elastic Bunch Graph Matching (EBGM) and Local Binary Pattern Feature (Agrawal *et al.,* 2014). The second approach is the global face recognition system which uses the whole face to identify a person. The principle of whole face method is to construct a subspace using algorithms like, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Random Projection (RP) or Non-negative Matrix Factorization (NMF). These are all dimensionality reduction algorithms that seek to reduce the large dimensional face image data to a smaller dimension for matching.

The object of data collection was frontal facial images from Cohn Kanade Facial Expressions database (CKFE), Japanese Female Facial Expressions database (JAFFE) and created Ghanaian Facial Expressions database (GFE).

These databases were selected to benchmark the recognition system. The Ghanaian Facial Expressions database (GFE) was created to invalidate the concern of database originality. Images were collected with preference to universal different facial expressions (Angry, Happy, Sad, Disgust and Surprise along with neutral pose). The collected images were resized into uniform dimensions. The recognition process comprises preprocessing of face image, facial feature extraction and modelling, face recognition and classification.

An important goal in image recognition is the ability to rate face recognition algorithms on the merit of efficiency and consistency in recognizing face images under variable constraints. Until now, a face recognition algorithm's rate, runtime, sensitivity (success rate), specificity and descriptive statistics are the basic means of rating face recognition algorithms' performance. The study focused on multivariate statistical evaluation of the recognition performance of PCA/SVD, Whitened PCA/SVD and Fast Fourier Transform-PCA/SVD under variable constraints (variable facial expressions). The study explored and compared techniques for automatically recognizing facial actions in sequence of images or detecting an "unknown" human face in input imagery and recognizing the faces under various constraints. The study used more intrinsic statistical methods (Multivariate methods) to assess the performance of face recognition algorithms under variable constraints.

1.1: Problem Statement

It was established by Turk and Pentland (1991) that, face recognition algorithms performance are restricted by constrained environments. Face recognition algorithm's rate, runtime, sensitivity and specificity are still the basic means of assessing the performance of face recognition algorithms. These evaluation methods only consider the rate of correct recognition, algorithm's speed and success rate. It is not possible to assess statistical significance from these numerical computations. That is, one cannot tell from these numerical computations whether a given constraint is significantly different from another when being recognized by an algorithm or whether significant difference exists between two algorithms in their recognition of a constraint. More importantly, one cannot assess significance of the variability in algorithms from these numerical computations.

The problem to look at, is to find a comparatively efficient (highest recognition rate) and consistent (lowest variation) face recognition algorithm based on statistical evaluation of the algorithms' performance under variable constraints. The statistical evaluation methods help to address the above mentioned limitations of the numerical evaluation methods. Specifically they help to compare mean recognition distances of algorithms and also assess algorithms' consistency (variations) on variable constraints.

1.2: Study Objectives

The main objective of the study is to identify an algorithm that is comparatively efficient and with the highest recognition performance in recognizing face image under certain constraints.

Specifically, the study seeks to ;

1. modify PCA/SVD face recognition algorithm at pre-processing phase usingWhitening and Fast Fourier Transform mechanisms.

2. statistically evaluate the performance of the face recognition algorithms inrecognizing face image under variable constraints and adjudge which algorithm is comparatively efficient and consistent.

1.3: Methodology

The research focused on running some template-based recognition algorithms on a created face database and subsequently evaluated the recognition performance of the algorithms.

Face image data were passed to face recognition modules as input for the system. The input image samples were local Ghanaian Facial Expressions (GFE),
Cohn Kanade Facial Expressions (CKFE) and Japanese Female Facial Expressions (JAFFE).
The face images passed, were transformed into operationally compatible format (resizing images into uniform dimension). The data type of the image samples were also changed into double precision and passed for preprocessing. The entire recognition exercise comprised a preprocessing stage, feature extraction stage and recognition stage. The adopted preprocessing procedures were basically, Mean Centering, Whitening and Fast Fourier Transform. This helped to reduce the noise level and made the estimation process simpler and better conditioned.
The various template based algorithms were used to train the image database. In the extraction unit, unique face image features were extracted and stored for recognition. The obtained facial features were passed to the classifier unit for classification of a given face query with knowledge created for the available database. For the implementation of the facial recognition, a real time created facial database was used. For the implementation of the proposed recognition design, the database samples were trained for the knowledge creation and classification. In the course of the training phase, when a new facial image was added to the system,

the features were calculated according to a particular recognition algorithm's procedure and aligned for the dataset information. The test face weight and the known weight in the database were compared by finding the norm of the difference between the test and known weights. A maximum and minimum difference signified poor and close match respectively. A detailed representation of the study methodology is discussed in chapter three.

1.3.1: Statistical Methods for Performance Evaluation

The statistical basis for evaluating the performance of the different face recognition algorithms are basically;

- The Repeated Measure Design: This test will be constructed for each recognition algorithm to ascertain whether there is a significant difference in the average distance for the various constraints from their neutral pose. This will reveal whether for a particular recognition algorithm, significant differences in mean exist between the various expressions or constraints from their neutral pose.

- Comparisons of two Multivariate Mean Vectors (Paired Comparisons): Multivariate data will be obtained from the computed euclidean norm across the variable constraints for the selected sample per each executed algorithm. For each algorithm, if there are k computed euclidean norms for a face image along the variable constraints, then there would be a $k - variate$ data for a sample of n individuals. The absolute deviations ($| \mathbf{X_k} - \bar{\mathbf{X}}_k |$) of these multivariate data are computed for both algorithms to create a common platform for comparison. A paired comparison test of equality of mean absolute deviation vectors is performed to confirm whether there is a significant difference in the mean absolute deviation of data from different algorithms.

Subject to the outcome of the test, a Bonferroni simultaneous confidence intervals for the difference in mean ($\mu_1 - \mu_2$, μ_1, μ_2 for algorithm 1 and 2 respectively) are computed. According to Johnson and Wichern (2007), these confidence intervals are developed by considering all possible linear combinations of the differences in the mean vectors. It is also assumed that the multivariate datasets are normal.

- The Box's M test for equality of covariance: This test is conducted for all the study recognition algorithms. The purpose of this test is to assess whether there exist significant difference in the covariance matrices of the multivariate recognition distance churned from the study algorithms.

- Profile Analysis: This test is constructed to check for parallel, coincident and level profiles. This test confirms whether or not the average distance of the various constraints from their neutral poses are parallel for the recognition algorithms. Subsequent check for coincident and level profiles will follow subject to the outcome of the test for parallel profile.

- Levene's Test for Equality of Variance: This is a univariate test to ascertain the direction of variation. This test helps to identify which constraints have significantly different variance and subsequently help to adjudge statistically, the well performing recognition algorithm on the study database.

- All the above listed statistical tests are preceded by the confirmation of their underlying assumptions.

  Detailed mathematical representations of these statistical methods are shown in chapter three.

1.3.2: Data Acquisition

The objects of data collection are labeled images in the wild and Frontal Ghanaian facial images captured under variable constraints. Specifically 294 facial images from 42 individuals are collected from a created Ghanaian Face Expressions database (GFE), Cohn-Kanade AU-Coded Facial Expressions database (CKFE) and Japanese Female Facial Expressions database (JAFFE) with preference to universally accepted facial expressions (Angry, Happy, Sad, Disgust, Fear and Surprise along with neutral pose). The collected images are resized into uniform dimensions and captured into a face database. Forty two of the images along neutral pose are trained into the database whereas the remaining 252 collected subject to variable constraints are used as test images during recognitions. *Plate 1.1* is an example of the face database constructed subject to variable constraints. A detailed description of the acquired data can be found in chapter three.

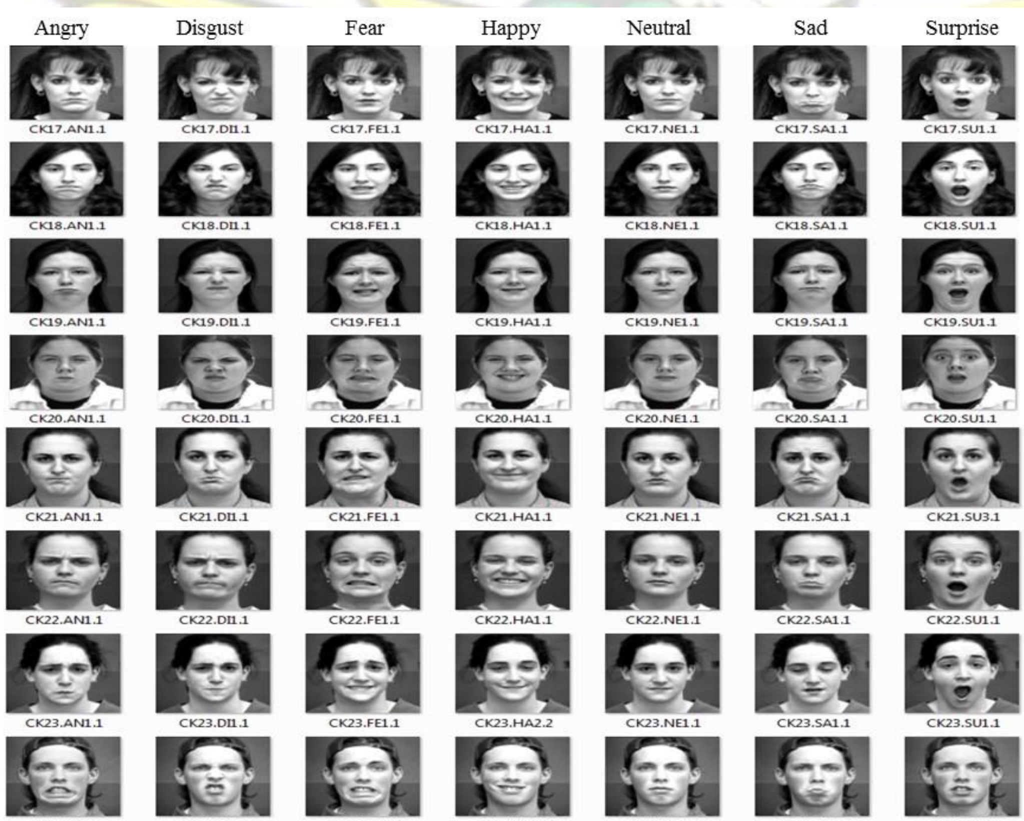


Plate 1.1: Example of face database constructed subject to variable constraints

1.3.3: Research Design

Figure 1.1 shows the various steps running from the pre-processing stages, through the feature extraction stage to the recognition stage. The database shown in the design contains the train image set which are trained per the recognition module and their corresponding informations stored in memory for recognition purpose. The unknown face shown in the design is also called the test image. This is introduced to be recognized in the database per the recognition module. The test image also is exposed to the entire recognition processes shown below and its important information kept in memory for comparison and identification. The ultimate goal of the process is to reduce the high dimension face image introduced, to small dimension thereby keeping important information for recognition purpose.

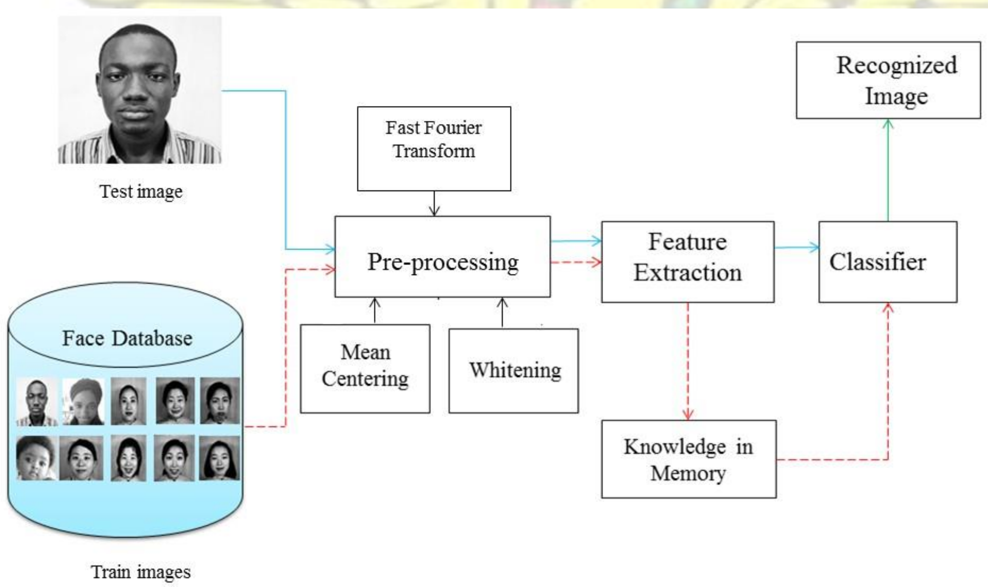


Figure 1.1 : Face Recognition Database Design

1.3.4: Computational Tool

The computational tool used for the image training and recognition and the statistical evaluation of the recognition algorithms is GNU-Octave 3.2.4. GNU Octave is a high-level language, primarily made for numerical computations. It provides a suitable interactive command line interface for solving numerical problems. It may also be used as a batch-oriented language for data processing. It also gives wide-ranging graphics capabilities for data visualization and manipulation. Octave is generally used through its interactive command line interface, but it can also be used to write non-interactive programs. The Octave language is quite similar to

Matlab in that most programs are easily portable.

1.4: Significance of Study

Despite the fact that there are other methods of identification (such as fingerprints, speech recognition, Iris recognitions, Iris scan etc) which are also accurate, it has been established that face recognition is a dedicated process in the brain (Marqu´*es*, 2010). Face recognition is a non-invasive method with a sense of primary identifi-

cation.

One of the main driving forces for face recognition is the ever growing number of applications that an efficient and resilient recognition technique addresses; for example, security systems based on biometric data and user-friendly human-machine interfaces. An example of the later category is smart rooms, which use cameras and microphone arrays to detect the presence of humans, decide on their identity and then react according to the predefined set of preferences for each person. *Table*

1.1 shows the various application areas of face recognition.

Table 1.1: Applications of face recognition.

Areas	Applications of Face Recognition.
Information Security	Accessing Security, Data Privacy, User authentication.
Access Management	Secure access authentication (restricted facilities).
Biometric	Personal Identification (National IDs, passports etc).
Law Enforcement	Video surveillance, Forensic Reconstruction of Face from remains.
Entertainment Leisure	Photo camera and home video games.

The study modified an existing recognition algorithm at its preprocessing phase.

The multivariate statistical methods introduced by the study helped to evaluate the performance of face recognition algorithms under variable constraints and subsequently identify a comparatively efficient and consistent face recognition algorithm under variable constraints.

1.5: Limitations of Study

The limitations of this study are constraints like; ageing, occlusions, face image in glasses and lightening conditions. Age is a time series data and its collection as a constraint takes relatively longer period since a particular face image has to be captured over some specified period of years.

There is also a challenge in developing a large Ghanaian face database along the proposed constraints for comparison. This accounted for the smaller Ghanaian face database used in the study. Age and large Ghanaian database constraints stated can be attributed to time and financial constraints respectively, since they could be adequately addressed with available time and sufficient financial support.

1.6: Research Contributions

This research modified Principal Component Analysis and Singular Value Decomposition (PCA/SVD) algorithm using different preprocessing mechanisms (Whitening and Fast Fourier Transform). These modifications helped to reduce the noise in images and made the images better conditioned for recognition.

Multivariate statistical methods for evaluating the performance of face recognition algorithms are also proposed. Using an existing numerical evaluation method (recognition rate) and the proposed multivariate methods, the study algorithms were rated on merit of efficiency (highest recognition rate) and consistency (lowest variation). These multivariate methods are new contributions to the face recognition evaluation system. They helped to distinguish between face recognition algorithms based on their recognition variations. They also helped to assess recognition algorithms performance on variables constraints (multivariate data). By using the proposed statistical evaluation methods, we can tell whether two or more recognition algorithms are significantly different in their mean recognition distances or in variations when used to recognize face images under variable constraints.

The modified face recognition algorithm FFT-PCA/SVD outperformed the existing PCA/SVD after the statistical evaluation. The research therefore reveals FFT-PCA/SVD as a viable face recognition algorithm for face recognition under variable constraints.

1.7: Structure of the Thesis

The study is structured into five chapters. Chapter one gives the background of the study, problem statement, objectives, brief description of the methodology, study relevance, limitations and research contributions. Chapter two reviews some relevant literature. In chapter two, some template-based and geometric-based algorithms with their mathematical representations are reviewed. Chapter three talks about the research methodology. In this chapter, all the methods employed in the study with their mathematical foundations are discussed. Chapter four presents the analytical results, their respective discussions and interpretations. Chapter five, summarises all the results and makes conclusions and recommendations.

CHAPTER TWO

Literature Review

2.0: Introduction

This chapter entails the review of some template-based face recognition algorithms. The subject of face recognition is as old as computer vision, because of its theoretical and practical importance. The earliest work on facial recognition was done in the 1950's in a psychology context by Bruner and Tagiuri (Marqu´*e*s, 2010). Engineering started to show interest in face recognition in the 1960's. One of the first engineering researchers on face recognition was Woodrow W. Bledsoe (Marqu´*e*s, 2010). Bledsoe designed and implemented a semi-automatic system. Some face coordinates were selected by a human operator. The administrator was required to locate features (such as ear, eye, nose and mouth) on the photograph before it calculated the distances and ratios to a common reference point. The computers used this information for recognition. He described most of the problems saying that, even in subsequent years face recognition might still suffer from effects of variations in illumination, head rotation, facial expressions and ageing (Marqu´*e*s, 2010).

It was also around the 1960's that automated face recognition was developed. In the 1970's, Goldstein, Harmon and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition process (Sharif *et al*., 2011). The shortcoming in both methods was that the measurements and locations were manually computed.

Principal Component Analysis was introduced as an algebraic manipulation to face recognition problem by Kirby and Sirovich (1990). This made it easy to calculate directly the so called eigenfaces. The limitation of this technique was that, fewer than 100 values were required to accurately code suitably aligned and normalised face images.

One of the famous examples of face recognition systems is the work by Kohonen (1989). Kohonen demonstrated how simple neural networks could perform face recognition for aligned and normalised face images. He employed computer face description to approximate the eigenvectors of the face image autocorrelation matrix. These eigenvectors are now known as eigenfaces. Kohonen's system seemed good because of the need for precise alignment and normalization but his system was not a practical success.

Other researchers tried face recognition schemes based on edge inter-feature distances and other neural network approaches. While several were successful on small database of aligned images, none successfully addressed the more realistic problem of large database with unknown location and scale of face.

Turk and Pentland (1991) demonstrated that while using eigenface technique, the residual error could be used to detect faces in cluttered natural imagery and to determine the precise location and scale of faces in the image. Turk and Pentland further demonstrated that this method of facial detection coupled with localizing faces with the eigenface recognition method could lead to achieving a reliable real-time recognition of faces in minimally constrained environment. Although the approach was to a certain extent restricted by environmental factors, it still created significant interest in advancing the developments of automated face recognition technologies.

Since the 1990's, the face recognition area has received a lot of attention with noticeable increase in the number of publications. Different Algorithms have emerged. Among them are PCA, ICA, LDA and their derivatives.

The Face Recognition Technology Evaluation (FERET) was sponsored by the Defense Advanced Research Products Agency (DARPA) from 1993 through 1997. It encouraged the development of face recognition algorithms and technology by assessing the prototype of face recognition systems. It propelled the recognition from its infancy to a market of commercial product.

Public attention was drawn to the technology from the media reaction to a trial implementation at the January 2001 Super Bowl, which captured surveillance images and compared them to a database of digital mugshot (National Science and Technology Council, 2006). This work initiated an ultimate attention on the technology to support natural needs while being considerate of public social and privacy concern.

Viola and Jones (2001) proposed a multi-stage classification procedure for face recognition that reduces the processing time substantially while achieving almost the same accuracy as compared to a much slower and more complex single stage classifier. Lienhart and Maydt (2002) extended their rapid object detection framework in two important ways: Firstly, their basic and over-complete set of haar-like feature was extended by an efficient set of 45° rotated features, which added additional domain-knowledge to the learning framework. Secondly, they derived a new post optimization procedure for a given boosted classifier that improved its performance significantly. The Face Recognition Vendor Tests (FRVT) were performed in 2000, 2002, and 2006. These evaluations built upon the work of FERET. The two goals were to assess the capabilities of commercially available facial recognition systems and to educate the public on how to properly present and analyse results. Delac, Grgic and Grgic (2005) used some descriptive statistics to measure performance of face recognition algorithms. In their paper, they introduced measures of central tendencies, measures of dispersion, skewness and kurtosis of some templatebased recognition algorithms and subsequently analysed the probability distribution of these algorithms. Beveridge *et al.,* (2001) also investigated only Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) in not as much detail using descriptive statistics. The Face Recognition Grand Challenge (FRGC) was introduced by Phillips, Flynn, Scruggs, Bowyer and Worek (2006), with the goal of improving the performance of face recognition algorithms by an order of magnitude over the best results in face recognition vendor test (FVRT) in 2002. High-resolution face images, 3D face scans, and iris images were used in the tests. The results indicated that the new algorithms are 10 times more accurate than face recognition algorithms of 2002 and 100 times more accurate than those of 1995. Some of the algorithms were able to outperform human participants in recognizing faces and could uniquely identify identical twins.

Today, face recognition technology is being used to combat passport fraud, support law enforcement, identify missing children and minimize benefit/identity fraud.

Zhang, Ding, Li and Liu (2015), also proposed an improved PCA approach based on facial expression recognition algorithm using Fast Fourier Transform (FFT) during the preprocessing stage. They combined the amplitude spectrum of one image with phase spectrum of another image as a mixed image. This research focuses primarily on the faces as attention in social intercourse and also playing a major role in conveying identity and emotion. We recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expressions, ageing and distractions such as glasses or changes in facial hair or hair style. As a consequence, the visual processing of human faces has fascinated philosophers and scientists for centuries.

Recent years have witnessed a growing interest of researchers in the field of face recognition problem. The remarkable interest is based on the increasing application areas of the face recognition system. Among these application areas are the Information Security (Data privacy, User authentication), Biometric system (National IDs, Passport checks, etc), Law enforcement (video surveillance, forensic reconstruction of face, criminal investigation, etc), Image and Film processing.

It is also notably clear that, these researchers studied this problem with different algorithms (from different point of views). Most of the face recognition algorithms can be classified into two groups; image template-based or geometry feature-based. Template-based methods compute a measure of correlation between new faces and a set of template models to estimate the face identity (Guillament & Vitri´*a*, 2002). Several well-known statistical techniques have been used to define a template model, such as Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and the Non-Negative Matrix Factorization (NMF).

These algorithms are mostly dimension reduction algorithms which seeks to reduce the dimensions of the face space, thereby bringing out only important points in the face for recognition and identification processes. In the field of face recognition, the dimension of the facial images is very high and requires considerable amount of computational time for classification (Thakur *et al.*, 2008).

Geometric feature-based methods analyse explicit local facial features, and their geometric relationships. Some examples of these methods are the active shape model, the elastic bunch graph matching algorithm for face recognition and the Local Feature Analysis (LFA) (Guillament & Vitri´*a*, 2002).

Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. One of the first automated face recognition systems was described by Kanade (1973); marker points (position of eyes, ears, nose, etc) were used to build a feature vector (distance between the points, angle between them, etc). The recognition was performed by calculating the euclidean distance between feature vectors of a probe and reference image. Such a method is robust against changes in illumination by its nature, but has a huge drawback; the accurate registration of the marker points is complicated, even with state of the art algorithms. Some of the latest work on geometric face recognition was carried out by Brunelli and Poggio (1993). A 22-dimensional feature vector was used but experiments on large datasets have shown that, geometric features alone do not carry enough information for face recognition.

Generally, the face space is very large with a lot of correlated variables. The classification and subsequent recognition time can be reduced by reducing dimension of the image data. There exist multiple methods that can be used for dimensionality reduction. Some common template based algorithms are reviewed below.

2.1: Review of some Recognition Algorithms .

In this section, the recognition algorithms classified into template-based (ICA, SVD, PCA, LDA, RP, NMF) and geometric-based (EBGM) are discussed.

2.1.1: Independent Component Analysis (ICA).

A basic problem in face recognition research, is finding a suitable representation of the large multivariate data or random vectors. For computational and conceptual simplicity, a linear transformation of the original data is required. Independent Component Analysis (ICA) is one of such linear transformations that seeks to separate the original data set into independent components that project the distinct features of the original data.

According to Gonzales and Wintz (1987), as cited in Hyv¨arinen, Karhunen and Oja (2001), a fundamental problem in digital signal processing is to find suitable representations for image, audio or other kind of data for tasks like compression and de-noising. Data representations are mostly based on linear transformations. It is always useful to estimate the linear transformation from the data itself, in which case the transform could be ideally adapted to the kind of data being processed.

ICA was originally developed to deal with problems that are closely related to Blind Source Separation (BSS).

Consider the voice of two people in a room simultaneously from two microphones. The microphones give two recorded time signals which could be denoted by $x_1(t)$ and $x_2(t)$, with x_1 and x_2 amplitudes and t time index. Each of these recorded signals is a weighted sum of speech signals emitted by the two speakers, denoted by $s_1(t)$ and $s_2(t)$. This is linearly expressed as

$$x_{1i}(t) = \sum_{i=1}^{2} a_{1i} s_i, \qquad (2.0a)$$

$$x_{2i}(t) = \sum_{i=1}^{2} a_{2i} s_i. \qquad (2.0b)$$

where a_{1i}, a_{2i}, $i = 1,2$ are parameters that depend on the distances of the microphones from the speakers. The problem now lies in estimating the two original speech signals $s_1(t)$ and $s_2(t)$ using only the recorded signals $x_1(t)$ and $x_2(t)$ (Hyva¨rinen *et al.,* 2001). The problem described above is known as the cocktail party problem.

The problem of source separation is an inductive inference problem. There is not enough information to deduce the solution, so one must use any available information to infer the most probable solution. The aim is to process these observations in such a way that the original source signals are extracted by the adaptive system (Naik & Kumar, 2011). In BSS, the word blind refers to the fact that, how the signals or images were mixed or generated is unknown. As such, the separation is in principle impossible. Allowing some relatively indirect and general constraints, however hold the term BSS valid, and separate under these conditions. That is, estimating the original source signals without knowing the parameters of mixing or filtering processes.

It should be noted that, the original source can be estimated up to certain indeterminacies. In mathematical terms, these indeterminacies and ambiguities can be expressed as arbitrary scaling, permutation and delay of estimated source signals

(Naik & Kumar, 2011). These indeterminacies preserve the original sources.

2.1.1.0: Blind Source Separation (BSS)

Considering a case like the Cocktail party problem, that is, many people speaking at the same time in a room, interfering electromagnetic waves from mobile phones or crosstalk from brain waves originating from different areas of the brain. The goal of BSS is to separate individual signals. The objective is to sample a mixture of spoken voices, with a given number of microphones, and then separate each voice into a separate speaker channel sources. The BSS is unsupervised and thought of as a black box method. In this we encounter many problems, e.g. time delay between microphones, echo, amplitude difference, voice order in speaker and under-determined mixture signal.

Jutten and Karhunen (2004), proposed that, in an artificial neural network (ANN) like architecture, the separation could be done by reducing redundancy between signals. This approach led to what is now known as independent component analysis

(ICA). A handful of researchers worked in this area until 1995. Bell and Sejnowski (1995), published a relatively simple approach to the problem named infomax, that many became aware of the potential of ICA. Since then a whole community has evolved around ICA, centralized around some large research groups on independent component analysis and blind signal separation.

2.1.1.1: Definition of Independent Component Analysis (ICA) Independent component analysis (ICA) is one of the most widely used BSS techniques for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA is essentially a method for extracting individual signals from mixtures. Its power resides in the physical assumptions that the different physical processes generate unrelated signals. The simple and generic nature of this assumption allows ICA to be successfully applied in diverse range of research

fields.

The general model for ICA is that, the sources are generated through a linear basis transformation, where additive noise can be present. According to Hyv¨arinen *et al.,* (2001), ICA is defined by using statistical ”latent variables” model. Assume that *n* linear mixtures $x_1, x_2, x_3, \dots, x_n$ of *n* independent components are observed such that,

$$x_j = \sum_{i=1}^{n} a_{ji} s_i, \forall j = 1, 2, 3, \dots, n \tag{2.1}$$

The time index is dropped in ICA model. It is assumed that each of the mixtures x_j as well as each independent component s_i is a random variable, instead of a proper time signal.

A fundamental aspect of the mixing process is that the sensors must be spatially separated (e.g. microphones that are spatially distributed around a room) so that each sensor records a different mixture of the sources (Naik & Kumar, 2011).

Having spatial separation assumption in mind, we can model the mixing process with matrix multiplication as follows:

$$\mathbf{X}_j = \mathbf{A}s_j, \tag{2.2}$$

where, $\mathbf{X}_j = [x_{1j},...,x_{nj}]^T$, $j = 1,...,M$ and $x_{ij}, i = 1,...,n$, $\mathbf{s}_j = [s_{1j},...,s_{nj}]^T$,

$j = 1,...,M$ and $s_{ij}, i = 1,...,n$,

$\mathbf{A} = [a_{1j},...,a_{nj}]^T$, $j = 1,...,M$ and $a_{ij}, i = 1,...,M$.

A is an unknown matrix called the mixing matrix and $\mathbf{X}_j$, $\mathbf{s}_j$ are the two vectors representing the observed signals and source signals respectively. The justification for the description of this ICA technique as blind is that we have no information on the mixing matrix **A**, or even on the sources themselves.

The ICA model is a generative model, which means that, it describes how the observed data are generated by a process of mixing the components $\mathbf{s}_j$. The independent components are latent variables, meaning that they cannot be directly observed. The starting point for ICA is the very simple assumption that the components $\mathbf{s}_j$ are statistically independent.

The general idea in ICA model, is to separate the mixtures, assuming that the original underlying source signals are mutually independently distributed. The objective is to recover the original signals, $\mathbf{s}_j$ from only the observed vector $\mathbf{X}_j$.

Estimates for the sources are obtained by first obtaining the "unmixing matrix" **W**, where, $\mathbf{W} = \mathbf{A}^{-1}$.

This enables an estimate, $\hat{\mathbf{s}}_j$, of the independent sources to be obtained as

$$\hat{\mathbf{s}}_j = \mathbf{W}\mathbf{X}_j. \tag{2.3}$$

The ICA model operation is illustrated in Figure 2.1.

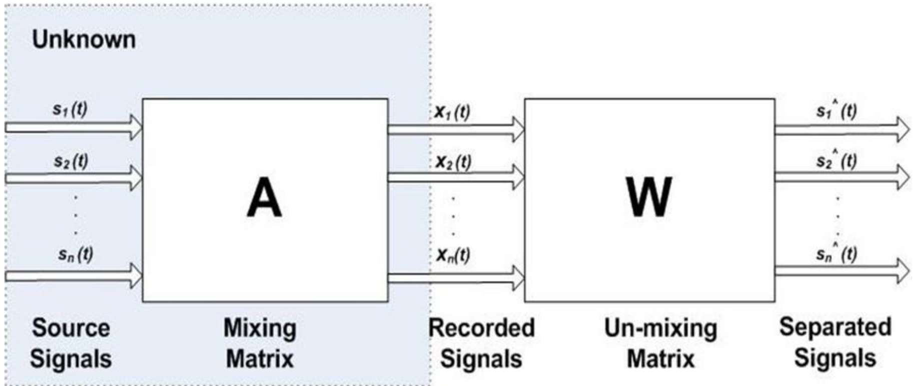


Figure 2.1 : ICA model block diagram
Source: (Naik & Kumar, 2011)

Regarding ICA, the basic framework for most researchers has been to assume that the mixing is instantaneous and linear, as in infomax. That is, performing ICA based on the algorithm developed by Bell and Sejnowski from the point of view of optimal information transfer functions. Barlett, Movellan and Sejnowski (2002), performed ICA on image set under two architectures. These are,

- Treating the images as random variables and the pixels as outcomes.
- Treating the pixels as random variables and the images as outcomes.

The Informax algorithm is motivated as follows;

Consider the image space X as a random variable of $n \times 1$ dimension (representing the distribution of input in the environment).

Let **W** be an $n \times n$ invertible matrix, **U** = **WX** and **Y** = f(**U**) an $n \times 1$ dimensional random variable representing the output of $n - neurons$. Each component of $f = (f_1,...,f_n)$ is an invertible squashing function, mapping real numbers into the [0,1] interval. Typically, the logistic function,

$$f_i(u) = \frac{1}{1+e^{-u}} \quad (2.4)$$

is used. The $U_1,...,U_n$ variables are linear combinations of inputs and can be interpreted as pre-synaptic activations of $n - neurons$.

The variables $Y_1,...,Y_n$ can be interpreted as post-synaptic activation rates and are bounded by the interval [0,1]. The goal in Bell and Sejnowski's algorithm is to maximize the mutual information between the environment **X** and the output of the neural network **Y**.

2.1.1.2: Ambiguities of Independent Component Analysis

There are two ambiguities of Independent Component Analysis. These are discussed below.

- The variances of the independent components cannot be determined; since the mixing matrix **A** and the independent components **s** are unknown, any scalar multiplier in one of the source $\mathbf{s}_j$ could be cancelled by dividing the corresponding column a_j of **A** by the same scalar. Consequently the magnitudes of the independent components is fixed as they are random variables. This is mostly done by assuming that each of the independent components has a unit variance, $E(\mathbf{s}^2_j) = 1$.

- The order of the independent components cannot be determined; since the mixing matrix **A** and the independent components **s** are unknown, the order of the independent components can always be changed with a permutation matrix **P**.

2.1.1.3: Definition of Independence

The foundation of independent component analysis is constituted on statistical independence. Consider two random variables s_1 and s_2, having a joint probability distribution $p(s_1,s_2)$. The two random variables s_1 and s_2, are said to be independent if $p(s_1,s_2) = p(s_1)p(s_2)$, where $p(s_1),p(s_2)$ are the respective marginal distributions of s_1 and s_2.

Generally, for $s_j, j = 1,2,...,M$ independent components;

$$p(s_1, s_2, \ldots, s_M) = \prod_{j=1}^{M} p(s_j). \quad (2.5)$$

2.1.1.4: Independence and Uncorrelated Variables

The two random variables s_1 and s_2, are said to be uncorrelated if their covariance is zero. That is;

$$Cov(s_1,s_2) = E(s_1s_2) - E(s_1)E(s_2) = 0. \quad (2.6)$$

Independence suffices uncorrelatedness, but uncorrelatedness does not imply independence. In ICA, the estimation process always gives uncorrelated estimates, $\hat{\mathbf{s}}$ of the independent components, $\mathbf{s}_j$.

2.1.1.5: Restrictions of ICA

Independent Component Analysis is restricted by the following conditions;

- The random variables s_j of unit variance are statistically independent. That is, $p(s_1, s_2, \ldots, s_M) = \prod_{j=1}^{M} p(s_j)$.
- Non-Gaussianity: In ICA, the independent components must be non-gaussian. This restriction makes it possible to choose the mixing matrix **A**. Now consider s_j to be Gaussian or normally distributed with some orthogonal mixing matrix **A**. This indicates that the source mixture x_j are also normally distributed with mean zero and unit variance, $N(0,1)$. The joint density function is given by;

$$
\begin{aligned}
p(x_1, x_2, \ldots, x_M) &= \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi}} exp(\frac{x_j^2}{2}), \\
&= (\frac{1}{\sqrt{2\pi}})^M exp(\sum_{j=1}^{M} \frac{x_j^2}{2}).
\end{aligned}
\tag{2.7}
$$

The case of two mixtures x_1 and x_2 following the Gaussian distribution is illustrated in *Figure 2.2*.

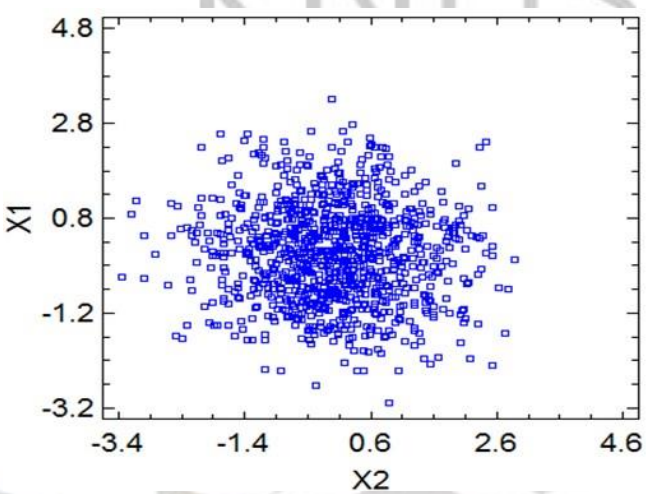


Figure 2.2: Plot of two normally distributed mixtures x_1 and x_2.

Figure 2.2 shows that the joint density of x_1 and x_2 is symmetric and does not contain any information on the direction of the columns of the mixing matrix **A**. This clearly explains why the mixing matrix **A** can not be estimated for Gaussian random variables.

2.1.1.6: Non-Gaussianity and Independence

By the central limit theorem, the distribution of a sum of independent signals with arbitrary distributions tends toward a Gaussian distribution under certain conditions (Naik & Kumar, 2011). The sum of two independent identically distributed signals usually has a distribution that is closer to Gaussian than distribution of the individual original signals.

2.1.1.7: Measure of Non-Gaussianity

Non-Gaussianity is an important and essential principle in ICA estimation. To use non-gaussianity in ICA estimation, there is the need to have a quantitative measure of non-gaussianity of a signal. Before using any measure of non-gaussianity, the signals should be normalised (eg. mean centered). Some of the commonly used measures are kurtosis, skewness and negentropy.

2.1.1.8: Kurtosis

Kurtosis is the classical method of measuring Non-Gaussianity.

$$kurt(X) = E(X^4) - 3\{E(X^2)\}^2. \tag{2.8}$$

When data is preprocessed to have unit variance, then $E(X^2) = 1$, and the kurtosis equation reduces to $kurt(X) = E(X^4) - 3$.

This shows that, kurtosis is equal to the fourth moment of the data. For a Gaussian data, the fourth moment, $E(X^4) = 3\{E(X^2)\}^2$. This gives a kurtosis value of zero for a Gaussian random variable. Kurtosis value can be negative or positive. Sub-Gaussian distributions have negative kurtosis value and Super-Gaussian distributions have positive kurtosis value.

According to Huber (1985), as cited in Hyva¨rinen, Karhunen and Oja (2001), the main problem of Kurtosis is that, it can be sensitive to outliers. Its value may depend on only few observations in the tails of the distribution.

2.1.1.9: Negentropy

The entropy of a random variable can be interpreted as the degree of information that the observed variables give. The more random or unpredictable and unstructured the variable, the larger its entropy (Hyv¨arinen & Oja, 2000). Negentropy is an information theory measure of the "distance" between a given random variable's probability density function (PDF) and the PDF of a Gaussian random variable (Brandt, 2010).

Entropy of a random variable, can be seen as the coding length of the random variable (Papoulis, 1991). The discrete case of entropy is represented mathematically as;

$$H(X) = -\sum_{\forall i} P(X = a_i) \log P(X = a_i). \quad (2.9a)$$

For continuous random variables, entropy is represented as;

$$H(X) = -\int f(x) \log f(x) dx. \quad (2.9b)$$

The Gaussian variable has the largest entropy among all random variables of equal variance (Naik & Kumar, 2011). This is to say, the Gaussian random variable is the most random or least structured of all the distribution. A useful measure of non-gaussianity of a random variable is measuring it against the Gaussian distribution. That is;

$$E(X) = H(X_{gaussian}) - H(X), \quad (2.10)$$

where $H(X_{gaussian})$ is the entropy of a Gaussian random variable and $H(X)$ is the entropy of any random variable from a different distribution.

2.1.2: Linear Discriminant Analysis (LDA)

The Linear Discriminant Analysis was invented by Sir R. A. Fisher, who successfully used it for classifying flowers in his 1936 paper, " The use of multiple measurements in taxonomic problems". Linear Discriminant Analysis (LDA) is a well-known scheme for feature extraction and dimension reduction. It has been used in many applications involving high-dimensional data, such as face recognition. Linear Discriminant Analysis (LDA) purposely plays a role in face recognition by reducing dimensions of input data using matrix representation. LDA is a classical method for feature extraction and dimensionality reduction which has been widely used in several classification problems.

The objective of LDA is to find out the optimal transformation matrix so that the ratio of between class scatter matrix and within class scatter matrix reaches its maximum (Neeta *et al.*, 2008). But why do we need another dimensionality reduction method, if the Principal Component Analysis (PCA) did such a good job?

The PCA finds a linear combination of features that maximizes the total variance in data. While this is clearly a powerful way to represent data, it does not consider any classes and so a lot of discriminative information may be lost when throwing components away. Imagine a situation where the variance is generated by an external source, let it be the light. The components identified by a PCA do not necessarily contain any discriminative information at all, so the projected samples are smeared together and a classification becomes impossible. In order to find the combination of features that separates best between classes, the Linear Discriminant Analysis maximizes the ratio of between-classes to within-classes scatter. Same classes cluster tightly together, whereas different classes separate as possible from each other.

However, in face recognition problems a critical issue using LDA is the small sample size (SSS) problem. This problem arrives when there are small number of training samples but the dimension of the feature space is large. This means that the within class scatter matrix would tend to be a singular matrix and so the algorithm fails. To overcome this problem, the enhancements in classical LDA was proposed. Among them, the most popular one is the use of PCA as a pre-processing step and then performing LDA so that dimensionality reduction occur during PCA phase. Bhat and Bhavikatti (2012), proposed PCA Combined with Fisher LDA for facial expression recognition. The LDA/QR method was also presented to overcome the

SSS problem by performing QR decomposition. Due to QR decomposition, dimensions of scatter matrices reduce because of orthonormal matrix produced by decomposition.

According to Neeta *et al.,* (2008), the class separability criteria proposed by classical LDA does not maximize the classification accuracy but it resulted in preserving the distance of previously well separated classes and an overlapping of neighbour classes also occurs. To overcome this problem an extended scheme is also proposed by applying weighting function in the estimation of scatter matrices (Loog, Duin, & Hach-Umbach, 2001).

2.1.2.1: Classical LDA

LDA is a technique for finding a set of projecting vector $\mathbf{W}_{LDA}$ that best discriminates different classes. The within-class scatter matrix $\mathbf{S}_w$ and the between-class scatter matrix $\mathbf{S}_b$ are defined as follows:

$$\mathbf{S}_w = \frac{1}{n}\sum_{j=1}^{c}\sum_{x\in c_j}(x-\bar{x}_j)(x-\bar{x}_j)^T, \tag{2.11}$$

$$\mathbf{S}_b = \frac{1}{n}\sum_{j=1}^{c} n_i(\bar{x}_j-\bar{x})(\bar{x}_j-\bar{x})^T, \tag{2.12}$$

where n_i denotes the sample size of class $c_j (j = 1,2,...,c)$ and n denotes the total sample size. $\bar{x}_j$ denotes the average of samples in c^{th}_j class, $\bar{x}$ denotes the average of all training samples.

One way to find the transformation matrix $\mathbf{W}_{LDA}$ is to use Fisher's criterion. It can be achieved by maximizing the ratio below;

$$J(\mathbf{W}_{LDA}) = \frac{\mathbf{W}_{LDA}^T \mathbf{S}_b \mathbf{W}_{LDA}}{\mathbf{W}_{LDA}^T \mathbf{S}_w \mathbf{W}_{LDA}}. \tag{2.13}$$

$\mathbf{W}_{LDA}$ can be constructed by the set of largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The maximum value of discriminative space is $c - 1$, where c denote the number of classes, since the rank of $\mathbf{S}_b$ is $c - 1$.

2.1.3: Random Projections (RP)

Random projection is also a powerful method for dimensionality reduction. Theoretical results indicate that the method preserves distances quite nicely.

Random projection is used as a dimensionality reduction tool in a number of cases, where the high dimensionality of the data would otherwise lead to burden computations. The RP is also an alternative method for projecting high-dimensional data onto a random lower-dimensional subspace.

In many applications of data mining, the high dimensionality of the data restricts the choice of data processing methods. Such application areas include the analysis of market basket data, text documents, image data and so on.

A statistically optimal way of dimensionality reduction is to project the data onto a lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. The best (in mean-square sense) and most widely used way to do this is principal component analysis (PCA) (Bingham & Mannila, 2001).

According to Bingham and Mannila, (2001), the RP is a computationally simple method of dimensionality reduction that does not introduce significant distortion in the data set.

In random projection (RP), the original high-dimensional data is projected onto a lower-dimensional subspace using a random matrix whose columns have unit lengths. RP has been found to be a computationally efficient, yet sufficiently accurate method for dimensionality reduction of high-dimensional data sets. While this method has attracted lots of interest, empirical results are sparse.

2.1.3.1: The Mathematics of Random Projection

In random projection, the original $d-dimensional$ data is projected to a $k-dimensional$ ($k << d$) subspace through the origin, using a random $k \times d$ matrix $\mathbf{R}$ whose columns have unit lengths. Using matrix notation where $\mathbf{X}_{d\times N}$ is the original set of N $d-dimensional$ observations,

$$\mathbf{X}_{RPk\times N} = \mathbf{R}_{k\times d}\mathbf{X}_{d\times N}. \quad (2.14)$$

$\mathbf{X}^{RP}{}_{k\times N}$ is the projection of the data onto a lower-dimensional subspace.

The preservative nature of this random mapping arises from the Johnson-Lindenstrauss lemma; if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved.

A linear mapping such as (2.14) can cause significant distortions in the data set if $\mathbf{R}$ is not orthogonal. Orthogonalizing $\mathbf{R}$ is unfortunately computationally expensive. This bring into mind the Hecht-Neilsen results; in a high-dimensional space, there exist a much larger number of almost orthogonal than orthogonal directions (Bingham & Mannila, 2001). Whence vectors having random directions might be sufficiently close to orthogonal, and equivalently $\mathbf{R}^T\mathbf{R}$ would approximate to the identity matrix.

It is instructive to see how the similarity of two vectors is distorted in dimensionality reduction when comparing the performance of random projection to other dimensionality reduction algorithms. The measure of similarity of data vectors can be done either by their Euclidean distance or by their inner product. In image data problems, the Euclidean distance is mostly used as a measure of similarity. Text documents, on the other hand, are generally compared according to the cosine of the angle between the document vectors; if document vectors are normalized to unit length, this corresponds to the inner product of the document vectors. The Euclidean distance, a, between two data vectors $\mathbf{x}_a$ and $\mathbf{x}_b$ in the original largedimensional space is given by

$$\epsilon_a = ||\mathbf{x}_a - \mathbf{x}_b|| \quad (2.15)$$

After the random projection (RP), this original distance is approximated by the scaled Euclidean distance, sa of these vectors in the reduced space as;

$$\epsilon_{sa} = \sqrt{\frac{d}{k}}||\mathbf{R}\mathbf{x}_a - \mathbf{R}\mathbf{x}_b|| \tag{2.16}$$

where d is the original and k the reduced dimensionality of the data set. The scaling term $\sqrt{\frac{d}{k}}$, takes care of the decrease in the dimensionality of the data. According to Johnson-Lindenstrauss lemma, the expected norm of a projection of a unit vector onto a random subspace through the origin is $\sqrt{\frac{k}{d}}$.

Now the next question of interest is the choice of a random matrix **R**. The r_{ij} element of **R** follows often the Normal or Gaussian distribution;

$$f_g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}} \tag{2.17}$$

A much simpler distribution is given as;

$$r_{ij} = \sqrt{3} \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -1 & \text{with probability } \frac{1}{6} \end{cases} \tag{2.18}$$

Practically all zero mean, unit variance distributions of r_{ij} would give a mapping that still satisfies the Johnson- Lindenstrauss lemma.

2.1.4: Elastic Bunch Graph Matching (EBGM)

All the above algorithms (template-based) have been used with high level of precision subjected to the various constraints. Differences in pose, lightening conditions and expressions, as well as partial face obstructions are only a few of the problems that such an algorithm would have to cope with. Geometric Featurebased approaches, on the other hand, rely on information about well-defined facial characteristics and the image area around these points to represent a face in the problem space and perform recognition (Stergious, 2003). Examples of these facial features are the eyes, nose, mouth, eyebrows etc. The exact coordinates of the eyes in particular are ideally given, although in practice the algorithm can only work with estimates obtained from a face detection and eye zone locator module that precedes the recognition process.

Elastic Bunch Graph Matching (EBGM) is a Geometric feature-based face identification method.

The algorithm assumes that the positions of certain fiducial points on the faces are known and stores information about the faces by convolving the images around the fiducial points with 2D Gabor wavelets of varying size (Stergious, 2003). According to Stergious (2003), the results of all convolutions form the Gabor jet for that fiducial point.

EBGM treats all images as graphs (called Face Graphs), with each jet forming a node. The training images are all stacked in a structure called the Face Bunch Graph (FBG), which is the model used for identification.

For each test image, the first step is to estimate the position of fiducial points on the face based on the known positions of fiducial points in the FBG. Afterwards, jets are extracted from the estimated points and the resulting Face Graph is compared against all training images in the FBG, using Gabor jet similarity measures to decide the identity of the person in the test image.

Gabor wavelets are fundamental to the EBGM algorithm and a background in wavelet analysis is necessary in order to understand the method's intricacies.

Wavelets are used (much like Fourier transforms) to analyse frequency space properties of an image, the difference being that wavelets operate on a localized image patch, while the Fourier transform affects the whole image (Stergious, 2003). Figure 2.3 is the flowchart of the EBGM algorithm;

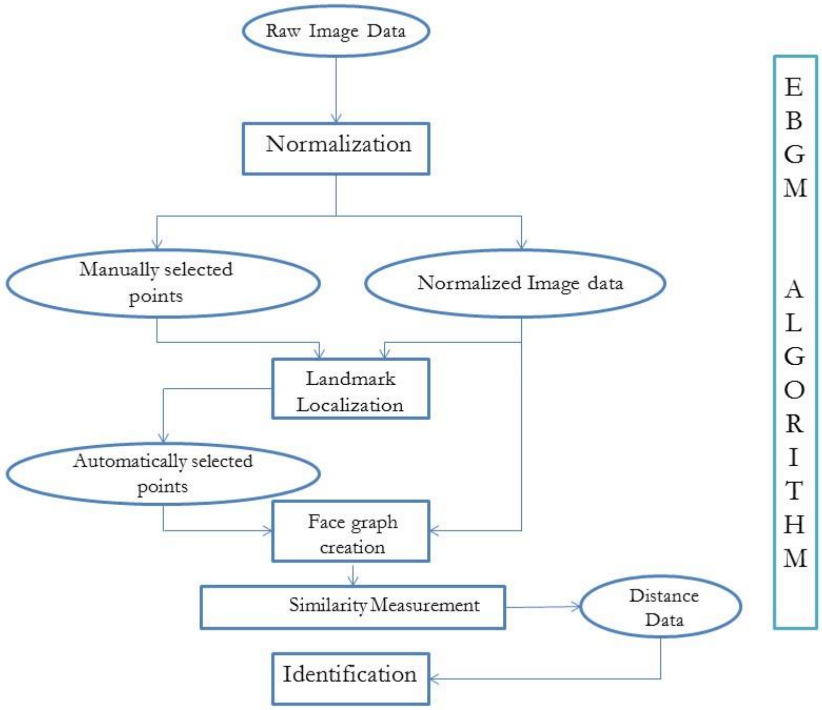


Figure 2.3: Flowchart of the Elastic Bunch Graph Matching Algorithm (Stergious, 2003).

The goal of Elastic graph matching is to find the fiducial points on a query image and thus to extract from the image a graph which maximizes the graph similarity function.

2.2: Summary

We have discussed some template-based algorithms (ICA, LDA and RP) and a geometric-based algorithm (EBGM) in this chapter. The gaps in the implementation of these algorithms for face image recognition have also been revealed with possible modifications to overcome their setbacks. All the template-based algorithms were described as dimensionality reduction algorithms which seek to reduce high dimensional image data to lower dimensions but preserving the components that carry important information about the observed images.

CHAPTER THREE

Materials and Methods

3.0: Introduction

This chapter presents a detailed review of Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). The recognition procedure comprises the pre-processing stage, feature extraction stage and categorization (classifying an input image as belonging to a face class or not) stage.

3.1: Data Acquisition

A real time face image database was created for the purpose of benchmarking the face recognition system. The image database is divided into two subsets, for separate training and testing purposes.

Two hundred and ninety four frontal facial images from 42 randomly selected individuals were acquired from Cohn Kanade, Japanese Female Facial Expressions database (JAFFE) at labeled faces in the wild and some local Ghanaian students facial database.

Of Two hundred and ninety four images, one hundred and eighty two facial expressions along the seven universally accepted principal emotions (Neutral, Angry, Happy, Fear, Disgust, Sad, Surprise) from 26 individuals were collected from the Cohn-Kanade AU-Coded Facial Expression database. Subjects in the available portion of the database were 26 university students enrolled in introductory psychology classes. They ranged in age from 18 to 30 years.

Forty two images from 6 randomly selected individuals were also from the Local Ghanaian database. In the creation of the database, the observation room was equipped with a chair for the subject and one canon camera. Only image data from the frontal camera were captured. Subjects were instructed by an experimenter to perform a series of 7 facial displays that included single action units.

Subjects began and ended each display from a neutral face. Before performing each display, an experimenter described and modelled the desired display. Six of the displays were based on descriptions of prototypic basic emotions (i.e., Happy, Surprise, Anger, Fear, Disgust, and Sadness). Image sequences from neutral to target display were digitized into 256 by 256 or with 8-bit precision for grayscale values. Seventy frontal face images from 10 randomly selected individuals were also collected from Japanese Female Facial Expressions database (JAFFE) along the principal emotional constraints.

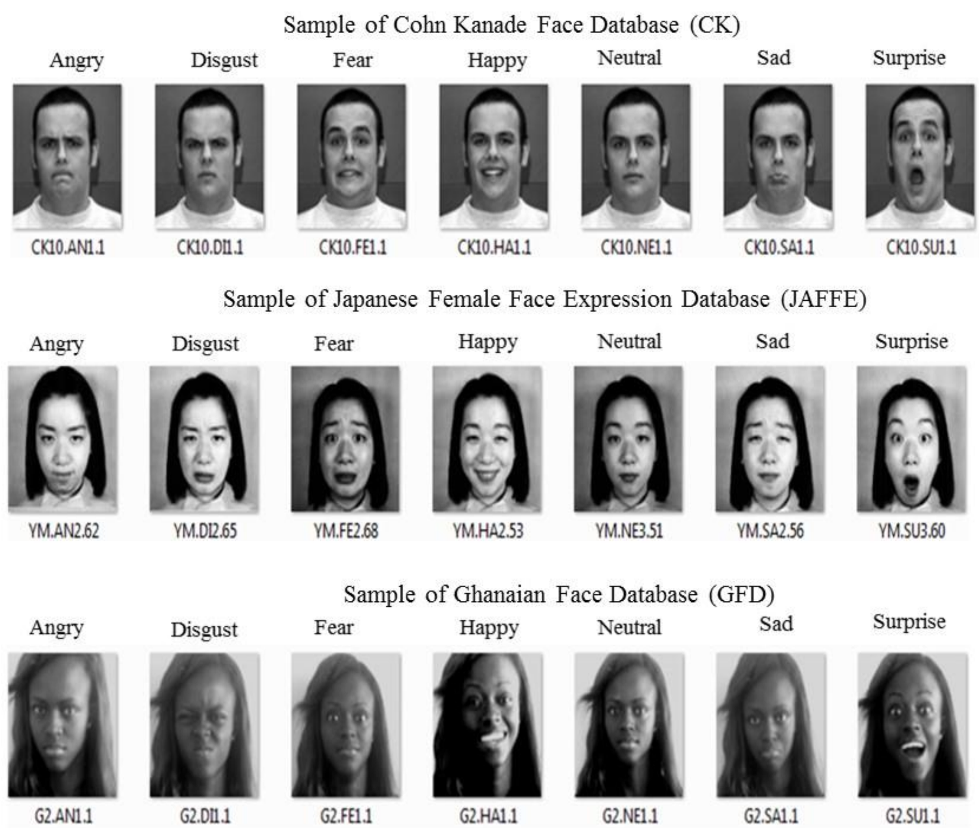


Plate 3.1: Sample of Research Database

Forty two of the face images representing the neutral poses of 42 individuals are captured into the train image database. These images are to be trained on the various recognition algorithms and their knowledge kept in memory for recognition.

Two hundred and Fifty two images representing the variable (Angry, Disgust, Fear, Happy, Sad and Surprise) poses of 42 individuals were captured into the test image database. These images are called the test images and used for testing the recognition algorithms. See *appendix 1.0* for entire study database.

3.2: Preprocessing of Frontal Face image.

Before applying any template-based algorithm on the image data to be trained, it is useful to do some preprocessing. In this work, preprocessing is basically, Mean Centering, Whitening and Fast Fourier Transform. This is to help reduce the noise level and make the estimation process simpler and better conditioned. As an illustration of preprocessing, *Plate 3.2* shows six images selected from Japanese Female Face Expression database (JAFFE) in labeled faces in the wild.

Plate 3.2: Six images from JAFFE

Define the image matrix, $\mathbf{M}_j$ as;

$$\mathbf{M}_j = (m_{jik})\ ;\ i,k = 1,2,...,p;\ j = 1,2,...,n,$$

$$= (\mathbf{m}_{j1},\mathbf{m}_{j2},...,\mathbf{m}_{jp}),\text{ where, } \mathbf{m}_{jk} = (m_{j1k},m_{j2k},...,m_{jpk})^T,$$

$$\mathbf{X}^j = (\mathbf{m}_{j1}^T, \mathbf{m}_{j2}^T, \ldots, \mathbf{m}_{jp}^T)^T, \quad (3.0)$$

where, p = the order of the image matrix,

n = the number of images to be trained.

From equation (3.0), suppose $\mathbf{X}_j$ is a column vector of dimension N given by;

$$\mathbf{X}_j = (X_{ji})_{N\times 1}. \qquad (3.1)$$

where X_{ji} replaces the m_{jik} position-wise. The preprocessing steps are based on the sample $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ whose elements are the vectorised form of the individual images in the study.

3.2.0: Centering

This is a simple preprocessing step, executed by subtracting the mean, $\bar{\mathbf{m}}_j = E(\mathbf{X}_j)$ of the data ($\mathbf{X}_j, j = 1,2,\ldots,n$), from the data.

$$\begin{aligned}\bar{\mathbf{m}}_j &= \frac{1}{N}\sum_{i=1}^{N} X_{ji},\\ &= \frac{1}{N}\sum_{i=1}^{p}\sum_{k=1}^{p} \mathbf{m}_{jik},\ (j = 1,2,3\ldots,n)\end{aligned} \qquad (3.2)$$

where $N = (p \times p)$, length (= rows of image × columns of image) of the image data, $\mathbf{M}_j$.

Define $\bar{\mathbf{X}}_j$ as a constant vector of order $(p \times p)$ with all elements same as $\bar{\mathbf{m}}_j$, $(j = 1,2,\ldots,n)$.

The centered mean is denoted by, $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_n)$.

$$\mathbf{w}_j = \mathbf{X}_j - \bar{\mathbf{X}}_j. \qquad (3.3)$$

Applying (3.3) to the images in *Plate 3.2*, generate the mean centered images as shown in *Plate 3.3*.

Plate 3.3 : Six mean centered images from JAFFE

3.2.1: Whitening

Whitening is a preprocessing technique that is used to remove noise factors in the observed image data, **X** so as to obtain a new image, $\mathbf{X}^{\sim}$ with uncorrelated components but equal unit variance. This is to say, the covariance of $\mathbf{X}^{\sim}$ is an identity matrix, *I*. A simple way to whiten images is to find the eigenvectors and eigenvalues of the observed images through eigenvalue decomposition (for symmetric image matrix) or singular value decomposition (for asymmetric image matrix) of the covariance matrix. Suppose the covariance matrix, **C** is given by;

$$\mathbf{C} = \frac{1}{n}\mathbf{W}^T\mathbf{W}. \tag{3.4}$$

Define matrix $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_n)$ where $\mathbf{u}_j, j = 1,2,\ldots n$ is the j^{th} eigenvector of the covariance matrix **C**. Let **D** (dimension $n \times n$) be the diagonal matrix whose entries $(\lambda_{jj}, j = 1,2,\ldots,n)$ are the eigenvalues corresponding to the eigenvectors $\mathbf{u}_j, j = 1,2,\ldots,n$.

The whitened images $\mathbf{X}^{\sim}$ are given by;

$$\vec{\mathbf{X}} = \mathbf{U}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{X}^T. \tag{3.5}$$

The covariance matrix $\mathbf{C}^{\sim}$, of $\mathbf{X}^{\sim}$ is given by;

$$\vec{\mathbf{C}} = \frac{1}{n}\vec{\mathbf{X}}^T\vec{\mathbf{X}} = \mathbf{I}. \tag{3.6}$$

This can simply be expressed as, $E(\vec{\mathbf{X}}\vec{\mathbf{X}}^T) = \mathbf{I}$.

Plate 3.4 shows the whitened outcome of the six images shown in *Plate 3.2*.

Plate 3.4: Whitened images

The whitened matrix $\mathbf{X}^{\sim}$, built from the eigenvalue decomposition of the covariance matrix **C** of the zero-mean observation, $\mathbf{w}_j$, creates a set of uncorrelated unit image variables.

3.2.2: Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is used as a noise reduction mechanism during image pre-processing. The FFT is a computationally efficient algorithm used to compute the Discrete Fourier Transform (DFT) and its inverse (IDFT). The FFT algorithm reduces the computational burden to $O(NlogN)$ arithmetic operations (Glynn, 2007).

The first stage in the execution of FFT during image preprocessing is to compute the Discrete Fourier Transform. The DFT of a of column vector, $\mathbf{m}_{jk}$ is represented mathematically as;

$$m^*_{jsk} = DFT\{\mathbf{m}_{jk}\} = \sum_{r=0}^{p-1} \mathbf{m}_{jk} e^{-i(2\pi sr/p)}, \quad (3.7)$$

$s = 0,1,...,p-1$, $j = 1,2,...,n$ and $i = \sqrt{-1}$, where, $\mathbf{m}_{jk}$ is the k^{th} column of the image matrix, $\mathbf{M}_j$. For an image matrix of order 4, $p = 4$ and $s = 0,1,2,3$. The DFT becomes;

$$m_{*j0k} = m_{j0k}e^{-0\cdot i\pi/2} + m_{j1k}e^{-0\cdot i\pi/2} + m_{j2k}e^{-0\cdot i\pi/2} + m_{j3k}e^{-0\cdot i\pi/2},$$

$$m_{*j1k} = m_{j0k}e^{-0\cdot i\pi/2} + m_{j1k}e^{-1\cdot i\pi/2} + m_{j2k}e^{-2\cdot i\pi/2} + m_{j3k}e^{-3\cdot i\pi/2},$$

$$m_{*j2k} = m_{j0k}e^{-0\cdot i\pi/2} + m_{j1k}e^{-2\cdot i\pi/2} + m_{j2k}e^{-4\cdot i\pi/2} + m_{j3k}e^{-6\cdot i\pi/2},$$

$_{j3k}$ $_{j0k}$ $_{j1k}$ $_{j2k}$ $_{j3k}$

$e^{-0\cdot i\pi/2}$ $e^{-0\cdot i\pi/2}$ $e^{-0\cdot i\pi/2}$ m_{j0k}

m_{*j0k} $e^{-0\cdot i\pi/2}e^{-1\cdot i\pi/2}$

$e^{-2\cdot i\pi/2}$ $e^{-2\cdot i\pi/2}$ $e^{-3\cdot i\pi/2}$ m_{j1k}

m_{*j1k} $e^{-0\cdot i\pi/2}$ $e^{-3\cdot i\pi/2}$,

$$\cdots\; e^{-4\cdot i\pi/2} \quad e^{-6\cdot i\pi/2} \; m_{j2k}$$

$$\cdots\; e^{-6\cdot i\pi/2} \quad e^{-9\cdot i\pi/2} \; m_{j3k}$$

$$m^*_{j3k} = m_{j0k}e^{-0\cdot i\pi/2} + m_{j1k}e^{-3\cdot i\pi/2} + m_{j2k}e^{-6\cdot i\pi/2} + m_{j3k}e^{-9\cdot i\pi/2}.$$

$$\begin{bmatrix} m^*_{j0k} \\ m^*_{j1k} \\ m^*_{j2k} \\ m^*_{j3k} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} m_{j0k} \\ m_{j1k} \\ m_{j2k} \\ m_{j3k} \end{bmatrix}.$$

The next stage is to compute the Inverse Discrete Fourier Transform(IDFT). The Inverse Discrete Fourier Transform (IDFT) is given by;

$$m_{jk} = IDFT\{\mathbf{m}^*_{jk}\} = \frac{1}{p}\sum_{r=0}^{p-1}\mathbf{m}^*_{jk}e^{i(2\pi sr/p)}, \qquad (3.8)$$

$s = 0,1,...,p-1$, $j = 1,2,...,n$ and $i = \sqrt{-1}$.

For $p = 4$, the IDFT is given by

$$\begin{bmatrix} m_{j0k} \\ m_{j1k} \\ m_{j2k} \\ m_{j3k} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} e^{0\cdot i\pi/2} & e^{0\cdot i\pi/2} & e^{0\cdot i\pi/2} & e^{0\cdot i\pi/2} \\ e^{0\cdot i\pi/2} & e^{1\cdot i\pi/2} & e^{2\cdot i\pi/2} & e^{3\cdot i\pi/2} \\ e^{0\cdot i\pi/2} & e^{2\cdot i\pi/2} & e^{4\cdot i\pi/2} & e^{6\cdot i\pi/2} \\ e^{0\cdot i\pi/2} & e^{3\cdot i\pi/2} & e^{6\cdot i\pi/2} & e^{9\cdot i\pi/2} \end{bmatrix}\begin{bmatrix} m^*_{j0k} \\ m^*_{j1k} \\ m^*_{j2k} \\ m^*_{j3k} \end{bmatrix},$$

$$\begin{bmatrix} m_{j0k} \\ m_{j1k} \\ m_{j2k} \\ m_{j3k} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & 1 & -i \\ 1 & 1 & -1 & 1 \\ 1 & -i & 1 & i \end{bmatrix} \begin{bmatrix} m^*_{j0k} \\ m^*_{j1k} \\ m^*_{j2k} \\ m^*_{j3k} \end{bmatrix}$$

The real components of the transformed images are extracted for the feature extraction stage whereas the imaginary components are discarded as noise. *Plate 3.5* shows the FFT preprocessed images of the six images shown in *Plate 3.2*.

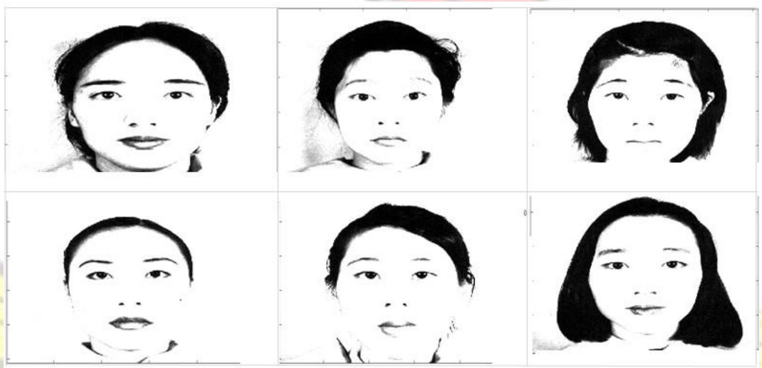


Plate 3.5 : Six FFT preprocessed images from JAFFE

3.3: Singular Value Decomposition(SVD)

Singular value decomposition (SVD) can be looked at from three mutually compatible points of view. On one hand, it can be seen as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation. This ties in to the third way of viewing SVD, which is that, once the most variation has been identified, it is possible to find the best approximation of the original data points using fewer dimensions (Baker, 2005). Hence, SVD can be seen as a method for data reduction or dimensionality reduction.

Deerwester, Dumais, Furnas and Landauer (1990), examined the dimensionality reduction problem in the context of information retrieval. Deerwester *et al.,* were trying to compare documents using the words they contained, and proposed the idea of creating features representing multiple words and then comparing those. The SVD is related to the familiar theory of diagonalizing a symmetric matrix.

If **A** is a symmetric real $n \times n$ matrix, then there is an orthogonal matrix **V** and a diagonal matrix **D** such that $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T$. The columns of **V** are seen here as the eigenvectors for **A** and they form an orthonormal basis for R^n.

The diagonal entries of **D** are the corresponding eigenvalues of **A**. The $\mathbf{V}\mathbf{D}\mathbf{V}^T$ is referred to as eigenvalue decomposition (EVD) for **A** (Kalman, 2002; Gentle, 1998).

Consider an arbitrary real $m \times n$ matrix **A**, then there are orthogonal matrices **U** and **V** and a diagonal matrix **Σ**, such that, $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where **U** is an $m \times n$ matrix, **V** is an $n \times n$ matrix and **Σ** is an $n \times n$ diagonal matrix with diagonal entries $\sigma_{ii} \geq 0$, $\forall i = 1,2,\ldots,n$ and $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{nn}$. The columns of **U** and **V** are called the singular vectors corresponding to the positive values (singular values) in the diagonal matrix **Σ**.

The analogy between the EVD for symmetric matrix and SVD for arbitrary matrix can be extended by thinking of matrices as linear transformations. For a symmetric matrix **A**, the transformation takes R^n to itself, and the columns of **V** define a basis. When vectors are expressed relative to the basis, the transformation simply dilates some components and contracts others according to the magnitude of the eigenvalues.

In the case of SVD for $m \times n$ matrix **A**, the transformation takes R^n to a different space R^m. The columns of **V** and **U** provides the basis for **A**. When these are used to represent vectors in the domain and range of transformation, the transformation simply dilates and contracts some components according to the magnitude of the singular values and possibly discard values and appends zeros as needed to account for a change in dimension. It is therefore clear that SVD tells how to choose orthonormal bases so that the transformation is represented by a matrix with the simplest possible form.

There is no difficulty in obtaining a diagonal representation. We need only the matrix $\sigma_i\mathbf{u}_i = \mathbf{A}\mathbf{v}_i$, which is easily arranged. So the challenge rests in finding the bases $\{v_1,v_2,...,v_n\}$ and $\{u_1,u_2,...,u_m\}$.

An orthonormal basis $\{v_1,v_2,...,v_n\}$ is selected for R^n so that the first k elements span the row space of $\mathbf{A}$ and the remaining $n - k$ elements span the null space of $\mathbf{A}$; where k is the rank of $\mathbf{A}$: Then for $1 \leq i \leq k$, define $\mathbf{u}_i$ to be a unit vector parallel to $\mathbf{A}\mathbf{v}_i$, and this is extended to a basis for R^m.

Suppose matrix $\mathbf{A}$ is symmetric, then $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{V}\mathbf{D}\mathbf{V}^T$; the eigenvalue decomposition of the $n \times n$ matrix $\mathbf{A}$.

The diagonal entries λ_i of $\mathbf{D}$ arranged in non-increasing order. Let the column of $\mathbf{V}$ (eigenvectors of $\mathbf{A}^T\mathbf{A}$) be the orthonormal basis $\{v_1,v_2,...,v_n\}$. Now,

$$\mathbf{A}\mathbf{v}_i\mathbf{A}\mathbf{v}_j = (\mathbf{A}\mathbf{v}_i)^T(\mathbf{A}\mathbf{v}_j) \text{ for } i \; 6{=} \; j,$$

$$= \mathbf{v}_{iT}\mathbf{A}_T\mathbf{A}\mathbf{v}_j,$$

$$= \mathbf{v}_i^T\lambda_j\mathbf{v}_j, \text{ where } \mathbf{A}^T\mathbf{A} = \lambda_j$$

$$= \lambda_j\mathbf{v}_i^T\mathbf{v}_j. \qquad (3.9)$$

So clearly the image $\{\mathbf{A}\mathbf{v}_1,\mathbf{A}\mathbf{v}_2,...,\mathbf{A}\mathbf{v}_n\}$ is orthogonal and the non-zero vectors in this form a basis for the range of $\mathbf{A}$. Hence the eigenvectors of $\mathbf{A}^T\mathbf{A}$ and their image under **A** provides orthogonal bases. When the vectors $\mathbf{Av}_i$ is normalized and given that $i = j$, we have,

$$|\mathbf{Av}_i|^2 = \lambda_i, \text{ which means } \lambda_i \geq 0$$

$$|\mathbf{Av}_i| = \sqrt{\lambda_i}.$$

Note that the eigenvalues λ_i are assumed to be arranged in non-increasing order; hence, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_k \geq 0$.

Clearly,

$$\frac{1}{|\mathbf{Av}_i|} = \frac{1}{\sqrt{\lambda_i}},$$

Hence,

$$\frac{\mathbf{Av}_i}{|A\mathbf{v}_i|} = \frac{\mathbf{Av}_i}{\sqrt{\lambda_i}} = \mathbf{u}_i, \; 1 \leq i \leq k. \quad (3.10)$$

If $k < m$; we extend this to an orthonormal basis for R^m. This generates the desired orthonormal bases for R^n and R^m. Setting $\sigma_i = \sqrt{\lambda_i}$, we get $\mathbf{Av}_i = \sigma_i u_i, \; \forall \; i \leq k$. (3.11)

Bringing together the $\mathbf{v}_i$ as the columns of a matrix **V** and the $\mathbf{u}_i$ to form **U**; it can be seen from (3.11) that $\mathbf{AV} = \mathbf{U\Sigma}$; where $\mathbf{\Sigma}$ has the same dimensions as **A** and has the entries σ_i along the main diagonal with all other entries being zero. Hence,

$$\mathbf{A} = \mathbf{U\Sigma V}^T. \quad (3.12)$$

Generally, the Singular Value Decomposition (SVD) is a well known matrix factorization technique that factors an $m \times n$ matrix **A** into three matrices as follows;

$$\underset{m\times n}{\overset{A}{\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}}} = \underset{m\times r}{\overset{U}{\begin{bmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mr} \end{bmatrix}}} \underset{r\times r}{\overset{\Sigma}{\begin{bmatrix} \sigma_{11} & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & \sigma_{rr} \end{bmatrix}}} \underset{r\times n}{\overset{V^T}{\begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{1r} & \cdots & v_{rn} \end{bmatrix}}} \qquad (3.13)$$

The matrix equation in (3.13) is called the Singular Value Decomposition of **A**. In summary, an $m \times n$ real matrix **A** can be expressed as the product $\mathbf{U\Sigma V}^T$, where **V** and **U** are orthogonal matrices and **Σ** is a diagonal matrix.

The matrix **V** is obtained from the diagonal factorization $\mathbf{A}^T\mathbf{A} = \mathbf{VDV}^T$; in which the diagonal entries of **D** (λ_i), appear in non-increasing order. The columns of **U** come from normalizing the non-vanishing images under **A** of the columns of **V** and extending (if necessary) to an orthonormal basis for $\mathbb{R}^m$; the non-zero entries (σ_i) of **Σ** are the respective square roots of corresponding diagonal entries (λ_i) of **D**.

3.3.1: Singular Value Decomposition and Eigen Value Decomposition. The mathematics illustrated above shows how to determine the SVD of a matrix **A** from the EVD of the symmetric matrix $\mathbf{A}^T\mathbf{A}$.

Conversely, it is possible to also recover the EVD of $\mathbf{A}^T\mathbf{A}$ from the SVD of **A**.
Given the SVD of $\mathbf{A} = \mathbf{U\Sigma V}^T$.

Clearly, $\mathbf{A}^T\mathbf{A} = \mathbf{V\Sigma}^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T$.

Now, since **U** is orthogonal matrix, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where **I** is the identity matrix.

Hence, $\mathbf{A}^T\mathbf{A} = \mathbf{V\Sigma}^T\mathbf{I\Sigma V}^T = \mathbf{V\Sigma}^T\mathbf{\Sigma V}^T$. (3.14)

Now in either order, the product of **Σ** and $\mathbf{\Sigma}^T$ is a square diagonal matrix whose first k diagonal entries are the σ_i and with any remaining diagonal entries equal to

0. Whence, $\mathbf{V\Sigma}^T\mathbf{\Sigma}V^T$ is the EVD of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{V\Sigma\Sigma}^T\mathbf{V}^T$ is the EVD of $\mathbf{AA}^T$.

3.3.2: Deductions from Singular Value Decomposition

The above computations portray the uniqueness of singular value decomposition.

In any SVD of **A**, the right singular vectors (columns of **V**) are the eigenvectors of $\mathbf{A}^T\mathbf{A}$, the left singular vectors (columns of **U**) are the eigenvectors of $\mathbf{AA}^T$, and the singular values are the square roots of the non-zero eigenvalues common to these two symmetric matrices from the diagonal matrix **Σ**. Thus, in possible orthogonal transformations in multidimensional eigenspaces of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{AA}^T$, the matrices **V** and **U** in the SVD are uniquely determined.

If **A** itself is square and symmetric, each eigenvector for **A** with eigenvalue λ, is an eigenvector for $\mathbf{A}^2 = \mathbf{A}^T\mathbf{A} = \mathbf{AA}^T$ with eigenvalue λ^2.

Hence the left and right singular vectors for **A** are simply the eigenvectors for **A**, and the singular values for **A** are the absolute values of its eigenvalues. That is, the EVD and SVD essentially coincide for symmetric **A**, and are actually identical if **A** has non-negative eigenvalues. Particularly, for any **A**, the SVD and EVD of $\mathbf{A}^T\mathbf{A}$ are the same. For detailed information on SVD operations on unitary matrices, refer to Golub and Van Loan, (1996).

3.4: Principal Component Analysis (PCA)

Factor analysis started at the beginning of the twentieth century with the efforts of psychologists to identify the factor or factors that make up intelligence. The person most responsible for pioneering this field was the psychologist, Charles Spearman. In a 1904 paper, Spearman analysed a series of exam scores at a preparatory school, using a correlation matrix. Based on the observed correlations, Spearman concluded that the test results provided evidence of common basic underlying functions. Further work by psychologists to identify the common factors that make up intelligence has led to development of an area of study known as factor analysis. An effective way to suppress redundant information and provide only one or two composite data from most of the information in the initial data is called Principal Component Analysis. Principal Component Analysis (PCA) was independently proposed in 1901 by Karl Pearson and in 1933 by Harold Hotelling to turn a set of possibly correlated variables into a smaller set of uncorrelated variables (Turk & Pentland, 1991). PCA is a statistical procedure concerned with elucidating the covariance structure of a set of variables. In particular it allows us to identify the principal directions in which the data varies. Shlens (2003), states that, PCA computes the most meaningful basis to re-express a noisy, garbled data set. The rationale behind this new basis is to filter out the noise and reveal hidden dynamics in the data set. PCA seeks to find a set of basis images which are uncorrelated, that is, they cannot be linearly predicted from each other and also yield projection directions that maximize the total scatter across all classes or across all face images. According to Barlett *et al.,* (2002), PCA can thus be seen as partially implementing Barlow's ideas: Dependencies that show up in the joint distribution of pixels are separated out into marginal distribution of PCA coefficients. Consider a set of *n* basis images each of which has *n* pixel with intensity 1, where each basis image has different active pixel. Any given image with *n* pixel can be decomposed as linear combination of the standard basis images (Barlett *et al.,* 2002).

Actually the pixel values of an image can be seen as the coordinate of that image with respect to the standard basis. However PCA can only separate pairwise linear dependencies between pixels. High-order dependencies will still show in the joint distribution of PCA coefficients, and, thus, will not be properly separated. Most of the successful representations for face recognition, such as eigenface and local feature analysis are based on PCA. In problems such as face recognition, much of the important information may be contained in the high-order relationships among the image pixels, and thus, it is important to investigate whether generalizations of the PCA which are sensitive to high-order relationships are advantageous or not and not just settling on second-order relationships.

In computational terms the principal components are found by calculating the eigenvectors and eigenvalues of the data covariance matrix. This process is equivalent to finding the axis system in which the covariance matrix is diagonal. The eigenvector with the largest eigenvalue is the direction of greatest variation, the one with the second largest eigenvalue is the (orthogonal) direction with the next highest variation and so on. The Principal Component Analysis (PCA) is one of the most successful techniques that has been used in image recognition and compression. In the field of face recognition, the dimension of facial images is very high and requires a considerable amount of computing time for classification. For example, two-dimensional $n \times n$ grayscale images span an $m = n^2$-dimensional vector space; so an image with 100 × 100 pixels lies in a 10,000-dimensional image space which is way too much for any computations. Our interest lies in the components that account for most of the information. The motivation is that, a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The classification and subsequent recognition time can be reduced by reducing the dimension of the image data. The purpose of PCA in this area is to reduce the large dimensionality of the data space (observed or correlated variable) to the smaller intrinsic dimensionality of feature space (independent or uncorrelated variables) which are needed to describe the data economically (Kim, 2000).

The PCA method finds the directions with the greatest variance in the data, called principal components. PCA does not only reduce the dimensionality of the image, but also retains some of the variations in the image data and provides a compact representation of a face image (Turk & Pentland, 1991).

Generally, the key idea of the PCA method is to transform the face image into a small set of characteristic feature images called eigenfaces which are the principal components of the initial training set of the face images.

The method of using PCA for face recognition which expresses the large 1-D vector pixels constructed from 2-D facial image into the compact principal components of the feature space is called the eigenspace projection (Kim, 2000). Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors).

3.4.1: Mathematics of Principal Component Analysis

A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into or along a thin vector.

Suppose we have n vectors of size N (= rows of image × columns of image) representing a set of sampled images.

Given the training set of images, $\mathbf{X}_j = (X_{ji})_{N\times 1}$ and $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ from (3.1), we demonstrate PCA, with three image matrices of uniform dimension,

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$

$$\mathbf{x}_1 = \begin{bmatrix} 2 & 5 & 6 \\ 1 & 7 & 9 \\ 6 & 4 & 3 \end{bmatrix}, (3 \times 3)\,pixels$$

$$\mathbf{x}_2 = \begin{bmatrix} 1 & 1 & 6 \\ 5 & 2 & 1 \\ 3 & 8 & 1 \end{bmatrix}, (3 \times 3) pixels$$

$$\mathbf{x}_3 = \begin{bmatrix} 0.5 & 3 & 4 \\ 0 & 8 & 4 \\ 5 & 2 & 3 \end{bmatrix}, (3 \times 3) pixels$$

then the images are vectorized to form a matrix

$$\mathbf{X} = \begin{bmatrix} 2 & 1 & 0.5 \\ 1 & 5 & 0 \\ 6 & 3 & 5 \\ 5 & 1 & 3 \\ 7 & 2 & 8 \\ 4 & 8 & 2 \\ 6 & 6 & 4 \\ & 1 & \\ & 1 & \end{bmatrix} (9 \times 3) pixels.$$

4

9

3

3

The images are further mean centred (normalized) by subtracting the mean image from each image vector. Let $\bar{\mathbf{m}}$ represent the mean image.

$$\bar{\mathbf{m}}_j = \frac{1}{n}\sum_{j=1}^{n} \mathbf{X}_{ji}, \tag{3.15}$$

where $\mathbf{X}_{ji}$, $j = 1,2,3,\ldots,n$ and $i = 1,2,\ldots,p$ is the training set of face images.

The mean matrix of $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3]$ is given by,

$$\bar{\mathbf{m}} = \begin{bmatrix} 1.1667 & 2.0000 & 4.6667 & 3.0000 & 5.6667 & 4.6667 & 5.3333 & 4.6667 & 2.3333 \end{bmatrix}^T$$

Let $\mathbf{W} = (\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_n)$ be defined as the mean centred image.

Define $\bar{\mathbf{X}}_j$ as a constant vector of order $(p \times p)$ with all elements same as $\bar{\mathbf{m}}_j$.

From (3.3), $\mathbf{w}_j = \mathbf{X}_j - \bar{\mathbf{X}}_j$.

The mean centred matrix $\mathbf{W}$ for the image matrix $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3]$ is given by;

$$\mathbf{W} = \begin{bmatrix} 0.83333 & -0.16667 & -0.66667 \\ -1.00000 & 3.00000 & -2.00000 \\ 1.33333 & -1.66667 & 0.33333 \\ 2.00000 & -2.00000 & 0.00000 \\ 1.33333 & -3.66667 & 2.33333 \\ -0.66667 & 3.33333 & -2.66667 \\ 0.66667 & 0.66667 & -1.33333 \\ 4.33333 & -3.66667 & -0.66667 \\ 0.66667 & -1.33333 & 0.66667 \end{bmatrix}$$

W is then subjected to principal component analysis, which seeks a set of n orthonormal vectors $\mathbf{e}_j$ which best describe the distribution of the image data. The t^{th} vector $\mathbf{e}_t$ is chosen such that;

$$\lambda_t = \frac{1}{n} \sum_{j=1}^{n} (\mathbf{e}_j^T \mathbf{w}_j)^2, \tag{3.16}$$

is maximum subject to the orthogonality constraints.

$$\mathbf{e}_l^T \mathbf{e}_t = \delta_{lt} = \begin{cases} 1 & \text{if, } l = t \\ 0 & \text{elsewhere} \end{cases} \tag{3.17}$$

The vectors $\mathbf{e}_t$ and the scalars λ_t are the eigenvectors and eigenvalues, respectively, of the covariance matrix

$$\mathbf{C} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{w}_j \mathbf{w}_j^T = \frac{1}{n} \mathbf{W}\mathbf{W}^T, \tag{3.18}$$

where the matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$.

The size of $\mathbf{C}$ $(n \times n)$ could be enormous and determining the eigenvectors and eigenvalues is an intractable task for typical image sizes.

A known theorem in linear algebra states that the vectors $\mathbf{e}_j$ and the scalars λ_j can be obtained by solving for the eigenvalues of $\mathbf{W}^T\mathbf{W}$, respectively.

$$\mathbf{W}^T\mathbf{W}\mathbf{d}_j = \mu_j\mathbf{d}_j \tag{3.19}$$

where $\mathbf{d}_j$ and μ_j are the eigenvectors and eigenvalues respectively of matrix $\mathbf{W}^T\mathbf{W}$. By multiplying both sides (from left) by $\mathbf{W}$, we obtain;

$$\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{d}_j = \mu_j(\mathbf{W}\mathbf{d}_j). \tag{3.20}$$

Which means that the first $n - 1$ eigenvectors $\mathbf{e}_j$ and eigenvalues λ_j of $\mathbf{W}\mathbf{W}^T$ are given by $\mathbf{W}\mathbf{d}_j$ and μ_j respectively. $\mathbf{W}\mathbf{d}_j$ needs to be normalized in order to be equal to $\mathbf{e}_j$. The rank of the covariance matrix cannot exceed $n - 1$. For the image matrix $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3]$, the covariance matrix from $\mathbf{W}^T\mathbf{W}$ is given by,

$$\mathbf{C} = \begin{bmatrix} 2.41667 & -3.37500 & 0.95833 \\ -3.37500 & 6.58333 & -3.20833 \\ 0.95833 & -3.20833 & 2.25000 \end{bmatrix}$$

To obtain the eigenvectors of the matrix $\mathbf{W}^T\mathbf{W}$, a singular value decomposition of its covariance matrix $\mathbf{C}$ is computed. Thus, the SVD of $\mathbf{C}$ will decompose $\mathbf{C}$ as,

$$\mathbf{U} = \begin{bmatrix} -0.420192 & -0.700075 & 0.577350 \\ 0.816379 & -0.013859 & 0.577350 \\ -0.396187 & 0.713934 & 0.577350 \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} 9.8775e+000 & 0 & 0 \\ 0 & 1.3725e+000 & 0 \\ 0 & 0 & 8.0473e-018 \end{bmatrix},$$

$$\mathbf{V} = \begin{bmatrix} -0.420192 & -0.700075 & 0.577350 \\ 0.816379 & -0.013859 & 0.577350 \\ -0.396187 & 0.713934 & 0.577350 \end{bmatrix}$$

SVD aids in decomposing the covariance matrix into two orthogonal matrices **U**,**V** and a diagonal matrix **S**. This is a form that reveals all the eigenvalues (arranged in descending order in the diagonal matrix **S**) and corresponding eigenvectors (shown in **U** as unit eigenvectors) of matrix **C**.

Hence eigenface of the image $X = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$,

$$\mathbf{e}_j = \mathbf{w}_j \times \mathbf{U}, \tag{3.21}$$

is given for $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ as;

$$\begin{bmatrix} 0.83333 & -0.16667 & -0.66667 \\ -1.00000 & 3.00000 & -2.00000 \\ 1.33333 & -1.66667 & 0.33333 \\ 2.00000 & -2.00000 & 0.00000 \end{bmatrix} \begin{bmatrix} -0.42019 & -0.70008 & 0.57735 \end{bmatrix}$$

$$
\mathbf{w}_j \times \mathbf{U} = \begin{bmatrix} -0.66667 & 3.33333 & -2.66667 \\ 0.66667 & 0.66667 & -1.33333 \\ 4.33333 & -3.66667 & -0.66667 \\ 0.66667 & -1.33333 & 0.66667 \\ 1.33333 & -3.66667 & 2.33333 \end{bmatrix} \begin{bmatrix} -0.39619 & 0.71393 & 0.57735 \\ 0.81638 & -0.01386 & 0.57735 \end{bmatrix},
$$

$$
= \begin{bmatrix} -0.22210 & -1.05704 & -0.00000 \\ 3.66170 & -0.76937 & 0.00000 \\ -2.05295 & -0.67236 & -0.00000 \\ -2.47314 & -1.37243 & 0.00000 \\ -4.47808 & 0.78323 & -0.00000 \\ 4.05789 & -1.48330 & -0.00000 \\ 0.79237 & -1.42787 & 0.00000 \\ -4.55009 & -3.45880 & -0.00000 \\ -1.63276 & 0.02772 & -0.00000 \end{bmatrix} \mathbf{e}
$$

The eigenvectors corresponding to non-zero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace within which most image data can be represented with a small amount of error.

The eigenvectors are sorted from high to low according to their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is one that reflects the greatest variance in the image. That is, the smallest eigenvalue is associated with the eigenvector that finds the least variance. We can project a facial image onto n' $(\ll n)$ dimensions by computing

$$\boldsymbol{\Omega} = [v_1 v_2 \ldots v_n]^T. \tag{3.22}$$

where $v_j = \mathbf{e}^T_j \mathbf{w}_j$, the j^{th} coordinate of the facial image in new space, which is referred to as principal component. The vectors $\mathbf{e}_j$ are also images, so called, eigenimages or eigenfaces. Hence the principal components for the image space

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ is given by,

$$\mathbf{\Omega} = \begin{bmatrix} -4.0484e+001 & 6.8725e+001 & -2.8241e+001 \\ -1.7641e+001 & 5.6107e+000 & 1.2030e+001 \\ -3.5661e-015 & 6.1892e-015 & -2.6231e-015 \end{bmatrix}.$$

$\mathbf{\Omega}$ describes the contribution of each eigenface in representing the facial image by treating the eigenfaces as a basis set for facial images.

For determining which face class provides the best description of an input facial image, we find the face class r that minimizes the Euclidean distance,

$$\varepsilon_r = \|\mathbf{\Omega} - \mathbf{\Omega}_r\|, \tag{3.23}$$

where $\mathbf{\Omega}_r$ is a vector describing the r^{th} face class. If ε_r is less than some predefined threshold Ψ_ε, a face is classified as belonging to the class r.

3.5: Feature Extraction Stage

Having these reduction algorithms in mind, it is now time to seek a set of n orthonormal vectors, $\mathbf{e}_j$, which best describes the distribution of the data. The t^{th} vector $\mathbf{e}_t$ is chosen such that

$$\lambda_t = \frac{1}{n} \sum_{j=1}^{n} (\mathbf{e}_j^T \mathbf{w}_j)^2,$$

is a maximum subject to the orthonormality constraints,

$$\mathbf{e}_l^T \mathbf{e}_t = \delta_{lt} = \begin{cases} 1 & \text{if, } l = t. \\ 0 & \text{elsewhere} \end{cases}$$

Referring to equation (3.16), and (3.17) respectively.

The vectors $\mathbf{e}_t$ and scalars λ_t are the eigenvectors and eigenvalues respectively, of the covariance matrix

$$\mathbf{C} = \frac{1}{M}\sum_{j=1}^{n} \mathbf{w}_j \mathbf{w}_j^T = \frac{1}{n}\mathbf{W}\mathbf{W}^T,$$

from (3.18), where the matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$.

We generate eigenvectors and eigenvalues from solving for the eigenvalues of $\mathbf{W}^T\mathbf{W}$.

This is done because the size of $\mathbf{C}$ from $\mathbf{W}\mathbf{W}^T$ (65536 × 65536) is enormous and determining the eigenvectors and eigenvalues is a very tedious task. Now considering the six input images, the covariance matrix from $\mathbf{W}^T\mathbf{W}$ is given by,

$$\mathbf{C} = \begin{pmatrix} 0.0967 & -0.0290 & -0.0264 & -0.0063 & -0.0064 & -0.0285 \\ -0.0290 & 0.1193 & -0.0229 & -0.0453 & -0.0425 & 0.0204 \\ -0.0264 & -0.0229 & 0.1316 & -0.0311 & -0.0204 & -0.0307 \\ -0.0063 & -0.0453 & -0.0311 & 0.1064 & 0.0076 & -0.0313 \\ -0.0065 & -0.0425 & -0.0204 & 0.0076 & 0.0906 & -0.0288 \\ -0.0285 & 0.0204 & -0.0307 & -0.0313 & -0.0288 & 0.0988 \end{pmatrix}.$$

Next is to run a singular value decomposition (SVD) of the matrix $\mathbf{C} = \mathbf{U\Sigma V}^T$ to ascertain its eigenvalues and their corresponding eigenvectors, as seen from equation (3.12). This will split the covariance matrix $\mathbf{C}$ into two orthogonal matrices

$\mathbf{U}$ and $\mathbf{V}$ and a diagonal matrix $\mathbf{\Sigma}$.

$$\mathbf{U} = \begin{pmatrix} -0.2444 & -0.1665 & -0.8452 & 0.0345 & 0.1742 & -0.4082 \\ 0.6343 & -0.1401 & -0.0662 & 0.2742 & -0.5760 & -0.4082 \\ 0.0076 & 0.9033 & 0.0415 & 0.0674 & 0.1054 & -0.4082 \\ -0.4587 & -0.0892 & 0.2041 & -0.6702 & -0.4421 & -0.4082 \\ -0.3729 & -0.0892 & 0.2041 & -0.6702 & -0.4421 & -0.4082 \\ 0.4341 & -0.2518 & 0.2426 & -0.3148 & 0.6508 & -0.4082 \end{pmatrix},$$

$$\mathbf{\Sigma} = \begin{pmatrix} 2.02e^{-001} & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.59e^{-001} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.09e^{-001} & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.0e^{-002} & 0 & 0 \end{pmatrix},$$

$$
\begin{matrix}
0 & 0 & 0 & 0 & 8.359.0e^{-002} & 0 \\
0 & 0 & 0 & 0 & 0 & 1.48e^{-017}
\end{matrix}
$$

V =

−0

0

0

−0

−0

0

Plate 3.6 is the image of the eigenvectors of the covariance matrix **C**. The extent of variability carried by each value is seen from the level of their brightness. The brighter the image, the greater the importance attached to the value in variability. This simply means the bright portions contains important information about the covariance matrix **C** than the dark portions. The hierarchy of importance is determined by the level of brightness.

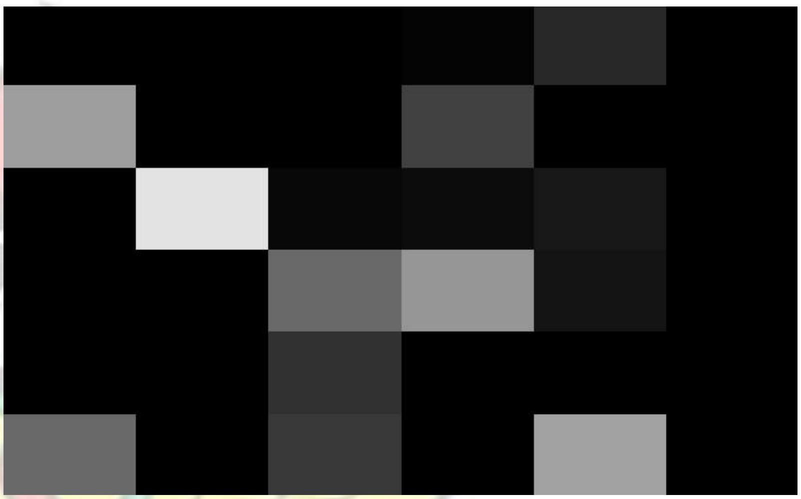


Plate 3.6 : Image of the unit eigenvectors **U.**

The diagonal values of the matrix **S** are the eigenvalues of the covariance matrix **C** and their corresponding unit eigenvectors are the elements of the matrix **U**. Here, the eigenvalues are arranged in descending order and this accounts for why portion of brightness fades as the values become smaller. This is illustrated in *Plate 3.7*.

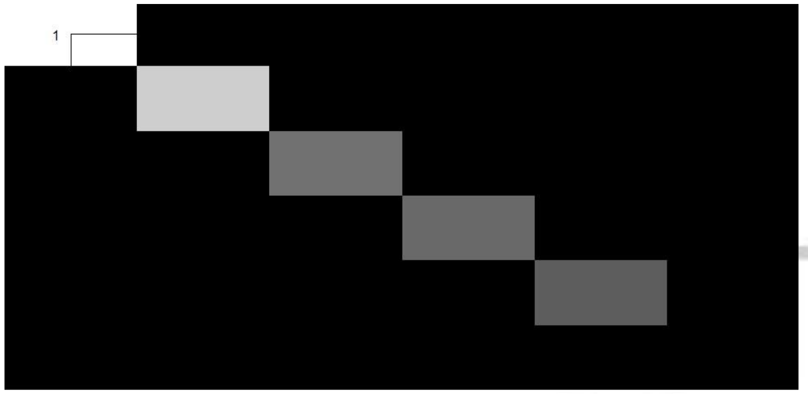


Plate 3.7: Image of the diagonal matrix **Σ**.

From equation (3.19), $\mathbf{WW}^T\mathbf{Wd}_j = \mu_j(\mathbf{Wd}_j)$

This means the first $n - 1$ eigenvectors e_j and eigenvalues λ_j of the $\mathbf{WW}^T$ are

given by $\mathbf{Wd}_j$ and μ_j respectively.

Hence;

$$\mathbf{e}^j = \sum_{j=1}^{n} \mathbf{w}_j \mathbf{d}_j. \quad (3.24)$$

where $\mathbf{d}_j$ and $\mathbf{w}_j$ are the columns from matrix **U** and **W** respectively and $j = 1,2,...,n$

Plate 3.8: The eigenfaces of the set of six input images

The principal component of the trained image set is determined by computing;

$$\nu_j = \mathbf{e}_j^T(\mathbf{x}_j - \bar{\mathbf{m}}), \quad (3.25)$$

and $\mathbf{\Omega}^T = [\nu_1,\nu_2,...,\nu_n]$.

The large correlated image dimensions are finally reduced to uncorrelated smaller intrinsic dimensions which display important characteristics of the image set. *Plate 3.9*, shows the principal components of the six trained images.

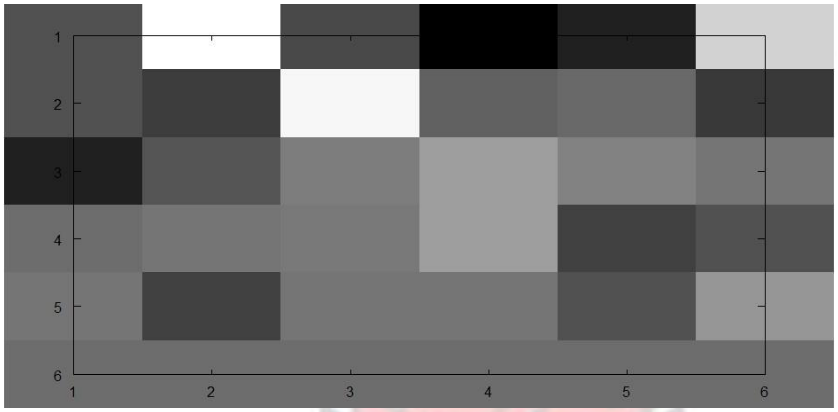


Plate 3.9: The Principle Component display of the trained images

Again, their brightness is tagged directly to their level of importance in describing the variability in the image. This means high level of brightness corresponds to high variability and therefore high importance of that point in carrying information about the image characteristics.

3.6: Recognition Procedure

This section gives a step by step approach to recognizing an unknown face in the trained database. An unknown input face is passed through the steps below before identification.

Following the steps in the feature extraction stage, a new face from the test image database is transformed into its eigenface components. First, the input image is compared with the mean image (trained images mean) in memory and their difference is multiplied with each eigenvector from $\mathbf{e}_j$. Each value represents a weight and is saved on a vector $\mathbf{\Omega}_r$.

$$\nu_j = \mathbf{e}^T{}_j\,(\mathbf{x}_r - \mathbf{m}^{-}), \qquad (3.26)$$

and $\mathbf{\Omega}_r^T = [\nu_1, \nu_2, \ldots, \nu_n]$

The next step is to determine which face class provides the best description for the input image from equation (3.23). This is done by looking for the face class that minimize the Euclidean distance; $\xi = k\mathbf{\Omega}-\mathbf{\Omega}_r k$. The input face is considered to belong to a class if $\varepsilon_m = min(\xi)$, $\xi = \{\varepsilon_1,\varepsilon_2,...,\varepsilon_n\}$ is below an established threshold Ψ_ξ. The face is then considered as a known face. Also if ε_m is above a given threshold Ψ_ξ but below a second threshold, then the image can be determined as an unknown face. Finally if ε_m is above all thresholds then the text image is classified as not a face image.

3.7: Statistical Evaluation of the Face Recognition Algorithms.

The recognition algorithms under study are PCA/SVD (Algorithm 1), Whitened PCA/SVD (Algorithm 2) and Fast Fourier Transform as preprocessing step of PCA/SVD (Algorithm 3). Figure 3.1 shows the flow diagram of the face recognition algorithms evaluated in this study.

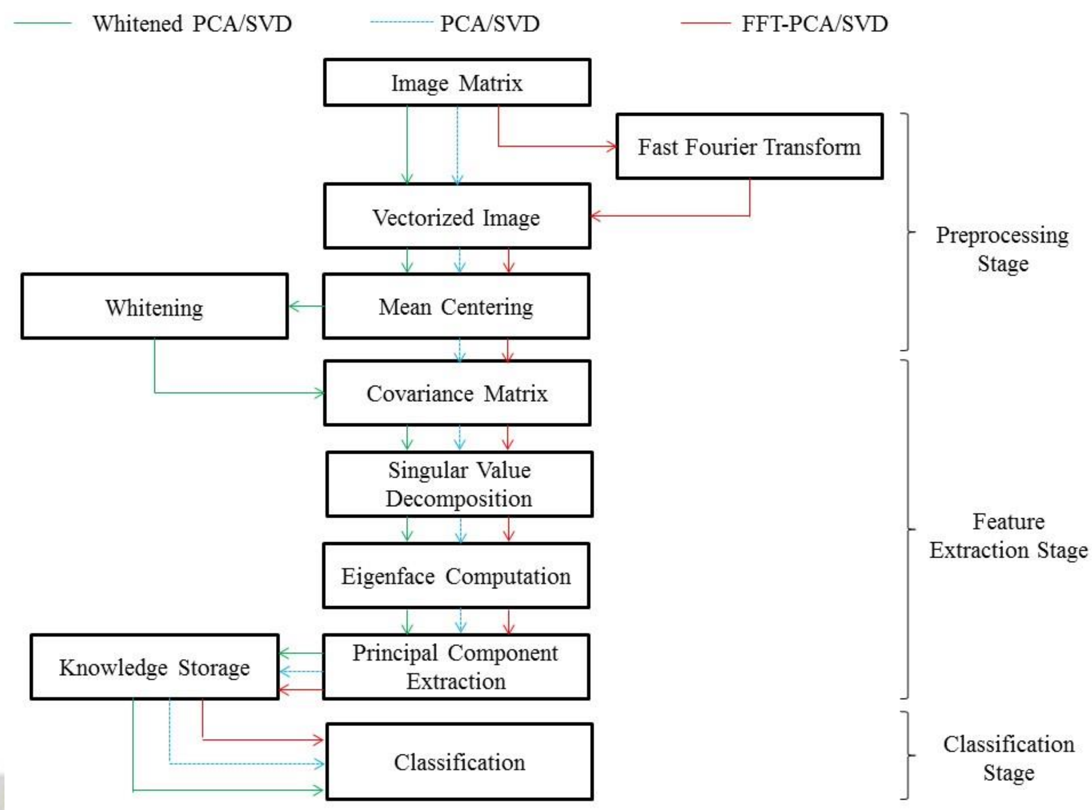


Figure 3.1: Flow Diagram of Study Algorithms

3.7.1: Assessing Multivariate Normality based on Squared Generalised distances.

From the study database, 6-variates are collected per each algorithm from the euclidean distance between the universally accepted principal emotions (Angry, Disgust, Fear, Happy, Sad and Surprise) and their neutral pose.

For each algorithm, we have $\mathbf{X}_{jk}$ dataset, $k = 1,2,...,p$ and $j = 1,2,...,n$. Where k is the number of constraints aside the neutral pose and j is the number of individuals in the research database.

Define the squared distance as,

$$\mathbf{d}^2 = \left(\mathbf{X}_{jk} - \bar{\mathbf{X}}_k\right) \boldsymbol{\Sigma}^{-1} \left(\mathbf{X}_{jk} - \bar{\mathbf{X}}_k\right)^T, \tag{3.27}$$

where, $\bar{\mathbf{X}}_k = \frac{1}{n}\sum_{j=1}^{n}\mathbf{X}_{jk}$ and $k = 1,2,...,p$.

Define $d^2{}_j$ as the diagonal entry of the j^{th} individual from matrix $\mathbf{d}^2$.

When the parent population is multivariate normal and n, $n - k$ are greater than 25 or 30, each of the squared distances $d^2{}_j$, $j = 1,2,...,n$ should behave like a chi-square random variable.

The resulting plot is called a chi-square plot or gamma plot because the chi-square distribution is a special case of the more general gamma distribution.

To construct the chi-square plot;

1. The squared distances $d^2{}_j$, $j = 1,2,...,n$ is arranged in ascending order as ; $d_1^2 \leq d_2^2 \leq d_3^2 \leq \cdots \leq d_n^2$.

$$\left\{ q_{c,k}\frac{(j-\frac{1}{2})}{n}, d_j^2 \right\} \qquad q_{c,k}\left(\frac{(j-\frac{1}{2})}{n}\right) \qquad \frac{(j-\frac{1}{n})}{n}$$

2. A graph ofwhereis the 100quan-tile of chi-square distribution with k degrees of freedom is constructed.

$$q_{c,k}\left(\frac{(j-\frac{1}{2})}{n}\right)$$

The quantileare related to the upper percentile of a chi-squared distribution and $q_{c,k}\left(\frac{(j-\frac{1}{2})}{n}\right) = \chi_k^2\left(\frac{(n-j+\frac{1}{2})}{n}\right)$.

3. The plot takes the form of a straight line through the origin having a unit slope.

4. A systematic curve pattern suggests lack of normality. One or two points farabove the line indicate a large distance or outlying observations, that merit further attention.

3.7.2: Repeated Measures Design

This test is used to compare experimental treatments. For this study, the treatments are seen as the constraints in the recognition database. The purpose of the test is to determine whether for each of the recognition algorithm under study, there exist a significant difference between the average distances of the various poses from their neutral pose. Let,

$$\mathbf{X}_j = \begin{bmatrix} X_{j1} \\ X_{j2} \\ \vdots \\ X_{jp} \end{bmatrix},$$

$j = 1,2,\dots,n$, where, X_{jk} is the measure of the k^{th} constraints on the j^{th} individual.

Repeated Measures stem from the fact that all treatments are administered to each individual (Johnson & Wichern, 2007; Rencher, 2002). That is, the various constraints are taken for each individual.

For comparative purpose, we consider contrasts of the components of $\mu = E(X_j)$ as,

$$\begin{bmatrix} \mu_1 - \mu_2 \\ \mu_2 - \mu_3 \\ \mu_3 - \mu_4 \\ \mu_4 - \mu_5 \\ \mu_5 - \mu_6 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{bmatrix} = \mathbf{C}\mu. \tag{3.28}$$

Where **C** is called the contrast matrix because the rows are linearly independent and each is a contrast vector. The nature of the design eliminates much of the influence of the unit to unit variation on the treatment comparisons (Johnson & Wichern, 2007).

The null hypothesis, H_0, states that, there are no differences in treatments (Equal treatment means). The alternative, H_1, states that there are difference in treatments.

$H_0 : \mathbf{C}\mu = \mathbf{0}$

$H_1 : \mathbf{C}\mu \neq \mathbf{0}$

The T^2- Statistic is given by;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}}, \tag{3.29}$$

where $\mathbf{\Sigma} = \frac{1}{n-1}\sum_{j=1}^{n}\left(\mathbf{X}_j - \bar{\mathbf{X}}\right)\left(\mathbf{X}_j - \bar{\mathbf{X}}\right)^T$ is the covariance matrix for a particular algorithm and $\bar{\mathbf{X}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{X}_{jk}$.

Reject H_0 if ;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}} > \frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(\alpha), \qquad (3.30)$$

where $F_{p-1,n-p+1}(\alpha)$ is the upper $(100\alpha)th$ percentile of an F-distribution with $p-1$ and $n-p+1$ degree of freedom.

A confidence region for contrasts $\mathbf{C}\mu$, is determined by the set of all $\mathbf{C}\mu$ such that,

$$n\left(\mathbf{C}\bar{\mathbf{X}} - \mathbf{C}\mu\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \left(\mathbf{C}\bar{\mathbf{X}} - \mathbf{C}\mu\right) \leq \frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(\alpha).$$

$100(1-\alpha)\%$ simultaneous confidence intervals are;

$$\mathbf{c}_i^T\mu : \mathbf{c}_i^T\bar{\mathbf{X}} \pm \sqrt{\frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(\alpha)} \sqrt{\frac{\mathbf{c}_i\mathbf{\Sigma}\mathbf{c}_i^T}{n}}, \qquad (3.31)$$

where $\mathbf{c}_i$, $i = 1,2,\ldots,p-1$ is the i^{th} row vector of the contrasts $\mathbf{C}$.

3.7.3: Paired Comparisons

Measurements are often recorded under different sets of experimental conditions to see whether the responses differ significantly over these sets. In the case of this study, the euclidean norms of various poses (Angry, Disgust, Fear, Happy, Sad and Surprise) along with their neutral pose are recorded by using three different recognition algorithms. One rational approach to comparing two treatments or the presence and absence of single treatment is to assign both treatment to the same or identical units (Johnson & Wichern, 2007). This concept is also captured in H¨ardle and Simar (2003).

Specifically for this study, 42 individuals were tested on the different recognition algorithms. The paired responses may then be analysed by computing their differences, thereby eliminating much of the influence of extraneous unit to unit variation.

In a single pose (univariate) case, let X_{j1} denote the computed euclidean norms from one algorithm and X_{j2} denote the computed euclidean norms from another algorithm for the j^{th} individual.

That is (X_{j1},X_{j2}) are computed euclidean norms on the j^{th} individual or the j^{th} pair of like units.

Now denote D_j, $j = 1,2,...,n$ (number of individuals) as the differences such that;

$$D_j = X_{j1} - X_{j2}. \tag{3.32}$$

Given that the differences D_j represent independent observations from N (μ,σ_d^2) distribution, the t- statistic is;

$$t = \frac{\bar{D} - \mu}{S_d/\sqrt{n}}, \tag{3.33}$$

where $\bar{D} = \frac{1}{n}\sum_{j=1}^{n} D_j$ and $S_d^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(D_j - \bar{D}\right)^2$ has a t-distribution with $n - 1$ degree of freedom.

Consequently, an α test of ;

$H_0 : \mu = 0$ (zero mean difference for algorithms),

$H_1 : \mu$ 6= 0, may be conducted by comparing | t | with $t_{n-1}\left(\frac{\alpha}{2}\right)$, the upper 100 $\left(\frac{\alpha}{2}\right)th$percentile of the t-distribution with $n - 1$ degree of freedom.

A 100(1 − α)% confidence interval for the mean difference $\mu = E(X_{j1} - X_{j2})$ is;

$$\mu : \bar{d} \pm t_{n-1}\left(\frac{\alpha}{2}\right)\frac{S_d}{\sqrt{n}}. \tag{3.34}$$

The multivariate case is motivated for p constraints, g algorithms and n experimental units. Label X_{jkl}, $j = 1,2,...,n$, $k = 1,2,...,p$, $l = 1,2,...,g$ as the euclidean norm of the k^{th} constraints under algorithm l for the jth individual.

The paired differences random variables become;

$$
\begin{aligned}
D_{j1} &= X_{j11} - X_{j12},\\
D_{j2} &= X_{j21} - X_{j22},\\
&\vdots \qquad \vdots\\
D_{jk} &= X_{jp1} - X_{jp2}.
\end{aligned}
\tag{3.35}
$$

Let $\mathbf{D}^T_j = [D_{j1}, D_{j2}, ..., D_{jp}]$ and assume for $j = 1,2,...,n$,

$$E(\mathbf{D}_j) = \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} \text{ and } Cov(\mathbf{D}_j) = \mathbf{\Sigma}_d.$$

In addition, $\mathbf{D}_j$, $j = 1,2,...,n$ are independent $N_p\ (\mu, \mathbf{\Sigma}_d)$ random vectors. Inferences about the vector of mean difference μ can be based on a T^2-statistic.

$$T^2 = n\left(\bar{\mathbf{D}} - \mu\right)^T \mathbf{\Sigma}_d^{-1}\left(\bar{\mathbf{D}} - \mu\right), \tag{3.36}$$

where $\bar{\mathbf{D}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{D}_j$ and $\mathbf{\Sigma}_d = \frac{1}{n-1}\sum_{j=1}^{n}\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)^T$.

Let $\mathbf{D}_j$, $j = 1,2,...,n$ be a random sample from a $N_p\ (\mu, \mathbf{\Sigma}_d)$ population, then $T^2 = n\left(\bar{\mathbf{D}} - \mu\right)^T \mathbf{\Sigma}_d^{-1}\left(\bar{\mathbf{D}} - \mu\right)$ is distributed as $\left[\frac{(n-1)\,p}{(n-p)}\right] F_{p,n-p}$ random variable, whatever the true value of μ and $\mathbf{\Sigma}_d$.

For n and $n-p$ both large, T^2 is approximately distributed as χ^2_p random variable, regardless of the form of the the underlying population of the differences. Given the observed differences, $\mathbf{d}^T_j = [d_{j1}, d_{j2}, ..., d_{jp}]$, $j = 1, 2, ..., n$ corresponding to the random variables in (3.35), an α- level test of, $H_0 : \mu = \mathbf{0}$ (zero mean difference between algorithms),

$H_1 : \mu \neq \mathbf{0}$.

For a $N_p(\mu, \Sigma_d)$ population, reject H_0 if the observed,

$$T^2 = n\bar{\mathbf{d}}^T \mathbf{\Sigma}_d^{-1} \bar{\mathbf{d}} > \left[\frac{(n-1)\,p}{(n-p)}\right] F_{p,n-p}(\alpha), \quad (3.37)$$

where $F_{p,n-p}(\alpha)$ is the upper (100α) percentile of an F- distribution with p and $n-p$ degree of freedom.

The Bonferroni $100(1-\alpha)\%$ simultaneous confidence intervals for the individual mean differences are;

$$\mu_i : \bar{d}_i \pm t_{n-1}\left(\tfrac{\alpha}{2p}\right)\sqrt{\frac{S^2_{di}}{n}}, \quad (3.38)$$

where $\bar{d}_i$ is the ith element of $\bar{\mathbf{d}}$, S_{di}^2 is the i^{th} diagonal element of $\mathbf{\Sigma}_d$ and $t_{n-1}\left(\frac{\alpha}{2p}\right)$ is the upper $100\left(\frac{\alpha}{2p}\right)th$ percentile of a t-distribution.

These confidence intervals will reveal specifically which constraints have significant differences in euclidean distances when the different face recognition algorithms are used.

3.7.4: Test for Equality of Covariance Matrices (Box's M)

This test is used as a measure of consistency between the recognition algorithms. The test will reveal whether the variations in distances across algorithms in the recognition of face images in the study database are equal or significantly different.

The most consistent algorithm should have lower variation in recognition distances.

For g populations, the null hypothesis, H_0 and alternative hypothesis, H_1 are;

$H_0 : \mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \cdots = \mathbf{\Sigma}_g = \mathbf{\Sigma},$

$H_1 : \mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j$ for $i \neq j$ (at least two unequal covariance matrices), where $\mathbf{\Sigma}_l$ is the covariance matrix for the lth population, $l = 1,2,...,g$. Here our populations are the measures from the recognition algorithms and $\mathbf{\Sigma}$ is the presumed common covariance matrix for the populations.

Assuming the collected samples under study are from multivariate normal populations, a likelihood ratio statistic for testing is given by;

$$\Lambda = \prod_{l=1}^{g} \left(\frac{| \mathbf{S}_l |}{| \mathbf{S}_{pooled} |} \right)^{(n_l-1)/2} . \tag{3.39}$$

Here, n_l, $l = 1,2,...,g$ is the sample size for the lth group (algorithm). $\mathbf{S}_l$ is the l^{th} group sample covariance matrix and $\mathbf{S}_{pooled}$ is the pooled sample covariance given by;

$$\mathbf{S}^{pooled} = \frac{1}{\sum_{l=1}^{g} (n_l - 1)} \sum_{l=1}^{g} (n_l - 1)\, \mathbf{S}_l . \tag{3.40}$$

The Box's test is based on the χ^2 approximation to the sampling distribution of $-2\ln\Lambda$.

setting $M = -2\ln\Lambda$, gives;

$$M = \sum_{l=1}^{g} (n_l - 1)\ln|\mathbf{S}_{pooled}| - \sum_{l=1}^{g} (n_l - 1)\ln|\mathbf{S}_l| . \tag{3.41}$$

Box's test for equality of covariance is motivated as;

$$U = \left[\sum_{l=1}^{g} \frac{1}{(n_l - 1)} - \frac{1}{\sum_{l=1}^{g} (n_l - 1)} \right] \left[\frac{2p^2 + 3p - 1}{6(p+1)(g-1)} \right], \tag{3.42}$$

where p is the number of constraints and g is the number of groups (Algorithms).

Now, $K = (1 - U)M$

has an approximate χ^2 distribution with $V = \frac{g}{2}p(p+1) - \frac{1}{2}p(p+1) = \frac{1}{2}p(p+1)(g-1)$ degree of freedom.

Reject H_0 at α (significance level) if, $K > \chi^2_{\frac{1}{2}p(p+1)(g-1)}(\alpha)$.

If H_0 is true, the algorithm sample covariance matrix are not expected to differ too much and consequently do not differ from the pooled covariance. In this case,

$$\left(\frac{|\ \boldsymbol{\Sigma}_1\ |}{|\ \boldsymbol{\Sigma}_2\ |}\right) = 1.$$

3.7.5: Profile Analysis

This analysis pertains to situation in which two tests are administered to two or more groups of subjects (Johnson & Wichern, 2007; Izenman, 2008). In performing this analysis, all responses must be expressed in similar units. It is also assumed that all responses for different groups are independent of one another.

Let $\boldsymbol{\mu_1}^T = [\mu_{11}, \mu_{12}, \ldots, \mu_{1p}]$ and $\boldsymbol{\mu_2}^T = [\mu_{21}, \mu_{22}, \ldots, \mu_{2p}]$ be the mean responses to p treatments for population 1 and 2 respectively.

The null hypothesis $H_0 : \boldsymbol{\mu_1} = \boldsymbol{\mu_2}$ implies that the treatments have the same average effect on the two populations.

In terms of population profiles, the stepwise process below is employed.

1. Are the profiles parallel?

   $H_{01} : \mu_{1i} - \mu_{1i-1} = \mu_{2i} - \mu_{2i-1},\ i = 1,2,\ldots,p.$

2. Assuming the profiles are parallel, are they coincident?
   $H_{02} : \mu_{1i} = \mu_{2i},\ i = 1,2,\ldots,p.$

3. Assuming the profiles are coincident, are they level?

   $H_{03} : \mu_{11} = \mu_{12} = \cdots = \mu_{1p} = \mu_{21} = \mu_{22} = \cdots = \mu_{2p}.$

3.7.5.1: Test for Parallel Profiles

The null hypothesis H_{01} can be written as $H_{01}: \mathbf{C}\mu_1 = \mathbf{C}\mu_2$
where **C** is the contrast matrix.

$$\mathbf{C} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ & & & 1 & -1 & \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

In testing for parallel profiles for two populations, reject $H_{01}: \mathbf{C}\mu_1 = \mathbf{C}\mu_2$ (parallel profiles) at α level if;

$$T^2 = (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)^T \mathbf{C}^T \left[\mathbf{C} \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2} \right) \mathbf{C}^T \right]^{-1} \mathbf{C}(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2) > c^2, \quad (3.43)$$

where $c^2 = \frac{(n_1 + n_2 - 2)(p-1)}{n_1 + n_2 - p} F_{p-1, n_1+n_2-p}(\alpha)$ and S_1, S_2 are covariance matrices of population 1 and 2 respectively.

When profiles are parallel, the first is either below the second ($\mu_{1i} < \mu_{2i}$) or the first above the second ($\mu_{1i} > \mu_{2i}$), for all i.

3.7.5.2: Test for Coincident Profiles

If H_{01} is tenable, then in testing for coincident profiles for two populations, reject

$H_{02}: \mathbf{1}^T\mu_1 = \mathbf{1}^T\mu_2$ (coincident profiles) at α level if;

$$T^2 = (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)^T \mathbf{1}^T \left[\mathbf{1} \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2} \right) \mathbf{1}^T \right]^{-1} \mathbf{1} (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2),$$

$$= \left[\frac{\mathbf{1}^T (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)}{\sqrt{\mathbf{1}^T \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2} \right) \mathbf{1}}} \right]^2 > t_{n_1+n_2-2}\left(\frac{\alpha}{2}\right) = F_{1,n_1+n_2-2}(\alpha) \quad . \tag{3.44}$$

3.7.5.3: Test for Level Profiles

When H_{01} and H_{02} are tenable, the common mean vector μ is estimated, using all $n_1 + n_2$ observations.

$$\bar{\mathbf{X}} = \frac{n_1}{(n_1 + n_2)} \bar{\mathbf{X}}_1 + \frac{n_2}{(n_1 + n_2)} \bar{\mathbf{X}}_2 \quad . \tag{3.45}$$

For two normal population, reject $H_{03} : \mathbf{C}\mu = \mathbf{0}$ (Level profiles) at α level of significance if;

$$(n_1 + n_2) \left(\bar{\mathbf{X}}^T \mathbf{C}^T \right)^T \left[\mathbf{C \Sigma C}^T \right]^{-1} \mathbf{C} \bar{\mathbf{X}} > c^2, \tag{3.46}$$

where $\mathbf{\Sigma}$ is the sample covariance based on $n_1 + n_2$ observations and

$$c^2 = \frac{(n_1 + n_2 - 1)(p - 1)}{n_1 + n_2 - p + 1} F_{p-1,n_1+n_2-p+1}(\alpha) \quad .$$

3.7.6: Levene's Test for Equality of Variance

The goal of this univariate test is to determine whether the populations under study have equal variance. The test is quite sensitive to the underlying assumption that, the samples being tested should come from a normal population.

Two independent normal populations each from the different study algorithms are collected. For example, angry pose data from algorithm 1 tested against angry pose data from algorithm 2. Let $\mathbf{X}_{jk1}$, $j = 1,2,...,n$ (individuals) and $k = 1,2,...,p$

(poses) be the datasets from algorithm 1 and $\mathbf{X}_{jk2}$, $j = 1,2,...,n$ (individuals) and $k = 1,2,...,p$ (poses) be the datasets from algorithm 2. Now consider two independent normal populations $\mathbf{X}_{jk1}$ and $\mathbf{X}_{jk2}$, with unknown variances σ_{k1}^2 and σ_{k2}^2. The null hypothesis, H_0 and alternative hypothesis, H_1 are given by;

$H_0 : \sigma_{k1}^2 = \sigma_{k2}^2$ (Equal variances),

$H_1 : \sigma_{k1}^2 \neq \sigma_{k2}^2$ (Unequal variances).

Consider samples of size n_{k1} from population 1 and n_{k2} from population 2 with their respective sample variance S_{k1}^2 and S_{k2}^2.

To test $H_0 : \sigma_{k1}^2 = \sigma_{k2}^2$, the test statistic is;

$$F = \frac{S_{k2}^2}{S_{k1}^2}. \tag{3.47}$$

Reject H_0 at α level of significance if;

$$F < F_{1-\frac{\alpha}{2},n_{2k}-1,n_{1k}-1} \text{ or } F > F_{\frac{\alpha}{2},n_{2k}-1,n_{1k}-1}. \tag{3.48}$$

A $100(1-\alpha)\%$ confidence interval is given by;

$$\frac{S_{k1}^2}{S_{k2}^2} F_{1-\frac{\alpha}{2},n_{2k}-1,n_{1k}-1} < \frac{\sigma_{k1}^2}{\sigma_{k2}^2} < \frac{S_{k1}^2}{S_{k2}^2} F_{\frac{\alpha}{2},n_{2k}-1,n_{1k}-1}. \tag{3.49}$$

If H_0 is not tenable and $\frac{S_{k1}^2}{S_{k2}^2} > 1$, then the variations in population 1 are greater than that of population 2. Hence population 2 is considered as comparatively consistent.

Also, if H_0 is not tenable and $\frac{S_{k1}^2}{S_{k2}^2} < 1$, then the variations in population 2 are greater than that of population 1 and hence population 1 is considered as comparatively consistent.

3.8: Numerical Evaluations

Numerically, the algorithms are evaluated through their recognition rate and runtime (computational time). The recognition rate of an algorithm is defined as the ratio of the total number of correct recognition by the algorithm to the total number of images in the test set for a single experimental run. Recognition performance has many measurement standards. The most important and popular formula is;

$$\text{Recognition rate} = \frac{n_r}{n_t}, \tag{3.50}$$

where n_r, n_t represents the number of recognized images and the number of test images respectively. Therefore, the average recognition rate, R_{avg}, is defined as;

$$R_{avg} = \frac{\sum_{i=1}^{q} n_{cls}^{i}}{q \times n_{tot}}, \tag{3.51}$$

where q is the number of experimental runs. The n^i_{cls} is the number of correct recognition in the i^{th} run and n_{tot} is the total number of faces under test in each run. Consequently, the average error rate, E_{avg}, may be defined as $100 - R_{avg}$.

3.9: Analytic Codes Documentation

The GNU Octave 3.24 was used as the computational tool for all numerical runs of the face recognition algorithms and all statistical methods used in this study. The codes in this study are categorised into two section namely;

- Numerical Computations: This part of the codes was used for the database training and recognition. The codes have been labelled according to the algorithms. Algorithm 1 represents the recognition process using PCA with SVD and only mean centering as the preprocessing step, Algorithm 2 and

Algorithm 3 are PCA with SVD using Whitening and Fast Fourier Transform (FFT) as the preprocessing steps respectively. Each algorithm first seeks to train the study database and subsequently recognize an unknown image subject to the variable constraints under study. See *appendix 4.0* for the codes of each recognition algorithm.

- Statistical Computations: GNU Octave codes were written and documented for all the statistical computations. These codes used data churned from the numerical computations as input data. See *appendix 5.0* for the codes on statistical computations (Statistical method specific).

3.10: Summary

In this chapter, we have discussed Principal Component Analysis and Singular Value Decomposition (PCA/SVD) face recognition algorithm from its preprocessing stage to the recognition stage. The three image preprocessing mechanisms (Mean Centering, Whitening and Fast Fourier Transform) adopted by the study have also been discussed with their mathematical representations. The chapter also presented the proposed multivariate statistical evaluation and the adopted numerical evaluation methods with their mathematical representations.

CHAPTER FOUR

Results and Discussion

4.0: Introduction

This chapter, present a detailed computational work on dataset churned from running the study algorithms on the available database using the statistical methods discussed in the previous chapter. The chapter also explains the reasoning behind every method used and the outcome of the analysis.

4.1: Multivariate Normality

Figure 4.1, *Figure 4.2* and *Figure 4.3* show the chi-square plots of the datasets from the study Algorithm 1, Algorithm 2 and Algorithm 3 respectively. The correlation, *r*, values 0.91359, 0.95846 and 0.91631 for Algorithm 1, Algorithm 2 and Algorithm 3 respectively are close to 1. These satisfy the assumption of a unit slope of the chi-square plot. Also Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) are linear transformations and as such preserve normality. (See *appendix 3.0* for the squared distances and quantiles data).

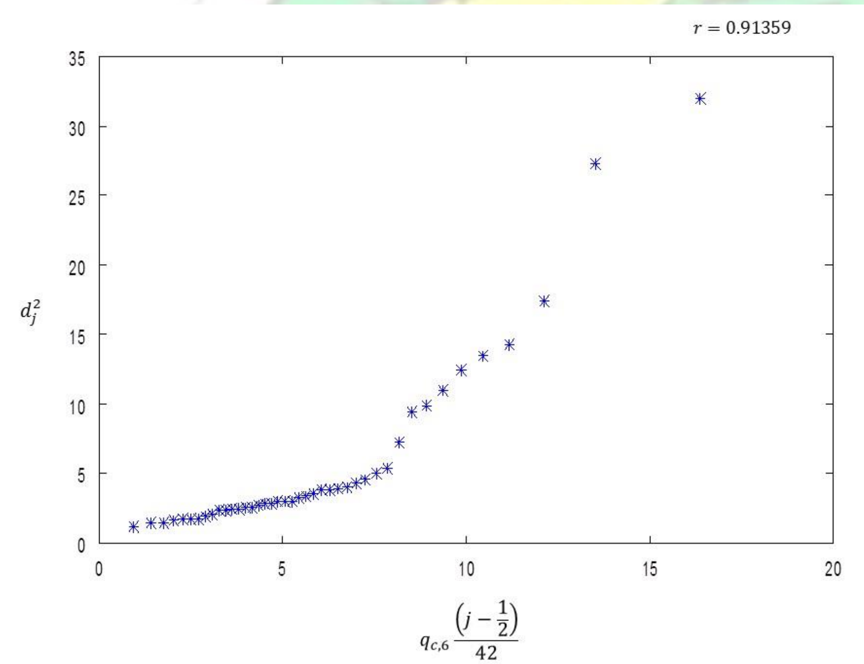


Figure 4.1: Chi-square plot for Algorithm 1

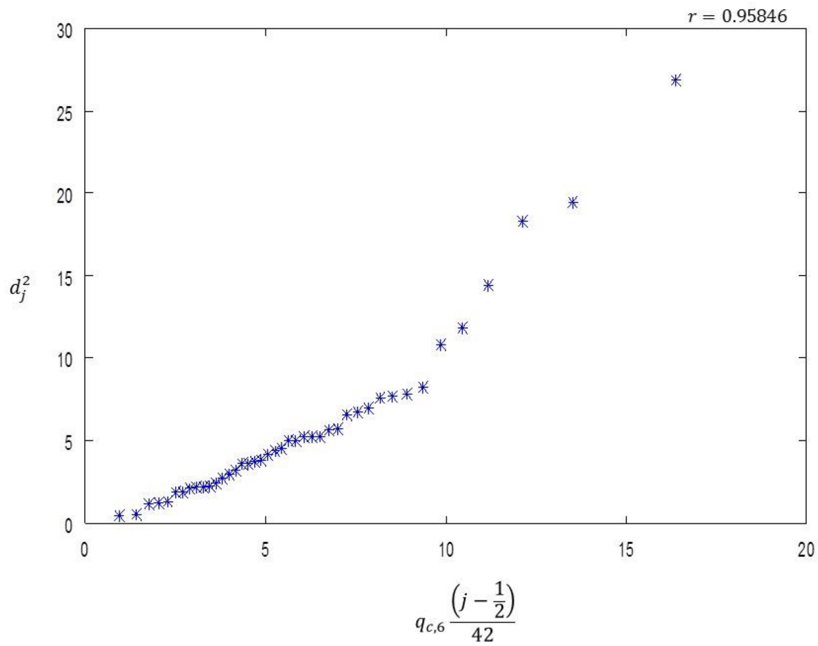


Figure 4.2: Chi-square plot for Algorithm 2

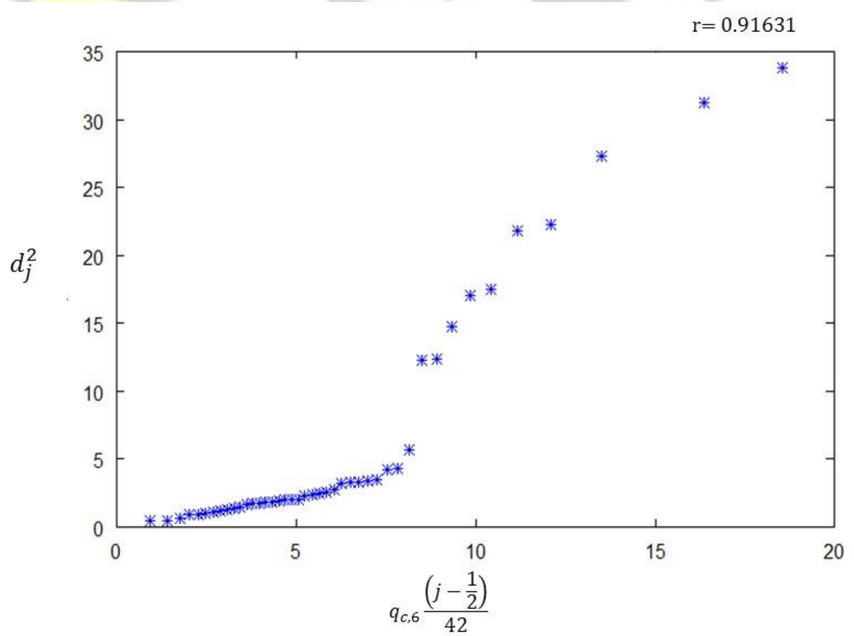


Figure 4.3: Chi-square plot for Algorithm 3

From *Figure 4.1*, *Figure 4.2* and *Figure 4.3*, the assumption of multivariate normality is not violated and hence can be assumed in subsequent statistical test that will be performed on the datasets.

4.2: Results from Repeated Measures Design

The repeated measures design is performed on the euclidean distances recorded after recognition. These datasets (Algorithm specific) are shown in *appendix 2.0*. The purpose of the test is to determine whether for each of the recognition algorithms under study, there exist a significant difference between the average distances of the various poses from their neutral pose.

Using the 6-variate dataset from Algorithm 1, we have;

$$n = 42,\quad \mathbf{C}\bar{\mathbf{X}} = \begin{bmatrix} 1003.917 \\ -98.952 \\ 278.283 \\ -96.397 \\ -1167.166 \end{bmatrix}$$

$$(\mathbf{C}\Sigma\mathbf{C}^T)^{-1} = \begin{bmatrix} 1.28e{-007} & 1.12e{-007} & 6.21e{-008} & -3.69e{-009} & 6.48e{-009} \\ 1.12e{-007} & 4.99e{-007} & 4.15e{-007} & 1.33e{-007} & 1.40e{-008} \\ 6.21e{-008} & 4.15e{-007} & 7.85e{-007} & 5.68e{-007} & 2.40e{-007} \\ -3.69e{-009} & 1.33e{-007} & 5.68e{-007} & 7.40e{-007} & 4.36e{-007} \\ & 1.40e{-008} & 2.40e{-007} & 4.36e{-007} & \end{bmatrix}$$

$6.48e_{-009}$ $5.44e_{-007}$

Hence, the T^2- Statistic from (3.29) is;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}},$$

$$= 35.095.$$

Now, $\frac{(n-1)(p-1)}{(n-p+1)} = 5.540541,\ F_{p-1,n-p+1}(0.05) = 2.4697$.

Hence, $\frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(0.05) = 13.6832$.

From equation (3.30), reject H_0 if ;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}} > \frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(\alpha).$$

35.095 > 13.683194. There is therefore enough evidence at 5% level of significance to reject H_0 and conclude on H_1 : $\mathbf{C}\mu \neq \mathbf{0}$. This means there exist significant difference in the average recognition distances of the various constraints from their neutral pose when Algorithm 1 is used for recognition.

The 95% simultaneous confidence intervals for the estimates of the mean differences are shown in *Table 4.1*.

Table 4.1: Simultaneous Confidence Intervals - Algorithm 1.

Constraints	**Mean Difference**	**Lower**	**Upper**
Angry vs Disgust	$\bar{X}_1 - \bar{X}_2 = 1003.9170$	−810.2300	2818.0636
Angry vs Fear	$\bar{X}_1 - \bar{X}_3 = 904.9650$	−949.4138	2759.3435
Angry vs Happy	$\bar{X}_1 - \bar{X}_4 = 1183.2480$	−537.6334	2904.1299
Angry vs Sad	$\bar{X}_1 - \bar{X}_5 = 1086.8520$	−879.9624	3053.6657
Angry vs Surprise	$\bar{X}_1 - \bar{X}_6 = -80.3140$	−1874.7483	1714.1202

Disgust vs Fear	$\bar{X}_2 - \bar{X}_3 = -98.9520$	-1489.3901	1291.4861
Disgust vs Happy	$\bar{X}_2 - \bar{X}_4 = 179.3300$	-925.4309	1284.0937
Disgust vs Sad	$\bar{X}_2 - \bar{X}_5 = 82.9350$	-1201.0698	1366.9394
Disgust vs Surprise	$\bar{X}_2 - \bar{X}_6 = -1084.200$	-2317.2119	148.7501
Fear vs Happy	$\bar{X}_3 - \bar{X}_4 = 278.2834$	-1290.6203	1847.1871
Fear vs Sad	$\bar{X}_3 - \bar{X}_5 = 181.8868$	-855.2611	1219.0347
Fear vs Surprise	$\bar{X}_3 - \bar{X}_6 = -985.2789$	-2024.8504	54.2926
Happy vs Sad	$\bar{X}_4 - \bar{X}_5 = -96.3970$	-1633.8636	1441.0704
*Happy vs Surprise	$\bar{X}_4 - \bar{X}_6 = -1263.600$	-2469.2859	-57.8386
*Sad vs Surprise	$\bar{X}_5 - \bar{X}_6 = -1167.200$	-2298.1676	-36.1638

$*$: *significant difference exist between constraints at* 5% *significance level.*

These simultaneous confidence intervals reveal the specific constraints that are significantly different in their average distances from their neutral poses. All constraints that contain 0 in their confidence interval are said to be equal and do not have significant differences from their neutral pose at 5% significance level.

It can be seen from *Table 4.1* that, for Algorithm 1, the Happy vs Surprise ($\mu_4 - \mu_6$) and Sad vs Surprise ($\mu_5 - \mu_6$) have confidence intervals, $[-2469.2859, -57.8386]$ and $[-2298.1676, -36.1638]$ respectively. Since these intervals do not contain 0, the respective constraints (Happy vs Surprise and Sad vs Surprise) have significant difference in their average distance from their neutral poses when Algorithm 1 is used as the recognition algorithm. PCA/SVD (Algorithm 1) had significantly higher average recognition distances in the recognition of Surprise poses as compared to Happy and Sad poses.

Using the 6-variate dataset from Algorithm 2, we have;

$n = 42$, $\left(\mathbf{C}\bar{\mathbf{X}}\right) = \begin{bmatrix} -157.19 \\ 440.23 \\ -271.21 \\ 865.49 \\ -2273.87 \end{bmatrix}$

$$\left(\mathbf{C\Sigma C}^T\right)^{-1} = \begin{bmatrix} 2.51e^{-007} & 8.90e^{-008} & 2.73e^{-008} & 7.67e^{-008} & 1.05e^{-007} \\ 8.90e^{-008} & 2.04e^{-007} & 1.47e^{-007} & 1.35e^{-007} & 4.95e^{-008} \\ 2.73e^{-008} & 1.47e^{-007} & 2.79e^{-007} & 1.53e^{-007} & -7.36e^{-009} \\ 7.67e^{-008} & 1.36e^{-007} & 1.53e^{-007} & 2.16e^{-007} & 7.32e^{-008} \\ 1.05e^{-007} & 4.95e^{-008} & -7.36e^{-009} & 7.32e^{-008} & 2.19e^{-007} \end{bmatrix}.$$

Hence, the T^2- Statistic from (3.29) is;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}},$$

$= 51.645.$

Now, $\frac{(n-1)(p-1)}{(n-p+1)} = 5.540541,\ F_{p-1,n-p+1}(0.05) = 2.4697$.

Hence, $\frac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(0.05) = 13.6832$.

From equation (3.30), reject H_0 if ;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1}\mathbf{C}\bar{\mathbf{X}} > \frac{(n-1)(p-1)}{(n-p+1)}F_{p-1,n-p+1}(\alpha).$$

51.645 > 13.6832. There is therefore enough evidence at 5% level of significance to reject H_0 and conclude on H_1 : $\mathbf{C}\mu$ 6= **0**. This means there exist significant difference in the average distances of the various constraints from their neutral pose when Algorithm 2 is used for recognition.

The 95% simultaneous confidence intervals for the estimates of the mean difference under Algorithm 2 are shown in *Table 4.2*.

Table 4.2: Simultaneous Confidence Intervals - Algorithm 2.

Constraints	**Mean Difference**	**Lower**	**Upper**
Angry vs Disgust	$\bar{X}_1 - \bar{X}_2 = 440.2300$	−922.7925	1803.2427
Angry vs Fear	$\bar{X}_1 - \bar{X}_3 = 169.0100$	−1805.9279	2143.9577
Angry vs Happy	$\bar{X}_1 - \bar{X}_4 = 1034.510$	−1114.9863	3184.0031
Angry vs Sad	$\bar{X}_1 - \bar{X}_5 = 877.310$	−828.3633	2582.9919
Angry vs Surprise	$\bar{X}_1 - \bar{X}_6 = -1396.55$	−3145.0346	351.9330
Disgust vs Fear	$\bar{X}_2 - \bar{X}_3 = -271.210$	−2150.1650	1607.7446
Disgust vs Happy	$\bar{X}_2 - \bar{X}_4 = 594.2800$	−1321.6768	2510.2434
Disgust vs Sad	$\bar{X}_2 - \bar{X}_5 = 437.0900$	−1083.5872	1957.7656
Disgust vs Surprise	$\bar{X}_2 - \bar{X}_6 = -1836.80$	−3780.7245	106.9097
Fear vs Happy	$\bar{X}_3 - \bar{X}_4 = 865.4900$	−739.2223	2470.2092
Fear vs Sad	$\bar{X}_3 - \bar{X}_5 = 708.3000$	−1109.0879	2525.6867
Fear vs Surprise	$\bar{X}_3 - \bar{X}_6 = -1565.60$	−3808.7245	677.5930
Happy vs Sad	$\bar{X}_4 - \bar{X}_5 = -157.190$	−2038.4331	1724.0450

*Happy vs Surprise	$\bar{X}_4 - \bar{X}_6 = -2431.10$	−4401.5625	−460.5558
*Sad vs Surprise	$\bar{X}_5 - \bar{X}_6 = -2273.90$	−3745.1675	−802.5626

∗ : *significant difference exist between constraints at* 5% *significance level.*

These simultaneous confidence intervals reveal the specific constraints that are significantly different in their average distances from their neutral poses. Where all confidence intervals that contain 0 are said to be equal and do not have significant differences from their neutral pose at 5% significance level.

It can be seen from *Table 4.2* that, for Algorithm 2, the Happy vs Surprise ($\mu_4 - \mu_6$) and Sad vs Surprise ($\mu_5 - \mu_6$) have confidence intervals, [−4401.5625,−460.5558] and [−3745.1675,−802.5626] respectively. Since these intervals do not contain 0, the respective constraints (Happy vs Surprise and Sad vs Surprise) have significant difference in their average distance from their neutral poses when Algorithm 2 is used as the recognition algorithm. Whitened PCA/SVD (Algorithm 2) had significantly higher average recognition distances in the recognition of Surprise poses as compared to Happy and Sad poses.

Using the 6-variate dataset from Algorithm 3, we have;

$$n = 42,\ \mathbf{C}\bar{\mathbf{X}}) = \begin{bmatrix} 331.919 \\ 30.585 \\ -15.824 \\ 76.751 \\ -520.819 \end{bmatrix},$$

$$\left(\mathbf{C\Sigma C}^{T}\right)^{-1} = \begin{bmatrix} 1.97e^{-006} & 1.68e^{-006} & 1.60e^{-006} & 8.88e^{-007} & 5.83e^{-009} \\ 1.68e^{-006} & 3.66e^{-006} & 3.08e^{-006} & 1.76e^{-006} & 5.46e^{-007} \\ 1.60e^{-006} & 3.08e^{-006} & 5.47e^{-006} & 4.25e^{-006} & 1.65e^{-006} \\ 8.88e^{-007} & 1.76e^{-006} & 4.25e^{-006} & 4.75e^{-006} & 2.04e^{-006} \\ 5.83e^{-009} & 5.46e^{-007} & 1.65e^{-006} & 2.04e^{-006} & 2.66e^{-006} \end{bmatrix}.$$

Hence, the T^2- Statistic from (3.29) is;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^{T}\left(\mathbf{C\Sigma C}^{T}\right)^{-1}\mathbf{C}\bar{\mathbf{X}},$$

$$= 36.695.$$

Now, $\dfrac{(n-1)(p-1)}{(n-p+1)} = 5.540541,\ F_{p-1,n-p+1}(0.05) = 2.4697.$

Hence, $\dfrac{(n-1)(p-1)}{(n-p+1)}F_{p-1,n-p+1}(0.05) = 13.6832.$

From equation (3.30), reject H_0 if ;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^{T}\left(\mathbf{C\Sigma C}^{T}\right)^{-1}\mathbf{C}\bar{\mathbf{X}} > \frac{(n-1)(p-1)}{(n-p+1)}F_{p-1,n-p+1}(\alpha).$$

36.695 > 13.6832. There is therefore enough evidence at 5% level of significance to reject H_0 and conclude on $H_1 : \mathbf{C}\mu \neq \mathbf{0}$. This means there exist significant difference in the average distances of the various constraints from their neutral pose when Algorithm 3 is used for recognition.

The 95% simultaneous confidence intervals for the estimates of the mean difference under Algorithm 3 are shown in *Table 4.3*.

Table 4.3: Simultaneous Confidence Intervals - Algorithm 3.

Constraints	**Mean Difference**	**Lower**	**Upper**

Angry vs Disgust	$\bar{X}_1 - \bar{X}_2 = 275.4800$	−204.2067	868.0441
Angry vs Fear	$\bar{X}_1 - \bar{X}_3 = 472.2400$	−194.1200	919.1281
Angry vs Happy	$\bar{X}_1 - \bar{X}_4 = 190.0800$	−173.5398	866.8997
Angry vs Sad	$\bar{X}_1 - \bar{X}_5 = 387.4100$	−53.8653	900.7276
Angry vs Surprise	$\bar{X}_1 - \bar{X}_6 = 167.9500$	−659.0025	464.2267
Disgust vs Fear	$\bar{X}_2 - \bar{X}_3 = 196.7600$	−457.9496	519.1203
Disgust vs Happy	$\bar{X}_2 - \bar{X}_4 = -85.400$	−506.0024	535.5248
Disgust vs Sad	$\bar{X}_2 - \bar{X}_5 = 111.9300$	−387.6305	570.6553
Disgust vs Surprise	$\bar{X}_2 - \bar{X}_6 = -107.53$	−907.4330	48.8198
Fear vs Happy	$\bar{X}_3 - \bar{X}_4 = -282.16$	−617.7883	586.1401
Fear vs Sad	$\bar{X}_3 - \bar{X}_5 = -84.830$	−329.8673	451.7214
*Fear vs Surprise	$\bar{X}_3 - \bar{X}_6 = -304.29$	−875.7119	−44.0719
Happy vs Sad	$\bar{X}_4 - \bar{X}_5 = 197.3300$	−480.1164	633.6187
Happy vs Surprise	$\bar{X}_4 - \bar{X}_6 = -22.130$	−1005.2381	117.1024
*Sad vs Surprise	$\bar{X}_5 - \bar{X}_6 = -219.46$	−957.8452	−83.7927

$*$: *significant difference exist between constraints at* 5% *significance level.*

These simultaneous confidence intervals reveal the specific constraints that are significantly different in their average distances from their neutral poses. All constraints that contain 0 in their confidence interval are said to be equal and do not have significant differences from their neutral pose at 5% significance level.

It can be seen from *Table 4.3* that, for Algorithm 3, the Fear vs Surprise ($\mu_3 - \mu_6$) and Sad vs Surprise ($\mu_5 - \mu_6$) have confidence intervals, [−875.7119,−44.0719] and [−957.8452,−83.7927] respectively. Since these intervals do not contain 0, the respective constraints (Fear vs Surprise and Sad vs Surprise) have significant difference in their average distance from their neutral poses when Algorithm 3 is used as the recognition algorithm. FFT-PCA/SVD (Algorithm 3) had significantly higher average recognition distance in the recognition Surprise poses as compared to Sad and Fear Constraints.

4.3: Results from Paired Comparisons

The multivariate case is motivated for Algorithm 1 and Algorithm 2, 6 constraints and 42 experimental units. From (3.35), the paired differences random variables become;

$$D_{j1} = X_{j11} - X_{j12},$$

$$D_{j1} = X_{j21} - X_{j22}, \ldots \ \ldots$$

$$D_{j1} = X_{j61} - X_{j62}.$$

Let $\mathbf{D}^T_j = [D_{j1}, D_{j2}, \ldots, D_{j6}]$ and assume for $j = 1,2,\ldots,42$.

$$\bar{\mathbf{D}} = \frac{1}{42}\sum_{j=1}^{42}\mathbf{D}_j = \begin{bmatrix} -896.43 \\ -2345.95 \\ -2369.30 \\ -2092.28 \\ -3060.31 \\ -2034.09 \end{bmatrix}$$ and $$\boldsymbol{\Sigma}_d = \frac{1}{42-1}\sum_{j=1}^{42}\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)^T.$$

$$\boldsymbol{\Sigma}_d = \begin{bmatrix} 1.91e{+007} & 9.05e{+006} & 3.26e^{+006} & 6.54e^{+006} & 7.94e^{+006} & 7.61e^{+006} \\ 9.05e{+006} & 1.66e{+007} & 5.38e^{+006} & 1.26e^{+007} & 9.51e^{+006} & 7.99e{+006} \\ 3.26e{+006} & 5.38e{+006} & 1.04e^{+007} & 7.33e^{+006} & 7.18e^{+006} & 3.23e^{+006} \\ 6.54e{+006} & 1.26e{+007} & 7.33e^{+006} & 2.04e^{+007} & 8.10e^{+006} & 9.54e{+006} \\ & & 7.18e^{+006} & 8.10e^{+006} & 1.34e^{+007} & \\ & & 3.23e^{+006} & 9.54e^{+006} & 9.35e^{+006} & \end{bmatrix}$$

$$\begin{bmatrix} 7.94e{+006} & 9.51e{+006} & \cdots & 9.35e{+006} \\ 7.61e{+006} & 7.99e{+006} & \cdots & 1.47e{+007} \end{bmatrix}$$

$$\mathbf{\Sigma}_d^{-1} = \begin{bmatrix} 7.89e{-008} & -3.10e{-008} & 4.54e{-009} & 8.32e{-009} & -1.97e{-008} & -1.79e{-008} \\ -3.10e{-008} & 1.65e{-007} & 3.23e{-008} & -8.17e{-008} & -8.43e{-008} & 2.62e{-008} \\ 4.54e{-009} & 3.23e{-008} & 2.03e{-007} & -7.19e{-008} & -1.40e{-007} & 7.15e{-008} \\ 8.32e{-009} & -8.17e{-008} & -7.19e{-008} & 1.30e{-007} & 5.89e{-008} & -6.59e{-008} \\ -1.97e{-008} & -8.43e{-008} & -1.40e{-007} & 5.89e{-008} & 2.72e{-007} & -1.24e{-007} \\ -1.79e{-008} & 2.620e{-008} & 7.15e{-008} & -6.59e{-008} & -1.24e{-007} & 1.69e{-007} \end{bmatrix}$$

In addition, $\mathbf{D}_j$, $j = 1,2,...,42$ are independent random individuals $N_6(\mu,\mathbf{\Sigma}_d)$.

Inferences about the mean difference vector, μ can be based on a T^2-statistic from (3.36), $T^2 = n\left(\bar{\mathbf{D}} - \mu\right)^T \mathbf{\Sigma}_d^{-1}\left(\bar{\mathbf{D}} - \mu\right)$.

Given the observed difference, $\mathbf{d}^T_j = [d_{j1}, d_{j2}, ..., d_{j6}]$, $j = 1,2,...,42$ corresponding to the random variables in (3.35), a 5%- level test has hypotheses,

$H_0 : \mu = \mathbf{0}$ (zero mean difference between algorithm 1 and algorithm 2)

$H_1 : \mu \neq \mathbf{0}$

The T^2 statistic is computed as;

$$T^2 = n\bar{\mathbf{d}}^T \mathbf{\Sigma}_d^{-1} \bar{\mathbf{d}} = 35.742.$$

$$\left[\frac{(n-1)\,p}{(n-p)}\right] F_{p,n-p}(\alpha) = \left[\frac{(42-1)\,6}{(42-6)}\right] F_{6,42-6}(0.05) = 16.152.$$

For a $N_6(\mu, \mathbf{\Sigma}_d)$ population, reject H_0 if the observed,

$$T^2(35.742) > \left[\frac{(42-1)\,6}{(42-6)}\right] F_{6,42-6}(0.05) = 16.152.$$

This means H_0 is not tenable, and we therefore have enough evidences at 5% significances level to reject H_0. It can be concluded that, there exist significant difference in the average distances of both algorithms with respect to the study constraints (pose-wise).

The Bonferroni 95% simultaneous confidence intervals for the individual mean differences from (3.38) are; $\mu_i : \bar{d}_i \pm t_{n-1}\left(\frac{\alpha}{2p}\right)\sqrt{\frac{S_{di}^2}{n}}$, where $\bar{d}_i$ is the *ith* element of $\bar{\mathbf{D}}$, S_{di}^2 is the *ith* diagonal element of $\mathbf{\Sigma}_d$ and $t_{n-1}\left(\frac{\alpha}{2p}\right)$ is the upper $100\left(\frac{\alpha}{2p}\right)th$ percentile of a *t*-distribution.

These confidence intervals reveal specifically which constraints have significant differences in euclidean distances when the different face recognition algorithms are used. *Table 4.4*, shows the confidence intervals of estimates for the average of the difference in distance.

Table 4.4: Bonferroni Simultaneous Confidence Intervals (Algorithm 1 & 2)

Constraints	**Average differences**	**Lower**	**Upper**
Angry poses	$\bar{d}_1 = -896.4300$	−2764.2480	971.3904
*Disgust poses	$\bar{d}_2 = -2345.9500$	−4086.8883	−605.0192

*Fear poses	$\bar{d}_3 = -2369.3000$	-3750.5892	-988.0069
*Happy poses	$\bar{d}_4 = -2092.2800$	-4025.6741	-158.8898
*Sad poses	$\bar{d}_5 = -3060.3100$	-4628.7849	-1491.8397
*Surprise poses	$\bar{d}_6 = -2034.0900$	-3673.1152	-395.0652

$*$: *significant difference exist between Algorithms at* 5% *significance level.*

From *Table 4.4*, it can seen that all the confidence intervals do not contain 0 except the confidence interval for Angry pose. This means for Algorithm 1 and Algorithm 2, there exist significant difference in their poses (Disgust, Fear, Happy, Sad and Surprise) recognition except their recognition of the angry pose. $\bar{d}_1$: $[-2764.2480, 971.3904]$ means, there is no significant difference in the average recognition distance on Angry pose between Algorithm 1 and Algorithm 2.

It can therefore be inferred that, at 5% level of significance, Algorithm 1 and Algorithm 2 have significantly different average recognition distances for all poses except Angry pose. PCA/SVD (Algorithm 1) had significantly lower average recognition distances in the recognition of all the study constraints except angry poses when compared to Whitened PCA/SVD (Algorithm 2).

The multivariate case is motivated for Algorithm 1 and Algorithm 3 , 6 constraints and 42 experimental units. From (3.35), the paired differences random variables become;

$$D_{j1} = X_{j11} - X_{j13},$$

$$D_{j1} = X_{j21} - X_{j23}, \ldots \ \ldots$$

$$D_{j1} = X_{j61} - X_{j63}.$$

Let $\mathbf{D}^T_j = [D_{j1}, D_{j2}, \ldots, D_{j6}]$ and assume for $j = 1,2,\ldots,42$.

$$\bar{\mathbf{D}} = \frac{1}{42}\sum_{j=1}^{42}\mathbf{D}_j = \begin{bmatrix} 2209.4 \\ 1801.4 \\ 1201.4 \\ 1426.8 \\ 1495.5 \\ 1141.0 \end{bmatrix}$$ and $$\boldsymbol{\Sigma}_d = \frac{1}{42-1}\sum_{j=1}^{42}\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)^T.$$

$$\boldsymbol{\Sigma}_d = \begin{bmatrix} 6.15e^{+006} & 1.30e^{+006} & 5.93e^{+005} & 1.46e^{+006} & 8.62e^{+004} & 7.34e^{+005} \\ 1.30e^{+006} & 2.15e^{+006} & 3.11e^{+004} & 1.32e^{+006} & 5.27e^{+005} & 4.87e^{+005} \\ 5.93e^{+005} & 3.11e^{+004} & 1.22e^{+006} & -1.24e^{+004} & 3.62e^{+005} & 8.47e^{+004} \\ 1.46e^{+006} & 1.32e^{+006} & -1.24e^{+004} & 2.54e^{+006} & 1.86e^{+005} & 5.90e^{+005} \\ 8.62e^{+004} & 5.27e^{+005} & 3.62e^{+005} & 1.86e^{+005} & 1.64e^{+006} & 3.61e^{+005} \\ 7.34e^{+005} & 4.87e^{+005} & 8.47e^{+004} & 5.90e^{+005} & 3.61e^{+005} & 1.93e^{+006} \end{bmatrix}$$

$$\boldsymbol{\Sigma}_d^{-1} = \begin{bmatrix} 2.14e^{-007} & -8.85e^{-008} & -1.18e^{-007} & -7.21e^{-008} & 6.10e^{-008} & -4.33e^{-008} \\ -8.85e^{-008} & 7.83e^{-007} & 8.78e^{-008} & -3.35e^{-007} & -2.24e^{-007} & -2.35e^{-008} \\ -1.18e^{-007} & 8.78e^{-008} & 9.45e^{-007} & 4.10e^{-008} & -2.39e^{-007} & 1.33e^{-008} \\ -7.21e^{-008} & -3.35e^{-007} & 4.10e^{-008} & 6.28e^{-007} & 5.16e^{-008} & -9.16e^{-008} \end{bmatrix}$$

$$
\begin{bmatrix}
6.10e^{-008} & -2.24e^{-007} & -2.39e^{-007} & 5.16e^{-008} & 7.52e^{-007} & -1.13e^{-007} \\
\vdots & & & & & \vdots \\
-4.33e^{-008} & -2.35e^{-008} & 1.33e^{-008} & -9.16e^{-008} & -1.13e^{-007} & 5.89e^{-007}
\end{bmatrix}
$$

$H_0 : \mu = \mathbf{0}$ (zero mean difference between algorithm 1 and algorithm 3)

$H_1 : \mu \neq \mathbf{0}$

The T^2 statistic is computed as;

$$T^2 = n\bar{\mathbf{d}}^T \mathbf{\Sigma}_d^{-1} \bar{\mathbf{d}} = 138.20.$$

$$\left[\frac{(n-1)\,p}{(n-p)}\right] F_{p,n-p}(\alpha) = \left[\frac{(42-1)\,6}{(42-6)}\right] F_{6,42-6}(0.05) = 16.152.$$

138.20 > 16.152. This means H_0 is not tenable, and we therefore have enough evidences at 5% significances level to reject H_0. It can be concluded that, there exist significant difference in the average distances of a Algorithm 1 and Algorithm 3 with respect to the study constraints (pose-wise).

The Bonferroni 95% simultaneous confidence intervals for the individual mean differences are;

Table 4.5: Bonferroni Simultaneous Confidence Intervals (Algorithm 1 & 3)

Constraints	**Average differences**	**Lower**	**Upper**
*Angry poses	$\bar{d}_1 = 2209.4$	1148.6406	3270.2178
*Disgust poses	$\bar{d}_2 = 1801.4$	1174.1298	2428.7090
*Fear poses	$\bar{d}_3 = 1201.4$	729.0442	1673.8369
*Happy poses	$\bar{d}_4 = 1426.8$	745.4911	2108.2032
*Sad poses	$\bar{d}_5 = 1495.50$	948.0762	2042.8308

*Surprise poses	$\bar{d}_6 = 1141.0$	546.5988	1735.4019

∗ : *significant difference exist between Algorithms at* 5% *significance level.*

Table 4.5, shows that all the confidence intervals do not contain 0. This means for Algorithm 1 and Algorithm 3, there exist significant difference in their poses (Angry, Disgust, Fear, Happy, Sad and Surprise) recognition. It can therefore be inferred that, at 5% level of significance, Algorithm 1 and Algorithm 3 have significantly different average recognition distances for all poses. PCA/SVD (Algorithm 1) had significantly higher average recognition distances in the recognition of all the study constraints when compared to FFT-PCA/SVD (Algorithm 3).

The multivariate case is motivated for Algorithm 2 and Algorithm 3, 6 constraints and 42 experimental units. From (3.35), the paired differences random variables become;

$$D_{j1} = X_{j12} - X_{j13},$$

$$D_{j1} = X_{j22} - X_{j23}, \ldots \ldots$$

$$D_{j1} = X_{j62} - X_{j63}.$$

Let $\mathbf{D}_j^T = [D_{j1}, D_{j2}, \ldots, D_{j6}]$ and assume for $j = 1,2,\ldots,42$.

$$\bar{\mathbf{D}} = \frac{1}{42}\sum_{j=1}^{42}\mathbf{D}_j = \begin{bmatrix} 3105.9 \\ 4147.4 \\ 3570.7 \\ 3519.1 \\ 4555.8 \\ 3175.1 \end{bmatrix}$$ and $$\mathbf{\Sigma}_d = \frac{1}{42-1}\sum_{j=1}^{42}\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)^T.$$

$$7.86e{+006} \quad 6.55e{+006} \quad 2.39e{+006} \quad 3.23e{+006} \quad 5.39e{+006} \quad 4.64e{+006}$$

$$\Sigma_d = \begin{bmatrix} 6.55e{+006} & 1.18e{+007} & 4.75e{+006} & 7.49e{+006} & 7.74e{+006} & 5.80e{+006} \\ 2.39e{+006} & 4.75e{+006} & 8.75e{+006} & 6.65e{+006} & 5.35e{+006} & 1.90e{+006} \\ 3.23e{+006} & 7.49e{+006} & 6.65e{+006} & 1.39e{+007} & 6.73e{+006} & 6.31e{+006} \\ 5.39e{+006} & 7.74e{+006} & 5.35e{+006} & 6.73e{+006} & 1.06e{+007} & 7.47e{+006} \\ 4.64e{+006} & 5.80e{+006} & 1.90e{+006} & 6.31e{+006} & 7.47e{+006} & 1.07e{+007} \end{bmatrix}$$

$$\Sigma_d^{-1} = \begin{bmatrix} 2.75e{-007} & -1.36e{-007} & -1.26e{-008} & 5.51e{-008} & -3.06e{-008} & -5.45e{-008} \\ -1.36e{-007} & 2.52e{-007} & 8.77e{-009} & -7.83e{-008} & -9.02e{-008} & 3.03e{-008} \\ -1.26e{-008} & 8.77e{-009} & 2.62e{-007} & -1.12e{-007} & -1.48e{-007} & 1.24e{-007} \\ 5.51e{-008} & -7.83e{-008} & -1.12e{-007} & 1.80e{-007} & 3.81e{-008} & -9.46e{-008} \\ -3.06e{-008} & -9.02e{-008} & -1.48e{-007} & 3.81e{-008} & 3.52e{-007} & -1.81e{-007} \end{bmatrix}$$

$H_0 : \mu = \mathbf{0}$ (zero mean difference between Algorithm 2 & Algorithm 3)

$H_1 : \mu \neq \mathbf{0}$

The T^2 statistic is computed as;

$$T^2 = n\bar{\mathbf{d}}^T \Sigma_d^{-1} \bar{\mathbf{d}} = 100.79.$$

$$\left[\frac{(n-1)\,p}{(n-p)}\right] F_{p,n-p}(\alpha) = \left[\frac{(42-1)\,6}{(42-6)}\right] F_{6,42-6}(0.05) = 16.152$$

100.79 > 16.152. This means H_0 is not tenable, and we therefore have enough evidences at 5% significances level to reject H_0. It can be concluded that, there exist significant difference in the average distances of Algorithm 2 and Algorithm 3 with respect to the study constraints (pose-wise). The Bonferroni 95% simultaneous confidence intervals for the individual mean differences are;

Table 4.6: Bonferroni Simultaneous Confidence Intervals (Algorithm 2 & 3)

Constraints	Average differences	Lower	Upper
*Angry poses	$\bar{d}_1 = 3105.9$	1906.5197	4305.1962
*Disgust poses	$\bar{d}_2 = 4147.4$	2680.4892	5614.2571
*Fear poses	$\bar{d}_3 = 3570.7$	2305.3044	4836.1728
*Happy poses	$\bar{d}_4 = 3519.1$	1926.3609	5111.8955
*Sad poses	$\bar{d}_5 = 4555.8$	3160.5832	5950.9483
*Surprise poses	$\bar{d}_6 = 3175.1$	1778.6157	4571.5654

$*$: *significant difference exist between Algorithms at* 5% *significance level.*

From *Table 4.6*, it can be seen that all the confidence intervals do not contain 0. This means for Algorithm 2 and Algorithm 3, there exist significant difference in their poses (Angry, Disgust, Fear, Happy, Sad and Surprise) recognition. It can therefore be inferred that, at 5% level of significance, Algorithm 2 and Algorithm 3 have significantly different average recognition distances for all poses. Whitened PCA/SVD (Algorithm 2) had significantly higher average recognition distances in the recognition of all the study constraints when compared to FFT-PCA/SVD (Algorithm 3).

4.4: Results from Box's M Test

For the study populations, the null hypothesis, H_0 and alternative hypothesis, H_1

are;

$H_0 : \mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}_3 = \mathbf{\Sigma},$

$H_1 : \mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j$ for $i \neq j$ (at least two unequal covariance matrices), where $\mathbf{\Sigma}_l$ is the covariance matrix for the lth population, $l = 1,2,3$. Here our populations are the measures from the recognition algorithms and $\mathbf{\Sigma}$ is the presumed common covariance matrix for the populations. The multivariate normality test confirmed that, the collected samples under study are from multivariate normal populations. The Box's test is based on the χ^2 approximation to the sampling distribution of M. Now, $\mathbf{S}_l$, $l = 1,2,3$ are given as;

$$\mathbf{S}_1 = \begin{pmatrix} 1.10e^{+007} & 2.67e^{+006} & 8.71e^{+005} & 3.87e^{+006} & 7.66e^{+005} & 1.71e^{+006} \\ 2.67e^{+006} & 4.47e^{+006} & -6.72e^{+004} & 3.30e^{+006} & 9.24e^{+005} & 1.07e^{+006} \\ 8.71e^{+005} & -6.72e^{+004} & 1.33e^{+006} & -1.82e^{+005} & 2.34e^{+005} & 1.70e^{+005} \\ 3.87e^{+006} & 3.30e^{+006} & -1.82e^{+005} & 5.86e^{+006} & 5.23e^{+005} & 1.86e^{+006} \\ 7.66e^{+005} & 9.24e^{+005} & 2.34e^{+005} & 5.23e^{+005} & 2.44e^{+006} & 4.20e^{+005} \\ 1.71e^{+006} & 1.07e^{+006} & 1.70e^{+005} & 1.86e^{+006} & 4.20e^{+005} & 2.33e^{+006} \end{pmatrix},$$

$$\mathbf{S}_2 = \begin{pmatrix} 6.80e^{+006} & 5.94e^{+006} & 1.91e^{+006} & 2.55e^{+006} & 4.27e^{+006} & 3.82e^{+006} \\ 5.94e^{+006} & 1.08e^{+007} & 4.48e^{+006} & 6.00e^{+006} & 7.19e^{+006} & 4.72e^{+006} \\ 1.91e^{+006} & 4.48e^{+006} & 8.99e^{+006} & 6.78e^{+006} & 4.77e^{+006} & 1.89e^{+006} \\ 2.55e^{+006} & 6.00e^{+006} & 6.78e^{+006} & 1.25e^{+007} & 6.15e^{+006} & 5.40e^{+006} \end{pmatrix},$$

$$\begin{bmatrix} \cdots \\ 3.82e^{+006} & 7.19e^{+006} & 4.77e^{+006} & 6.15e^{+006} & 1.07e^{+007} & 7.14e^{+006} \\ 4.27e^{+006} & 4.72e^{+006} & 1.89e^{+006} & 5.40e^{+006} & 7.14e^{+006} & 1.02e^{+007} \end{bmatrix}$$

$$\mathbf{S}_3 = \begin{bmatrix} 9.36e^{+005} & 3.23e^{+005} & 6.90e^{+004} & 6.07e^{+005} & 2.68e^{+005} & 2.28e^{+005} \\ 3.23e^{+005} & 5.93e^{+005} & 6.13e^{+003} & 4.34e^{+005} & 9.29e^{+004} & 1.90e^{+005} \\ 6.90e^{+004} & 6.13e^{+003} & 1.53e^{+005} & 7.43e^{+004} & -9.00e^{+003} & 5.46e^{+004} \\ 6.07e^{+005} & 4.34e^{+005} & 7.43e^{+004} & 1.11e^{+006} & 2.27e^{+005} & 3.14e^{+005} \\ 2.68e^{+005} & 9.29e^{+004} & -9.00e^{+003} & 2.27e^{+005} & 2.98e^{+005} & 9.96e^{+004} \\ 2.28e^{+005} & 1.89e^{+005} & 5.46e^{+004} & 3.14e^{+005} & 9.96e^{+004} & 4.87e^{+005} \end{bmatrix}.$$

The pooled sample covariance, $\mathbf{S}_{pooled}$ from (3.40);

$$\mathbf{S}_{pooled} = \frac{1}{\sum_{l=1}^{g}(n_l-1)} \sum_{l=1}^{g}(n_l-1)\,\mathbf{S}_l.$$

$$\mathbf{S}_{pooled} = \begin{bmatrix} 6.23e^{+006} & 2.98e^{+006} & 9.49e^{+005} & 2.34e^{+006} & 1.77e^{+006} & 1.92e^{+006} \\ 2.98e^{+006} & 5.28e^{+006} & 1.47e^{+006} & 3.24e^{+006} & 2.74e^{+006} & 1.99e^{+006} \\ 9.49e^{+005} & 1.47e^{+006} & 3.49e^{+006} & 2.23e^{+006} & 1.67e^{+006} & 7.05e^{+005} \\ 2.34e^{+006} & 3.24e^{+006} & 2.23e^{+006} & 6.48e^{+006} & 2.30e^{+006} & \\ 1.77e^{+006} & 2.74e^{+006} & 1.67e^{+006} & 2.30e^{+006} & 4.47e^{+006} & \\ & 1.99e^{+006} & 7.05e^{+005} & 2.53e^{+006} & 2.55e^{+006} & \end{bmatrix}$$

S 2.53e^{+006}

2.34e^{+006}

2.55e^{+006}

$^{+006}$

1.77e

4.35e^{+006}

1.92e^{+006}

From the above, $|\mathbf{S}_1| = 5.6378e^{+038}$, $|\mathbf{S}_2| = 3.8460e^{+040}$, $|\mathbf{S}_3| = 3.0936e^{+033}$, $|\mathbf{S}_{pooled}| = 1.930e^{+039}$ and

$$M = \sum_{l=1}^{g}(n_l - 1)\ln|\mathbf{S}_{pooled}| - \sum_{l=1}^{g}(n_l - 1)\ln|\mathbf{S}_l|$$

$= 474.93.$

From (3.42), Box's test for equality of covariance is motivated as;

$$U = \left[\sum_{l=1}^{g}\frac{1}{(n_l-1)} - \frac{1}{\sum_{l=1}^{g}(n_l-1)}\right]\left[\frac{2p^2+3p-1}{6(p+1)(g-1)}\right],$$

$= 0.068912$, where $p = 6$ is the number of constraints and $g = 3$ is the number of groups (Al-

gorithms).

Now,

$K = (1 - U)M = (1 - 0.068912)474.93,$

$= 442.21$ has an approximate χ^2 distribution with $V = \frac{g}{2}p(p+1) - \frac{1}{2}p(p+1) = \frac{1}{2}p(p+1)(g-1)$ degree of freedom.

Reject H_0 at 0.05% (significance level) if, $K > \chi^2_{\frac{1}{2}p(p+1)(g-1)}(0.05)$.

$$\chi^2_{\frac{1}{2}p(p+1)(g-1)}(0.05) = \chi^2_{42}(0.05)$$

$= 58.1240.$

Clearly, $442.21 > 58.1240$, hence H_0 is not tenable at 5% level of significance. We can therefore conclude that, the covariance matrices of the recognition distances associated with the study algorithms are not the same. That is, $H_1 : \mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j$ for $i \neq$

j is accepted. This means, significant difference exist in the variations of algorithms recognition distances.

4.5: Results from Profile Analysis

For small sample size, profile analysis depends on the normality assumption. The datasets under study are multivariate normal, hence this assumption of normality is satisfied.

Profile analysis also works on the premise of equality of covariance matrices. Here, the pooled covariance is then used as the common covariance for the populations under study. The Box's M test showed that, the covariance matrices of the algorithms under study are unequal. According to Johnson and Wichern (2007), with equal sample sizes, some differences in covariance matrices have have little effect on the test. Also Mettle, Yeboah and Asiedu (2014), demonstrated that profile analysis is still feasible when the H_0 of the Box's M test is not tenable. That is, profile analysis can continue when unequal covariance exist. In this case the separate covariance matrices are used in the computation.

Let $\boldsymbol{\mu_1}^T = [\mu_{11}, \mu_{12}, ..., \mu_{1p}]$ and $\boldsymbol{\mu_2}^T = [\mu_{21}, \mu_{22}, ..., \mu_{2p}]$ be the mean responses to p treatments for Algorithm 1 and 2 respectively.

The null hypothesis $H_0 : \mu_1 = \mu_2$ implies that the treatments have the same average effect on the two populations.

For parallel profiles,

$H_0 : \mu_{1i} - \mu_{1i-1} = \mu_{2i} - \mu_{2i-1},\ i = 1,2,...,p$

H_0 can be written as $H_{01} : \mathbf{C}\boldsymbol{\mu_1} = \boldsymbol{\mu_2}$ where $\mathbf{C}$ is the contrast matrix.

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Now from the study datasets, we have;

$$\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1)^T = \begin{bmatrix} 896.43 & 2345.95 & 2369.30 & 2092.28 & 3060.31 & 2034.09 \end{bmatrix},$$

$$\mathbf{S}_1 = \begin{bmatrix} 1.10e^{+007} & 2.67e^{+006} & 8.71e^{+005} & 3.87e^{+006} & 7.66e^{+005} & 1.71e^{+006} \\ 2.67e^{+006} & 4.47e^{+006} & -6.72e^{+004} & 3.29e^{+006} & 9.24e^{+005} & 1.07e^{+006} \\ 8.71e^{+005} & -6.72e^{+004} & 1.33e^{+006} & -1.82e^{+005} & 2.34e^{+005} & 1.70e^{+005} \\ 3.87e^{+006} & 3.29e^{+006} & -1.82e^{+005} & 5.86e^{+006} & 5.23e^{+005} & 1.86e^{+006} \\ 7.66e^{+005} & 9.24e^{+005} & 2.34e^{+005} & 5.23e^{+005} & 2.44e^{+006} & 4.20e^{+005} \\ 1.71e^{+006} & 1.07e^{+006} & 1.70e^{+005} & 1.86e^{+006} & 4.20e^{+005} & 2.33e^{+006} \end{bmatrix},$$

$$
= \mathbf{S}_2 = \begin{bmatrix}
6.79e^{+006} & 5.94e^{+006} & 1.91e^{+006} & 2.55e^{+006} & 4.27e^{+006} & 3.82e^{+006} \\
5.94e^{+006} & 1.08e^{+007} & 4.48e^{+006} & 6.00e^{+006} & 7.19e^{+006} & 4.72e^{+006} \\
1.91e^{+006} & 4.48e^{+006} & 8.99e^{+006} & 6.78e^{+006} & 4.77e^{+006} & 1.89e^{+006} \\
2.55e^{+006} & 6.00e^{+006} & 6.78e^{+006} & 1.25e^{+007} & 6.15e^{+006} & 5.40e^{+006} \\
4.27e^{+006} & 7.19e^{+006} & 4.77e^{+006} & 6.15e^{+006} & 1.07e^{+007} & 7.14e^{+006} \\
3.82e^{+006} & 4.72e^{+006} & 1.89e^{+006} & 5.40e^{+006} & 7.14e^{+006} & 1.02e^{+007}
\end{bmatrix}.
$$

The sample sizes, $n_1 = n_2 = 42$.

$$
\begin{aligned}
T^2 &= (\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1)^T \mathbf{C}^T \left[\mathbf{C} \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2} \right) \mathbf{C}^T \right]^{-1} \mathbf{C}(\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1) \\
&= 10.459.
\end{aligned}
$$

$$
\begin{aligned}
c^2 &= \left[\frac{(n_1 + n_2 - 2)(p-1)}{n_1 + n_2 - p} \right] F_{p-1, n_1+n_3-p}(0.05) \\
&= \left[\frac{(42 + 42 - 2)(6-1)}{42 + 42 - 6} \right] F_{5,78}(0.05), \\
&= 2.451.
\end{aligned}
$$

Clearly, 10.459 > 2.451 and hence $H_0 : \mu_{1i} - \mu_{1i-1} = \mu_{2i} - \mu_{2i-1}$, $i = 1,2,...,6$ (parallel profiles) is not tenable at 5% significance level. It can therefore be concluded that, the profiles of Algorithm 1 and Algorithm 2 are not parallel. This also means that, the profiles are not coincident and subsequently not level. *Figure 4.4* shows a mean plot of the recognition algorithms.

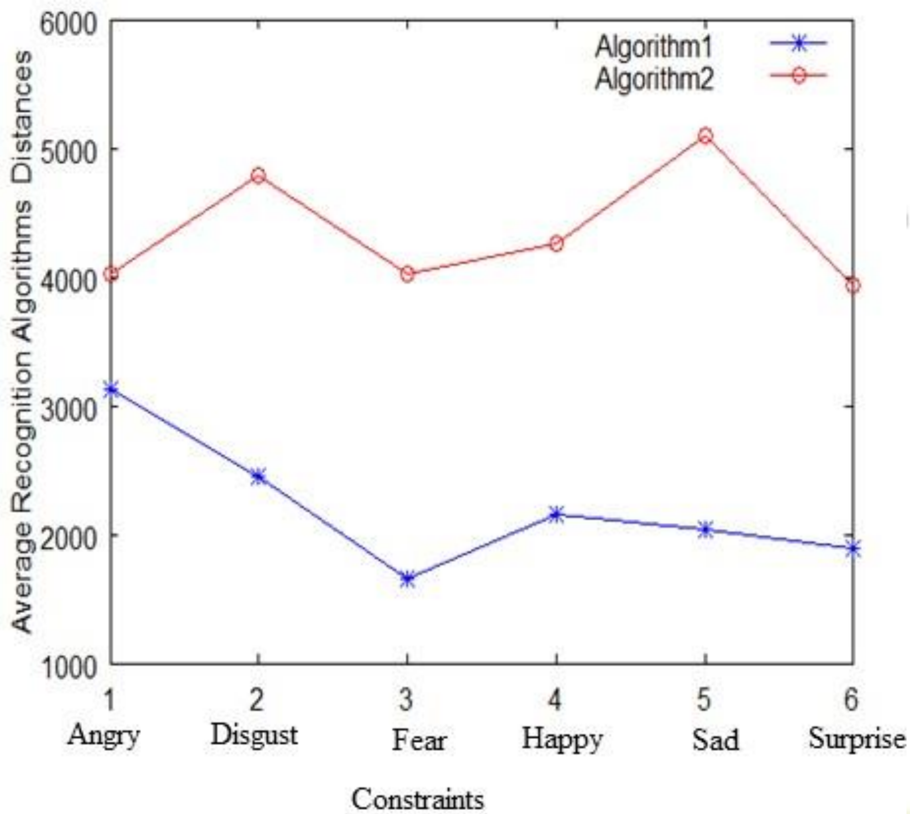


Figure 4.4: Mean plot of recognition distances. (Algorithm 1 & Algorithm 2)

Let $\mu_{\mathbf{1}}{}^{T} = [\mu_{11}, \mu_{12}, \ldots, \mu_{1p}]$ and $\mu_{\mathbf{3}}{}^{T} = [\mu_{31}, \mu_{32}, \ldots, \mu_{3p}]$ be the mean responses to p treatments for Algorithm 1 and 3 respectively.

The null hypothesis $H_0 : \mu_1 = \mu_3$ implies that the treatments have the same average effect on the two populations.

For parallel profiles,

$H_0 : \mu_{1i} - \mu_{1i-1} = \mu_{3i} - \mu_{3i-1}$, $i = 1,2,\ldots,p$

H_0 can be written as $H_{01} : \mathbf{C}\mu_{\mathbf{1}} = \mu_{\mathbf{3}}$ where **C** is the contrast matrix.

Now from the study datasets, we have;

$$\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_1)^T = \begin{bmatrix} -2209.4 & -1801.4 & -1201.4 & -1426.8 & -1495.5 & -1141.0 \end{bmatrix},$$

$$
S_1 = \begin{bmatrix}
1.10e^{+007} & 2.67e^{+006} & 8.71e^{+005} & 3.87e^{+006} & 7.66e^{+005} & 1.71e^{+006} \\
2.67e^{+006} & 4.47e^{+006} & -6.72e^{+004} & 3.29e^{+006} & 9.24e^{+005} & 1.07e^{+006} \\
8.71e^{+005} & -6.72e^{+004} & 1.33e^{+006} & -1.82e^{+005} & 2.34e^{+005} & 1.70e^{+005} \\
3.87e^{+006} & 3.29e^{+006} & -1.82e^{+005} & 5.86e^{+006} & 5.23e^{+005} & 1.86e^{+006} \\
7.66e^{+005} & 9.24e^{+005} & 2.34e^{+005} & 5.23e^{+005} & 2.44e^{+006} & 4.20e^{+005} \\
1.71e^{+006} & 1.07e^{+006} & 1.70e^{+005} & 1.86e^{+006} & 4.20e^{+005} & 2.33e^{+006}
\end{bmatrix}
$$

$$\mathbf{S}_3 = \begin{pmatrix} 9.36e^{+005} & 3.23e^{+005} & 6.90e^{+004} & 6.07e^{+005} & 2.68e^{+005} & 2.28e^{+005} \\ 3.23e^{+005} & 5.93e^{+005} & 6.13e^{+003} & 4.34e^{+005} & 9.29e^{+004} & 1.90e^{+005} \\ 6.90e^{+004} & 6.13e^{+003} & 1.53e^{+005} & 7.43e^{+004} & -9.00e^{+003} & 5.46e^{+004} \\ 6.07e^{+005} & 4.34e^{+005} & 7.43e^{+004} & 1.11e^{+006} & 2.27e^{+005} & 3.14e^{+005} \\ 2.68e^{+005} & 9.29e^{+004} & -9.00e^{+003} & 2.27e^{+005} & 2.98e^{+005} & 9.96e^{+004} \\ 2.28e^{+005} & 1.89e^{+005} & 5.46e^{+004} & 3.14e^{+005} & 9.96e^{+004} & 4.87e^{+005} \end{pmatrix}.$$

The sample sizes, $n_1 = n_3 = 42$.

$$T^2 = (\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_1)^T \mathbf{C}^T \left[\mathbf{C} \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_3}{n_3} \right) \mathbf{C}^T \right]^{-1} \mathbf{C}(\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_1)$$
$$= 7.4215.$$

$$c^2 = \left[\frac{(n_2 + n_3 - 2)(p-1)}{n_1 + n_3 - p} \right] F_{p-1, n_2+n_3-p}(0.05)$$

$$= \left[\frac{(42 + 42 - 2)(6-1)}{42 + 42 - 6} \right] F_{5,78}(0.05),$$
$$= 2.451.$$

Clearly, $7.4215 > 2.451$ and hence $H_0 : \mu_{1i} - \mu_{1i-1} = \mu_{3i} - \mu_{3i-1}$, $i = 1,2,...,6$ (parallel profiles) is not tenable at 5% significance level. It can therefore be concluded that, the profiles of Algorithm 1 and Algorithm 3 are not parallel. This also means that, the profiles are not coincident and subsequently not level. *Figure 4.5* shows a mean plot of the recognition algorithms.

=

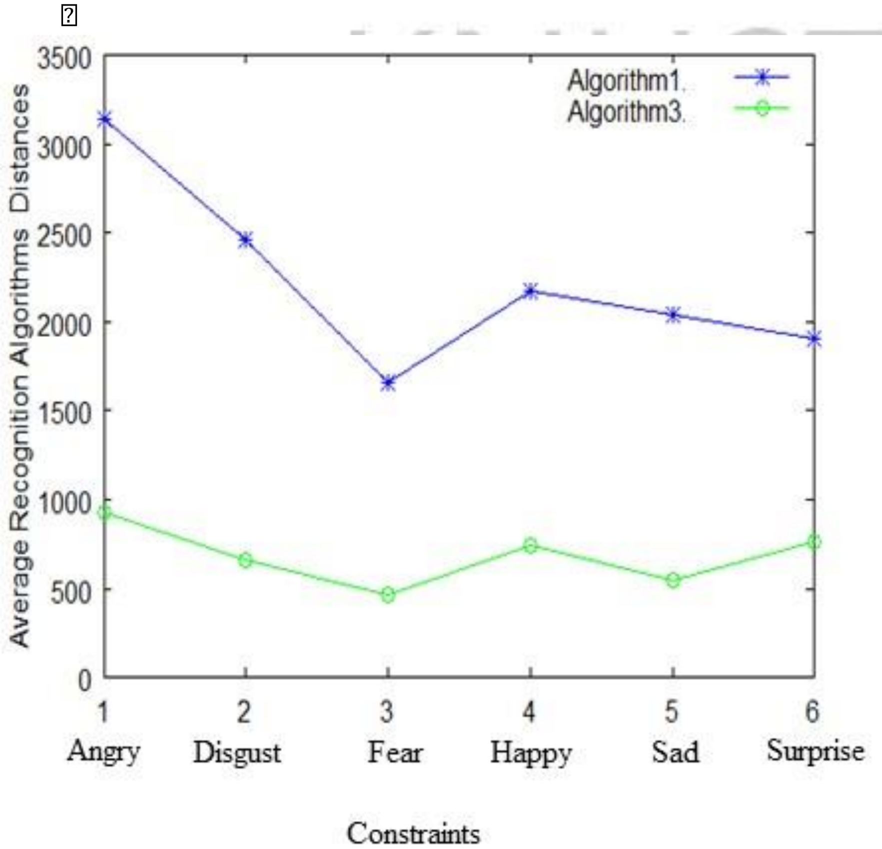


Figure 4.5 : Mean plot of recognition distances. (Algorithm 1 & Algorithm 3)

Let $\boldsymbol{\mu_1}^T = [\mu_{11}, \mu_{12}, \ldots, \mu_{1p}]$ and $\boldsymbol{\mu_3}^T = [\mu_{31}, \mu_{32}, \ldots, \mu_{3p}]$ be the mean responses to p treatments for Algorithm 1 and 3 respectively.

The null hypothesis $H_0 : \mu_1 = \mu_3$ implies that the treatments have the same average effect on the two populations.

For parallel profiles,

$H_0: \mu_{1i} - \mu_{1i-1} = \mu_{3i} - \mu_{3i-1},\ i = 1,2,\ldots,p$

H_0 can be written as $H_{01}: \mathbf{C}\boldsymbol{\mu}_1 = \boldsymbol{\mu}_3$ where $\mathbf{C}$ is the contrast matrix.

Now from the study datasets, we have;

$$(\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_2)^T = \begin{bmatrix} -3105.9 & -4147.4 & -3570.7 & -3519.1 & -4555.8 & -3175.1 \end{bmatrix},$$

$$\mathbf{S}_2 = \begin{bmatrix} 6.79e^{+006} & 5.94e^{+006} & 1.91e^{+006} & 2.55e^{+006} & 4.27e^{+006} & 3.82e^{+006} \\ 5.94e^{+006} & 1.08e^{+007} & 4.48e^{+006} & 6.00e^{+006} & 7.19e^{+006} & 4.72e^{+006} \\ 1.91e^{+006} & 4.48e^{+006} & 8.99e^{+006} & 6.78e^{+006} & 4.77e^{+006} & 1.89e^{+006} \\ 2.55e^{+006} & 6.00e^{+006} & 6.78e^{+006} & 1.25e^{+007} & 6.15e^{+006} & 5.40e^{+006} \\ 4.27e^{+006} & 7.19e^{+006} & 4.77e^{+006} & 6.15e^{+006} & 1.07e^{+007} & 7.14e^{+006} \\ 3.82e^{+006} & 4.72e^{+006} & 1.89e^{+006} & 5.40e^{+006} & 7.14e^{+006} & 1.02e^{+007} \end{bmatrix},$$

$$\mathbf{S}_3 = \begin{bmatrix} 9.36e^{+005} & 3.23e^{+005} & 6.90e^{+004} & 6.07e^{+005} & 2.68e^{+005} & 2.28e^{+005} \\ 3.23e^{+005} & 5.93e^{+005} & 6.13e^{+003} & 4.34e^{+005} & 9.29e^{+004} & 1.90e^{+005} \\ 6.90e^{+004} & 6.13e^{+003} & 1.53e^{+005} & 7.43e^{+004} & -9.00e^{+003} & 5.46e^{+004} \\ 6.07e^{+005} & 4.34e^{+005} & 7.43e^{+004} & 1.11e^{+006} & 2.27e^{+005} & 3.14e^{+005} \\ 2.68e^{+005} & 9.29e^{+004} & -9.00e^{+003} & 2.27e^{+005} & 2.98e^{+005} & 9.96e^{+004} \\ 2.28e^{+005} & 1.89e^{+005} & 5.46e^{+004} & 3.14e^{+005} & 9.96e^{+004} & 4.87e^{+005} \end{bmatrix}.$$

The sample sizes, $n_2 = n_3 = 42$.

$$T^2 = (\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_2)^T \mathbf{C}^T \left[\mathbf{C} \left(\frac{\mathbf{S}_2}{n_2} + \frac{\mathbf{S}_3}{n_3} \right) \mathbf{C}^T \right]^{-1} \mathbf{C}(\bar{\mathbf{X}}_3 - \bar{\mathbf{X}}_2)$$
$$= 17.267.$$

$$c^2 = \left[\frac{(n_2 + n_3 - 2)(p-1)}{n_2 + n_3 - p} \right] F_{p-1, n_2+n_3-p}(0.05)$$

$$= \left[\frac{(42 + 42 - 2)(6-1)}{42 + 42 - 6} \right] F_{5,78}(0.05),$$
$$= 2.451.$$

Clearly, 17.267 > 2.451 and hence $H_0 : \mu_{2i} - \mu_{2i-1} = \mu_{3i} - \mu_{3i-1}$, $i = 1, 2, ..., 6$ (parallel profiles) is not tenable at 5% significance level. It can therefore be concluded that, the profiles of Algorithm 2 and Algorithm 3 are not parallel. This also means that, the profiles are not coincident and subsequently not level. *Figure 4.6* shows a mean plot of the recognition algorithms.

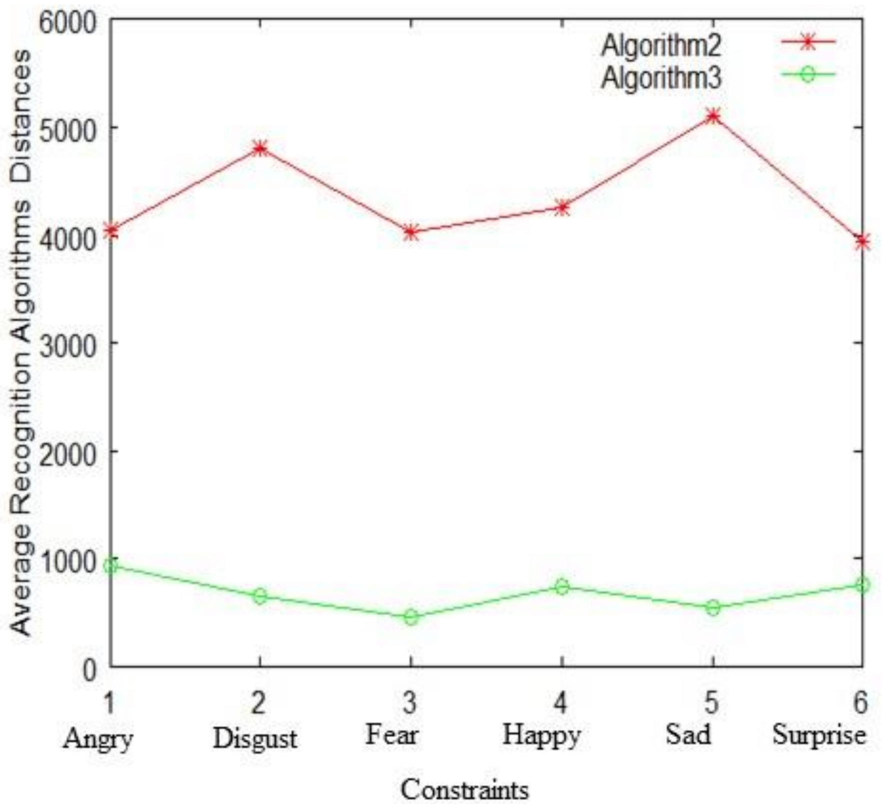


Figure 4.6 : Mean plot of recognition distances. (Algorithm 2 & Algorithm 3)

4.6: Results from Levene's Test of Equality of variances.

The goal of this test is to determine if the Algorithms under study have equal variance in their recognition distances. The test is quite sensitive to the underlying assumption that, the samples been tested should come from a normal population. In this study, two independent normal populations each from the different study algorithms are collected. For example, angry pose data from Algorithm 1 tested against angry pose data from Algorithm 2. Let X_{jk1}, $j = 1,2,\ldots,42$ (individuals) and $k = 1,2,\ldots,6$ (poses) be the datasets from Algorithm 1 and X_{jk2}, $j = 1,2,\ldots,42$ (individuals) and $k = 1,2,\ldots,6$ (poses) be the datasets from Algorithm 2. Now consider two independent normal populations X_{jk1} and X_{jk2}, with unknown variances $\sigma_k{}^2{}_1$ and $\sigma_k{}^2{}_2$.

With samples of size $n_{k1} = 42$, from Algorithm 1, $n_{k2} = 42$, from Algorithm 2 and

their respective sample variance $S_k{}^2{}_1$ and $S_k{}^2{}_2$.

To test $H_0 : \sigma_k{}^2{}_1 = \sigma_k{}^2{}_2$, a 95% confidence interval from (3.49), is given by;

$$\frac{S_{k1}^2}{S_{k2}^2}\ F_{1-\frac{\alpha}{2},n_{2k}-1,n_{1k}-1} < \frac{\sigma_{k1}^2}{\sigma_{k2}^2} < \frac{S_{k1}^2}{S_{k2}^2}\ F_{\frac{\alpha}{2},n_{2k}-1,n_{1k}-1}$$

$$\frac{S_{k1}^2}{S_{k2}^2} = \begin{bmatrix} 1.6142 \\ 0.4141 \\ 0.1479 \\ 0.4697 \\ 0.2283 \\ 0.2274 \end{bmatrix}.$$

$$F_{1-\frac{\alpha}{2},n_{2k}-1,n_{1k}-1} = F_{0.975,41,41} = 0.$$ 5375 and

$$F_{\frac{\alpha}{2},n_{2k}-1,n_{1k}-1} = F_{0.025,41,41} = 1.8604$$.

The 95% confidence intervals for the estimates of the ratio of variances are shown in *Table 4.7*.

Table 4.7: Confidence Intervals for the ratio of variances

(Algorithm 1 & Algorithm 2) .

Constraints	**Ratio of Variances**	**Lower**	**Upper**
Angry poses	6142	0.8676	3.0029
*Disgust poses	4141	0.2226	0.7704
*Fear poses	1479	0.0795	0.2752

*Happy poses	$\frac{S_{11}}{S_{12}} = 1.$ $\frac{S_{21}}{S_{22}} = 0.$ $\frac{S_{31}}{S_{32}} = 0.$ $\frac{S_{41}}{S_{42}} = 0.$ $\frac{S_{51}}{S_{52}} = 0.$ $\frac{S_{61}}{S_{62}} = 0.$4697	0.2525	0.8738
*Sad poses	2283	0.1227	0.4247
*Surprise poses	2274	0.1222	0.4230

$*$: *significant difference exist between Algorithms at* 5% *significance level.*

Reject $H_0 : \sigma_k{}^2{}_1 = \sigma_k{}^2{}_2$, if 1 does not belong to the confidence interval for $\frac{S_{k1}^2}{S_{k2}^2}$, k = 1,2,...,6. Clearly from *Table 4.7*, the confidence interval for Angry poses [0.8676,3.0029] contains 1 and hence H_0 is tenable at 5% significance level. This means the variance of the recognition distances for Angry poses are equal for both algorithms. The remaining constraints (Disgust, Fear, Happy, Sad and Surprise) have confidence intervals that do not contain 1. Here, H_0 is not tenable. This means, the variances of the recognition distances for these poses are not equal for both algorithms.

If H_0 is not tenable and $\frac{S_{k1}^2}{S_{k2}^2} >$ 1, then the variations in Algorithm 1 are greater than that of Algorithm 2 and hence Algorithm 2 is considered as comparatively consistent. Also, if H_0 is not tenable and $\frac{S_{1k}^2}{S_{2k}^2} <$ 1, then the variations in Algorithm 2 are greater than that of Algorithm 1 and hence Algorithm 1 is considered as comparatively consistent.

Now considering the constraints for which H_0 was not tenable (Disgust, Fear, Happy, Sad and Surprise), $\frac{S_{k1}^2}{S_{k2}^2}$ values are given as; 0.4141, 0.1479, 0.4697, 0.2283 and 0.2274 respectively. All these ratios are less than 1 and hence it can be concluded that, the variations in Algorithm 2 are greater than that of Algorithm 1 for all these constraints recognition distance. Subsequently, Algorithm 1 is considered as comparatively consistent in the recognition of Disgust, Fear, Happy, Sad and Surprise poses.

Also, consider two independent normal populations X_{jk1} and X_{jk3}, with unknown

variances $\sigma_k{}^2{}_1$ and $\sigma_k{}^2{}_3$.

With samples of size $n_{k1} = 42$, from Algorithm 1, $n_{k3} = 42$, from Algorithm 3 and their respective sample variance $S_k{}^2{}_1$ and $S_k{}^2{}_3$.

To test $H_0 : \sigma_k{}^2{}_1 = \sigma_k{}^2{}_3$, a 95% confidence interval is given by;

$$\frac{S_{k1}^2}{S_{k3}^2}\ F_{1-\frac{\alpha}{2},n_{k3}-1,n_{k1}-1} < \frac{\sigma_{k1}^2}{\sigma_{k3}^2} < \frac{S_{k1}^2}{S_{k3}^2}\ F_{\frac{\alpha}{2},n_{k3}-1,n_{k1}-1}$$

$$\frac{S_{k1}^2}{S_{k3}^2} = \begin{bmatrix} 11.7120 \\ 7.5465 \\ 8.7200 \\ 5.2894 \\ 8.1794 \\ 4.7753 \end{bmatrix}$$

.

$F_{1-\frac{\alpha}{2},n_{k3}-1,n_{k1}-1} = F_{0.975,41,41} = 0.$ 5375 and

$F_{\frac{\alpha}{2},n_{k3}-1,n_{1k}-1} = F_{0.025,41,41} = 1.8604$.

The 95% confidence intervals for the estimates of the ratio of variances are shown in *Table 4.8*.

Table 4.8: Confidence Intervals for the ratio of variances

(Algorithm 1 & Algorithm 3) .

Constraints	**Ratio of Variances**	**Lower**	**Upper**
*Angry poses	7120	6.2955	21.7888
*Disgust poses	5465	4.0564	14.0394
*Fear poses	7200	4.6872	16.2225

*Happy poses	$\frac{S_{11}}{S_{13}} = 11.$ $\frac{S_{21}}{S_{23}} = 7.$ $\frac{S_{31}}{S_{33}} = 8.$ $\frac{S_{41}}{S_{43}} = 5.$ $\frac{S_{51}}{S_{53}} = 8.$ $\frac{S_{61}}{S_{63}} = 4.$ 2894	2.8432	9.8403
*Sad poses	1794	4.3966	15.2167
*Surprise poses	7753	2.5668	8.8838

$*$: *significant difference exist between Algorithms at* 5% *significance level.*

Reject $H_0 : \sigma_k{}^2{}_1 = \sigma_k{}^2{}_3$, if 1 does not belong to the confidence interval for $\frac{S_{k1}^2}{S_{k3}^2}$, k = 1,2,...,6. Clearly from *Table 4.8*, all the constraints (Angry, Disgust, Fear,

Happy, Sad and Surprise) have confidence intervals that do not contain 1. Here, H_0 is not tenable. This means, the variances of the recognition distances for all the poses are not equal.

Now, $\frac{S_{k1}^2}{S_{k3}^2}$ values are given as; 11.7120, 7.5465, 8.7200, 5.2894, 8.1794 and 4.7753 respectively. All these ratios are greater than 1 and hence it can be concluded that, the variations in Algorithm 1 are greater than that of Algorithm 3 for all constraints recognition distance. Subsequently, Algorithm 3 is considered as comparatively consistent in the recognition of Angry, Disgust, Fear, Happy, Sad and Surprise poses.

4.7: Results from Numerical Evaluations

From equation (3.49), the total number of correct recognition $\sum_{i=1}^{q} n_{cls}^{i}$, for Algorithm 1, Algorithm 2 and Algorithm 3 is 223, 201 and 225 respectively.

The total number of experimental runs, q = 42.

The total number of images in a single experimental runs, n_{tot} = 6.
Hence;

The average recognition rate of Algorithm 1,

$$R_{avg1} = \frac{223}{(42)(6)} \times 100$$
$$= 88.49\%.$$

The average recognition rate of Algorithm 2,

$$R_{avg2} = \frac{201}{(42)(6)} \times 100$$
$$= 79.76\%.$$

The average recognition rate of Algorithm 3,

$$R_{avg3} = \frac{225}{(42)(6)} \times 100$$
$$= 89.29\%.$$

The average runtime of the Algorithm 1, Algorithm 2 and Algorithm 3 in the recognition of the 252 test images are 42 seconds, 184 seconds and 75 seconds respectively.

4.8: Summary

In this chapter, we have presented the results of the assessment of face recognition algorithms considered in this study using the proposed multivariate statistical evaluation methods and the adopted numerical methods. We have also given detailed discussions of the study findings.

CHAPTER FIVE

Conclusion and Recommendations

5.0: Summary

The study modified PCA/SVD algorithm using Whitening and Fast Fourier Transform mechanisms. The Algorithms were numerically evaluated from their recognition rates and run-times. The study also used multivariate statistical methods to evaluate the face recognition algorithms performance on variable constraints (facial expressions).

Given the numerical evaluations in chapter 4, the recognition rates of Principal Component Analysis and Singular Value Decomposition (PCA/SVD), Whitened Principal Component Analysis and Singular Value Decomposition (Whitened-PCA/SVD) and Principal Component

Analysis and Singular Value Decomposition with Fast Fourier Transform as preprocessing step (FFT-PCA/SVD) are 88.49%, 79.76% and 89.29% respectively.

The average runtime of PCA/SVD (Algorithm 1), Whitened PCA/SVD (Algorithm 2) and FFT-PCA/SVD (Algorithm 3) in the recognition of the 252 test images are 42 seconds, 184 seconds and 75 seconds respectively. The time used by Whitened PCA/SVD and FFT-PCA/SVD algorithms in preprocessing accounts for the differences in the algorithms' runtime (speed).

It can be concluded from the results of the repeated measures design that, significance difference exist in each of the study algorithms in recognizing the face images under the study constraints. Specifically, PCA/SVD (Algorithm 1) and Whitened PCA/SVD (Algorithm 2) had significantly higher average recognition distances in the recognition of Surprise poses as compared to Happy and Sad poses. FFTPCA/SVD (Algorithm 3) had significantly higher average recognition distance in the recognition Surprise poses as compared to Sad and Fear Constraints.

The paired comparison results also showed that, PCA/SVD (Algorithm 1) has significantly lower average recognition distances in the recognition of all the study constraints except angry poses when compared to Whitened PCA/SVD (Algorithm 2). PCA/SVD (Algorithm 1) and Whitened PCA/SVD (Algorithm 2) has no significant difference in the recognition of angry poses. PCA/SVD (Algorithm 1) has significantly higher average recognition distances in the recognition of all the study constraints when compared to FFT-PCA/SVD (Algorithm 3). Whitened PCA/SVD (Algorithm 2) also has significantly higher average recognition distances in the recognition of all the study constraints when compared to FFT-PCA/SVD (Algorithm 3).

The Box's M test reported a significant difference in covariance matrix of the study algorithms. This confirms unequal variations among the study algorithms in their recognition of face images under the study constraints.

The Profile Analysis revealed that, the study algorithms' profiles are not parallel. It can be inferred from this results that, the study algorithms have different recognition patterns.

The Levene's test also revealed that, PCA/SVD (Algorithm 1) is comparatively consistent (has lower variation) to Whitened PCA/SVD (Algorithm 2) in the recognition of Disgust, Happy, Fear, Sad and Surprise expressions. No significant difference exist in variation between the Algorithm 1 and Algorithm 2 in recognition of angry pose. In assessing the variation in PCA/SVD (Algorithm 1) and FFTPCA/SVD (Algorithm 3), the Levene's test of equality of variance revealed that the variations in FFT-PCA/SVD (Algorithm 3) are lower than the variations in PCA/SVD (Algorithm 1) in the recognition of all the study constraints.

5.1: Conclusion and Recommendations

The study successfully implemented two modifications of the Principal Component Analysis and Singular Value Decomposition (PCA/SVD) algorithm. These were executed by using:

1. Whitening
2. Fast Fourier Transform

as noise removal mechanisms during the face image preprocessing stage.

To assess the statistically significant differences (which is a gap in almost all researches in the area) among the algorithms, the study successfully used multivariate statistical methods to evaluate the performance of the face recognition algorithms. The study showed that, FFT-PCA/SVD (Algorithm 3) is the most efficient (highest recognition rate) and consistent (lowest variation) in the recognition of face images under variable constraints, followed by PCA/SVD (Algorithm 1) and then Whitened PCA/SVD (Algorithm 2).

Fast Fourier Transform is therefore recommended as a viable noise removal mechanism, which should be adopted during preprocessing stages in image or pattern recognition.

Further studies could be carried out to explore some noise removal mechanisms like, Discrete wavelet transforms on face images that could be adopted for image preprocessing. Another area for future research could be a study on other constraints like; ageing, occlusions, face images in glasses and lightening conditions.

References

Agrawal, S., Khatri, P., & Gupta, S. (2014). Facial Expression Recognition Techniques: A survey. *International Conference on Electrical, Electronic, Computer Science and Mathematics Physical Education & Management, (ICEECMPE),* 6-11. Gwalior, india.

Baker, K. (2005). *Singular value decomposition tutorial.* Retrieved from Dec. 3, 2014 http://www.cs.wits.ac.za/v michael/ SVDTut.pdf

Bartlett, M. S., Donato, G., Ekman, P., Hager, J. C., & Sejnowski, T.J. (1999). Classifying Facial Actions. *IEEE Trans.Pattern Analysis and Machine Intelligence. 21*(10), 974-989.

Barlett, M. S., Movellan, J., & Sejnowski, T. J. (2002). Face recognition by independent component analysis: *IEEE Trans. on Neural networks, 13*(6), 1450-1464.

Bell, A. J., & Sejnowski, T. J. (1995). An information maximization approach to blind separation and blind deconvolution.*Neural Computing, 7*(6), 11291159.

Beveridge, J.R., She, K., Draper, B., & Givens, G. (2001). Parametric and Non-parametric Methods for the Statistical Evaluation of Human ID Algorithms. *IEEE Third Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, HI, USA.

Bhat, N., & Bhavikatti, S. (2012). Facial Expression Recognition using Principal Component Analysis Combined with Fisher Linear Discriminant Analysis. *International Journal of Advanced Research in Electronics and Communication Engineering, 1* (2), India.

Bingham, E., & Manilla, H. (2001). Random projection in dimensionality reduction: Application to image and text data. *In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD),*1-6.

Brandt, D. (2010). Investigation of GPGPU for use in processing of EEG in real-time. *Thesis, RIT Scholar Works.*

Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus template. *IEEE Trans. PAMI, 15*(10), 1042-1062.

Chellapa P., Wilson C., and Sirohey S. (1995). Human and Machine Recognition of Faces: A Survey. *Proceedings of IEEE,* 83(5), 705-740.

Deerwester, S., Dumais, S.T., Furnas, G.W., & Landauer, T. K. (1990). Indexing by latent semantic analysis; *Journal of the Am. Soc. for Information Science, 41*(6), 391-407.

Delac, K., Grgic, M., & Grgic, S. (2005). Statistics in face Recognition: Analyzing probability distributions of PCA, ICA and LDA performance. *Proceedings 4th international symposium on image and signal processing and Analysis. ISPA*, 289-294.

Gentle, J. E.(1998). Singular Value Factorization in Numerical Linear Algebra for Applications in Statistics.102-103. Springer-Verlag, Berlin.

Glynn, E. F. (2007). *Fourier Analysis and Image Processing.* Retrieved Aug.12, 2015 from http://research.stowers-institute.org/efg/Report/FourierAnalysis.pdf

Golub, G. H. & Van Loan, C. F. (1996). The Singular Value Decomposition and Unitary Matrices in Matrix Computations, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 70-71 and 73.

Guillament, D., & Vitri´*a*, J. (2002). Classifying face with non-negative matrix factorization. *Proc. 5th Catalan conference for artificial intelligence,* 336344.

H¨*a*rdle, W., & Simar, L. (2003). *Applied Multivariate Statistical Analysis*, Springer-Verlag, Berlin.

Hyvarinen, A., Karhunen, J. & Oja, E. (2001). Independent Component Analysis, *Neural Networks, 13*(4-5), 411-430.

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques*; Springer, New York, USA.

Johnson, A. R., & Wichern, W. D. (2007). *Applied Multivariate Statistical Analysis.* 6th Ed.: Pearson Prentice Hall, New Jersey.

Jutten, C., & Karhunen, J. (2004). Advance in blind source separation (bss) and independent component analysis (ica) for nonlinear mixtures.*International Journal of Neural System 14(5)*, 267-292.

Kalman, D. (2002). *A singular valuable decomposition: The SVD of a matrix*, Washington, DC: The American University.

Kanade, T. (1973). *Picture processing by computer complex and recognition of human faces, PhD thesis*: Kyoto University.

Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (FG'00), Grenoble, France, 46-53.

Kim, K. (2000). *Face recognition using principal component analysis*, University of Maryland, USA.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Lo´eve procedure for the characterization of human faces *IEEE Trans. Pattern Analysis and Machine Intelligence, 12* (1), 103-108.

Kohonen, T. (1989). *Self-organization and associative memory.* SpringerVerlag, Berlin.

Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. *IEEE ICIP, 1,* 900-903.

Loog, M., Duin, R. P. W., & Hacb-Umbach, R. (2001). Multi class linear dimension reduction by weighted pairwise Fisher criteria.*Pattern recognition and machine intelligence, 23*, 762-766.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB)* San Francisco, USA, 94-101.

McKenna S.J., Gong S. (1997). Tracking Facial Feature Points with Gabor Wavelets and Shape Models. Lecture Notes in Computer Science. *Proc. 1st Int. Conf. on Audio- and Video-based Biometric Person Authentication,* Springer Verlag, Berlin.

Marqu´es, I. (2010). *Face recognition algorithms; Proyecto Fin de, Universidad del Pais Vasco* 1-5.

Mettle, F. O., Yeboah, E., & Asiedu, L. (2014). Profile Analysis of Spatial Differential of Inflation in Ghana. *International Journal of Statistics and Analysis, 4*(2), 245-259.

Naik, R. G., & Kumar, K. D. (2011). An Overview of Independent Component Analysis and its Applications. *Informatica,35*, 63-81.

National Science and Technology Council. (2006). Face Recognition.Retrieved Oct. 6, 2014 from http://www.biometrics.gov/Documents/FaceRec.pdf

Neeta, N., Prashant, G., Nitish, A., Rakesh, P., & Subhash, C. (2008). *Face recognition using PCA and LDA with singular value decomposition(SVD) using 2DLDA, 1* ,1-4.

Phillips, J. P., Flynn, J. P., Scruggs, T., Bowyer, W. K., & Worek, W. (2006). Preliminary Face Recognition Grand Challenge Results. *In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR' 06), IEEE*, 7695-2503.

Rahman, M. U. (2013). *A comparative study on face recognition techniques and neural network, vol abs/*1210-1916

Rencher, A. (2002). *Methods of Multivariate Analysis*; John Wiley & Sons, New York.

Samal, D., & Starovoitov, V. (1998). Approaches and methods to face recognition. A survey, *Institute of Engineering Cybernetics*, Preprint 8, Minsk, 54 pages, (in Russian).

Sharif, M., Khalid, A., Raza, M., & Mohsin, S. (2011). Face recognition using Gabor filters. *Journal of applied computer science & mathematics, 11*(5),
1-5.

Shlens, J. (2003). *A Tutorial on Principal Component Analysis, Derivation, Discussion and Singular Value Decomposition*, version 1, 1-15, Princeton, USA.

Starovoitov, V., Samal, D., Votsis, G., & Kollias, S. (1999). Face recognition by geometric features. *Proceedings of 5-th Pattern Recognition and Information Analysis Conference*, Minsk.

Stergious, A. (2003). *Elastic bunch graph matching face recognition: Performance and comparison with subspace projection methods.* Anthens: Anthens Information Technology.

Thakur, S., Sing, J. K., Basu, D. K., Nasipuri, M., & Kundu, M. (2008). *Face recognition using principal component analysis and RBF neural networks, 10*(5), 1-9.

Turk, M. & Pentland, A. (1991). Eigenface for recognition.*Journal of Cognitive Neuroscience, 3*(1), 71-86.

Voila, P., & Jones, M. J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Feature.*In Proceedings IEEE, Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wagner, P. (2012). *Face recognition with python.* Retrieved November 3, 2014 from http:// www.bytefish.de

Zhang, D., Ding, D., Li, J., & Liu, Q. (2015). PCA Based Extracting Feature Using Fast Fourier Transform for Facial Expression Regression. *Transaction on Engineering Technologies*, 413 - 424. Springer Netherlands.

Appendix 1.0

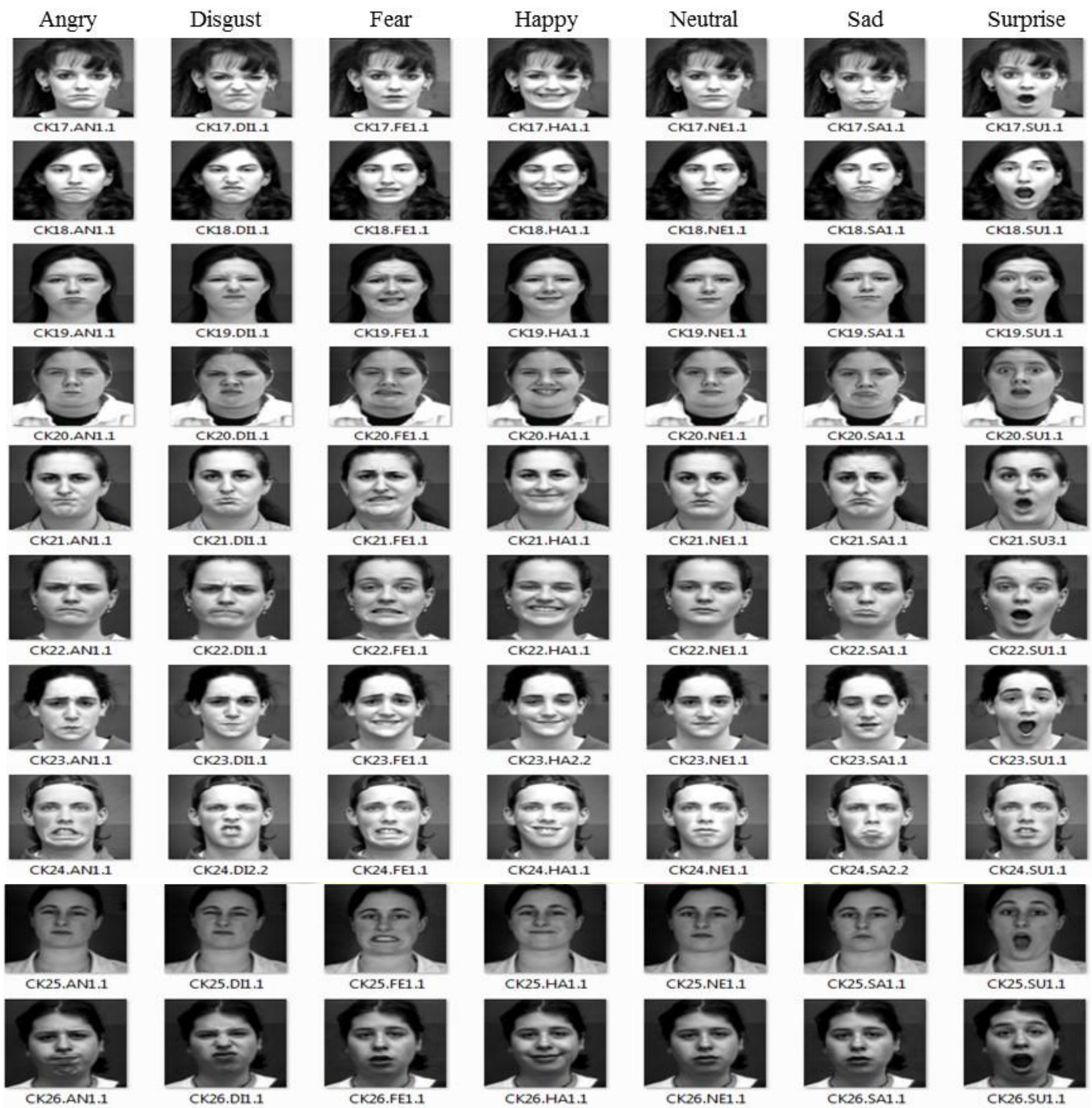
This section contains all the study database collected along the various constraints. The six constraints (Angry, Disgust, Fear, Happy, Sad and Surprise) were used for testing the algorithms whereas the neutral expressions were trained and knowledge captured in memory for recognition.

Appendix 1.0a: CKFE database with the various study constraints.

Source: Kanade, Cohn and Tian (2000); Lucey et al.,(2010)

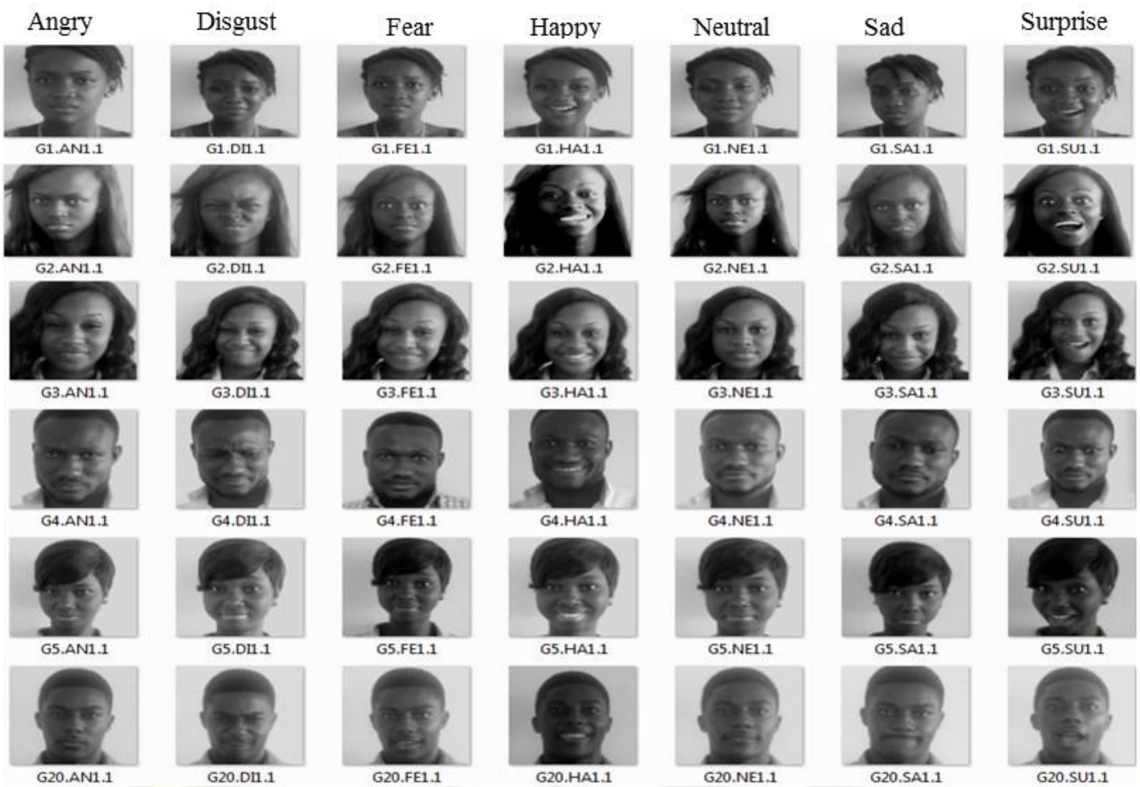
Appendix 1.0a: CKFE database with the various study constraints.

Source: Kanade, Cohn and Tian (2000); Lucey et al.,(2010)

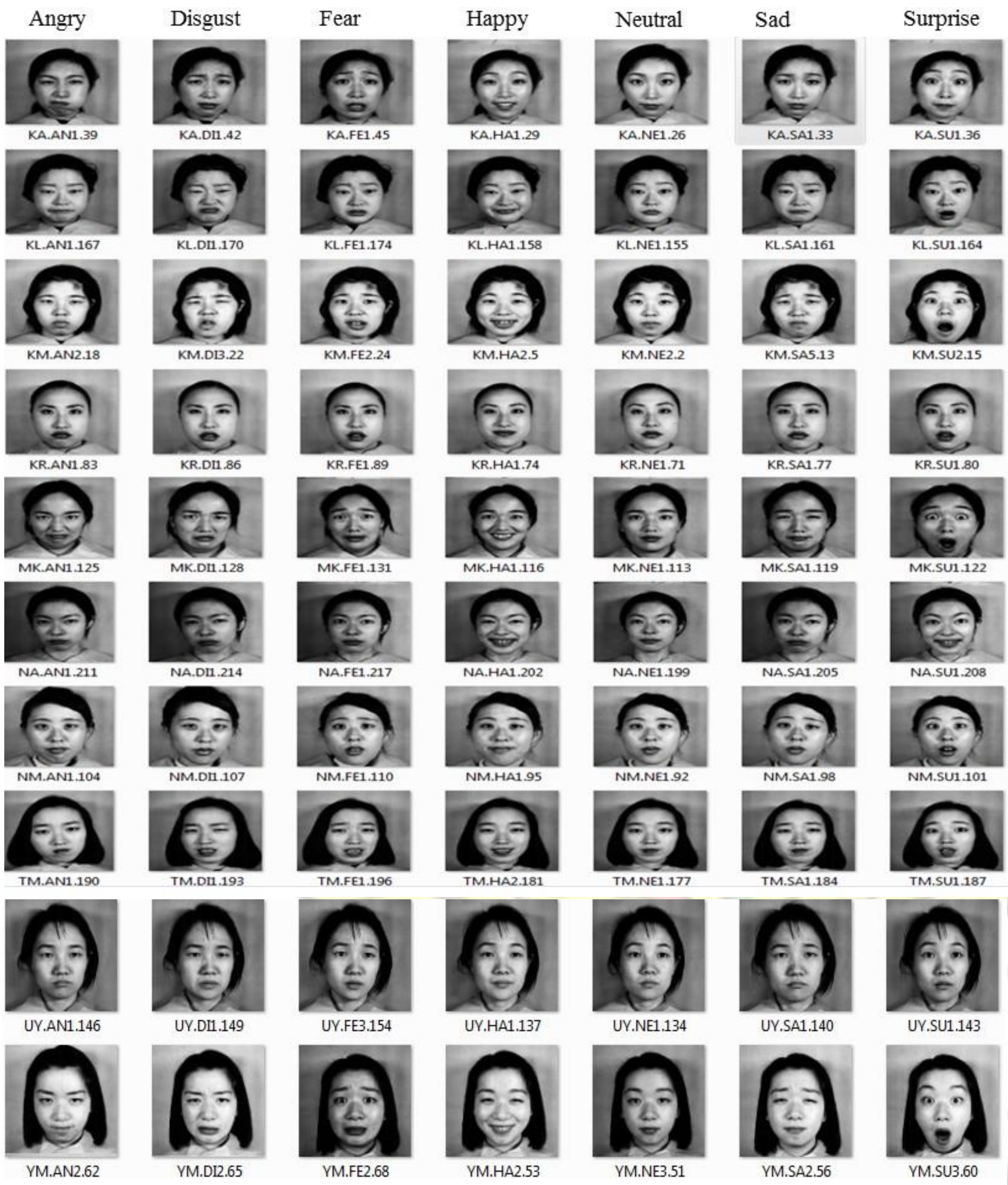


Appendix 1.0a: CKFE database with the various study constraints.

Source: Kanade, Cohn and Tian (2000); Lucey et al.,(2010)



Appendix 1.0b: GFE database with the various study constraints.



Appendix 1.0c: JAFFE database with the various study constraints.

Source: Labeled faces in the wild

Appendix 2.0

This section contains the multivariate data from each of the study algorithms. Here the data been presented are the absolute deviations from the variates means.

Appendix 2.0a: Multivariate data from Algorithm 1

Indiv.	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**

1	3517.10	3312.40	3138.90	382.56	3200.90	3416.70
2	3648.50	3188.90	1369.80	1270.00	977.53	1200.80
3	1066.70	2292.30	944.91	1754.10	153.65	503.84
4	1887.20	1565.20	1968.20	650.66	1592.40	3163.70
5	1310.00	1786.40	714.62	1555.70	1041.40	19.27
6	3344.00	2233.90	396.92	3125.50	1639.30	457.94
7	2026.00	3846.00	2079.90	2066.40	3774.70	1257.60
8	3938.20	1352.10	287.93	603.52	2255.80	2297.10
9	4511.80	2741.50	2001.40	612.89	343.57	555.98
10	3911.80	2740.30	431.02	1892.20	1288.20	3438.50
11	4061.40	3992.90	2417.50	3841.70	3093.60	3190.70
12	1007.30	311.05	1354.20	2343.10	60.55	4302.20
13	641.44	1039.40	1502.30	2711.90	20.16	439.59
14	4498.90	213.60	1725.30	381.99	3122.70	389.05
15	1291.70	649.23	2096.00	2441.90	2137.40	1073.80
16	604.40	662.43	2188.90	450.77	38.29	150.02
17	1592.40	726.54	972.45	3186.60	2050.10	1685.60
18	2443.20	2156.40	968.90	2.48	1099.10	1322.60
19	2367.30	2632.20	1758.50	3362.80	602.98	2763.20
20	4202.90	1733.30	2035.50	802.91	3368.70	2945.40

Appendix 2.0a: Multivariate data from Algorithm 1

Indiv.	Angry	Disgust	Fear	Happy	Sad	Surprise
21	2649.60	2807.10	2181.80	388.48	2203.00	1093.80
22	435.71	1929.40	115.94	1088.90	1845.60	5610.80
23	974.95	1838.10	21.98	2655.70	3448.70	713.35
24	2468.70	2045.50	3403.00	961.96	1571.80	2048.30

25	3914.60	2149.00	2617.90	2952.00	1991.80	2543.70
26	1960.60	832.35	986.10	1915.10	716.68	1099.80
27	8971.90	2449.40	249.59	878.30	3907.00	517.22
28	13519.00	10854.00	421.22	14982.00	1501.90	6786.50
29	4172.60	2069.40	2502.50	1358.10	900.82	1927.70
30	16692.00	703.38	4804.20	2626.10	751.38	2277.00
31	497.09	3554.50	537.24	1834.10	3030.60	978.56
32	577.03	1869.10	6.08	1428.30	342.60	487.20
33	1877.00	2814.10	3018.80	2190.90	430.11	2250.20
34	1876.80	3680.20	3528.10	2197.50	1940.20	2460.50
35	540.55	977.60	3713.80	1454.10	4722.70	3596.40
36	598.35	1004.10	247.22	1411.10	662.00	1216.50
37	2560.70	1592.40	365.71	657.07	2981.10	1623.50
38	4335.30	6486.20	2552.90	1564.10	4242.90	680.90
39	3086.10	9049.10	1717.60	3712.90	3998.40	1340.50
40	490.52	239.92	1581.30	296.72	2493.80	1591.70
41	312.47	1362.20	2670.10	4459.20	2594.10	180.96
42	7593.30	3786.50	2211.30	6672.80	7580.70	4450.40

Appendix 2.0b: Multivariate data from Algorithm 2

Indiv.	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
1	3929.30	1918.00	2907.80	984.55	1742.00	2248.90
2	1637.00	4053.60	921.14	152.62	122.54	114.62
3	2989.60	945.27	1693.90	84.026	3856.20	3642.90
4	1741.70	1329.00	2337.70	272.14	2404.70	4884.50
5	5219.00	7353.30	5542.90	7505.90	6778.40	4785.00
6	4900.50	4418.60	1258.30	7069.30	2382.70	1736.00

7	2098.50	6856.20	64.71	749.39	4640.60	1429.50
8	8407.70	7262.50	1341.10	2030.00	8003.20	7156.00
9	3915.70	5193.40	5625.60	6103.2	5826.00	4427.00
10	3906.40	1869.00	2432.60	42.76	254.16	1532.70
11	7144.20	7174.70	7001.70	7722.80	7771.10	6110.50
12	177.04	562.65	1252.10	1229.10	884.50	3956.10
13	4545.30	6205.10	3990.90	1141.50	8371.80	5793.50
14	6672.50	5363.10	2682.00	3198.50	5694.50	2677.70
15	1094.50	3998.40	3502.70	2484.90	1843.00	2839.90
16	4918.50	5932.70	1625.50	3949.60	4863.00	2854.80
17	2627.60	4213.10	2539.80	751.27	6783.00	352.22
18	12174.0	14077.00	3140.70	8239.00	10686.00	17268.00
19	3018.20	1995.50	2996.90	1304.10	4832.00	2564.10
20	1505.10	2684.60	977.25	4505.30	1772.70	1420.10

Appendix 2.0b: Multivariate data from Algorithm 2

Indiv.	Angry	Disgust	Fear	Happy	Sad	Surprise
21	6934.20	10366.00	8812.90	3080.20	5346.20	907.24
22	5300.40	11536.00	13774.00	8873.70	9510.60	2815.20
23	3083.10	10613.00	5415.70	11820.00	14835.00	116040
24	3338.90	3070.30	3024.40	1157.20	1415.90	574.77
25	7904.70	5899.90	7441.70	7210.00	7234.50	5957.00
26	5393.30	2711.00	3402.50	6448.5	2097.50	2860.90
27	5950.30	6829.80	1126.50	951.82	5932.90	5839.20
28	1158.00	881.27	1109.70	5.1759	1951.90	1178.90
29	9067.70	8645.50	9289.10	9909.8	8146.80	6254.00
30	330.90	6931.30	8995.50	13328	1857.60	2752.80

31	3205.90	3275.20	8366.80	7883.2	7951.60	4761.10
32	3818.00	4746.10	8554.90	3525.5	9534.00	4869.30
33	3071.80	6427.10	3628.70	6806.8	7390.10	2400.40
34	3422.80	4109.70	2993.20	3249.8	3521.30	1940.90
35	2826.60	703.15	20.18	2048	373.76	864.58
36	1199.30	719.29	6833.20	9573.9	6802.50	9171.20
37	6985.20	7310.30	4305.40	7072.3	7992.70	5140.80
38	737.83	119.12	4402.50	931.8	1390.30	3810.90
39	4518.20	6239.70	3104.10	4036.8	4293.30	2421.00
40	304.12	25.30	1876.70	4268.7	6646.20	6396.30
41	4235.90	4675.40	5187.60	2814.8	5867.40	2581.40
42	4217.00	2560.20	3816.80	4485.8	4647.40	2584.20

Appendix 2.0c: Multivariate data from Algorithm 3

Indiv.	Angry	Disgust	Fear	Happy	Sad	Surprise
1	914.00	1059.20	603.90	132.60	869.70	1878.10
2	1339.10	750.80	766.20	550.50	345.90	794.80
3	423.90	769.30	292.50	519.60	34.80	205.90
4	776.80	491.10	481.10	116.30	534.20	860.30
5	603.60	8.10	99.40	457.60	198.40	373.40
6	1164.00	778.40	90.90	1031.80	588.10	64.90
7	490.20	981.40	346.80	410.80	849.80	574.70
8	1045.00	198.00	176.30	269.90	883.40	733.10
9	1494.30	950.80	93.00	194.70	159.70	432.30
10	1081.10	548.20	618.60	364.10	24.10	1002.00
11	1022.70	1278.00	239.10	1104.40	731.40	1046.00
12	545.80	312.10	572.50	851.90	54.40	1056.60

13	9.00	74.10	337.50	577.70	290.80	194.10
14	1437.70	22.70	444.10	50.20	841.60	69.20
15	845.70	491.50	675.50	1034.20	821.60	738.50
16	73.70	434.20	695.20	22.10	805.00	580.00
17	325.90	223.60	70.90	819.30	690.40	11.70
18	957.50	787.40	781.00	184.60	121.80	1161.20
19	402.40	728.50	207.50	1015.80	124.50	907.20
20	790.40	34.90	343.5	314.90	604.10	663.70

Appendix 2.0c: Multivariate data from Algorithm 3

Indiv.	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
21	20.10	666.30	24.70	611.80	3.90	95.70
22	406.10	646.60	570.70	376.70	179.60	233.30
23	300.40	419.30	535.80	760.30	963.60	221.40
24	178.80	319.80	1797.50	72.60	97.80	458.20
25	1034.10	549.60	668.50	799.60	284.20	1100.00
26	1175.80	10.40	102.90	599.40	66.90	488.90
27	3338.70	798.90	20.00	588.10	2042.40	90.70
28	4395.08	3716.90	913.30	6307.70	1258.80	2771.80
29	882.10	110.70	776.80	2643.50	147.90	84.50
30	3465.90	86.30	1294.60	314.60	138.40	1335.70
31	784.10	1110.00	1543.00	1315.40	1010.50	717.30
32	42.40	811.50	218.50	637.40	782.90	3437.00
33	144.80	164.00	244.50	338.40	466.20	204.70
34	686.10	218.10	432.10	672.80	492.00	729.30
35	451.30	233.70	488.70	209.70	540.60	985.10
36	76.30	38.10	324.50	398.00	184.10	866.70

37	766.00	332.30	263.50	428.20	487.90	650.40
38	420.10	856.60	368.60	52.40	445.60	214.00
39	837.40	3610.70	23.20	41.30	15.90	303.70
40	135.00	364.30	56.10	556.20	837.50	833.40
41	776.50	330.40	322.80	616.70	115.30	1195.80
42	3120.10	1294.20	421.40	2834.10	2774.40	1760.80

Appendix 3.0

This section contains the squared distances and their respective quantiles for the chi-square plot.

The datasets for multivariate normality evaluation of the study algorithms are shown below.

Appendix 3.0a: Squared distances and quantiles data.

j	$q_{c,6}\left(\frac{(j-\frac{1}{2})}{42}\right)$	$d^2{}_j$(Algorithm 1)	$d^2{}_j$(Algorithm 2)	$d^2{}_j$(Algorithm 3)
1	0.9309	1.1169	0.4006	0.4561
2	1.4256	1.4555	0.5101	0.4571
3	1.7590	1.4709	1.1284	0.6100
4	2.0327	1.5880	1.1826	0.8726
5	2.274	1.6572	1.2899	0.9009
6	2.4948	1.7222	1.8178	0.9253
7	2.7018	1.7367	1.8638	1.0820
8	2.8989	1.8584	2.0949	1.1538
9	3.0888	2.1090	2.1368	1.2055
10	3.2736	2.3079	2.1700	1.3280
11	3.4546	2.3546	2.2330	1.3859
12	3.6331	2.4071	2.3981	1.5639
13	3.8099	2.4314	2.6938	1.6791
14	3.9860	2.5096	2.9791	1.7177
15	4.162	2.5188	3.2173	1.8008

16	4.3386	2.7458	3.5903	1.8212
17	4.5165	2.8275	3.6107	1.8569
18	4.6961	2.8328	3.7163	1.9964
19	4.8782	2.9505	3.8409	2.0061
20	5.0633	3.0049	4.1705	2.0081

Appendix 3.0a: Squared distances and quantiles data.

j	$q_{c,k}\left(\frac{(j-\frac{1}{2})}{n}\right)$	$d^2{}_j$(Algorithm 1)	$d^2{}_j$(Algorithm 2)	$d^2{}_j$(Algorithm 3)
21	5.2521	3.0146	4.3411	2.2089
22	5.4453	3.3013	4.5537	2.3006
23	5.6435	3.3816	4.9833	2.4504
24	5.8476	3.5698	4.9972	2.5268
25	6.0585	3.8197	5.2058	2.7402
26	6.2773	3.8536	5.2160	3.2037
27	6.5051	3.8903	5.2509	3.2489
28	6.7433	3.9978	5.5978	3.2715
29	6.9937	4.2519	5.6825	3.3139
30	7.2583	4.5491	6.5734	3.4713
31	7.5396	4.9629	6.7241	4.1323
32	7.8408	5.3763	6.9436	4.2687
33	8.1660	7.2484	7.5550	5.6077
34	8.5207	9.3790	7.6428	12.2082
35	8.9123	9.8304	7.8031	12.3034
36	9.3513	11.0057	8.2394	14.7236
37	9.8536	12.4019	10.8247	17.0344
38	10.4444	13.4871	11.8157	17.4526
39	11.1676	14.3062	14.4085	21.8080

40	12.1117	17.4006	18.3022	22.2702
41	13.5026	27.3446	19.4550	27.3479
42	16.3690	32.0215	26.8394	31.2795

Appendix 4.0: Numerical Codes

4.1: Recognition Codes for PCA/SVD (Algorithm 1)

```
function[A,faces,imgtrace,Run_time]=Output1(num_indv,num_constraints)
                  tic();
       train_imgDir = 'C:\Octave\3.2.4_gcc-4.4.0\bin\trainimg\';
           imgtrain = dir(fullfile( train_imgDir, '*.tiff'));
                    %im_orig = [];
               data_o = [];
               for i = 1:length(imgtrain)              img =
   imread([train_imgDir imgtrain(i).name]);
                     %im_orig = [im_orig img];
                  data_o = [data_o img(:)];
                  end
                    % normalizing the images
     datat=double(data_o./255);
                         dat=datat';
                       m=mean(dat);
                    meanimg=reshape(m',256,256);
     w=bsxfun(@minus,datat,m');

   %Looking for the eigenvectors through singular value decomposition
                       c=cov(w);
```

```
[U S V] =svd(c); % Singular Value Decomposition
% eigenface display
ei=w*U;
%eigenface=[];
%for k=1:length(imgtrain)
%eface=reshape([ei(:,k)],256,256);
%eigenface=[eigenface eface];
%end

%Principal components of the trained images
faces= ei'*w;

% Recognition Procedure
% Get list of ALL TIFF files in this directory
test_imgDir = 'C:\Octave\3.2.4_gcc-4.4.0\bin\testimg\';
test_images = dir(fullfile( test_imgDir, '*.tiff'));

%To count the number of TIFF files in the directory
my_counter = length(test_images);
A = [];
C = [];
for i=1:my_counter            currentfilename =
test_images(i).name;          currentimage = imread([test_imgDir
currentfilename]);
x7 = currentimage;
```

```
d7=x7(:);
d7t=double(d7./255);
w7=bsxfun(@minus,d7t,m');                    face7=ei'*w7;
rmat =bsxfun(@minus,faces,face7);
imgtrace= [];
for r = 1:length(imgtrain)
    imgtrace = [imgtrace norm(rmat(:,r))];
end

%Categorization process
idvalue = min(imgtrace);
A = [A idvalue];                    for
l=1:length(imgtrace)                    if
idvalue==imgtrace(:,l)
img=data_o(:,l);
img1=reshape(img,256,256);                    C = [C
currentimage,img1];
disp('correct match')
        else
            disp ('Wrong match')
    end
end
    disp ("\n")
  end
A = reshape(A,num_constraints,num_indv);
```

```
                A = A';
                %imagesc(C);
                dlmwrite('boxesm\Algorithm1.csv',A);
           Run_time = toc();
```

4.2: Recognition Codes for Whitened PCA/SVD (Algorithm 2)

```
function[A,faces,imgtrace,Run_time]=White1(num_indv,num_constraints)
                tic();
                %reading of input images
    train_imgDir = 'C:\Octave\3.2.4_gcc-4.4.0\bin\trainimg\';
         imgtrain = dir(fullfile( train_imgDir, '*.tiff'));
              %im_orig = [];
         data_o = [];
              for i = 1:length(imgtrain)                   img =
  imread([train_imgDir imgtrain(i).name]);
                   %im_orig = [im_orig img];
                   data_o = [data_o img(:)];
                end
              % normalizing the images
         datat=double(data_o./255);
    m=mean(datat);
    w=bsxfun(@minus,datat,m);

  %Looking for the eigenvectors through singular value decomposition
```

```
            c=cov(w);
            [E, D] =eig(c);
         % Whitening Preprocessing
b=E*sqrt(inv(D))*E';
            B=b*datat';
            %Whiteimg=[ ];
           %for j=1:length(imgtrain)
                %White_B=reshape([B'(:,j)],256,256);
                %Whiteimg=[Whiteimg White_B ];
         %end

         %Display of eigenfaces mb=
         mean(B');
            Bnew =bsxfun(@minus,B,mb');
            Ev = Bnew*Bnew';
            [U S V]=svd(Ev); % Singular Value Decomposition
            ei= Bnew'*U;
            %eigenface=[];
            %for k=1:length(imgtrain)
                 %eface=reshape([ei(:,k)],256,256);
                 %eigenface=[eigenface eface];
            %end

         %Principal components of the trained images
faces = ei'*Bnew';
```

```
% Recognition Procedure
%Get list of ALL TIFF files in this directory
test_imgDir = 'C:\Octave\3.2.4_gcc-4.4.0\bin\testimg\';
test_images = dir(fullfile( test_imgDir, '*.tiff'));

%To count the number of TIFF files in the directory
my_counter = length(test_images);
A = [];
C = [];
for i=1:my_counter  currentfilename =
test_images(i).name; currentimage = imread([test_imgDir
currentfilename]);
x7 = currentimage;
d7=x7(:);                    d7t=double(d7./255);
Wnew =b*(repmat(d7t,1,length(imgtrain)))';
w7=bsxfun(@minus,Wnew,mb');
face7=ei'*w7';
rmat =bsxfun(@minus,faces,face7);
imgtrace= [];
for r = 1:length(imgtrain )
imgtrace = [imgtrace norm(rmat(:,r))];
end
```

```
%Categorization process
idvalue = min(imgtrace);
A = [A idvalue];                    for
l=1:length(imgtrace)                          if
idvalue==imgtrace(:,l)
img=data_o(:,l);
img1=reshape([img],256,256);
C = [C  currentimage,img1];
disp("correct match")
else
disp ("Wrong match")      endif

end
disp ("\n")
end
A = reshape(A,num_constraints,num_indv);
A = A';
%imagesc(C);
dlmwrite('boxesm\Algorithm2.csv',A);
Run_time = toc();
```

4.3: Recognition Codes for FFT-PCA/SVD (Algorithm 3)

```
function[A, faces,imgtrace,
Run_time]=fftpca(num_indv,num_constraints)
tic();
```

```
%reading of input images
train_imgDir = 'C:\Octave\3.2.4_gcc-4.4.0\bin\trainimg\';
imgtrain = dir(fullfile( train_imgDir, '*.tiff'));

%im_orig = [];
%Fourier Transform Preprocesing
data_o = [];
for i = 1:length(imgtrain)                    img =
imread([train_imgDir imgtrain(i).name]);
%im_orig = [im_orig img];
img1=fft(img);
ft1=real(ifft(img1));                    data_o =
[data_o ft1(:)];
end

datat=double(data_o./255);

%Fourier=[ ];
%for j=1:length(imgtrain)
    %Fourier_B=reshape([datat(:,j)],256,256);
    %Fourier=[Fourier Fourier_B ];
%end

%Display of eigenfaces
mb= mean(datat');
```

```
Bnew =bsxfun(@minus,datat,mb');
Ev = Bnew'*Bnew;
[U S V]=svd(Ev); % Singular Value Decomposition
ei= Bnew*U;
%eigenface=[];
%for k=1:length(imgtrain)
    %eface=reshape([ei(:,k)],256,256);
    %eigenface=[eigenface eface];
%end

%Principal components of the trained images
faces = ei'*Bnew;

% Recognition Procedure
%Get list of ALL TIFF files in this directory test_imgDir =
'C:\Octave\3.2.4_gcc-4.4.0\bin\testimg\'; test_images =
dir(fullfile( test_imgDir, '*.tiff'));
%To count the number of TIFF files in the directory
my_counter = length(test_images);
A = [];
C = [];            for i=1:my_counter
currentfilename = test_images(i).name;          currentimage =
imread([test_imgDir currentfilename]);
```

```
x7 = currentimage;
dd7=fft(x7);                    ft2=real(ifft(dd7));
d7=ft2(:);
d7t= double(d7./255);
Wnew = repmat(d7t,1,length(imgtrain));
w7=bsxfun(@minus,Wnew,mb');
face7=ei'*w7;
rmat =bsxfun(@minus,faces,face7);
imgtrace= [];
for r = 1:length(imgtrain )
imgtrace = [imgtrace norm(rmat(:,r))];
end

%Categorization process
idvalue = min(imgtrace);
A = [A idvalue];
for l=1:length(imgtrace)
if idvalue==imgtrace(:,l)
imgt=data_o(:,l);
img2=reshape([imgt],256,256);
C = [C  currentimage, img2];
disp("correct match")
else
disp("Wrong match")
```

```
                        endif
    end
                        disp ("\n")
                  end
                  A = reshape(A,num_constraints,num_indv);
               A = A';
%imagesc(C);
                  dlmwrite('boxesm\Algorithm3.csv',A);
                  Run_time = toc();
```

Appendix 5.0: Statistical Codes 5.1: Assessing Multivariate Normality

```
function [ r chisq dsquares ] = Normality(filename)
  a = csvread(['boxesm\',filename]);      m=mean(a);
w=abs(bsxfun(@minus,a,m));      w1=bsxfun(@minus,w,mean(w));
  s=cov(w);
       ins=inv(s);   d=w1*ins*w1';   A =
     diag(d);   dsquares = sort(A);
     chisq = load(['tables\'
     'chisq.csv']);   r= cor(chisq,
     dsquares);   plot(chisq,
     dsquares,'markersize',3,'*');
     xlabel('quantiles');
     ylabel('dsquares');
     title('Evaluating Multivariate
     Normality');     legend(filename);
```

```
end
```

5.2: Repeated Measures Design

```
function[tsquare F_table] = RepeatedMeasures(filename)
          X = csvread(['boxesm\' filename]);
          U = mean(X);
          X = abs(bsxfun(@minus,X,U))

                  C = [ 1 -1  0  0  0  0;
                  0  1 -1  0  0  0;
                  0  0  1 -1  0  0;
                  0  0  0  1 -1  0;
                  0  0  0  0  1 -1;
                  ];
          CX_bar = C*U';
       s = cov(X);
       [n p] = size(X);
       csc = C*cov(X)*C';   ins_csc =
     inverse(csc);   tsquare =
     n*CX_bar'*ins_csc*CX_bar;   F_table =
     13.683194;
            if tsquare > F_table
disp('Reject the null hypothesis Ho');
               else
                  disp('Fail to reject the null hypothesis Ho');
          end
```

```
        CIs = [];

        CI = [];

  C = [

                1 -1  0  0  0  0;

                1  0 -1  0  0  0;

                1  0  0 -1  0  0;

                1  0  0  0 -1  0;

                1  0  0  0  0 -1;

                ];

        for i=1:p-1

      lower = C(i,:)*U' - sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);

      upper = C(i,:)*U' + sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);

                if i == 1

b = ['Angry vs Disgust ' sprintf("%f ",lower) sprintf("%f
",upper)];

                elseif i == 2

b = ['Angry vs Fear ' sprintf("%f  ",lower) sprintf("%f    ",upper)];

              elseif i == 3

b = ['Angry vs Happy ' sprintf("%f ",lower) sprintf("%f
",upper)];

                elseif i == 4

b = ['Angry vs Sad  ' sprintf("%f ",lower)  sprintf("%f
",upper)];

                elseif i == 5

b = ['Angry vs Surprise ' sprintf("%f ",lower) sprintf("%f
",upper)];

                 end
```

```
          CI = [CI; b];
end

           CIs = [CIs; CI];
            CIs = [CIs; [ ' '  ' '  ' ']];
            CI = [];
            C = [
                 0  0  0  0  0  0;
                 0  1 -1  0  0  0;
                 0  1  0 -1  0  0;
                 0  1  0  0 -1  0;
                 0  1  0  0  0 -1;
                 ];
          for i=2:p-1

      lower = C(i,:)*U' - sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);
      upper = C(i,:)*U' + sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);
                 if i == 2
b = ['Disgust vs Fear ' sprintf("%f ",lower) sprintf("%f
",upper)];
                      elseif i == 3
b = ['Disgust vs Happy ' sprintf("%f ",lower) sprintf("%f
",upper)];
                      elseif i == 4
b = ['Disgust vs Sad ' sprintf("%f  ",lower) sprintf("%f
```

```
",upper)];

                       elseif i == 5

b = ['Disgust vs Surprise ' sprintf("%f ",lower) sprintf("%f
",upper)];

                  end


                CI = [CI; b];

end


               CIs = [CIs; CI];

               CIs = [CIs; [ ' '  ' '  '']];

               CI = [];


               C = [

                       0  0  0  0  0  0;

                       0  0  0  0  0  0;

                       0  0  1 -1  0  0;

                       0  0  1  0 -1  0;

                       0  0  1  0  0 -1;

                       ];


for i=3:p-1

lower = C(i,:)*U' - sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n); upper =

C(i,:)*U' + sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);

                  if i == 3

b = [ 'Fear vs Happy  ' sprintf("%f ",lower) sprintf("%f
```

```
",upper)];

                        elseif i == 4

b = [ 'Fear vs Sad     ' sprintf("%f ",lower) sprintf("%f
",upper)];

                        elseif i == 5

b = [ 'Fear vs Surprise ' sprintf("%f ",lower) sprintf("%f
",upper)];

                   end


                CI = [CI; b];

end


          CIs = [CIs; CI];

          CIs = [CIs; [ ' '  ' '  '']];

          CI = [];


             C = [

                  0  0  0  0  0  0;

                  0  0  0  0  0  0;

                  0  0  0  0  0  0;

                  0  0  0  1 -1  0;

                  0  0  0  1  0 -1;

                  ];

           for i=4:p-1

lower = C(i,:)*U' - sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n); upper =

C(i,:)*U' + sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);
```

```
                  if i == 4

b = [ 'Happy vs Sad ' sprintf("%f  ",lower) sprintf("%f
",upper)];

                       elseif i == 5

b = [ 'Happy vs Surprise  ' sprintf("%f ",lower) sprintf("%f
",upper)];

                  end

                CI = [CI; b];

end

          CIs = [CIs; CI];

          CIs = [CIs; [ ' '  ' '  '']];

          CI = [];

             C = [

                  0  0  0  0  0  0;

                  0  0  0  0  0  0;

                  0  0  0  0  0  0;

                  0  0  0  0  0  0;

                  0  0  0  0  1 -1;

                  ];

             for i=5:p-1


lower = C(i,:)*U' - sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n); upper =

C(i,:)*U' + sqrt(F_table)*sqrt((C(i,:)*s*C(i,:)')/n);

                  if i == 5

b = [ 'Sad vs Surprise ' sprintf("%f ",lower) sprintf("%f
",upper)];

               end
```

```
            CI = [CI; b];       end

        CIs = [CIs; CI]    end
```

5.3: Paired Comparison function[Hotteling_Tsq F] = PairedComparison(data1,data2)

```
     X1 = csvread(['boxesm\' data1]);   X2 =
csvread(['boxesm\' data2]);   m1=mean(X1);
m2=mean(X2);
        X1=abs(bsxfun(@minus,X1,m1));
     X2=abs(bsxfun(@minus,X2,m2));      d =
X1-X2;    [n p] = size(d);    k = ((n-
1)*p)/(n-p);

        %Using 5% level of significance
     F_table = 2.363750958;   F=
k*F_table;     bar_d = mean(d);    sd =
cov(d)    ins = inv(sd)
 Hotteling_Tsq   = n*bar_d*ins*bar_d'; if Hotteling_Tsq > F
     disp('Reject the null hypothesis Ho');
     else      disp('Fail to reject the null hypothesis Ho'); end
var = diag(sd);
     %Setting confidence interval      CI_Bonferroni = [];
student_t = 2.77235376;       for i=1:p          lower =
bar_d(i)-student_t*sqrt(var(i,:)/n);         upper =
bar_d(i)+student_t*sqrt(var(i,:)/n);
```

```
            if i == 1
b = [ 'Angry pose ' sprintf('%f    ',lower) sprintf('%f
',upper)];
                 elseif i == 2
b = [ 'Disgust pose  ' sprintf('%f  ',lower) sprintf('%f
',upper)];
                 elseif i == 3
b = [ 'Fear pose     ' sprintf('%f  ',lower) sprintf('%f
',upper)];
                 elseif i == 4
b = [ 'Happy pose    ' sprintf('%f  ',lower) sprintf('%f
',upper)];
                 elseif i == 5
b = [ 'Sad pose      ' sprintf('%f   ',lower) sprintf('%f
',upper)];
                 elseif i == 6
b = [ 'Surprise pose ' sprintf('%f  ',lower) sprintf('%f
',upper)];
            end

          CI_Bonferroni = [CI_Bonferroni; b]; end
CI_Bonferroni end
```

5.4: Box's M Test of Equality of Covariance Matrices

```
function [C
chisquare] = BoxM   boxesm = dir('boxesm\*.csv');      data =
csvread(['boxesm\' boxesm(1).name]);
     [n p] = size(data);      j =
length(boxesm);
```

```
     s = [];   n = [];
        G=zeros(p,p);

     for i=1:j          data = csvread(['boxesm\' boxesm(i).name]);
             m=mean(data);
          X=abs(bsxfun(@minus,data,m));          [r p]
= size(X);         n = [n (r-1)];          s = [s;
det(cov(X))];       g= n(i)*cov(X);
             G=G+g;

     end spooled = (1/sum(n))-G;
     H=[];
        for i=1:j
 K = n(i)*log(s(i));  H= [H  K];
        end
        M=(sum(n))*log(det(spooled))-sum(H);
   U =(sum(n.^(-1))- sum(n).^(-1))*(( 2*p^2 + 3*p - 1)/(6*(p+1)*(j-
1)))
     C = (1-U)*M    chisquare = 58.12403768;      if C >
chisquare          disp('Reject the null hypothesis Ho');
        else
             disp('Fail to reject the null hypothesis Ho');
        end
end
```

5.5: Profile Analysis function [tsquare F_table] = ProfileAnalysis boxesm = dir('boxesm*.csv'); j = length(boxesm); data = csvread(['boxesm\' boxesm(1).name]);

```
     [n p] = size(data);          ss_ad =
zeros(p,p); d = zeros(1,p);

     X_bars = [];
     Fnames = [];

     for i=1:j          data = csvread(['boxesm\' boxesm(i).name]);
             m=mean(data);
             X=abs(bsxfun(@minus,data,m));
             [n p] = size(X);
          X_bar = mean(X);          X_bars = [X_bars; X_bar];
     d = X_bar  - d          ss_ad = ss_ad +  cov(X)/n;
Fnames = [Fnames; boxesm(i).name];
        end
        C = [
             1 -1  0  0  0  0;
             0  1 -1  0  0  0;
             0  0  1 -1  0  0;
             0  0  0  1 -1  0;
             0  0  0  0  1 -1;
          ];
```

```
     csc = C'*inv(C*ss_ad*C')*C ;       tsquare =
d*csc*d';
     F_table = 2.451314875;     if tsquare > F_table
disp('Reject the null hypothesis Ho');   else
      disp('Fail to reject the null hypothesis Ho');       end
plot((1:p), X_bars(1,:),'markersize', 3,'-*',(1:p),
X_bars(2,:),'markersize', 3,'color','r','-o')     legend(Fnames)
xlabel('Constraints');   ylabel('Average Recognition Algorithms
Distances');   title('Mean Plot of Recognition Algorithms
Distances'); end
```

5.6: Levene's Test of Equality of Variance

```
function [Var_ratio
Confidence_Intervals] = VarianceTest    boxesm =
dir('boxesm\*.csv');     j = length(boxesm);      Xs = [];
        Fnames = [];

     for i=1:j           data = csvread(['boxesm\'
boxesm(i).name]);
             m=mean(data);
             X=abs(bsxfun(@minus,data,m));
             Xs=[Xs X];
             Fnames = [Fnames; boxesm(i).name];
           [n p] = size(X);

        end
```

```
     Ss = []; Vs = []; for
     i = 1:p
          C = [ Xs(:,i) Xs(:,i+p)];          v =
[var(Xs(:,i)) var(Xs(:,i+p))];          Vs = [Vs; v];
end

        Fmin = 0.537523654;
 Fmax   =   1.860383247;      Var_ratio   =
Vs(:,1)./Vs(:,2)    lower  =  Var_ratio*Fmin;
upper = Var_ratio*Fmax;  CI = [lower  upper];
        [r c] = size(CI);
        D = [];


     for i=1:r           pose
= '';          if i == 1
b = [ 'Angry pose ' sprintf('%f ',CI(i,1)) sprintf('%f
',CI(i,2))];
               pose = 'Angry pose';          elseif
i == 2
b = [ 'Disgust pose ' sprintf('%f ',CI(i,1))sprintf('%f
',CI(i,2))];
               pose = 'Disgust pose';   elseif i == 3

b = [ 'Fear pose  ' sprintf('%f  ',CI(i,1)) sprintf('%f
',CI(i,2))];
               pose = 'Fear pose';
elseif i == 4
```

```
b = [ 'Happy pose  ' sprintf('%f  ',CI(i,1)) sprintf('%f
',CI(i,2))];

               pose = 'Happy pose';

elseif i == 5

b = [ 'Sad pose '  sprintf('%f  ',CI(i,1)) sprintf('%f
',CI(i,2))];

               pose = 'Sad pose';

elseif i == 6

b = [ 'Surprise pose ' sprintf('%f ',CI(i,1)) sprintf('%f
',CI(i,2))];

                  pose = 'Surprise pose';

        end

         D = [D;  b];         if CI(i,1) < 1 &&  1 < CI(i,2)

     disp('Fail to reject the null hypothesis Ho');

                  disp(cstrcat('Variance of ', Fnames(1,:),' ',pose,
' is equal to the variance of ',Fnames(2,:),' ',pose,"\n" ) ) ;

             else

               disp('Reject the null hypothesis Ho');

     if Var_ratio(i,:) < 1

disp(cstrcat('Variance of ', Fnames(1,:),' ',pose, ' is less than the
variance of ',Fnames(2,:),' ',pose,"\n"  ) ) ;

                  else

                       disp(cstrcat('Variance of ', Fnames(1,:),'
',pose, ' is greater than the variance of ',Fnames(2,:),' ',pose,"\n"
) ) ;

                  end

             end

     end
```

```
Confidence_Intervals = D; end
```


Appendix 6.0: Publications from Thesis

Asiedu, L., Adebanji, A.O., Oduro, F., & Mettle, F.O. (2015). Statistical Evaluation of Face Recognition Techniques under variable Environmental Constraints. *International Journal of Statistics and Probability,* **4**(4), 93-111.

Asiedu, L., Adebanji, A.O., Oduro, F., & Mettle, F.O. (2015). Statistical Assessment of PCA/SVD and FFT-PCA/SVD under variable Facial Expressions. *British Journal of Mathematics and Computer Science,* **12**(6), 1-23.

Asiedu, L., Adebanji, A.O., Oduro, F., & Mettle, F.O. (2016). Statistical Assessment Whitened PCA/SVD under variable Environmental Constraints, International Journal of Ecological Economics and Statistics **37**(1), 63-79.