British Journal of Mathematics & Computer Science
12(6): 1-23, 2016, Article no.BJMCS.22141
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Statistical Assessment of PCA/SVD and FFT-PCA/SVD on Variable Facial Expressions

Louis Asiedu[1*], Atinuke O. Adebanji[2], Francis T. Oduro[2] and Felix O. Mettle[1]

[1] *Department of Statistics, University of Ghana, Legon-Accra, Ghana.*
[2] *Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.*

Article Information

DOI: 10.9734/BJMCS/2016/22141
Editor(s):
(1) Victor Carvalho, Polytechnic Institute of Cvado and Ave, Portuguese Catholic University and Lusiada University, Portugal.
(2) Paul Bracken, Department of Mathematics, The University of Texas-Pan American, Edinburg, TX 78539, USA.
Reviewers:
(1) Anonymous, Universiti Sains Malaysia, Malaysia.
(2) Hammad Khalil, University of Malakand, Pakistan.
Complete Peer review History: http://sciencedomain.org/review-history/12139

Original Research Article

Received: 19 September 2015
Accepted: 16 October 2015
Published: 07 November 2015

Abstract

Face recognition is a dedicated process in the human brain. Automatic face recognition is rewarding since an efficient and resilient recognition system is useful in many application areas. Recent face recognition algorithms are still faced with the challenge of recognizing face image under variable environmental constraints. This paper presents a statistical evaluation of the performance of two face recognition algorithms namely, Principal Component Analysis with Singular Value Decomposition (PCA/SVD) and Principal Component Analysis with Singular Value Decomposition using Fast Fourier Transform for preprocessing (FFT-PCA/SVD) on variable facial expressions (Angry, Disgust, Fear, Happy, Sad and Surprise) along with their neutral expressions. We considered 42 individuals from Cohn Kanade Facial Expressions database, Japanese Female Facial Expressions (JAFFE) and a created Ghanaian Face database for recognition runs. Multivariate statistical methods were used in the assessment of the face recognition algorithms. GNU Octave was used to perform all numerical runs and statistical evaluation of the recognition algorithms. The results of the statistical evaluation show that, FFT-PCA/SVD is comparatively consistent (Low variation) and efficient (Higher recognition rate) than PCA/SVD algorithm in the recognition of variable facial expressions. The paper also proposes Fast Fourier Transform as a viable noise removal mechanism that should be adopted during image preprocessing.

**Corresponding author: E-mail: lasiedu@ug.edu.gh*

Keywords: Fast fourier transform; multivariate; principal component analysis; singular value decomposition.

2010 Mathematics Subject Classification: 62H25; 62H35h.

1 Introduction

The intricacy of a face features originate from continuous changes in the facial features that take place over time. Regardless of these changes, we are able to recognize a person very easily [1]. 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 [2].

It was shown by David Hubel and Torsten Wiesel as cited by [3] that, our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Automatic face recognition is all about extracting these meaningful features from an image, putting them into a useful representation and performing some kind of classification on them [3]. The idea of imitating this skill inherent in human beings by machines can be very rewarding though the idea of developing an intelligent and self-learning system may require supply of sufficient information to the machine. An efficient and resilient face recognition technique is worthwhile in a number of application areas. These include, criminal identification, access management, law enforcement, information security and entertainment or leisure.

Principal Component Analysis was introduced as an algebraic manipulation to face recognition problem by [4]. This made it easy to calculate directly the eigenfaces. The limitations of this techniques was that, fewer than 100 values were required to accurately code suitably aligned and normalised face images. [5], 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.

[5], 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 [6]. Their approach created significant interest in advancing the developments of automated face recognition technologies, although it was somehow restricted by environmental factors.

Currently, all face recognition techniques adopt two approaches. One is local face recognition system which uses facial features (nose, mouth, eyes) of a face. That is to consider the fiducial points in the face to associate the face with a person. The local-feature method computes the descriptor from parts of the face and gathers information into one descriptor such as in Local Feature Analysis (LFA), Garbor Features, Elastic Bunch Graph Matching (EBGM) and Local Binary Pattern Feature [7]. The second approach is global face recognition system which uses the whole face to identify a person by constructing a subspace using dimensionality reduction methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Random Projection (RP), or Non-negative Matrix Factorization (NMF).

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 environmental constraints. Previous methods of evaluation focussed only on a face recognition algorithm's rate, runtime, sensitivity and descriptive statistics as the basic means of rating face recognition algorithms' performance [8].

Current attention of researchers is drawn to finding a comparatively efficient and consistent algorithm in recognizing face images under variable environmental constraints. [9], proposed an improved approach of PCA 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.

[8], proposed multivariate statistical methods to evaluate the performance of face recognition techniques under variable environmental constraints. The performance of Principal Component Analysis and Singular Value Decomposition (PCA/ SVD) and Principal Component Analysis and Singular Value Decomposition using whitening for image preprocessing (Whitened PCA/SVD) were assessed. Their evaluation criteria adjudged PCA/SVD as comparatively consistent and efficient than the Whitened PCA/SVD.

This paper seeks to statistically assess, Principal Component Analysis and Singular Value Decomposition (PCA/SVD) and Principal Component Analysis and Singular Value Decomposition with Fast Fourier transform for preprocessing (FFT-PCA/SVD) face recognition algorithms under variable facial expression.

2 Materials and Methods

The paper focuses on running PCA/SVD (Algorithm 1) and FFT-PCA/SVD (Algorithm 2) recognition algorithms on a created face database and evaluates the recognition performance of the algorithms as well as comparing their results on the face database.

Face image samples from local Ghanaian Facial Expressions (GFE), Cohn Kanade database (CK) and Japanese Female Facial Expression database (JAFFE) are used as inputs.

The input face images, are resized into uniform dimension and the data types changed into double precision for preprocessing. The whole recognition process comprises a preprocessing stage, feature extraction stage and classification stage [6]. The adopted preprocessing procedures are mainly, mean centering and Fast Fourier Transform. This is to help reduce the noise level and make the estimation process simpler and better conditioned [8].

The PCA/SVD (Algorithm 1) and FFT-PCA/SVD (Algorithm 2) algorithms are used to train the image database. In the extraction unit, unique face image features are extracted and stored for recognition. The obtained facial features are passed to the classifier unit for classification of a given face query with the knowledge created for the available database.

For the implementation of the facial recognition, a real time GFE, CK and JAFFE database is used. All three databases were combined in the study. This helped to evaluate the face recognition algorithms on large and different databases. The new created GFE accounted for the originality of the study database.

For the implementation of the proposed recognition design, the database samples are trained for the knowledge creation and classification. In the course of the training phase, when a new facial image is added to the system, the features are 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 are compared by finding the norm of the difference between the test and known weights. A maximum and minimum difference signifies poor and close match respectively. Fig. 1 is a flow diagram of the study algorithms.

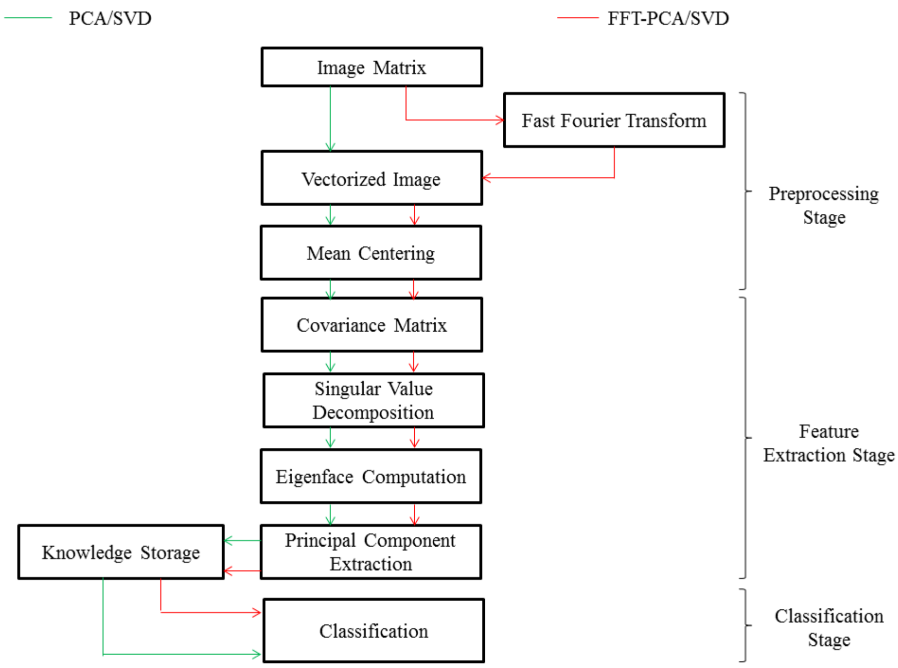

Fig. 1. Flow diagram of PCA/SVD & FFT-PCA/SVD

2.1 Preprocessing of Face Images.

Before extracting features from face images using PCA/SVD algorithm, it is useful to do some preprocessing. As mentioned earlier, this is to help reduce the noise level and make the estimation process simpler and better conditioned. In this paper, preprocessing was basically, Mean Centering and Fast Fourier Transform. Fig. 2. shows six images selected from Japanese Female Face Expression database (JAFFE).

Fig. 2. Six images from JAFFE

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

$$\begin{aligned}\mathbf{M}_j &= (m_{jik}) \ ; i, k = 1, 2, \ldots, p; \ j = 1, 2, \ldots, n, \\ &= (\mathbf{m}_{j1}, \mathbf{m}_{j2}, \ldots, \mathbf{m}_{jp}), \text{ where } \mathbf{m}_{jk} = (m_{j1k}, m_{j2k}, \ldots, m_{jpk})^T.\end{aligned}$$

$$\mathbf{X}_j = (\mathbf{m}_{j1}^T, \mathbf{m}_{j2}^T, \ldots, \mathbf{m}_{jp}^T)^T, \tag{2.1}$$

where,
p = the order of the image matrix.
n = the number of images to be trained.
Now from equation (2.1), suppose $\mathbf{X}_j$ is a column vector of dimension N given by

$$\mathbf{X}_j = (X_{ji})_{N\times 1}, \tag{2.2}$$

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

Mean 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.

$$\begin{aligned}\bar{\mathbf{m}}_j &= \frac{1}{N}\sum_{i=1}^{N} X_{ji}, \\ &= \frac{1}{N}\sum_{i=1}^{p}\sum_{k=1}^{p} \mathbf{m}_{jik},\ (j = 1, 2, 3\ldots, n)\end{aligned} \tag{2.3}$$

where $N = (p \times p)$, length (= rows of image × columns of image) of the image data, $\mathbf{X}_j$.
Define $\bar{\mathbf{X}}_j$ as a constant vector of order $(p \times p)$ with all elements same as $\bar{\mathbf{m}}_j$, $(j = 1, 2, \ldots, n)$ [[8]].

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

$$\mathbf{w}_j = \mathbf{X}_j - \bar{\mathbf{X}}_j. \tag{2.4}$$

Applying (2.4) to the images in Fig. 1, generate the mean centered images as shown in Fig. 3.

Fig. 3. Six mean centered images from JAFFE

Fast Fourier Transform: The Fast Fourier Transform (FFT) is used as a noise reduction mechanism during image preprocessing. 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 [10]. The FFT is a computationally efficient method of generating a Fourier transform. It is computationally faster than DFT and gets its speed by decreasing the number of calculations needed to analyze an image data. FFT is therefore preferred to DFT when working with large data.

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

$$m_{jsk}^{*} = DFT\{\mathbf{m}_{jk}\} = \sum_{r=0}^{p-1} \mathbf{m}_{jk} e^{-i(2\pi sr/p)}, \tag{2.5}$$

where, $s = 0, 1, \ldots, p-1$, $j = 1, 2, \ldots, n$ 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;

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

Therefore,

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

and

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

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

$$m_{jk} = IDFT\{\mathbf{m}^*_{jk}\} = \frac{1}{p}\sum_{r=0}^{p-1} \mathbf{m}^*_{jk} e^{i(2\pi sr/p)}, \tag{2.6}$$

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

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

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

and

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

The real components of the transformed images are extracted for the feature extraction stage whereas the imaginary components are discarded as noise. Fig. 4 shows the FFT preprocessed images of the six images shown in Fig. 2.

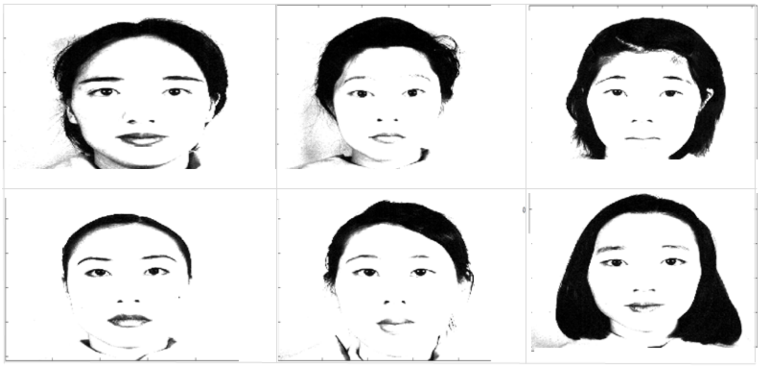

Fig. 4. Six FFT preprocessed images from JAFFE

2.2 Singular Value Decomposition (SVD)

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 $\mathbb{R}^n$. The diagonal entries of **D** are the corresponding eigenvalues of **A**.

In the case of SVD for $m \times n$ matrix **A**, the transformation takes $\mathbb{R}^n$ to a different space $\mathbb{R}^m$. 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.

2.3 Principal Component Analysis (PCA)

PCA elucidates the covariance structure of a set of variables. In particular it allows us to identify the principal directions in which the data varies. 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 [11], PCA can 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. In particular it allows us to identify the principal directions in which the data varies. An effective way to suppress redundant information and provide only one or two composite data from most of the information from the initial data is called Principal Component Analysis.

2.4 Feature Extraction

According [6], we 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;

$$\lambda_t = \frac{1}{n} \sum_{j=1}^{n} (\mathbf{e}_j^T \mathbf{w}_j)^2, \tag{2.7}$$

is a maximum subject to the orthonormality constraints,

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

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

$$\mathbf{C} = \frac{1}{n}\mathbf{W}\mathbf{W}^T, \tag{2.8}$$

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

Next is to run a singular value decomposition (SVD) of the matrix, $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ to ascertain its eigenvalues and their corresponding eigenvectors.

This splits the covariance matrix $\mathbf{C}$ into two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ and a diagonal matrix $\mathbf{\Sigma}$.

$$\mathbf{e}_j = \textstyle\sum_{j=1}^{n} \mathbf{w}_j \mathbf{d}_j, \tag{2.9}$$

where $\mathbf{d}_j$ is the jth column vector of $\mathbf{U}$.

The Principal components of the trained image set are determined by computing;

$$\nu_j = \mathbf{e}_j^T(\mathbf{x}_j - \bar{\mathbf{m}}), \tag{2.10}$$

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

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 Ω.

$$\nu_j = \mathbf{e}_j^T(\mathbf{x}_r - \bar{\mathbf{m}}), \text{ and } \mathbf{\Omega}_r^T = [\nu_1, \nu_2, \ldots, \nu_n].$$

The recognition distances (euclidean distance) are computed as;

$$\xi = \|\mathbf{\Omega} - \mathbf{\Omega}_r\|. \tag{2.11}$$

$\epsilon_m = min[\xi]$ is chosen as the distance at which a test image is recognized in the trained image database.

2.5 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 [12].

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) [8].

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 (10 individuals) were also collected from Japanese Female Facial Expressions database (JAFFE) along the principal emotional constraints. Fig. 5 shows a sample of the study database.

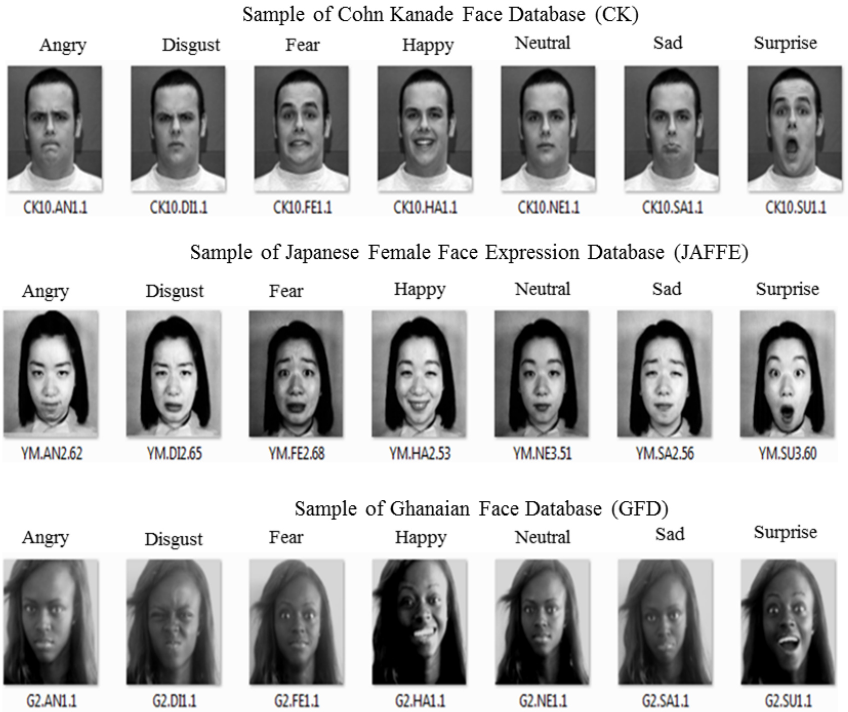

Fig. 5. Sample of Study Database
Source: [8]

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 two 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 are captured into the test image database. These images are called the test images and used for testing the recognition algorithms.

3 Results and Discussion

This section contains detailed results of the multivariate statistical evaluation methods proposed by [8], on dataset churned from running the study algorithms on the available database. The section also interprets the outcome of the statistical analysis.

3.1 Assessing Multivariate Normality

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, $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, n$, where k is the number of constraints and j is the number of individuals in the research database.

Define the squared distance as;

$$\mathbf{d}_{ij}^2 = \left(\mathbf{X}_{jk} - \bar{\mathbf{X}}_k\right) \mathbf{\Sigma}^{-1} \left(\mathbf{X}_{jk} - \bar{\mathbf{X}}_k\right)^T, \qquad (3.1)$$

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

Define $d_j^2 = \mathbf{d}_{ij}^2$, for $i = j$. That is d_j^2 is the diagonal entry of the jth individual from matrix $\mathbf{d}_{ij}^2$.

Fig. 6 and Fig. 7 show plots of $\left\{ q_{c,k} \frac{(j - \frac{1}{2})}{n}, d_j^2 \right\}$ for Algorithm 1 and Algorithm 2 respectively. $q_{c,k} \left(\frac{(j - \frac{1}{2})}{n} \right)$ is the $100 \frac{(j - \frac{1}{n})}{n}$ quantile of chi-square distribution with k degrees of freedom.

The correlation, r, values 0.91359 and 0.91631 for Algorithm 1 and Algorithm 2 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.

Multivariate normality exist and hence can be assumed in subsequent statistical tests that will be performed on the datasets.

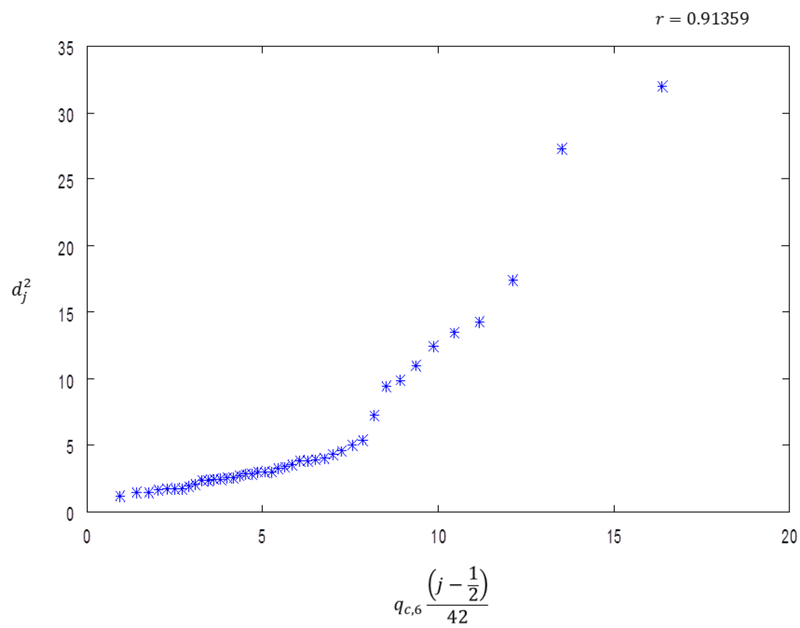

Fig. 6. Chi-square plot for Algorithm1

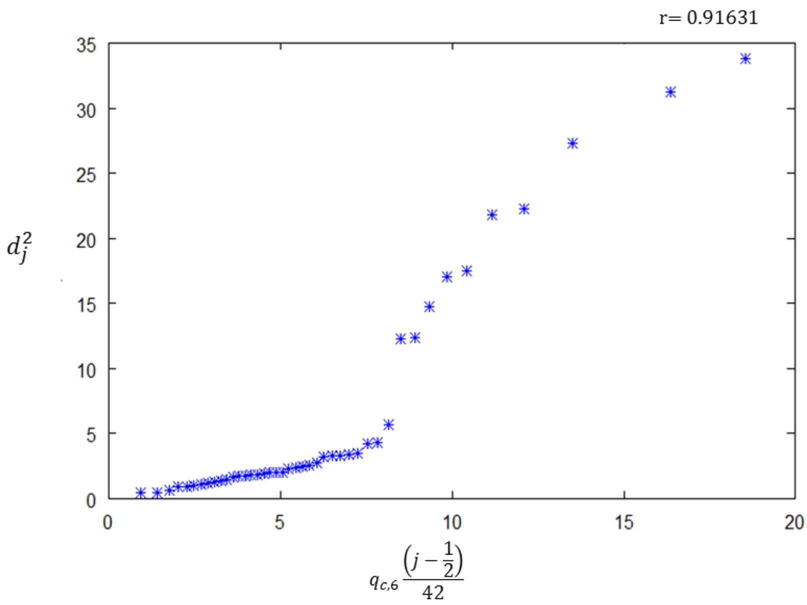

Fig. 7. Chi-square plot for Algorithm 2

3.2 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 1.0*. 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.

Using the 6-variate dataset from PCA/SVD (Algorithm 1), we have;

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

and

$$\left(C\Sigma C^T\right)^{-1} = \begin{bmatrix} 1.28e^{-007} & 1.12e^{-007} & 6.21e^{-008} & -3.69e^{-009} & 6.48e^{-009} \\ 1.12e^{-007} & 4.99e^{-007} & 4.15e^{-007} & 1.33e^{-007} & 1.40e^{-008} \\ 6.21e^{-008} & 4.15e^{-007} & 7.85e^{-007} & 5.68e^{-007} & 2.40e^{-007} \\ -3.69e^{-009} & 1.33e^{-007} & 5.68e^{-007} & 7.40e^{-007} & 4.36e^{-007} \\ 6.48e^{-009} & 1.40e^{-008} & 2.40e^{-007} & 4.36e^{-007} & 5.44e^{-007} \end{bmatrix}.$$

Hence, the T^2- Statistic is;

$$T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C}\mathbf{\Sigma}\mathbf{C}^T\right)^{-1}\mathbf{C}\bar{\mathbf{X}}, \tag{3.2}$$
$$= 35.095.$$

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

and $F_{p-1,n-p+1}(0.05) = 2.4697$,

where $F_{p-1,n-p+1}$ is the F-distribution with $(p-1, n-p+1)$ degree of freedom with p and n being the total number of constraints and individuals respectively.

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

Reject H_0 if ; $\qquad T^2 = n\left(\mathbf{C}\bar{\mathbf{X}}\right)^T \left(\mathbf{C\Sigma C}^T\right)^{-1} \mathbf{C}\bar{\mathbf{X}} > \dfrac{(n-1)(p-1)}{(n-p+1)} F_{p-1,n-p+1}(\alpha)$.

$35.095 > 13.683194$. 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 distances of the various constraints from their neutral pose when Algorithm 1 is used for recognition.

Similarly, using the 6-variate dataset from Algorithm 2, we obtain

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

and

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

Hence, the T^2- Statistic from (3.2) is;

$$T^2 = 36.695.$$

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

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

$36.695 > 13.6832$. 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 2 is used for recognition.

The 95% simultaneous confidence intervals for the estimates of the mean differences are shown in Table 1.

Table 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	$\bar{X}_2 - \bar{X}_6 = -1084.2000$	-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	$\bar{X}_4 - \bar{X}_6 = -1263.6000$	-2469.2859	-57.8386
*Sad vs Surprise	$\bar{X}_5 - \bar{X}_6 = -1167.2000$	-2298.1676	-36.1638

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

These simultaneous confidence intervals show 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 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.

The 95% simultaneous confidence intervals for the estimates of the mean difference under Algorithm 2 is shown in Table 2 below.

Table 2. Simultaneous Confidence Intervals - Algorithm 2.

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	$\bar{X}_2 - \bar{X}_4 = -85.4000$	-506.0024	535.5248
Disgust vs Sad	$\bar{X}_2 - \bar{X}_5 = 111.9300$	-387.6305	570.6553
Disgust vs Surprise	$\bar{X}_2 - \bar{X}_6 = -107.5300$	-907.4330	48.8198
Fear vs Happy	$\bar{X}_3 - \bar{X}_4 = -282.1600$	-617.7883	586.1401
Fear vs Sad	$\bar{X}_3 - \bar{X}_5 = -84.8300$	-329.8673	451.7214
*Fear vs Surprise	$\bar{X}_3 - \bar{X}_6 = -304.2900$	-875.7119	-44.0719
Happy vs Sad	$\bar{X}_4 - \bar{X}_5 = 197.3300$	-480.1164	633.6187
Happy vs Surprise	$\bar{X}_4 - \bar{X}_6 = -22.1300$	-1005.2381	117.1024
*Sad vs Surprise	$\bar{X}_5 - \bar{X}_6 = -219.4600$	-957.8452	-83.7927

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

It can be seen from Table 2 that, for Algorithm 2, the Fear vs Surprise ($\mu_3 - \mu_6$) and Sad vs Surprise ($\mu_5 - \mu_6$) have confidence intervals, $[-957.8452, -83.7927]$ and $[-875.7119, -44.0719]$ 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 2 is used as the recognition algorithm.

3.3 Paired Comparison

The multivariate case is motivated for PCA/SVD (Algorithm 1) and FFT-PCA/SVD (Algorithm 2), 6 constraints and 42 experimental units. The paired differences random variables become;

$$\begin{aligned} D_{j1} &= X_{j11} - X_{j12}, \\ D_{j1} &= X_{j21} - X_{j22}, \\ \vdots & \qquad \vdots \\ D_{j1} &= X_{j61} - X_{j62}. \end{aligned} \tag{3.3}$$

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

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

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

and

$$\mathbf{\Sigma}_d^{-1} = \begin{bmatrix} 2.14e^{-007} & -8.85e^{-008} & -1.18e^{-007} & -7.21e^{-008} & 6.10e^{-008} & -4.33e^{-008} \\ -8.85e^{-008} & 7.83e^{-007} & 8.78e^{-008} & -3.35e^{-007} & -2.24e^{-007} & -2.35e^{-008} \\ -1.18e^{-007} & 8.78e^{-008} & 9.45e^{-007} & 4.10e^{-008} & -2.39e^{-007} & 1.33e^{-008} \\ -7.21e^{-008} & -3.35e^{-007} & 4.10e^{-008} & 6.28e^{-007} & 5.16e^{-008} & -9.16e^{-008} \\ 6.10e^{-008} & -2.24e^{-007} & -2.39e^{-007} & 5.16e^{-008} & 7.52e^{-007} & -1.13e^{-007} \\ -4.33e^{-008} & -2.35e^{-008} & 1.33e^{-008} & -9.16e^{-008} & -1.13e^{-007} & 5.89e^{-007} \end{bmatrix}.$$

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;

$$T^2 = n\bar{\mathbf{d}}^T\mathbf{\Sigma}_d^{-1}\bar{\mathbf{d}}, \tag{3.4}$$

$$= 138.20,$$

and

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

where $F_{p,n-p}$ is the $F-$ distribution with $(p, n-p)$ degree of freedom.

$138.20 > 16.152$. This means H_0 is not tenable, and we therefore have enough evidences at 5% significances level to reject H_0. It can concluded that, there exist significant difference in the average recognition distances of algorithm 1 and algorithm 2 with respect to the study constraints (pose-wise).

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

Table 3. Bonferroni Simultaneous Confidence Intervals (Algorithm 1 & 2)

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	$\bar{d}_5 = 1495.5$	948.0762	2042.8308
*Surprise poses	$\bar{d}_6 = 1141.0$	546.5988	1735.4019

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

From Table 3, it can seen that all the confidence intervals do not contain 0. This means for Algorithm 1 and Algorithm 2, 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 2 have significantly different average recognition distances for all poses.

3.4 Box's M Test of Equality of Covariance

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

$H_0 : \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$
$H_1 : \boldsymbol{\Sigma}_i \neq \boldsymbol{\Sigma}_j$ for $i \neq j$ (at least two unequal covariance matrices).

where $\boldsymbol{\Sigma}_l$ is the covariance matrix for the l^{th} population, $l = 1, 2$. Here our populations are the measures from the recognition algorithms and $\boldsymbol{\Sigma}$ 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, $\mathbf{S}_l$, $l = 1, 2$ are given as;

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

and

$$\mathbf{S}_2 = \begin{bmatrix} 9.36e^{+005} & 3.23e^{+005} & 6.90e^{+004} & 6.07e^{+005} & 2.68e^{+005} & 2.28e^{+005} \\ 3.23e^{+005} & 5.93e^{+005} & 6.13e^{+003} & 4.34e^{+005} & 9.29e^{+004} & 1.90e^{+005} \\ 6.90e^{+004} & 6.13e^{+003} & 1.53e^{+005} & 7.43e^{+004} & -9.00e^{+003} & 5.46e^{+004} \\ 6.07e^{+005} & 4.34e^{+005} & 7.43e^{+004} & 1.11e^{+006} & 2.27e^{+005} & 3.14e^{+005} \\ 2.68e^{+005} & 9.29e^{+004} & -9.00e^{+003} & 2.27e^{+005} & 2.98e^{+005} & 9.96e^{+004} \\ 2.28e^{+005} & 1.89e^{+005} & 5.46e^{+004} & 3.14e^{+005} & 9.96e^{+004} & 4.87e^{+005} \end{bmatrix}.$$

The pooled sample covariance ; $\mathbf{S}_{pooled} = \dfrac{1}{\sum_{l=1}^{g}(n_l-1)} - \sum_{l=1}^{g}(n_l-1)\,\mathbf{S}_l$.

Now,

$$\mathbf{S}_{pooled} = \begin{bmatrix} -4.88e^{+008} & -1.23e^{+008} & -3.86e^{+007} & -1.84e^{+008} & -4.24e^{+007} & -7.93e^{+007} \\ -1.23e^{+008} & -2.08e^{+008} & 2.50e^{+006} & -1.53e^{+008} & -4.17e^{+007} & -5.14e^{+007} \\ -3.86e^{+007} & 2.50e^{+006} & -6.08e^{+007} & 4.40e^{+006} & -9.23e^{+006} & -9.21e^{+006} \\ -1.84e^{+008} & -1.53e^{+008} & 4.40e^{+006} & -2.86e^{+008} & -3.07e^{+007} & -8.93e^{+007} \\ -4.24e^{+007} & -4.17e^{+007} & -9.22e^{+006} & -3.07e^{+007} & -1.12e^{+008} & -2.13e^{+007} \\ -7.93e^{+007} & -5.14e^{+007} & -9.21e^{+006} & -8.93e^{+007} & -2.13e^{+007} & -1.15e^{+008} \end{bmatrix}.$$

From the above, $|\mathbf{S}_1| = 5.6378e^{+038}$, $|\mathbf{S}_2| = 3.0936e^{+033}$, $|\mathbf{S}_{pooled}| = 6.3196e^{+048}$ and

$$M = \sum_{l=1}^{g}(n_l-1)\,ln|\mathbf{S}_{pooled}| - \sum_{l=1}^{g}(n_l-1)\,ln|\mathbf{S}_l|. \quad (3.5)$$
$$= 2394.1$$

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

$$U = \left[\sum_{l=1}^{g}\frac{1}{(n_l-1)} - \frac{1}{\sum_{l=1}^{g}(n_l-1)}\right]\left[\frac{2p^2+3p-1}{6(p+1)(g-1)}\right],$$
$$= 0.077626,$$

where $p = 6$ is the number of constraints and $g = 2$ is the number of groups (Algorithms).

Now,

$$K = (1-U)M = (1-0.077626)2394.1,$$

$$= 2208.5,$$

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

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

$$\chi^2_{\frac{1}{2}p(p+1)(g-1)}(0.05) = \chi^2_{21}(0.05),$$

$$= 32.6706.$$

Clearly, $2208.5 > 32.6706$, 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.

3.5 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 [13]. Here, the pooled covariance is then used as the common covariance for the populations under study. The Box's M test revealed that, the covariance matrices of the algorithms under study are unequal.

According [14], the 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 $\boldsymbol{\mu_1}^T = [\mu_{11}, \mu_{12}, \ldots, \mu_{1p}]$ and $\boldsymbol{\mu_2}^T = [\mu_{21}, \mu_{22}, \ldots, \mu_{2p}]$ be the mean responses to p treatments for Algorithm 1 and 2 respectively.

The contrast matrix given as

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

Now from the study datasets, we have;

$$\left(\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1\right)^T = \begin{bmatrix} -2209.4 & -1801.4 & -1201.4 & -1426.8 & -1495.5 & -1141.0 \end{bmatrix},$$

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

and

$$\mathbf{S}_2 = \begin{bmatrix} 9.36e^{+005} & 3.23e^{+005} & 6.90e^{+004} & 6.07e^{+005} & 2.68e^{+005} & 2.28e^{+005} \\ 3.23e^{+005} & 5.93e^{+005} & 6.13e^{+003} & 4.34e^{+005} & 9.29e^{+004} & 1.90e^{+005} \\ 6.90e^{+004} & 6.13e^{+003} & 1.53e^{+005} & 7.43e^{+004} & -9.00e^{+003} & 5.46e^{+004} \\ 6.07e^{+005} & 4.34e^{+005} & 7.43e^{+004} & 1.11e^{+006} & 2.27e^{+005} & 3.14e^{+005} \\ 2.68e^{+005} & 9.29e^{+004} & -9.00e^{+003} & 2.27e^{+005} & 2.98e^{+005} & 9.96e^{+004} \\ 2.28e^{+005} & 1.89e^{+005} & 5.46e^{+004} & 3.14e^{+005} & 9.96e^{+004} & 4.87e^{+005} \end{bmatrix}.$$

The sample sizes are, $n_1 = n_2 = 42$. In testing for parallel profiles for two populations,

$$T^2 = (\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1)^T \mathbf{C}^T \left[\mathbf{C} \left(\frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2} \right) \mathbf{C}^T \right]^{-1} C(\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1), \qquad (3.6)$$
$$= 7.4215$$

and

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

Clearly, $7.4215 > 2.451$ and hence $H_0 : \mu_{1i} - \mu_{1i-1} = \mu_{2i} - \mu_{2i-1}$, $i = 1, 2, \ldots, 6$ (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. Fig. 8 shows a mean plot of the recognition algorithms.

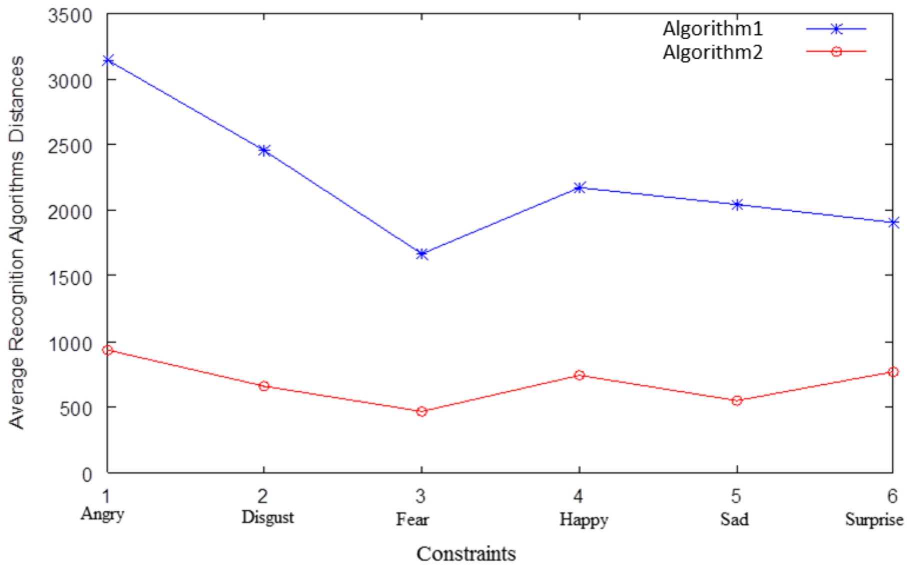

Fig. 8. Mean plot of recognition distances. (Algorithm 1 & Algorithm 2)

3.6 Levene's Test of Equality of Variances

The goal of this test is to determine if the algorithms under study have equal variances in their pose-wise 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 σ^2_{k1} and σ^2_{k2}. Also, consider samples of sizes $n_{k1} = 42$, from Algorithm 1, $n_{k2} = 42$, from Algorithm 2 and their respective sample variances S^2_{k1} and S^2_{k2}.

To test $H_0 : \sigma^2_{k1} = \sigma^2_{k2}$, a 95% confidence interval is constructed from,

$$\frac{S^2_{k1}}{S^2_{k2}} \, F_{1-\frac{\alpha}{2}, n_{2k}-1, n_{1k}-1} < \frac{\sigma^2_{k1}}{\sigma^2_{k2}} < \frac{S^2_{k1}}{S^2_{k2}} \, F_{\frac{\alpha}{2}, n_{2k}-1, n_{1k}-1},$$

and shown in Table 4.

Table 4. Confidence Intervals for the ratio of variances (Algorithm 1 & Algorithm 2)

Constraints	*Ratio of Variances*	*Lower*	*Upper*
*Angry poses	$\frac{S_{11}}{S_{12}} = 11.7120$	6.2955	21.7888
*Disgust poses	$\frac{S_{21}}{S_{22}} = 7.5465$	4.0564	14.0394
*Fear poses	$\frac{S_{31}}{S_{32}} = 8.7200$	4.6872	16.2225
*Happy poses	$\frac{S_{41}}{S_{42}} = 5.2894$	2.8432	9.8403
*Sad poses	$\frac{S_{51}}{S_{52}} = 8.1794$	4.3966	15.2167
*Surprise poses	$\frac{S_{61}}{S_{62}} = 4.7753$	2.5668	8.8838

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

Reject $H_0 : \sigma^2_{k1} = \sigma^2_{k2}$, if 1 does not belong to the confidence interval for $\frac{S^2_{k1}}{S^2_{k2}}$, $k = 1, 2, \ldots, 6$. Clearly from Table 4, 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, $\frac{S^2_{k1}}{S^2_{k2}}, k = 1, 2, \ldots, 6$ 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 2 for all constraints recognition distance. Subsequently, Algorithm 2 is considered as comparatively consistent in the recognition of Angry, Disgust, Fear, Happy, Sad and Surprise poses.

3.7 Numerical Evaluations

The recognition rate 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 average recognition rate, R_{avg}, of our method is defined as

$$R_{avg} = \frac{\sum_{i=1}^{q} n^i_{cls}}{q \times n_{tot}}, \tag{3.7}$$

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 $100 - R_{avg}$.

The total number of correct recognition $\sum_{i=1}^{q} n^i_{cls}$, for Algorithm 1 and Algorithm 2 is 223 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,

$$R_{avg2} = \frac{225}{(42)(6)} \times 100,$$
$$= 89.29\%.$$

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

4 Summary

From the above numerical evaluations, the recognition rates of Principal Component Analysis and Singular Value Decomposition (PCA/SVD) and Principal Component Analysis and Singular Value Decomposition with Fast Fourier Transform as preprocessing step (FFT-PCA/SVD) are 88.49% and 89.29% respectively.

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

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

From the results of the paired comparison, PCA/SVD (Algorithm 1) has significantly higher average recognition distances in the recognition of all the expressions when compared to FFT-PCA/SVD (Algorithm 2).

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

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

The Levene's test also revealed that, the variations in PCA/SVD (Algorithm 1) are higher than the variations in FFT-PCA/SVD (Algorithm 2) in the recognition of all the principal emotions (expressions).

5 Conclusions

From the above statistical evaluation, FFT-PCA/SVD (Algorithm 2) is adjudged comparatively the most consistent algorithm in the recognition of face images under variable environmental constraints.

This confirms the computed recognition rates of the algorithms.

FFT-PCA/SVD (Algorithm 2) is therefore the most efficient (highest recognition rate) and consistent (lowest variation) in the recognition of face images under variable facial expressions. It can also be inferred from the statistical evaluation results that, Fast Fourier Transform improved PCA/SVD algorithm when used as a noise removal mechanism during image preprocessing.

Competing Interests

The authors declare that no competing interests exist.

References

[1] Rahman MU. A comparative study on face recognition techniques and neural network; 2013 vol abs/ 1210-1916.

[2] Bartlett MS, Donato G, Ekman P, Hager JC, Sejnowski TJ. Classifying facial actions. IEEE Trans.Pattern Analysis and Machine Intelligence. 1999;21(10):974-989.

[3] Wagner P. Face recognition with python; 2012.
Available: http:// www.bytefish.de

[4] Kirby M, Sirovich L. Application of the Karhunen-Loéve procedure for the characterization of human faces. IEEE Trans. Pattern Analysis and Machine Intelligence.1990;12(1):103-108.

[5] Turk M, Pentland A. Eigenface for recognition. Eigenface for Recognition. 1991;3(1):71-86.

[6] Asiedu L, Mettle FO, Nortey ENN. Recognition of facial expressions using principal component analysis and singular value decomposition. International Journal of Statistics and Systems. 2014;(9)2:157-172.

[7] Agrawal S, Khatri P, Gupta S. Facial expression recognition techniques: A survey. ITM University. Gwalior, India; 2014.

[8] Asiedu L, Adebanji AO, Oduro F, Mettle FO. Statistical evaluation of face recognition techniques under variable environmental constraints. International Journal of Statistics and Probability. 2015;4(4):93-111.

[9] Zhang D, Ding D, Li J, Liu Q. PCA based extracting feature using fast fourier transform for facial expression. Transaction on Engineering Technologies. Springer Netherlands. 2015;413 - 424.

[10] Glynn EF. Fourier analysis and image processing. Stowers Institute for Medical Research; 2007.

[11] Barlett MS, Movellan J, Sejnowski TJ. Face recognition by independent component analysis. IEEE Trans. on Neural Networks. 2002;13(6):1450-1464.

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

[13] Johnson AR, Wichern WD. Applied multivariate statistical analysis. 6th Ed. New Jersey Pearson Prentice Hall; 2007.

[14] Mettle FO, Yeboah E, Asiedu L. Profile analysis of spatial differential of inflation in ghana. International Journal of Statistics and Analysis. 2014;4(2):245-259.

Appendix

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 1.0a: Multivariate data from PCA/SVD (Algorithm 1)

Indiv.	*Angry*	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*
1	3517.1	3312.4	3138.9	382.56	3200.9	3416.7
2	3648.5	3188.9	1369.8	1270.0	977.53	1200.8
3	1066.7	2292.3	944.91	1754.1	153.65	503.84
4	1887.2	1565.2	1968.2	650.66	1592.4	3163.7
5	1310.0	1786.4	714.62	1555.7	1041.4	19.272
6	3344.0	2233.9	396.92	3125.5	1639.3	457.94
7	2026.0	3846.0	2079.9	2066.4	3774.7	1257.6
8	3938.2	1352.1	287.93	603.52	2255.8	2297.1
9	4511.8	2741.5	2001.4	612.89	343.57	555.98
10	3911.8	2740.3	431.02	1892.2	1288.2	3438.5
11	4061.4	3992.9	2417.5	3841.7	3093.6	3190.7
12	1007.3	311.05	1354.2	2343.1	60.553	4302.2
13	641.44	1039.4	1502.3	2711.9	20.156	439.59
14	4498.9	213.6	1725.3	381.99	3122.7	389.05
15	1291.7	649.23	2096.0	2441.9	2137.4	1073.8
16	604.4	662.43	2188.9	450.77	38.288	150.02
17	1592.4	726.54	972.45	3186.6	2050.1	1685.6
18	2443.2	2156.4	968.9	2.4806	1099.1	1322.6
19	2367.3	2632.2	1758.5	3362.8	602.98	2763.2
20	4202.9	1733.3	2035.5	802.91	3368.7	2945.4
21	2649.6	2807.1	2181.8	388.48	2203.0	1093.8
22	435.71	1929.4	115.94	1088.9	1845.6	5610.8
23	974.95	1838.1	21.981	2655.7	3448.7	713.35
24	2468.7	2045.5	3403.0	961.96	1571.8	2048.3
25	3914.6	2149.0	2617.9	2952.0	1991.8	2543.7
26	1960.6	832.35	986.1	1915.1	716.68	1099.8
27	8971.9	2449.4	249.59	878.3	3907.0	517.22
28	13519	10854	421.22	14982	1501.9	6786.5
29	4172.6	2069.4	2502.5	1358.1	900.82	1927.7
30	16692	703.38	4804.2	2626.1	751.38	2277
31	497.09	3554.5	537.24	1834.1	3030.6	978.56
32	577.03	1869.1	6.0795	1428.3	342.6	487.2
33	1877.0	2814.1	3018.8	2190.9	430.11	2250.2
34	1876.8	3680.2	3528.1	2197.5	1940.2	2460.5
35	540.55	977.6	3713.8	1454.1	4722.7	3596.4
36	598.35	1004.1	247.22	1411.1	662.0	1216.5
37	2560.7	1592.4	365.71	657.07	2981.1	1623.5
38	4335.3	6486.2	2552.9	1564.1	4242.9	680.9
39	3086.1	9049.1	1717.6	3712.9	3998.4	1340.5
40	490.52	239.92	1581.3	296.72	2493.8	1591.7
41	312.47	1362.2	2670.1	4459.2	2594.1	180.96
42	7593.3	3786.5	2211.3	6672.8	7580.7	4450.4

Appendix 1.0b: Multivariate data from FFT-PCA/SVD (Algorithm 2)

Indiv.	*Angry*	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*
1	914.0	1059.2	603.9	132.6	869.7	1878.1
2	1339.1	750.8	766.2	550.5	345.9	794.8
3	423.9	769.3	292.5	519.6	34.8	205.9
4	776.8	491.1	481.1	116.3	534.2	860.3
5	603.6	8.1	99.4	457.6	198.4	373.4
6	1164	778.4	90.9	1031.8	588.1	64.9
7	490.2	981.4	346.8	410.8	849.8	574.7
8	1045.0	198.0	176.3	269.9	883.4	733.1
9	1494.3	950.8	93.0	194.7	159.7	432.3
10	1081.1	548.2	618.6	364.1	24.1	1002.0
11	1022.7	1278	239.1	1104.4	731.4	1046.0
12	545.8	312.1	572.5	851.9	54.4	1056.6
13	9.0	74.1	337.5	577.7	290.8	194.1
14	1437.7	22.7	444.1	50.2	841.6	69.2
15	845.7	491.5	675.5	1034.2	821.6	738.5
16	73.7	434.2	695.2	22.1	805	580.0
17	325.9	223.6	70.9	819.3	690.4	11.7
18	957.5	787.4	781.0	184.6	121.8	1161.2
19	402.4	728.5	207.5	1015.8	124.5	907.2
20	790.4	34.9	343.5	314.9	604.1	663.7
21	20.1	666.3	24.7	611.8	3.9	95.7
22	406.1	646.6	570.7	376.7	179.6	233.3
23	300.4	419.3	535.8	760.3	963.6	221.4
24	178.8	319.8	1797.5	72.6	97.8	458.2
25	1034.1	549.6	668.5	799.6	284.2	1100.0
26	1175.8	10.4	102.9	599.4	66.9	488.9
27	3338.7	798.9	20.0	588.1	2042.4	90.7
28	4395.8	3716.9	913.3	6307.7	1258.8	2771.8
29	882.1	110.7	776.8	2643.5	147.9	84.5
30	3465.9	86.3	1294.6	314.6	138.4	1335.7
31	784.1	1110.0	1543.0	1315.4	1010.5	717.3
32	42.4	811.5	218.5	637.4	782.9	3437.0
33	144.8	164.0	244.5	338.4	466.2	204.7
34	686.1	218.1	432.1	672.8	492.0	7 29.3
35	451.3	233.7	488.7	209.7	540.6	985.1
36	76.3	38.1	324.5	398.0	184.1	866.7
37	766.0	332.3	263.5	428.2	487.9	650.4
38	420.1	856.6	368.6	52.4	445.6	214.9
39	837.4	3610.7	23.2	41.3	15.9	303.7
40	135.0	364.3	56.1	556.2	837.5	833.4
41	776.5	330.4	322.8	616.7	115.3	1195.8
42	3120.1	1294.2	421.4	2834.1	2774.4	1760.8

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/12139