# FACIAL EXPRESSION DETECTION

A Thesis
submitted to the Department of Computer Science and Engineering
of United International University
In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Science and Engineering (CSE)

Sohana Mahmud, Gazi Muhammad Samiul Hoque, Md. Feroj Bepari
March 2011

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Bachelor of Science in Computer Science and Engineering (CSE).

(Professor Dr. Chowdhury Mofizur Rahman) Principal Adviser

Signature of the Candidates

(Sohana Mahmud)

(Gazi Muhammad Samiul Hoque)

(Md. Feroj Bepari)

# Abstract

Facial expression analysis is rapidly becoming an area of intense interest in computer science and human-computer interaction design communities. The most expressive way humans display emotions is through facial expressions. This thesis presents the development of a facial expression recognition system that takes any frontal image of any human face and finds the emotion expressed by that person. We have developed a method which is implemented using eigenface-based approach for the extraction of intransient facial features and recognition of five facial expressions. The algorithm implements principal component analysis and further uses for feature extraction and creates eigenface representation of the face, followed by classification into one of the expression classes. The algorithm achieves an accuracy of 77% for facial expression recognition for normalised color image.

# Acknowledgement

First and foremost, we would like to thank God Almighty for life itself. All that we have is due to his grace and we give all glory to him. Even though we have let him down many times in our life, he has always carried us through. We would like to thank our parents, our other family members. Without our family, we could not have made it to where we are today. We would also like to thank all those who have encouraged us in our work here at the United International University: Professor Dr. Chowdhury Mofizur Rahman, Dr. Hasan Sarwar, Dr. S M Monzurur Rahman, Dr. Mohammad Nurul Huda and Mr. Feroz Ahmed Siddiky. They not only encouraged us but also helped us as much as possible. There are many friends who have helped us a great deal in our work and life whom we have not mentioned. However, we want to thank them all, and they know who they are, for all they have done for us.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Facial expression recognition is a great advancement in machine learning. Of the five senses - vision, hearing, smell, taste, and touch - vision is undoubtedly the one that we have come to depend upon above all others, and indeed the one that provides most of the data we receive. Human can percepts other peoples expression / emotion using their vision, in which field machines are far behind than humans. We tried to formulate a method which will try to give the capabilities to a machine to percept human facial expressions.

The facial recognition system incorporates a series of steps. It begins by extracting the face from the image, generate eigenface by principal component alanysis (pca), takeing top most features to use with the test images. Then the test images are also normalised using the feature vector and comparing each test image with all training images to find the similarity between them. Each test image has the same emotion expressed of the most similar train image. The face is detected from any frontal view image is out of the scope of this thesis, but we incorporated an well developed face detection technique to formulate an advanced system which will be able to recognize facial expression from a human face in any complex color image. We have used GeorgiaTech face database, CalTech face database, and local peoples faces collected by ourselves. Among the images, some are used as the training images. The corresponding expression for any training image is saved in a file.

The face detection technique used lighting compensation to improve the performance of color-based scheme, and reduce the computation complexity of feature based scheme. This method is effective on facial variations such as dark/bright vision, close eyes, open mouth, a half profile face, and pseudo faces. After detecting the face region, we have cropped the face and resize them into a fixed size to be processed by our proposed system.

The feature extraction method locates top most features by first converting the images into eigenface using the pca. All the images are added in an array and their mean is deducted from each image. Then principal component analysis is performed over the dataset. From the coefficient matrix provided by the dataset, we took 30 to 40 features to perform the expression detection algorithm to the test images.

The test images are also went through some pre-processing before recognizing the actual expression on the image. Face are detected from the test images and cropped to the specific size. Then their mean image is subtracted from each of them. The feature vector is then projected on them to be compared with the training images. Each test image is first compared with neutral images and their euclidean distance is taken. Then each test image is compared with all the training images and the minimum euclidean distance is taken. By comparing the distances among neutral distance and other distance, we were able to find in which face space the test image is likely fall. The test image has the same emotion expressed in it as the most similar train image.

# Chapter 2

# Background Study

In the recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. This emerging field has been a research interest for scientists from several different scholastic tracks, i.e., computer science, engineering, psychology, and neuroscience. These studies focus not only on improving computer interfaces, but also on improving the actions the computer takes based on feedback from the user. Feedback from the user has traditionally been given through the keyboard and mouse. Other devices have also been developed for more application specific interfaces, such as joysticks, trackballs, datagloves, and touch screens. The rapid advance of technology in recent years has made computers cheaper and more powerful, and has made the use of microphones and PC-cameras affordable and easily available. The microphones and cameras enable the computer to “see” and “hear,” and to use this information to act. A good example of this is the “Smart-Kiosk”.

Psychologists and engineers alike have tried to analyze facial expressions in an attempt to understand and categorize these expressions. This knowledge can be for example used to teach computers to recognize human emotions from video images acquired from built-in cameras. In some applications, it may not be necessary for computers to recognize emotions. For example, the computer inside an automatic teller machine or an airplane probably does not need to recognize emotions. However, in applications where computers take on a social role such as an “instructor,” “helper,” or even “companion,” it may enhance their functionality to be able to recognize users emotions. In her book, Picard suggested several applications where it is beneficial for computers to recognize human emotions. For example, knowing the users emotions, the computer can become a more effective tutor. Synthetic speech with emotions in the voice would sound more pleasing than a monotonous voice. Computer "agents" could learn the user's preferences through the users' emotions. Another application is to help the human users monitor their stress level. In clinical settings, recognizing a person's inability to express certain facial expressions may help diagnose early psychological disorders.

Although the automated recognition of facial expressions has been studied with much interest in the past 10 years, it is still a challenging task for a computer program. State-based representation of facial expressions has been investigated by some researchers. However, the impact of such representation on recognition robustness has not received much attention. In general there are two approaches to represent the face and consequently the facial features to perform facial expression analysis: the geometric feature-based methods and appearance-based methods. The geometric facial feature-based methods present the shape, texture and/or location information of prominent components such as the mouth, eyes, nose, eyebrow, and chin, which can cover the variation in the appearance of the facial expression. The appearance-based methods, on the other hand, using image filters such as Gabor wavelets, generate the facial feature for either the whole-face or specific regions in a face image. Fiducial points are a set of facial salient points, usually located on the corners of the eyes, corners of the eyebrows, corners and outer mid points of the lips, corners of the nostrils, tip of the nose, and the tip of the chin. Automatically detecting fiducial points can extract the prominent characteristics of facial expressions with the distances between points and the relative sizes of the facial components and form the feature vector. Using fiducial points to model the position of the prominent features one can symbolize the face geometry in a local manner. The number of fiducial points used varies and mainly depends on the desired representation, as it is reported that different positions hold different information regarding the expressions. Additionally, choosing the feature points should represent the most important characteristics on the face and be extracted easily. In other words, the number of feature points should represent enough information and not be too many.

Some researchers used a local parameterized model of image motion obtained from optical flow analysis. They utilize a planar model for rigid facial motion and an affineplus- curvature model for non rigid motion. Essa and Pentland first locate the nose, eyes and mouth. Then, from two consecutive normalized frames, a 2D spatio-temporal motion energy representation of facial motion is used as a dynamic face model. Another researchers use feature points that are automatically tracked using hierarchical optical flow method. The feature vectors, used for the recognition, are created by calculating the displacement of the facial points. The displacement of a point is obtained by subtracting its normalized position in the first frame from its current normalized position. Some proposed a feature-based method, which uses geometric and motion facial features and detects transient facial features. The extracted features (mouth, eyes, brows and cheeks) are represented with geometric and motion parameters. The furrows are also detected using a Canny edge detector to measure orientation and quantify their intensity. The parameters of the lower and upper face are then fed into separate neural networks trained to recognize AUs. In most facial expression recognizers, facial feature extractions followed by classification into an expression class.

# Chapter 3

# PCA and Eigenface Based Expression Recognition

## 3.1 Introduction to PCA and Eigenface

The Principal Component Analysis (PCA) is one of the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables. The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc.

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgement of features, and use this information to encode and compare individual face images.

In mathematical terms, we wish to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as a point (or vector) in a very high dimensional space. The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images.

These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face which are the eigenface. Each eigenface deviates from uniform color where some facial feature differs among the set of training face, they are a sort of map of the variations between faces.

Each individual face can be represented exactly in terms of a linear combination of eigenfaces. Each face can also be approximated using only the "best" eigenfaces - those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images. The best M eigenfaces span an M-dimensional subspace - "face space" - of all possible images.

Face recognition and expression recognition has many applicable areas. Moreover, it can be categorized into face identification, face classification, sex determination, or emotion detection. The most useful applications contain crowd surveillance, video content indexing, personal identification (ex. drivers licence), mug shots matching, entrance security, etc. The main idea of using PCA for facial expression recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors). The details are described in the following section.

## 3.2 Mathematics of PCA

A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. Lets suppose we have $M$ vectors of size $N$ (= rows of image $\times$ columns of image) representing a set of sampled images. $p_j$'s represent the pixel values.

$$x_i = [p_1 \dots p_N]^T, i = 1, \dots, M \tag{3.1}$$

The images are mean centered by subtracting the mean image from each image vector. Let $m$ represent the mean image.

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{3.2}$$

And let $w_i$ be defined as mean centered image

$$w_i = x_i - m \tag{3.3}$$

Our goal is to find a set of $e_i$'s which have the largest possible projection onto each of the $w_i$'s. We wish to find a set of $M$ orthonormal vectors $e_i$ for which the quantity

$$\lambda_i = \frac{1}{M} \sum_{n=1}^{M} (e_i^T w_n)^2 \tag{3.4}$$

is maximized with the orthonormality constraint

$$e_l^T e_k = \delta_{lk} \tag{3.5}$$

It has been shown that the $e_i$'s and $\lambda_i$'s are given by the eigenvectors and eigenvalues of the covariance matrix

$$C = WW^T \tag{3.6}$$

where $W$ is a matrix composed of the column vectors $w_i$ placed side by side. The size of $C$ is $N \times N$ which could be enormous. For example, images of size $64 \times 64$ create the covariance matrix of size $4096 \times 4096$. It is not practical to solve for the eigenvectors of $C$ directly. A common theorem in linear algebra states that the vectors $e_i$ and scalars $\lambda_i$ can be obtained by solving for the eigenvectors and eigenvalues of the $M \times M$ matrix $W^TW$. Let $d_i$ and $\mu_i$ be the eigenvectors and eigenvalues of $W^TW$, respectively.

$$W^TWd_i = \mu_i d_i \tag{3.7}$$

By multiplying left to both sides by $W$

$$WW^T(Wd_i) = \mu(Wd_i) \tag{3.8}$$

which means that the first $M-1$ eigenvectors $e_i$ and eigenvalues $\lambda_i$ of $WW^T$ are given by $Wd_i$ and $\mu_i$, respectively. $Wd_i$ needs to be normalized in order to be equal to $e_i$. Since we only sum up a finite number of image vectors, $M$, the rank of the covariance matrix cannot exceed $M-1$ (The -1 come from the subtraction of the mean vector $m$).

The eigenvectors corresponding to nonzero 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. They decrease in exponential fashion, meaning that the roughly 90the total variance is contained in the first 5

A facial image can be projected onto $M'(<< M)$ dimensions by computing

$$\Omega = [v_1 v_2 \dots v_{M'}]^T \tag{3.9}$$

where $v_i = e_i^T w_i$. $v_i$ is the $i^{th}$ coordinate of the facial image in the new space, which came to be the principal component. The vectors $e_i$ are also images, so called, *eigenimages*, or *eigenfaces* in our case, which was first named by [1]. They can be viewed as images and indeed look like faces. So, $\Omega$ describes the contribution of each eigenface in representing the facial image by treating the eigenfaces as a basis set for facial images. The simplest method for determining which face class provides the best description of an input facial image is to find the face class $k$ that minimizes the Euclidean distance

$$\epsilon_k = \|(\Omega - \Omega_k)\| \tag{3.10}$$

where $\Omega_k$ is a vector describing the $k^{th}$ face class. If $\epsilon_k$ is less than some predefined threshold $\theta_\epsilon$, a face is classified as belonging to the class $k$.

## 3.3 Facial Expression Recognition Method

### 3.3.1 Training Phase

1. Detect the face from input image. We took several types of images such as semi normalized and complex. But we need to get the face from those images. That's why we used an algorithm for getting face from images.

2. Crop and resize the face for specific dimension. From those complex images, we took specific dimension of images by using cropping and resizing.

3. Attach the resized images to create the dataset. After that, we created a dataset using these resized images.

4. Find the mean of the images. Then, we got mean of the images using the following equation:
$$m = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{3.11}$$

5. Subtract the mean from each image. Subtract mean image from each image and got new dataset. Now it is ready to apply PCA.

6. Apply PCA over the whole dataset. We used two formulas for getting Eigen

faces and Eigen value.

$$\lambda_i = \frac{1}{M} \sum_{n=1}^{M} (e_i^T w_n)^2 \tag{3.12}$$

$$C = WW^T \tag{3.13}$$

7. From the coefficient matrix, choose the specific number of features for eigenvalues. Make a coefficient matrix and set the number of feature which will be use as Eigen values.

### 3.3.2 Testing Phase

1. Detect the face from a test image. We detect the face with the same procedure we used for the training images.

2. Crop and resize the face for specific dimension. From those training images, we took specific dimension of images by using cropping and resizing.

3. Attach the resized images to create the dataset. After that, we created a dataset using these resized images.

4. Find the mean of the images. Then, we got mean of the images

5. Subtract the mean from each image. Subtract mean image from each image and got new dataset. Now it is ready to apply PCA.

6. Project the feature vector over the normalised test images. Then we project the feature vector we got from the co-varience matrix of the training images.

7. Get the mean of Neutral images. We calculate the mean of neutral image.

8. Find the Euclidean distance between each image and Neutral Image. Now we just calculate the Euclidean distance between each image and neutral image.

9. For each test image, take the minimum euclidean distance from all the training images. Compare testing image with each training image. Minimum Euclidean distance will be selected.

10. Compare the neutral distance and other distance, and take the minimum. Compare the neutral distance with other expression of the images and took the minimum.

11. If neutral is minimum, the expression is neutral. If neutral is minimum then we will be able to take decision is testing image is neutral. Otherwise we compare with other images and took minimum and take decision.

12. Otherwise, the expression on the test image is similar to the minimum distanced train image.

# Chapter 4

# Test Result Analysis

We have simulated our algorithm over three facial datasets. The normalised dataset was from GeorgiaTech Face Database. It had 81 faces of a person with 5 different expressions. All the images are taken exclusively for facial expression research.

The second dataset we used are a collection of photo's that are taken by ourselves of Bangladeshi people who worked at Ergo Ventures Limited. These images are semi normalised as they are taken in front of a single background. This dataset consists of 55 images.

The third dataset we get from California Institute of Technology (CalTech) face dataset. These are very complex images of people with various background. Most of these images are not taken for any expression recognition analysis, but we used it to find out how well our method works against such complex face image. This dataset had 350 images on it.

We have performed two types of tests for each of the datasets. At first we simulated our method by using a feature vector of length 30. The output files are saved and the analysis are done manually, by checking each image to find whether the result is correct or not. Then we calculated our success rate.

Then we have increased the feature length to 40 features. This increase in feature length increases our method's accuracy, but the execution time was little higher. All the test results for 40 features are included in Appendix A.

## 4.1 Analysis on tests using 30 facial features

Using 30 features vector we tested three set of data which are normalized, semi-normalized and complex.

### 4.1.1 GeorgiaTech Face Dataset (Normalised)

Table 4.1: Result on normalised dataset with 30 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***5*** | | | | |
| Disgust | 1 | ***4*** | 1 | 3 | 1 |
| Anger | | 1 | ***5*** | 1 | |
| Sad | | 1 | 1 | ***2*** | |
| Neutral | 2 | 1 | 1 | | ***1*** |
| **Accuracy:** | 62.50% | 57.14% | 62.50% | 33.33% | 50.00% |
| **Total Images:** | **31** | | | | |
| **Total Matched:** | **17** | | | | |
| **Overall Accuracy:** | **54.84%** | | | | |

### 4.1.2 Local Face Dataset (Semi-Normalised)

Table 4.2: Result on semi-normalised dataset with 30 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***6*** | | | 1 | 3 |
| Disgust | 2 | ***9*** | 1 | | 2 |
| Anger | 1 | 5 | ***3*** | 1 | 3 |
| Sad | 2 | 2 | 1 | ***1*** | |
| Neutral | 3 | 2 | 1 | | ***6*** |

Table 4.2: *continued*

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| **Accuracy:** | 42.85% | 50.00% | 50.00% | 33.33% | 42.86% |
| **Total Images:** | **55** | | | | |
| **Total Matched:** | **25** | | | | |
| **Overall Accuracy:** | **45.45%** | | | | |

### 4.1.3 CalTech Face Dataset (Complex)

Table 4.3: Result on complex dataset with 30 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***30*** | 5 | 4 | 7 | 18 |
| Disgust | 16 | ***19*** | 11 | 12 | 21 |
| Anger | 5 | 9 | ***12*** | 9 | 13 |
| Sad | 7 | 3 | 5 | ***19*** | 11 |
| Neutral | 24 | 17 | 10 | 17 | ***46*** |
| **Accuracy:** | 36.59% | 35.85% | 28.57% | 29.69% | 42.20% |
| **Total Images:** | **350** | | | | |
| **Total Matched:** | **126** | | | | |
| **Overall Accuracy:** | **36.00%** | | | | |

### 4.1.4 Comparison of three dataset

The above analysis can be shown in the following chart to show comparison between them.

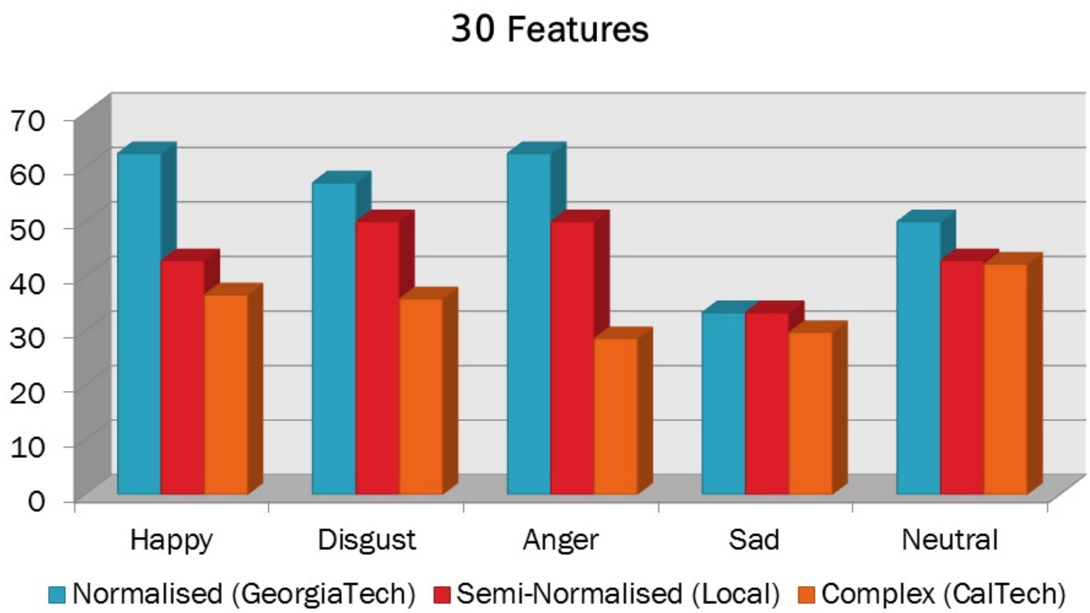


Figure 4.1: Comparison between three dataset using 30 features

Most of the cases normalized data give more accurate result from semi-normalized and semi normalized gives more accurate result from complex. In figure 4.1 we observed that normalized data gives maximum accurate result. So, if we use normalized data we will get more accurate result.

## 4.2 Analysis on tests using 40 facial features

Similarly, using 40 features vector we tested three set of data which are normalized, semi-normalized and complex.

### 4.2.1 GeorgiaTech Face Dataset (Normalised)

Table 4.4: Result on normalised dataset with 40 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***8*** | | | | |
| Disgust | | ***5*** | 2 | 2 | |
| Anger | | 2 | ***5*** | | |
| Sad | | | | ***4*** | |
| Neutral | | | 1 | | ***2*** |
| **Accuracy:** | 100.00% | 71.43% | 62.50% | 66.67% | 100.00% |
| **Total Images:** | **31** | | | | |
| **Total Matched:** | **24** | | | | |
| **Overall Accuracy:** | **77.42%** | | | | |

### 4.2.2 Local Face Dataset (Semi-Normalised)

Table 4.5: Result on semi-normalised dataset with 40 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***9*** | 2 | 1 | 1 | 2 |
| Disgust | 4 | ***14*** | | | 4 |
| Anger | | 2 | ***3*** | | |
| Sad | | | 1 | ***2*** | |
| Neutral | 1 | | 1 | | ***8*** |

Table 4.5: *continued*

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| **Accuracy:** | 64.29% | 77.78% | 50.00% | 66.67% | 57.14% |
| **Total Images:** | **55** | | | | |
| **Total Matched:** | **36** | | | | |
| **Overall Accuracy:** | **65.45%** | | | | |

### 4.2.3 CalTech Face Dataset (Complex)

Table 4.6: Result on complex dataset with 40 features

| | **Happy** | **Disgust** | **Anger** | **Sad** | **Neutral** |
|---|---|---|---|---|---|
| Happy | ***36*** | 4 | 4 | 5 | 13 |
| Disgust | 16 | ***22*** | 11 | 12 | 19 |
| Anger | 2 | 9 | ***16*** | 8 | 9 |
| Sad | 6 | 3 | 2 | ***22*** | 6 |
| Neutral | 22 | 15 | 9 | 17 | ***62*** |
| **Accuracy:** | 43.90% | 41.51% | 38.10% | 34.38% | 56.88% |
| **Total Images:** | **350** | | | | |
| **Total Matched:** | **158** | | | | |
| **Overall Accuracy:** | **45.14%** | | | | |

### 4.2.4 Comparison of three dataset

The above analysis can be shown in the following chart to show comparison between them.

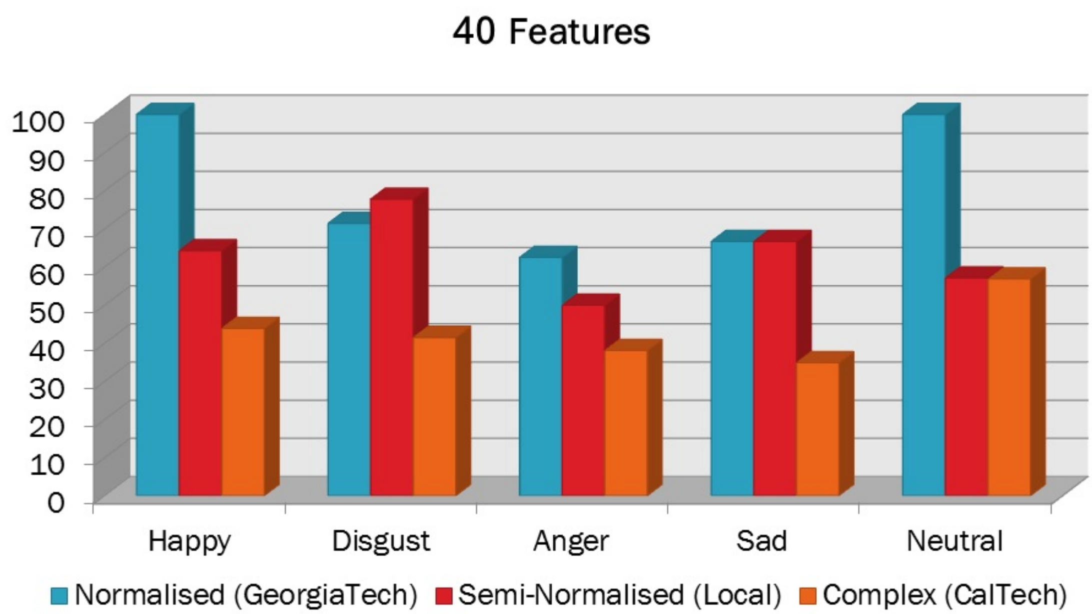


Figure 4.2: Comparison between three dataset using 40 features

Similarly, normalized data give more accurate result from semi-normalized and semi normalized gives more accurate result from complex in 40 features. In figure 4.2 we observed that normalized data gives maximum accurate result. So if we use normalized data we will get more accurate result.

## 4.3 Performance comparison of the feature vectors

We can show the performance graph between two set of features.

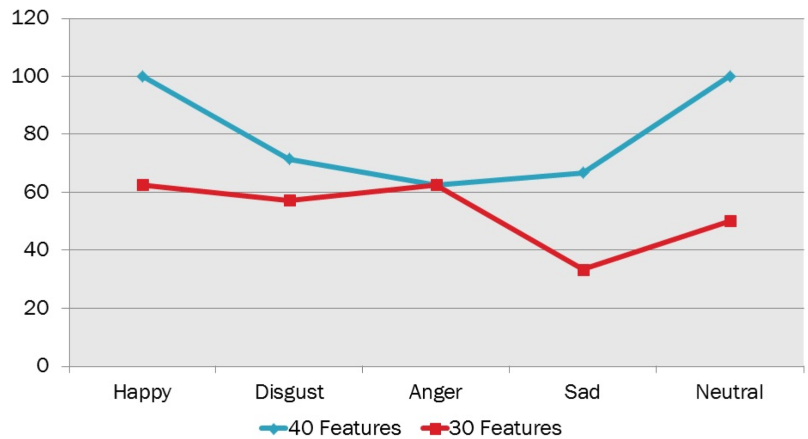


Figure 4.3: Comparison between two feature-lengths in Normalised dataset

In the normalised dataset, 40 features gives more accurate result than 30 features.

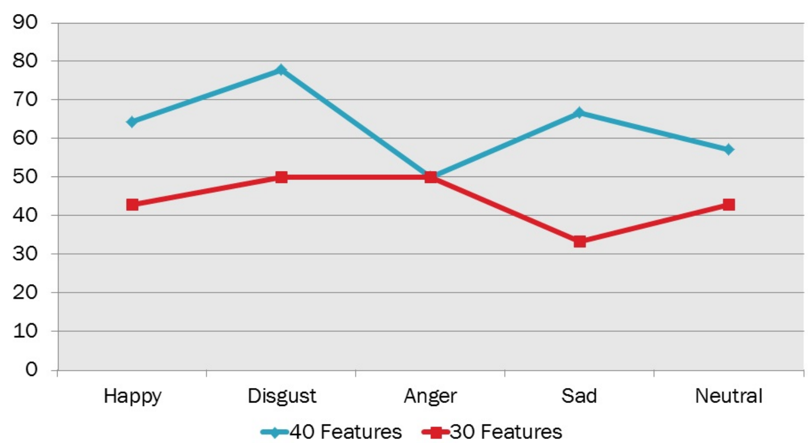


Figure 4.4: Comparison between two feature-lengths in Semi-Normalised dataset

In the semi-normalised dataset, 40 features gives even more accurate result than 30 features.

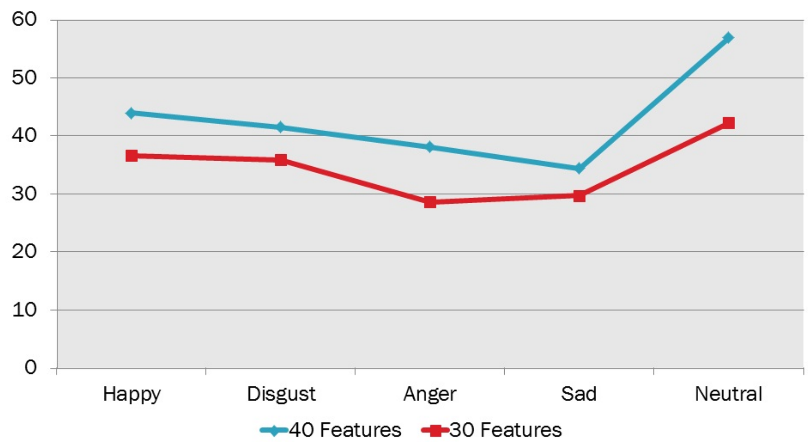


Figure 4.5: Comparison between two feature-lengths in Complex dataset

In the complex dataset, 40 features gives little more accurate result than 30 features.

Of course normalized data is better than complex. But if we change the number of features, then accuracy will be changed. In our testing we again test with 40 features. In this stage we got more accurate result from previous 30 features dataset for three types of data. So we can say that if we increase the number of features, of course the accuracy will be increased. So number of feature is a factor of accuracy.

# Chapter 5

# Conclusion

We tried to develop a facial expression recognition system which gives better performance. Since we have used the top most features of the images and by that feature vector images are normalized, we think it has acceptable performance rate for a facial expression recognition system. We have used some complex images in which the emotion expressed in them are recognized by our system successfully. We tried with 40 and 30 features to simulate the expression detection algorithm to the test images and we have achieved better performance for 40 features than 30 features. So we can say, by using more features, our system will give the result more accurate.

# Appendix A

# Test Data

## A.1 Test result for 40 features over normalised dataset (GeorgiaTech Face Database)

Table A.1: Test data from normalised dataset

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 1 | Image001 | Neutral | Neutral | 1 |
| 2 | Image002 | Happy | Happy | 1 |
| 3 | Image003 | Disgust | Disgust | 1 |
| 4 | Image004 | Anger | Anger | 1 |
| 5 | Image005 | Anger | Anger | 1 |
| 6 | Image006 | Happy | Happy | 1 |
| 7 | Image007 | Unhappy | Unhappy | 1 |
| 8 | Image008 | Happy | Happy | 1 |
| 9 | Image009 | Happy | Happy | 1 |
| 10 | Image010 | Happy | Happy | 1 |
| 11 | Image011 | Unhappy | Unhappy | 1 |
| 12 | Image012 | Anger | Anger | 1 |

Table A.1: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 13 | Image013 | Happy | Happy | 1 |
| 14 | Image014 | Happy | Happy | 1 |
| 15 | Image015 | Happy | Happy | 1 |
| 16 | Image016 | Unhappy | Unhappy | 1 |
| 17 | Image017 | Anger | Neutral | 0 |
| 18 | Image018 | Anger | Disgust | 0 |
| 19 | Image019 | Anger | Disgust | 0 |
| 20 | Image020 | Anger | Anger | 1 |
| 21 | Image021 | Disgust | Anger | 0 |
| 22 | Image022 | Disgust | Disgust | 1 |
| 23 | Image023 | Disgust | Disgust | 1 |
| 24 | Image024 | Unhappy | Unhappy | 1 |
| 25 | Image025 | Unhappy | Disgust | 0 |
| 26 | Image026 | Neutral | Neutral | 1 |
| 27 | Image027 | Disgust | Disgust | 1 |
| 28 | Image028 | Unhappy | Disgust | 0 |
| 29 | Image029 | Disgust | Anger | 0 |
| 30 | Image030 | Disgust | Disgust | 1 |
| 31 | Image031 | Anger | Anger | 1 |
| | | | **Total Match:** | **24** |
| | | | **Accuracy:** | **77.42%** |

## A.2 Test result for 40 features on semi-normalised dataset (Local People Faces)

Table A.2: Test data from semi-normalised dataset

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 1 | DSC09434 | Neutral | Disgust | 0 |
| 2 | DSC09435 | Happy | Disgust | 0 |
| 3 | DSC09436 | Disgust | Disgust | 1 |
| 4 | DSC09437 | Disgust | Disgust | 1 |
| 5 | DSC09438 | Unhappy | Happy | 0 |
| 6 | DSC09439 | Happy | Happy | 1 |
| 7 | DSC09440 | Disgust | Happy | 0 |
| 8 | DSC09441 | Disgust | Disgust | 1 |
| 9 | DSC09442 | Happy | Happy | 1 |
| 10 | DSC09443 | Happy | Happy | 1 |
| 11 | DSC09444 | Happy | Disgust | 0 |
| 12 | DSC09445 | Neutral | Happy | 0 |
| 13 | DSC09446 | Neutral | Neutral | 1 |
| 14 | DSC09447 | Happy | Happy | 1 |
| 15 | DSC09448 | Unhappy | Unhappy | 1 |
| 16 | DSC09449 | Happy | Happy | 1 |
| 17 | DSC09450 | Disgust | Disgust | 1 |
| 18 | DSC09451 | Disgust | Angry | 0 |
| 19 | DSC09452 | Angry | Angry | 1 |
| 20 | DSC09453 | Disgust | Disgust | 1 |
| 21 | DSC09454 | Happy | Disgust | 0 |
| 22 | DSC09455 | Happy | Happy | 1 |
| 23 | DSC09456 | Disgust | Disgust | 1 |
| 24 | DSC09457 | Angry | Unhappy | 0 |

Table A.2: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 25 | DSC09459 | Disgust | Disgust | 1 |
| 26 | DSC09460 | Disgust | Angry | 0 |
| 27 | DSC09461 | Neutral | Happy | 0 |
| 28 | DSC09462 | Neutral | Neutral | 1 |
| 29 | DSC09463 | Happy | Happy | 1 |
| 30 | DSC09464 | Happy | Happy | 1 |
| 31 | DSC09465 | Disgust | Disgust | 1 |
| 32 | DSC09466 | Disgust | Happy | 0 |
| 33 | DSC09467 | Disgust | Disgust | 1 |
| 34 | DSC09468 | Angry | Neutral | 0 |
| 35 | DSC09469 | Neutral | Neutral | 1 |
| 36 | DSC09470 | Neutral | Disgust | 0 |
| 37 | DSC09471 | Happy | Happy | 1 |
| 38 | DSC09472 | Happy | Disgust | 0 |
| 39 | DSC09473 | Disgust | Disgust | 1 |
| 40 | DSC09474 | Disgust | Disgust | 1 |
| 41 | DSC09475 | Angry | Angry | 1 |
| 42 | DSC09476 | Angry | Happy | 0 |
| 43 | DSC09477 | Disgust | Disgust | 1 |
| 44 | DSC09478 | Unhappy | Unhappy | 1 |
| 45 | DSC09479 | Neutral | Disgust | 0 |
| 46 | DSC09480 | Neutral | Disgust | 0 |
| 47 | DSC09481 | Happy | Neutral | 0 |
| 48 | DSC09482 | Neutral | Neutral | 1 |
| 49 | DSC09483 | Disgust | Disgust | 1 |
| 50 | DSC09484 | Angry | Angry | 1 |
| 51 | DSC09485 | Neutral | Neutral | 1 |

Table A.2: *continued*

| **No** | **Image Name** | **Actual Expression** | **System Output** | **Match?** |
|---|---|---|---|---|
| 52 | DSC09486 | Neutral | Neutral | 1 |
| 53 | DSC09487 | Disgust | Disgust | 1 |
| 54 | DSC09488 | Neutral | Neutral | 1 |
| 55 | DSC09489 | Neutral | Neutral | 1 |
| | | | **Total Match:** | **36** |
| | | | **Accuracy:** | **65.45%** |

## A.3 Test result for 40 features on complex dataset (CalTech Face Database)

Table A.3: Test data from complex dataset

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 1 | image_0001 | Happy | Disgust | 0 |
| 2 | image_0002 | Disgust | Disgust | 1 |
| 3 | image_0003 | Anger | Anger | 1 |
| 4 | image_0004 | Neutral | Unhappy | 0 |
| 5 | image_0005 | Disgust | Happy | 0 |
| 6 | image_0006 | Happy | Happy | 1 |
| 7 | image_0007 | Disgust | Unhappy | 0 |
| 8 | image_0008 | Anger | Disgust | 0 |
| 9 | image_0009 | Neutral | Happy | 0 |
| 10 | image_0010 | Happy | Unhappy | 0 |
| 11 | image_0011 | Neutral | Neutral | 1 |
| 12 | image_0012 | Disgust | Happy | 0 |
| 13 | image_0013 | Happy | Neutral | 0 |
| 14 | image_0014 | Anger | Anger | 1 |
| 15 | image_0015 | Happy | Happy | 1 |
| 16 | image_0016 | Neutral | Disgust | 0 |
| 17 | image_0017 | Happy | Disgust | 0 |
| 18 | image_0018 | Anger | Anger | 1 |
| 19 | image_0019 | Disgust | Disgust | 1 |
| 20 | image_0020 | Disgust | Neutral | 0 |
| 21 | image_0021 | Neutral | Unhappy | 0 |
| 22 | image_0022 | Happy | Disgust | 0 |
| 23 | image_0023 | Happy | Happy | 1 |
| 24 | image_0024 | Neutral | Neutral | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 25 | image_0025 | Unhappy | Anger | 0 |
| 26 | image_0026 | Unhappy | Happy | 0 |
| 27 | image_0027 | Disgust | Disgust | 1 |
| 28 | image_0028 | Anger | Disgust | 0 |
| 29 | image_0029 | Disgust | Anger | 0 |
| 30 | image_0030 | Neutral | Disgust | 0 |
| 31 | image_0031 | Neutral | Happy | 0 |
| 32 | image_0032 | Unhappy | Neutral | 0 |
| 33 | image_0033 | Happy | Neutral | 0 |
| 34 | image_0034 | Disgust | Disgust | 1 |
| 35 | image_0035 | Anger | Unhappy | 0 |
| 36 | image_0036 | Neutral | Unhappy | 0 |
| 37 | image_0037 | Neutral | Neutral | 1 |
| 38 | image_0038 | Unhappy | Happy | 0 |
| 39 | image_0039 | Happy | Disgust | 0 |
| 40 | image_0040 | Disgust | Neutral | 0 |
| 41 | image_0041 | Anger | Disgust | 0 |
| 42 | image_0042 | Unhappy | Disgust | 0 |
| 43 | image_0043 | Disgust | Happy | 0 |
| 44 | image_0044 | Neutral | Happy | 0 |
| 45 | image_0045 | Unhappy | Disgust | 0 |
| 46 | image_0046 | Disgust | Neutral | 0 |
| 47 | image_0047 | Happy | Neutral | 0 |
| 48 | image_0048 | Anger | Anger | 1 |
| 49 | image_0049 | Disgust | Disgust | 1 |
| 50 | image_0050 | Neutral | Unhappy | 0 |
| 51 | image_0051 | Neutral | Disgust | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 52 | image_0052 | Unhappy | Disgust | 0 |
| 53 | image_0053 | Happy | Happy | 1 |
| 54 | image_0054 | Neutral | Anger | 0 |
| 55 | image_0055 | Unhappy | Neutral | 0 |
| 56 | image_0056 | Unhappy | Disgust | 0 |
| 57 | image_0057 | Happy | Neutral | 0 |
| 58 | image_0058 | Anger | Disgust | 0 |
| 59 | image_0059 | Neutral | Happy | 0 |
| 60 | image_0060 | Disgust | Disgust | 1 |
| 61 | image_0061 | Unhappy | Happy | 0 |
| 62 | image_0062 | Unhappy | Anger | 0 |
| 63 | image_0063 | Unhappy | Disgust | 0 |
| 64 | image_0064 | Neutral | Neutral | 1 |
| 65 | image_0065 | Disgust | Disgust | 1 |
| 66 | image_0066 | Anger | Happy | 0 |
| 67 | image_0067 | Unhappy | Neutral | 0 |
| 68 | image_0068 | Neutral | Disgust | 0 |
| 69 | image_0069 | Happy | Unhappy | 0 |
| 70 | image_0070 | Anger | Disgust | 0 |
| 71 | image_0071 | Neutral | Neutral | 1 |
| 72 | image_0072 | Disgust | Happy | 0 |
| 73 | image_0073 | Unhappy | Anger | 0 |
| 74 | image_0074 | Unhappy | Disgust | 0 |
| 75 | image_0075 | Unhappy | Neutral | 0 |
| 76 | image_0076 | Neutral | Anger | 0 |
| 77 | image_0077 | Happy | Anger | 0 |
| 78 | image_0078 | Anger | Anger | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 79 | image_0079 | Neutral | Unhappy | 0 |
| 80 | image_0080 | Disgust | Disgust | 1 |
| 81 | image_0081 | Unhappy | Unhappy | 1 |
| 82 | image_0082 | Unhappy | Happy | 0 |
| 83 | image_0083 | Unhappy | Neutral | 0 |
| 84 | image_0084 | Unhappy | Disgust | 0 |
| 85 | image_0085 | Happy | Happy | 1 |
| 86 | image_0086 | Anger | Happy | 0 |
| 87 | image_0087 | Neutral | Unhappy | 0 |
| 88 | image_0088 | Disgust | Disgust | 1 |
| 89 | image_0089 | Anger | Disgust | 0 |
| 90 | image_0090 | Neutral | Happy | 0 |
| 91 | image_0091 | Unhappy | Anger | 0 |
| 92 | image_0092 | Neutral | NeutraL | 1 |
| 93 | image_0093 | Happy | Unhappy | 0 |
| 94 | image_0094 | Anger | Anger | 1 |
| 95 | image_0095 | Disgust | Anger | 0 |
| 96 | image_0096 | Neutral | Disgust | 0 |
| 97 | image_0097 | Unhappy | Unhappy | 1 |
| 98 | image_0098 | Happy | Neutral | 0 |
| 99 | image_0099 | Unhappy | Unhappy | 1 |
| 100 | image_0100 | Disgust | Disgust | 1 |
| 101 | image_0101 | Happy | Disgust | 0 |
| 102 | image_0102 | Anger | Happy | 0 |
| 103 | image_0103 | Neutral | Disgust | 0 |
| 104 | image_0104 | Unhappy | Happy | 0 |
| 105 | image_0105 | Neutral | Happy | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 106 | image_0106 | Unhappy | Anger | 0 |
| 107 | image_0107 | Disgust | Disgust | 1 |
| 108 | image_0108 | Happy | Happy | 1 |
| 109 | image_0109 | Anger | Happy | 0 |
| 110 | image_0110 | Neutral | Disgust | 0 |
| 111 | image_0111 | Disgust | Neutral | 0 |
| 112 | image_0112 | Happy | Disgust | 0 |
| 113 | image_0113 | Neutral | Anger | 0 |
| 114 | image_0114 | Neutral | Disgust | 0 |
| 115 | image_0115 | Happy | Neutral | 0 |
| 116 | image_0116 | Anger | Disgust | 0 |
| 117 | image_0117 | Unhappy | Unhappy | 1 |
| 118 | image_0118 | Unhappy | Unhappy | 1 |
| 119 | image_0119 | Disgust | Disgust | 1 |
| 120 | image_0120 | Happy | Happy | 1 |
| 121 | image_0121 | Neutral | Disgust | 0 |
| 122 | image_0122 | Unhappy | Neutral | 0 |
| 123 | image_0123 | Happy | Disgust | 0 |
| 124 | image_0124 | Unhappy | Anger | 0 |
| 125 | image_0125 | Neutral | Happy | 0 |
| 126 | image_0126 | Disgust | Anger | 0 |
| 127 | image_0127 | Happy | Neutral | 0 |
| 128 | image_0128 | Anger | Disgust | 0 |
| 129 | image_0129 | Disgust | Neutral | 0 |
| 130 | image_0130 | Neutral | Neutral | 1 |
| 131 | image_0131 | Unhappy | Disgust | 0 |
| 132 | image_0132 | Neutral | Anger | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 133 | image_0133 | Anger | Anger | 1 |
| 134 | image_0134 | Neutral | Disgust | 0 |
| 135 | image_0135 | Unhappy | Unhappy | 1 |
| 136 | image_0136 | Happy | Neutral | 0 |
| 137 | image_0137 | Neutral | Unhappy | 0 |
| 138 | image_0138 | Anger | Anger | 1 |
| 139 | image_0139 | Neutral | Neutral | 1 |
| 140 | image_0140 | Happy | Disgust | 0 |
| 141 | image_0141 | Neutral | Unhappy | 0 |
| 142 | image_0142 | Disgust | Anger | 0 |
| 143 | image_0143 | Anger | Neutral | 0 |
| 144 | image_0144 | Happy | Neutral | 0 |
| 145 | image_0145 | Neutral | Happy | 0 |
| 146 | image_0146 | Unhappy | Unhappy | 1 |
| 147 | image_0147 | Neutral | Neutral | 1 |
| 148 | image_0148 | Unhappy | Disgust | 0 |
| 149 | image_0149 | Unhappy | Neutral | 0 |
| 150 | image_0150 | Neutral | Anger | 0 |
| 151 | image_0151 | Happy | Disgust | 0 |
| 152 | image_0152 | Anger | Unhappy | 0 |
| 153 | image_0153 | Disgust | Neutral | 0 |
| 154 | image_0154 | Happy | Happy | 1 |
| 155 | image_0155 | Neutral | Disgust | 0 |
| 156 | image_0156 | Unhappy | Neutral | 0 |
| 157 | image_0157 | Unhappy | Disgust | 0 |
| 158 | image_0158 | Neutral | Unhappy | 0 |
| 159 | image_0159 | Unhappy | Neutral | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 160 | image_0160 | Disgust | Disgust | 1 |
| 161 | image_0161 | Anger | Neutral | 0 |
| 162 | image_0162 | Happy | Disgust | 0 |
| 163 | image_0163 | Disgust | Neutral | 0 |
| 164 | image_0164 | Neutral | Happy | 0 |
| 165 | image_0165 | Happy | Happy | 1 |
| 166 | image_0166 | Neutral | Disgust | 0 |
| 167 | image_0167 | Disgust | Unhappy | 0 |
| 168 | image_0168 | Happy | Neutral | 0 |
| 169 | image_0169 | Anger | Disgust | 0 |
| 170 | image_0170 | Neutral | Neutral | 1 |
| 171 | image_0171 | Happy | Happy | 1 |
| 172 | image_0172 | Disgust | Disgust | 1 |
| 173 | image_0173 | Neutral | Anger | 0 |
| 174 | image_0174 | Unhappy | Neutral | 0 |
| 175 | image_0175 | Happy | Happy | 1 |
| 176 | image_0176 | Disgust | Anger | 0 |
| 177 | image_0177 | Anger | Anger | 1 |
| 178 | image_0178 | Neutral | Disgust | 0 |
| 179 | image_0179 | Disgust | Neutral | 0 |
| 180 | image_0180 | Neutral | Happy | 0 |
| 181 | image_0181 | Neutral | Neutral | 1 |
| 182 | image_0182 | Unhappy | Neutral | 0 |
| 183 | image_0183 | Happy | Happy | 1 |
| 184 | image_0184 | Disgust | Neutral | 0 |
| 185 | image_0185 | Happy | Neutral | 0 |
| 186 | image_0186 | Anger | Anger | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 187 | image_0187 | Neutral | Neutral | 1 |
| 188 | image_0188 | Neutral | Disgust | 0 |
| 189 | image_0189 | Happy | Neutral | 0 |
| 190 | image_0190 | Neutral | Neutral | 1 |
| 191 | image_0191 | Neutral | Disgust | 0 |
| 192 | image_0192 | Disgust | Neutral | 0 |
| 193 | image_0193 | Anger | Disgust | 0 |
| 194 | image_0194 | Neutral | Neutral | 1 |
| 195 | image_0195 | Happy | Disgust | 0 |
| 196 | image_0196 | Unhappy | Neutral | 0 |
| 197 | image_0197 | Neutral | Neutral | 1 |
| 198 | image_0198 | Neutral | Neutral | 1 |
| 199 | image_0199 | Disgust | Disgust | 1 |
| 200 | image_0200 | Anger | Anger | 1 |
| 201 | image_0201 | Happy | Happy | 1 |
| 202 | image_0202 | Neutral | Neutral | 1 |
| 203 | image_0203 | Neutral | Anger | 0 |
| 204 | image_0204 | Happy | Happy | 1 |
| 205 | image_0205 | Disgust | Disgust | 1 |
| 206 | image_0206 | Anger | Neutral | 0 |
| 207 | image_0207 | Neutral | Happy | 0 |
| 208 | image_0208 | Neutral | Neutral | 1 |
| 209 | image_0209 | Happy | Neutral | 0 |
| 210 | image_0210 | Disgust | Anger | 0 |
| 211 | image_0211 | Anger | Anger | 1 |
| 212 | image_0212 | Happy | Disgust | 0 |
| 213 | image_0213 | Neutral | Neutral | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 214 | image_0214 | Unhappy | Neutral | 0 |
| 215 | image_0215 | Unhappy | Disgust | 0 |
| 216 | image_0216 | Neutral | Neutral | 1 |
| 217 | image_0217 | Neutral | Disgust | 0 |
| 218 | image_0218 | Happy | Happy | 1 |
| 219 | image_0219 | Happy | Unhappy | 0 |
| 220 | image_0220 | Happy | Happy | 1 |
| 221 | image_0221 | Disgust | Neutral | 0 |
| 222 | image_0222 | Anger | Anger | 1 |
| 223 | image_0223 | Neutral | Neutral | 1 |
| 224 | image_0224 | Neutral | Neutral | 1 |
| 225 | image_0225 | Happy | Happy | 1 |
| 226 | image_0226 | Happy | Happy | 1 |
| 227 | image_0227 | Disgust | Anger | 0 |
| 228 | image_0228 | Neutral | Neutral | 1 |
| 229 | image_0229 | Happy | Happy | 1 |
| 230 | image_0230 | Anger | Disgust | 0 |
| 231 | image_0231 | Unhappy | Anger | 0 |
| 232 | image_0232 | Neutral | Neutral | 1 |
| 233 | image_0233 | Happy | Neutral | 0 |
| 234 | image_0234 | Neutral | Neutral | 1 |
| 235 | image_0235 | Anger | Anger | 1 |
| 236 | image_0236 | Happy | Happy | 1 |
| 237 | image_0237 | Neutral | Neutral | 1 |
| 238 | image_0238 | Anger | Anger | 1 |
| 239 | image_0239 | Happy | Happy | 1 |
| 240 | image_0240 | Neutral | Disgust | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 241 | image_0241 | Unhappy | Neutral | 0 |
| 242 | image_0242 | Neutral | Anger | 0 |
| 243 | image_0243 | Happy | Happy | 1 |
| 244 | image_0244 | Anger | Neutral | 0 |
| 245 | image_0245 | Unhappy | Unhappy | 1 |
| 246 | image_0246 | Neutral | Neutral | 1 |
| 247 | image_0247 | Happy | Happy | 1 |
| 248 | image_0248 | Anger | Neutral | 0 |
| 249 | image_0249 | Neutral | Neutral | 1 |
| 250 | image_0250 | Unhappy | Unhappy | 1 |
| 251 | image_0251 | Neutral | Neutral | 1 |
| 252 | image_0252 | Happy | Disgust | 0 |
| 253 | image_0253 | Disgust | Disgust | 1 |
| 254 | image_0254 | Anger | Neutral | 0 |
| 255 | image_0255 | Happy | Neutral | 0 |
| 256 | image_0256 | Neutral | Neutral | 1 |
| 257 | image_0257 | Unhappy | Unhappy | 1 |
| 258 | image_0258 | Neutral | Neutral | 1 |
| 259 | image_0259 | Unhappy | Unhappy | 1 |
| 260 | image_0260 | Neutral | Neutral | 1 |
| 261 | image_0261 | Happy | Happy | 1 |
| 262 | image_0262 | Disgust | Anger | 0 |
| 263 | image_0263 | Anger | Neutral | 0 |
| 264 | image_0264 | Neutral | Neutral | 1 |
| 265 | image_0265 | Happy | Happy | 1 |
| 266 | image_0266 | Disgust | Disgust | 1 |
| 267 | image_0267 | Anger | Neutral | 0 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 268 | image_0268 | Neutral | Neutral | 1 |
| 269 | image_0269 | Happy | Neutral | 0 |
| 270 | image_0270 | Neutral | Neutral | 1 |
| 271 | image_0271 | Unhappy | Unhappy | 1 |
| 272 | image_0272 | Neutral | Neutral | 1 |
| 273 | image_0273 | Unhappy | Unhappy | 1 |
| 274 | image_0274 | Unhappy | Disgust | 0 |
| 275 | image_0275 | Neutral | Neutral | 1 |
| 276 | image_0276 | Unhappy | Unhappy | 1 |
| 277 | image_0277 | Happy | Happy | 1 |
| 278 | image_0278 | Neutral | Neutral | 1 |
| 279 | image_0279 | Anger | Anger | 1 |
| 280 | image_0280 | Happy | Happy | 1 |
| 281 | image_0281 | Disgust | Unhappy | 0 |
| 282 | image_0282 | Unhappy | Unhappy | 1 |
| 283 | image_0283 | Neutral | Neutral | 1 |
| 284 | image_0284 | Happy | Neutral | 0 |
| 285 | image_0285 | Happy | Happy | 1 |
| 286 | image_0286 | Disgust | Disgust | 1 |
| 287 | image_0287 | Anger | Neutral | 0 |
| 288 | image_0288 | Neutral | Neutral | 1 |
| 289 | image_0289 | Happy | Disgust | 0 |
| 290 | image_0290 | Disgust | Neutral | 0 |
| 291 | image_0291 | Neutral | Happy | 0 |
| 292 | image_0292 | Happy | Happy | 1 |
| 293 | image_0293 | Happy | Happy | 1 |
| 294 | image_0294 | Neutral | Neutral | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 295 | image_0295 | Unhappy | Unhappy | 1 |
| 296 | image_0296 | Disgust | Disgust | 1 |
| 297 | image_0297 | Neutral | Neutral | 1 |
| 298 | image_0298 | Happy | Neutral | 0 |
| 299 | image_0299 | Neutral | Neutral | 1 |
| 300 | image_0300 | Neutral | Neutral | 1 |
| 301 | image_0301 | Unhappy | Unhappy | 1 |
| 302 | image_0302 | Happy | Happy | 1 |
| 303 | image_0303 | Neutral | Neutral | 1 |
| 304 | image_0304 | Unhappy | Unhappy | 1 |
| 305 | image_0305 | Happy | Happy | 1 |
| 306 | image_0306 | Disgust | Anger | 0 |
| 307 | image_0307 | Neutral | Neutral | 1 |
| 308 | image_0308 | Happy | Neutral | 0 |
| 309 | image_0309 | Disgust | Disgust | 1 |
| 310 | image_0310 | Neutral | Neutral | 1 |
| 311 | image_0311 | Unhappy | Unhappy | 1 |
| 312 | image_0312 | Neutral | Neutral | 1 |
| 313 | image_0313 | Happy | Happy | 1 |
| 314 | image_0314 | Neutral | Neutral | 1 |
| 315 | image_0315 | Unhappy | Unhappy | 1 |
| 316 | image_0316 | Happy | Neutral | 0 |
| 317 | image_0317 | Neutral | Neutral | 1 |
| 318 | image_0318 | Unhappy | Unhappy | 1 |
| 319 | image_0319 | Neutral | Neutral | 1 |
| 320 | image_0320 | Unhappy | Neutral | 0 |
| 321 | image_0321 | Unhappy | Unhappy | 1 |

Table A.3: *continued*

| No | Image Name | Actual Expression | System Output | Match? |
|---|---|---|---|---|
| 322 | image_0322 | Neutral | Neutral | 1 |
| 323 | image_0323 | Happy | Unhappy | 0 |
| 324 | image_0324 | Neutral | Neutral | 1 |
| 325 | image_0325 | Happy | Happy | 1 |
| 326 | image_0326 | Neutral | Neutral | 1 |
| 327 | image_0327 | Unhappy | Neutral | 0 |
| 328 | image_0328 | Disgust | Disgust | 1 |
| 329 | image_0329 | Happy | Disgust | 0 |
| 330 | image_0330 | Neutral | Neutral | 1 |
| 331 | image_0331 | Unhappy | Neutral | 0 |
| 332 | image_0332 | Neutral | Neutral | 1 |
| 333 | image_0333 | Happy | Happy | 1 |
| 334 | image_0334 | Neutral | Neutral | 1 |
| 335 | image_0335 | Neutral | Neutral | 1 |
| 336 | image_0336 | Happy | Neutral | 0 |
| 337 | image_0337 | Disgust | Neutral | 0 |
| 338 | image_0338 | Neutral | Neutral | 1 |
| 339 | image_0339 | Happy | Happy | 1 |
| 340 | image_0340 | Disgust | Neutral | 0 |
| 341 | image_0341 | Happy | Anger | 0 |
| 342 | image_0342 | Neutral | Unhappy | 0 |
| 343 | image_0343 | Happy | Neutral | 0 |
| 344 | image_0344 | Neutral | Neutral | 1 |
| 345 | image_0345 | Unhappy | Anger | 0 |
| 346 | image_0346 | Happy | Unhappy | 0 |
| 347 | image_0347 | Disgust | Neutral | 0 |
| 348 | image_0348 | Neutral | Neutral | 1 |

Table A.3: *continued*

| **No** | **Image Name** | **Actual Expression** | **System Output** | **Match?** |
|---|---|---|---|---|
| 349 | image_0349 | Neutral | Neutral | 1 |
| 350 | image_0350 | Happy | Anger | 0 |
| | | | **Total Match:** | **158** |
| | | | **Accuracy:** | **45.14%** |

# Appendix B

# MATLAB Code for our System

## B.1 Face Detection

```
% Originaly by Tolga Birdal
% Implementation of the paper:
% "A simple and accurate face detection algorithm
% in complex background"
% by Yu-Tang Pai, Shanq-Jang Ruan, Mon-Chau Shie,
% Yi-Chi Liu

% Additions by Tolga Birdal:
%  Minimum face size constraint
%  Adaptive theta thresholding (Theta is thresholded by
% mean2(theata)/4
%  Parameters are modified by to detect better. Please
% check the paper for
%  parameters they propose.
% Check the paper for more details.

% usage:
%  I=double(imread('c:\Data\girl1.jpg'));
```

```
%  detect_face(I);
% The function will display the bounding box if a
% face is found.

function [aa,SN_fill,FaceDat]=detect_face(I)

close all;

% No faces at the beginning
Faces=[];
numFaceFound=0;

I=double(I);

H=size(I,1);
W=size(I,2);

%%%%%%%%%%%%%%%%%% LIGHTING COMPENSATION %%%%%%%%%%%%%

C=255*imadjust(I/255,[0.3;1],[0;1]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% EXTRACT SKIN %%%%%%%%%%%%%%%%%
YCbCr=rgb2ycbcr(C);
Cr=YCbCr(:,:,3);

S=zeros(H,W);
[SkinIndexRow,SkinIndexCol] =find(10<Cr & Cr<255);
for i=1:length(SkinIndexRow)
    S(SkinIndexRow(i),SkinIndexCol(i))=1;
```

```
end

m_S = size(S);
S(m_S(1)-7:m_S(1),:) = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%% REMOVE NOISE %%%%
SN=zeros(H,W);
for i=1:H-5
    for j=1:W-5
        localSum=sum(sum(S(i:i+4, j:j+4)));
        SN(i:i+5, j:j+5)=(localSum>20);
    end
end

Iedge=edge(uint8(SN));

SE = strel('square',9);
SN_edge = (imdilate(Iedge,SE));

SN_fill = imfill(SN_edge,'holes');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%% FIND SKIN COLOR BLOCKS %%%%

[L,lenRegions] = bwlabel(SN_fill,4);
AllDat  = regionprops(L,'BoundingBox','FilledArea');
AreaDat = cat(1, AllDat.FilledArea);
[maxArea, maxAreaInd] = max(AreaDat);
```

```
FaceDat = AllDat(maxAreaInd);
FaceBB = [FaceDat.BoundingBox(1),FaceDat.BoundingBox(2),...
    FaceDat.BoundingBox(3),FaceDat.BoundingBox(4)];

aa=imcrop(rgb2gray(uint8(I)).*uint8(SN_fill),FaceBB);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
```

## B.2 Facial Expression Recognition

```
%% #################### Facial Expression Recognition #####
%% Many conceptions here are taken from:
%% M. Turk and A. Pentland, "Eigenfaces for Recognition",
%% Journal of Cognitive Neuroscience, March 1991
%%
%% This code requires the following toolboxes:
%%          1. Image Processing Toolbox (For Resizing Image)
%%          2. Statistics Toolbox (For PCA)
%% #########################################################

function [isSucceed] =
EigenFace(strTrainPath, strLabelFile, strTestPath)
isSucceed = 0;
if (exist('strTrainPath')==0)
    strTrainPath = input('Enter Train Folder Name:','s');
end
if (exist('strLabelFile')==0)
    strLabelFile = input('Enter Label File Name:','s');
end
```

```
if (exist('strTestPath')==0)
    strTestPath = input('Enter Test Folder Name:','s');
end

fid=fopen(strLabelFile);
imageLabel=textscan(fid,'%s %s','whitespace',',');
fclose(fid);

NeutralImages=[];
for i=1:length(imageLabel{1,1})
    if (strcmp(lower(imageLabel{1,2}{i,1}),'neutral'))
        NeutralImages=[NeutralImages,i];
    end
end
if (length(NeutralImages)==0)
  disp('ERROR: Neutral Expression is not available in training');
  return;
end

structTestImages = dir(strTestPath);
numImage = length(imageLabel{1,1});
% Total Observations: Number of Images in training set
lenTest = length(structTestImages);

if (lenTest==0)
    disp('Error:Invalid Test Folder');
    return;
end

TrainImages='';
for i = 1:numImage
```

```
TrainImages{i,1} =
  strcat(strTrainPath,'\',imageLabel{1,1}(i));
end

j=0;
for i = 3:lenTest
     if ((~structTestImages(i).isdir))
         if  (structTestImages(i).name(end-3:end)=='.jpg')
             j=j+1;
             TestImages{j,1} =
               [strTestPath,'\',structTestImages(i).name];
         end
     end
end
numTestImage = j; % Number of Test Images
clear ('structTestImages','fid','i','j');

imageSize = [280,180];
% All Images are resized into a common size

%% ################# Load Train Data & Preprocess  ##########
%% Load training images & preparing for PCA by subtracting mean

img = zeros(imageSize(1)*imageSize(2),numImage);
for i = 1:numImage
    aa = imresize(detect_face(imresize(imread(
      cell2mat(TrainImages{i,1})),[375,300])),imageSize);
    img(:,i) = aa(:);
    disp(sprintf('Loading Train Image # %d',i));
end
meanImage = mean(img,2);
```

```
img = (img - meanImage*ones(1,numImage))';
% img is the input to PCA
imshow(img(:,2))
    Willexit = input('Press Enter to Quit ...','s');
%% ##############################################################

%% ################# Low Dimension Face Space Construction ###
[C,S,L]=princomp(img,'econ');            % Performing PCA Here
EigenRange=[1:40];% Defines which Eigenvalues will be selected

C = C(:,EigenRange);
%% ##############################################################


%% ############# Load Test Data and project on Face Space ####
img = zeros(imageSize(1)*imageSize(2),numTestImage);
for i = 1:numTestImage
    aa = imresize(detect_face(imresize(imread(TestImages{i,1}),
      [375,300])),imageSize);
    img(:,i) = aa(:);
    disp(sprintf('Loading Test Image # %d',i));
end
meanImage = mean(img,2);
img = (img - meanImage*ones(1,numTestImage))';
Projected_Test = img*C;
%% ##############################################################


%% ################# Calculation of Distance from Neutral ####
meanNutral = mean(S(NeutralImages,EigenRange)',2);
```

```
for Dat2Project = 1:numTestImage
    TestImage = Projected_Test(Dat2Project,:);
    % Picking the image #Dat2Project

    Eucl_Dist(Dat2Project) = sqrt((TestImage'-meanNutral)'*
      (TestImage'-meanNutral));
    % Here, the distance between the expression under test and
    % the mean neutral expressions is being calculated
end
%Eucl_Dist = Eucl_Dist/max(Eucl_Dist);
%% ##############################################################

%% ################# Calculation of other Distances ##########
Other_Dist = zeros(numTestImage,numImage);
for Dat2Project = 1:numTestImage
    TestImage = Projected_Test(Dat2Project,:);
    % Picking the image #Dat2Project
    for i = 1:numImage
      Other_Dist(Dat2Project,i) = sqrt(
      (TestImage'-S(i,EigenRange)')'*
      (TestImage'-S(i,EigenRange)'));
    end
end
[Min_Dist,Min_Dist_pos] = min(Other_Dist,[],2);
%% ###############################################################


%% ########################## Save Result ####################
fid = fopen('Results.txt','w');
fprintf(fid,'//Test Image, Expression, Best Match\r\n');
```

```
for i = 1:numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid,'%s, %s, %s\r\n',Test_Image,Expr,Best_Match);
end
fclose(fid);
%% ##############################################################
isSucceed = 1;
disp('Done')
disp('Output File = .\Results.txt');
Willexit = input('Press Enter to Quit ...','s');
end
```

# References


[1] **Classification, Parameter Estimation and State Estimation**
An Engineering Approach using MATLAB
*F. van der Heijden, R.P.W. Duin, D. de Ridder, D.M.J. Tax*
Published in 2004 at John Wiley & Sons Ltd

[2] **Eigenfaces for Recognition**
*Matthew Turk and Alex Pentland*
Published in 1991 at Journal of Cognitive Neuroscience - Volume 3, Number 1

[3] **Practical Algorithms for Image Analysis**
*Michael Seul, Lawrence O'Gorman and Michael J. Sammon*
Published in 2000 at Cambridge University Press

[4] **Automatic Pixel Selection for Optimizing Facial Expression Recognition using Eigenfaces**
*Carmen Frank and Elmar Noth*
Published in 2003 at DAGM 2003, Springer-Verlag

[5] **Facial Expression Recognition: A Brief Tutorial Overview**
*Claude C. Chibelushi, Fabrice Bourel*
Published in 2002

[6] **Automatic Facial Expression Recognition Using Linear and Nonlinear Holistic Spatial Analysis**
*Rui Ma, Jiaxin Wang*

[7] **Mood Detection: Implementing a facial expression recognition system**
*Neeraj Agrawal, Rob Cosgriff and Ritvik Mudur*

[8] **Automated Facial Expression Recognition System**
*Andrew Ryan, Jeffery F. Cohn, Simon Lucey, Jason Saragih, Patrick Lucey, & Fernando De la Torre, Adam Rossi*
Published in 2009 at IEEE

[9] **2D/3D Facial Feature Extraction**
*Hatice nar Akakn, Albert Ali Salah, Lale Akarun, Blent Sankur*
Published in 2006 at Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, SPIE Vol. 6064

[10] **Computational and performance aspects of PCA-based face-recognition algorithms**
*Hyeonjoon Moon, P Jonathon Phillips*
Published in 2001

[11] **FACIAL FEATURE EXTRACTION USING TOPOLOGICAL METHODS**
*Aysegul Gunduz, Hamid Krim*
Published in 2003 at IEEE

[12] **Multi-Attribute Robust Facial Feature Localization**
*Oya C eliktutan, Hatice C nar Akakn, Bulent Sankur*

[13] **A Simple and Efficient Supervised Method for Spatially Weighted PCA in Face Image Analysis**
*Carlos E. Thomaz, Gilson A. Giraldi, Joaquim F. P. da Costa, Duncan F. Gillies*
Published in 2010

[14] **Analysis of Facial Expression using Gabor and SVM**
*Praseeda Lekshmi.V, Dr.M.Sasikumar*
Published in 2009 at International Journal of Recent Trends in Engineering, Vol. 1, No. 2

[15] **A Novel Feature Extraction Method for Facial Expression Recognition**
*Xiaoyi Feng, Baohua Lv, Zhen Li, Jiling Zhang*

[16] **Facial Feature Extraction and Principal Component Analysis for Face Detection in Color Images**
*MobiSyM*

[17] **Local versus Global Face Analysis for Facial Expression Recognition**
*Jacob Whitehill, Christian Omlin*

[18] **Walsh-Hadamard Transform for Facial Feature Extraction in Face Recognition**
*M. Hassan, I. Osman, and M. Yahia*
Published in 2007 at World Academy of Science, Engineering and Technology

[19] **Feature Extraction for Facial Expression Recognition based on Hybrid Face Regions**
*Seyed M. LAJEVARDI, Zahir M. HUSSAIN*
Published in 2009 at Advances in Electrical and Computer Engineering V9, N3

[20] **Machine Vision - Theory, Algorithms, Practicalities**
*E. R. Davies*
Published at Morgan Kaufmann

[21] **Machine Learning in Computer Vision**
*N. Sebe, Ira Cohen, Ashutosh Garg, Thomas S. Huang*
Published in 2005 at Springer

[22] **Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction**
*Marian Stewart Bartlett, Gwen Littlewort, Ian Fase, Javier R. Movellan*

[23] **Facial Expression Recognition**
*Neeta Sarode, Prof. Shalini Bhatia*
Published in 2010 at International Journal on Computer Science and Engineering

[24] **Comprehensive Database for Facial Expression Analysis**
*Takeo Kanade, Jeffrey F. Cohn, Yingli Tian*

[25] **Facial Expression Recognition: A Fully Integrated Approach**
*Roberto Valenti, Nicu Sebe, Theo Gevers*
Published in 2007 at VMDL07

[26] **Vision Systems - Segmentation and Pattern Recognition**
*Goro Obinata and Ashish Dutta*
Published in 2007 at I-Tech Education and Publishing

[27] **A Simple and Accurate Color Face Detection Algorithm in Complex Background**
*Yu-Ting Pai, Shanq-Jang Ruan, Mon-Chau Shie, Yi-Chi Liu*
Published in 2006 at ICME, IEEE

[28] **Effectiveness of Eigenspaces for Facial Expressions Recognition**
*G. R. S. Murthy, R.S.Jadon*
Published in 2009 at International Journal of Computer Theory and Engineering, Vol. 1, No. 5

[29] **Face Recognition: A Comparison of Appearance-Based Approaches**
*Thomas Heseltine, Nick Pears, Jim Austin, Zezhi Chen*
Published in 2003 at Proc. VIIth Digital Image Computing: Techniques and Applications

[30] **Face Recognition using Principle Component Analysis, Eigenface and Neural Network**
*Mayank Agarwal, Nikunj Jain, Mr. Manish Kumar, Himanshu Agrawal*
Published at Sensors, Signals, Visualization, Imaging, Simulation and Materials

[31] **Optimizing Eigenfaces by Face Masks for Facial Expression Recognition**
*Carmen Frank and Elmar Noth*
Published in 2003 at Springer-Verlag Berlin Heidelberg