Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EMOTION RECOGNITION USING FACIAL EXPRESSION ANALYSIS

[1]ALI GHALI [2]Dr. MHD BASSAM KURDY

[1]Research Scholar, Department of Web Sciences, Syrian Virtual University, Syria

[2]Professor, Department of Artificial Intelligence, Syrian Virtual University, Syria

E-mail: [1]ali_67398@svuonline.org, [2]t_bkurdy@svuonline.org

ABSTRACT

Emotion recognition is a natural capability in human beings. However, if we are to ever create a humanoid "robot" that can interact and emote with its human companions, the difficult task of emotion recognition will have to be solved. The ability for a computer to recognize human emotion has many highly valuable real world applications. Consider the domain of therapy robots which are designed to provide care and comfort for infirm and disabled individuals. These machines could lever information on a patient's current and evolving state of mind, in order to tailor personalized strategies for patient care and interaction. For example, when a patient is upset or unhappy, a more effective strategy may be take a moment to recognize the emotion and offer sympathy.

Even outside of the realm of robotics, working with computers that have the ability to sense and respond to emotional state can go a long way to improve the quality of human-computer interaction (HCI). By designing HCI to be more like human-human interaction, we have the ability to create more natural, fulfilled, and productive working relationships with our machines.

In this research we explain how to recognize emotions through digital images using Android application, and we will identify seven types of emotions (neutral- happy- sad- surprised- afraid- angry- disgusted).

We designed this work based on a popular library called OpenCv, and the Fisherfaces algorithm that consists of (PCA) principle component analysis algorithm and (LDA) the linear discriminate analysis algorithm, in addition, we built the server using Java language to implement the android application, also we compare the coordinates of eyes and mouth in test image with the coordinates in the database to take the highest similarity and show the result.

The language used to build this work is the Java language using NetBeans IDE 8.0.2, and the use of android studio to design android application.

Keywords: *OpenCv Library, Fisherfaces Algorithm, (PCA) Principle Components Analysis Algorithm,(LDA) Linear Discriminate Analysis, Server Architecture, Android Application.*

1. INTRODUCTION

The identification of facial expressions plays a key role in identifying patterns and image processing, and identifying facial expressions through three main stages: face detection, extraction features and classification.

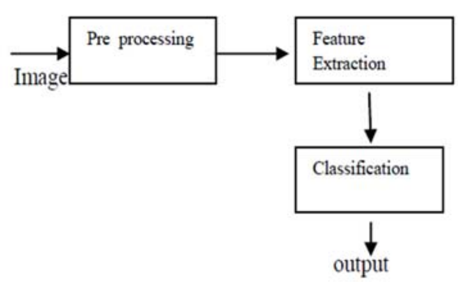

Figure 1: Phases Of Recognition Of Facial Expressions

Fear, surprise, sadness, happiness, anger and disgust are six basic emotions that universally accepted [7]. These emotions can be classified as negative and positive emotions. Fear, anger, disgust

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and sadness are negative emotions and the majority of people do not like them whereas happiness is a positive emotion and everybody wishes to enjoy it. Anger is the most dangerous emotion and at some point in this emotion a person can hurt other purposefully.

In this research, emotion detection system using facial expressions has been implemented. Firstly, video frames are captured using a built in webcam of mobile. Then face extraction and cropping is carried out from these video frames and a training and test database is prepared. Then a low dimensional face space is constructed of training database using Principle Component Analysis (PCA) and emotions are detected using Linear Discriminate Analysis (LDA) between various feature points of test image to the train images.

2. RELATED WORK:

A facial expression analysis method required to deal with three basic problems that are face detection, face extraction and facial expressions classification from still images or image sequences.

Facial expression recognition deals with the problem of classifying facial images into expression classes.

Viola & Jones present an algorithm to find the exact location of face in the image or video frames.

The basic principle behind the Viola-Jones algorithm is to scan a sub-window that is capable enough to detect faces in the input image or video frames.

Then Eigenfaces are used to evaluate these blobs via principal component analysis (PCA).

The classification based on facial expression recognition method uses Fisherfaces system.

Viola-Jones algorithm based on OpenCv library is applied to extract the features of the Karolinska Directed Emotional Faces (KDEF) database and the feature size reduction is done by Principal Component Analysis (PCA).

Another case study for face expression recognition uses a feature selection via linear programming (FSLP) method. This technique performs simultaneous feature selection and classifier training. Another facial expression recognition method uses 2 D DCT and K means algorithm . In this, the images of both men and women are considered and it shows that the mean recognition rate can be as high as 95% for the two databases. Another method uses two-dimensional (2D) discrete cosine transform (DCT) over the entire face image as a feature detector and a constructive one-hidden layer feed forward neural network as a facial expression classifier .

3. RESEARCH OBJECTIVES:

The primary goal of this research is to design, implement and evaluate a novel facial expression recognition system using various statistical learning techniques. This goal will be realized through the following objectives:

- To advance the knowledge on facial expression recognition (FER) systems.
- Developing robust computer vision techniques for the analysis and recognition of facial expressions from static images, image sequences and videos.
- Improving and optimizing both feature extraction and selection methods for facial expression recognition.
- To overcome some of the disadvantages and limitations of existing facial expression recognition methods.

4. RESEARCH VARIABLES:

This research is based on the recognition of emotions using digital images. there are seven basic emotion expressions that are universal for people of different nations and cultures. Those basic emotions are neutral, happy, sad, angry, fear, disgust and surprised. In which any user should be allowed to work with the system, despite of skin color, age, gender or nation. The latter is connected with handling the complex background and variety in lightning conditions.

5. RESEARCH HYPOTHESES:

The goal of Facial Expression Recognition System (FERS) is to imitate the human visual system in the most similar way. This is very challenging task in the area of computer vision because not only it requires efficient image/video analysis techniques but also well-suited feature vector used in machine learning process. The first principle of (FER) system is that it should be effortless and efficient. That relates to full automation, so that no additional manual effort is required. It is also preferred for such system to be real-time which is especially important in both: human-computer interaction and human-robot interaction applications. Furthermore, the subject of study should be allowed to act spontaneously while data is being captured for analysis. System should be designed to avoid limitations on body and head

Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

movements which could also be an important source of information about displayed emotion. The constraints about facial hair, glasses or additional make-up should be reduced to minimum. Moreover, handling the occlusions problem seems to be a challenge for a system and it should also be taken into consideration.

6. RESEARCH METHODOLOGY:

To make our model run in real time, we lever the Viola-Jones Object Detection framework (VJ-ODF) [2] described in 2.3, as implemented in OpenCV [9], for the task of face detection and tracking. The algorithm is robust, and fast enough to run in real time (roughly fifteen frames per second when implemented on a conventional 700 MHz Intel Pentium III [10].
The VJ-OD executes on live video stream and identifies segments of frames that are likely to contain faces. In our research, faces identified using the VJ-ODF are aligned with the help of OpenCV feature detectors, and are subsequently classified based on emotional content.
In phase 1 of our Emotion Classification System (ECS) we use the VJ-ODF as is. Our minor contributions occur in phase 2 with proper alignment and preprocessing of detected faces. Our main contributions for our ECS occur in phase 3 with a hierarchical Fisherface classification model.

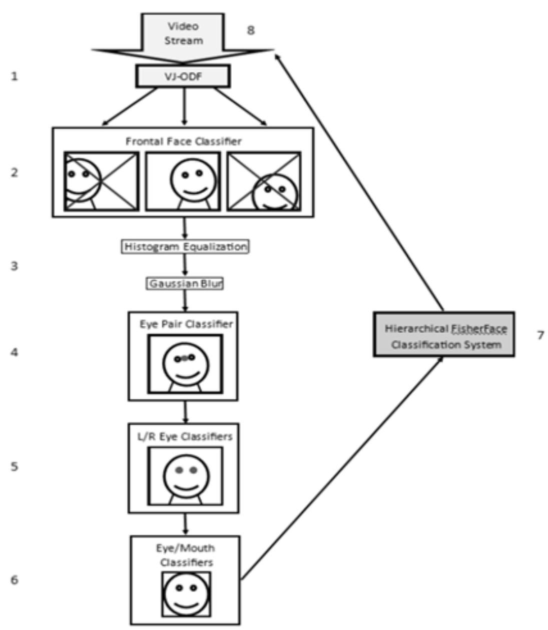

Figure 2: Real Time Emotion Classification System Methodology.

7. RESEARCH DISCUSSION:

This research describes how to build a system to detect emotions through facial digital images using android application, so that the user can take a picture of his face to be processed and analyzed and gives to the user the desired result, and we will try to detect seven types of emotions: (neutral-happy-sad-surprised-afraid-angry and disgusted).
The recognition will perform by detecting face, eyes, and mouth that compared with data set to take the highest similarity and gives results. The results are consisting of expression of face written as a word and an expressive photo.
We will illustrate the following stages in order to obtain the desired results to identify the emotions:

7.1. Database Generation And Training:

Data base consist of 1000 face images that stored in computer. Images obtained from the Karolinska Directed Emotional Faces (KDEF) were used to train various Fisherface models [11]. We train these photos using OpenCv library for detecting faces, eyes and mouth using cascade classifier. In our research, faces identified using the Viola Jones Object Detection are aligned with the help of OpenCV feature detectors, and are subsequently classified based on emotional content. The training stage will take time depending on the number of samples we want to train.

7.2. Face Detection And Tracking:

The box diagram shown in (Figure 3) will show face detection and tracking steps:

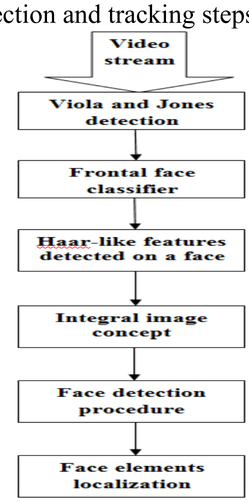

Figure 3:Box diagram for face detection and tracking.

First part of my system is module for face detection and landmark localization in the image. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm for face detection is based on work by Viola and Jones. In this approach image is represented by a set of Haar-like features [8]. Possible types of features are two-, three- and four rectangular features.

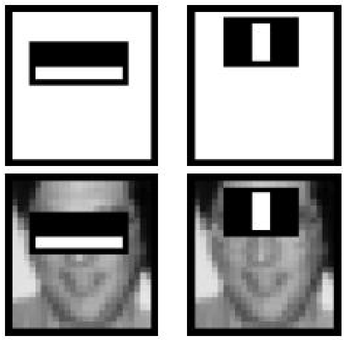

Figure 4: Haar-like features detected on a face.

Feature value is calculated by subtracting sum of the pixels covered by white rectangle from sum of pixels under gray rectangle. Two rectangular features detect contrast between two vertically or horizontally adjacent regions. Three rectangular features detect contrasted region placed between two similar regions and four rectangular features detect similar regions placed diagonally.

The method is widely used in area of face detection. However, it can be trained to detect any object. What is more, this algorithm is quick and efficient and could be used in real time applications. In proposed system, the algorithm is applied for face, eyes and mouth localization with use of already trained classifiers from OpenCV library.

The face detection procedure includes some steps which are consecutively performed on the input image. Procedure flow is illustrated by the output of each function called on input image. Firstly, the classifier trained for face detection searches for a face in the image. In case when face is not found in the image, further processing is omitted and system returns appropriate error message.

If the face is located, the classifiers for eye detections are employed only on the upper part of the face. The left and right eyes are detected separately – in left and right upper face regions (Figure 5). Finally, the mouth region is located with the fourth classifier which searches in the lower part of the face. The search area of facial elements detectors is narrowed in order to improve the time efficiency of the algorithm.

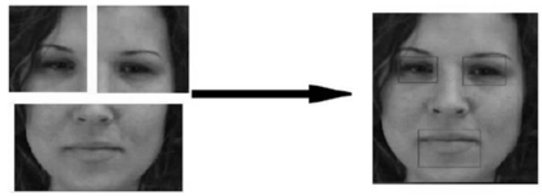

Figure 5: Face elements localization.

Having locations of the face and facial landmarks, the face representation can be formed. If there are more faces detected in the image, the algorithm takes the biggest one for further processing.

The following (Figure 6) shows how face is detected and tracked using our system, and we will see how the face is determined by drawing a yellow square around it, then The face will be cut off automatically under the word (**Captured Image**).:

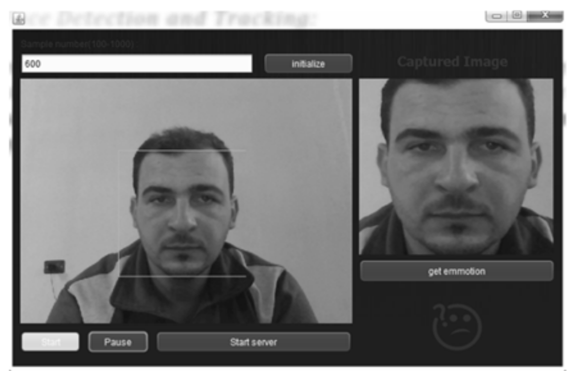

Figure 6: Face detection and tracking in our system.

7.3. Feature Extraction:

The box diagram shown in (Figure 7) will show feature extraction steps after face detection and tracking:

Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

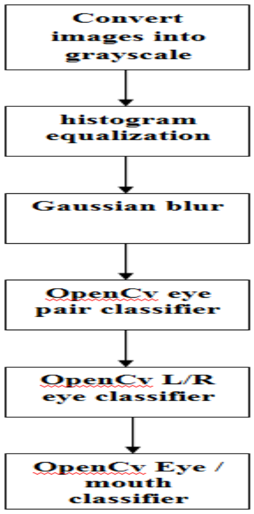

Figure 7: Box diagram for feature extraction.

7.3.1. Convert images into grayscale:

Modern descriptor-based image recognition systems often operate on grayscale images, with little being said of the mechanism used to convert from color-to-grayscale.

This is because most researchers assume that the color-to-grayscale method is of little consequence when using robust descriptors. However, since many methods for converting to grayscale have been employed in computer vision, we believe it is prudent to assess whether this assumption is warranted.

The main reason why grayscale representations are often used for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements. Indeed, color may be of limited benefit in many applications and introducing unnecessary information could increase the amount of training data required to achieve good performance.

7.3.2. Histogram equalization:

Histogram equalization is contrast enhancement technique in a spatial domain in image processing using histogram of image [4]. Histogram equalization usually increases the global contrast of the processing image. This method is useful for the images which are bright or dark.

The main reasons to use histograms equalization in our system are:

1- Simple to calculate
2- Give information about the kind (global appearance) of image and its properties.
3- Used for Image enhancement
4- Used for Image compression
5- Used for Image segmentation
6- Can be used for real time processing

Mathematically, the following equation is used to obtain histogram equalization:

$$Sk = T(rk) = \sum_{j=0}^{k} \frac{nj}{n} * (L-1)$$

K = 0,1,2,…L-1
L: is the number of gray levels in the image,
Nj: is the number of times the gray level is shown in the image
N: is the total number of pixels in the image.
The following figure illustrates the process of applying the histogram equalization equation in our system:

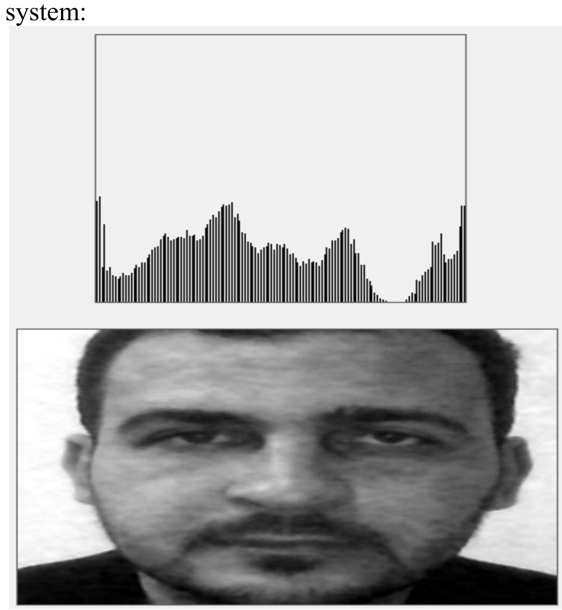

Figure 8: applying the histogram equalization equation in our system

7.3.3. Gaussian blur:

In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function . It is a widely used effect in graphics software, typically to

Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales—see scale space representation and scale space implementation.

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. This is also known as a two-dimensional Weierstrass transform. By contrast, convolving by a circle (i.e., a circular box blur) would more accurately reproduce the bokeh effect. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components, a Gaussian blur is thus a low pass filter.

Mathematically: In the beginning the transformation of images into numbered matrices is applied and then the use of Gaussian function to obtain the Gaussian blur by the following equation:

$$g(c) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2 + y^2}{2\sigma^2})$$

The following figure illustrates the process of applying Gaussian blur in our system:

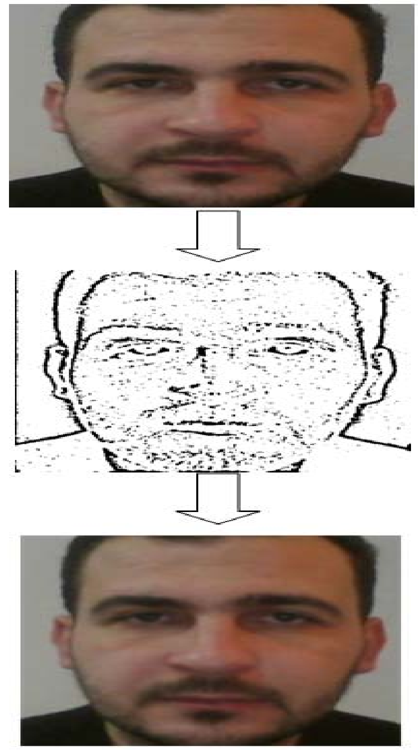

Figure 9:applying Gaussian blur in our system

7.3.4. OpenCv eye pair classifier:

Use OpenCV eye pair classifier to locate the eyes. Calculate the midpoint between the eyes and use this data to vertically and horizontally align the face within the image.

Figure 10: Eye pair classifier

OpenCV (Open Source Computer Vision) [9] is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.

Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7.3.5. OpenCv L/R eye classifier:

Use OpenCV left and right eye detectors on bounded regions of interest. Use this location data to correct any facial rotation.

Figure 11: L/R eye classifier.

7.3.6. OpenCv eye/mouth classifier:

Use OpenCV eye pair and mouth classifiers. Use this location data to obtain a tightly cropped facial image that will align with the template images used to build the Fisherface models.

Figure 12: Eye / mouth classifier.

7.4. Emotion Classification:

The last stage of my system is devoted to facial expressions recognition. This task requires classifier training with a set of images with particular emotions displayed. The box diagram shown in (Figure 11) will show emotion classification steps after generating database, face detection and tracking, and feature extraction, in addition to, all trained face images and test image will go through those steps:

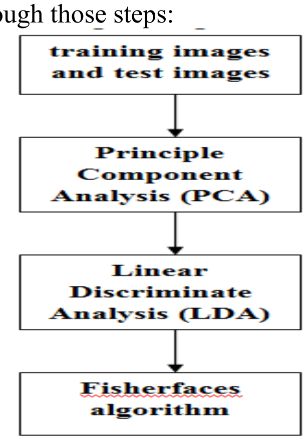

Figure 13: Box diagram for emotion classification.

7.4.1. Principle component analysis (PCA):

The main purposes of a principal component analysis are the analysis of data to identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information [5].

Here, our desired outcome of the principal component analysis is to project a feature space (our dataset consisting of nn dd-dimensional samples) onto a smaller subspace that represents our data "well". A possible application would be a pattern classification task, where we want to reduce the computational costs and the error of parameter estimation by reducing the number of dimensions of our feature space by extracting a subspace that describes our data "best".

For the following example, we will generate 40 3-dimensional samples randomly drawn from a multivariate Gaussian distribution. Here, we will assume that the samples stem from two different classes, where one half (i.e., 20) samples of our data set are labeled ω1 (class 1) and the other half ω2 (class 2).

$$\mu 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mu 2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

(sample means)

$$£1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad £2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(covariance matrices)

7.4.2. Linear discriminate analysis (LDA):

Linear Discriminate Analysis (LDA) is used to solve dimensionality reduction for data with higher attributes and pre-processing step for pattern-classification and machine learning applications. It used for feature extraction and linear transformation that maximize the separation between multiple classes [5].

Listed below are the 5 general steps for performing a linear discriminate analysis, we will explore them in more detail in the following sections.

1- Compute the dimensional mean vectors for the different classes from the dataset.

2- Compute the scatter matrices (in-between-class and within-class scatter matrix).
3- Compute the eigenvectors and corresponding eigenvalues for the scatter matrices.
4- Sort the eigenvectors by decreasing eigenvalues and choose eigenvectors with the largest eigenvalues to form a dimensional matrix (where every column represents an eigenvector).
5- Use this eigenvector matrix to transform the samples onto the new subspace.

7.4.3. LDA vs. PCA:

Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are linear transformation techniques that are commonly used for dimensionality reduction. PCA can be described as an "unsupervised" algorithm, since it "ignores" class labels and its goal is to find the directions (the so-called principal components) that maximize the variance in a dataset. In contrast to PCA, LDA is "supervised" and computes the directions ("linear discriminate") that will represent the axes that maximize the separation between multiple classes. Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known, this might not be always the case.

For example [3], comparisons between classification accuracies for image recognition after using PCA or LDA show that PCA tends to outperform LDA if the number of samples per class is relatively small. In practice, it is also not uncommon to use both LDA and PCA in combination: E.g., PCA for dimensionality reduction followed by an LDA.

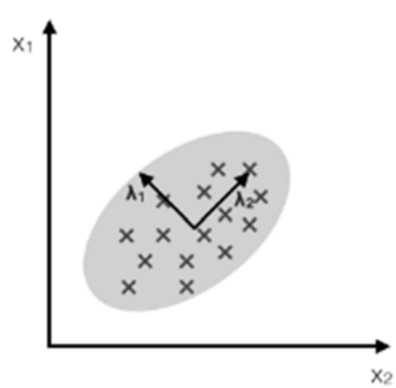

Figure 14: PCA: component axes that maximize the variance.

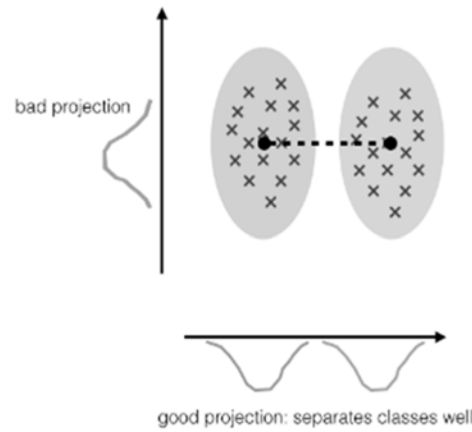

Figure 15: LDA: maximizing the component axes for class-separation.

7.4.4. Fisherfaces algorithm:

Fisherfaces were first described by Belhumeur, Hespanha, and Kriegman [11] in their paper "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection." The authors note that with respect to a set of images, variation within classes lies in a linear subspace of the image space indicating that the classes are convex, and thus linearly separable. In the Fisherfaces method, the task of classification is simplified using of Fisher's Linear Discriminate (FLD) which attains a greater between-class scatter than PCA. In order to obtain tightly clustered well separated classes, LDA maximizes the ratio of the determinant of between-class to within-class scatter.

The Fisherfaces technique takes a pattern classification approach considering each pixel in an image as a coordinate in the high-dimensional image space. The algorithm begins by creating a matrix wherein each column vector (consisting of pixel intensities) represents an image. A corresponding class vector containing class labels is also created. The image matrix is projected into (n-c)-dimensional subspace (where n is the number of images and c is the number of classes). The between-class and within-class scatter of the projection is calculated and LDA is applied. For our purposes here, we levered functionality available within the libraries of OpenCV (2) to implement LDA using the Fisherface methodology.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This method requires that all images, both in the training and testing set, be equal in size. Method performance is the highest when all images are full frontal head shots with major features aligned. This technique does not work on an image directly rather it converts images into grayscale vector matrices and works with the vector form. Ultimately, each image is represented by a weight vector which indicates the percentage of each Fisherface it contains. It is this weight vector representing unique image attributes that is used in a nearest neighbor search of the training set to predict the identity of an unknown face. Written a bit more formally, we have an image I which is represented by a weight vector w = {w1, w2, … , wk} where the image is defined by:

I = mean face + w1x1 + w2x2 + … + wkxk (1)

such that x1..xk are Fisherface templates (k < number of training images). Refer to (Figure 16) for a graphical representation of (1).

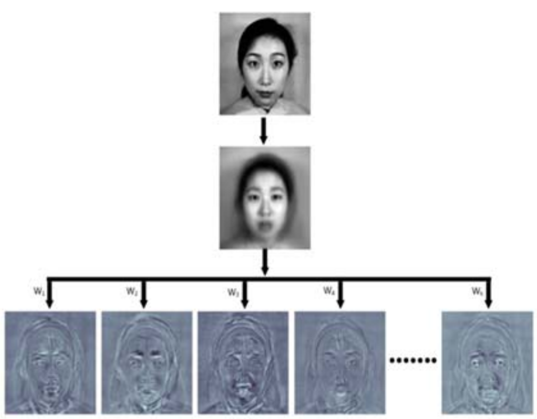

Figure 16: High-level visualization of how an image is represented in the Fisherface model. Each image is comprised of the average face and a weighted sum of the Fisherface templates. Images courtesy of the JAFFE database.

In one experiment reported on in the Belhumeur, Hespanha, and Kriegman paper, the researchers applied the Fisherfaces method to a set images depicting faces with or without glasses. Their results indicated that "PCA had recognition rates near chance, since, in most cases, it classified both images with and without glasses to the same class. On the other hand, the Fisherface methods can be viewed as deriving a template which is suited for finding glasses and ignoring other characteristics of the face." (4) They also commented that "it is expected that the same techniques could be applied to identifying facial expressions where the set of training images is divided into classes based on the facial expression." (4) The two main approaches to content interpretation, or classification, are template-based and rule-based. In a template-based approach, an unknown facial image is compared to a set of templates and a "best match" predicts the displayed emotion. One of the main drawbacks to a template-based methodology lies in the variability of facial expressions and intensities both within and between different individuals. Another challenge for a template-based methodology is quantification of a given expression. A rule-based system is comprised of a set of premises that hold for each class. In this approach, an unknown facial image, described in terms of facial actions, is compared to the prototypical facial actions for a given emotion and a "best match" predicts the displayed emotion.

8. IMPLEMENTATION:

We designed our system using Java based on the NetBeans program, through which many of the classes can be defined. So, we defined the following classes:

1- Export.java class
2- App.java class
3- Predictor.java class
4- FaceDetection.java class

In addition, we designed an Android application for the user to get the desired results. First, the user runs the server to obtain the IP address from the black screen CMD by typing the following instruction: **IPconfig** and then write the address in the place assigned to the interface of the Android application for the communication between the server and the application. Second, the user will take a face image and then send it to the server, which in turn analyzes, processes and resends the result to the user. In the end we will get a written statement expressing the resulting emotion, for example happy word in addition to the emergence of a picture that reflects this situation.

9. THE RESULTS:

Our results suggest that the Fisherface model can be successful in recognizing human emotions in facial images. In the beginning we will review the application interface and explain its parts:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 17: Application interface.

The application interface consists of the IP address that we can get from the black **CMD** screen as shown in the (Figure 16). At the beginning we must press the **start server** button on the main application interface and then open the black screen and write the instruction **ipconfig,** to get the desired IP address.

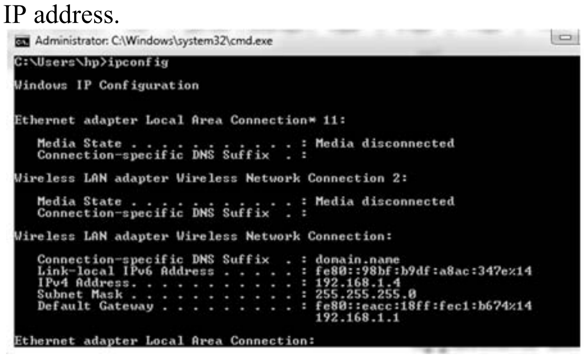

Figure 18: Black screen (cmd.exe) to get the IP address.

In addition to, the application interface contains a display screen to activate the camera mobile for taking pictures after pressing the button **TAKE PHOTO,** then the program will connect to the server so that it receives the taken image to be processed and sent back the result to the program as shown in the following figures.

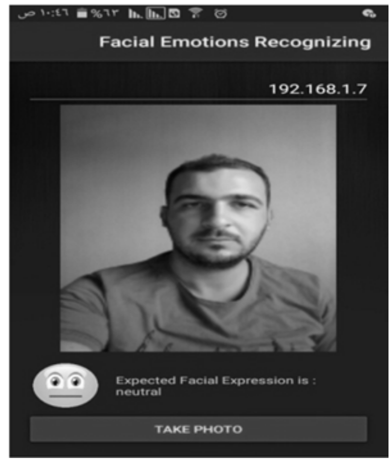

Figure 19: Neutral emotion using Android application.

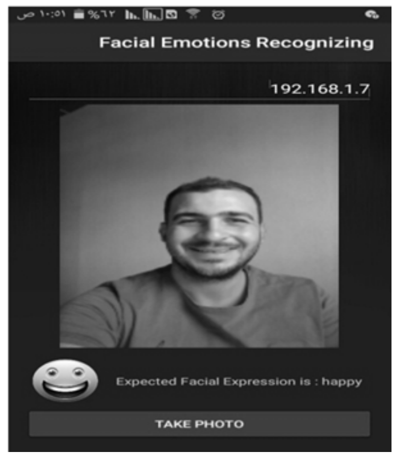

Figure 20: Happy emotion using Android application.

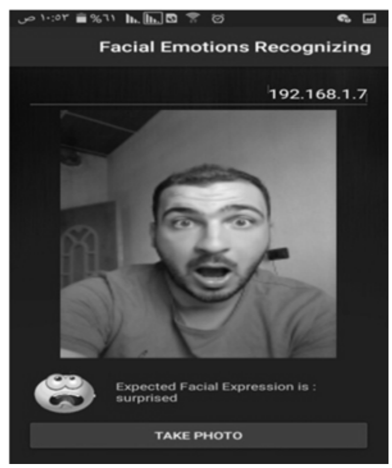

Figure 21: Surprised emotion using Android application.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

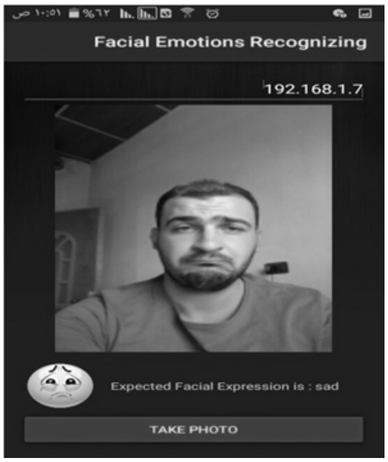

Figure 22: Sad emotion using Android application.

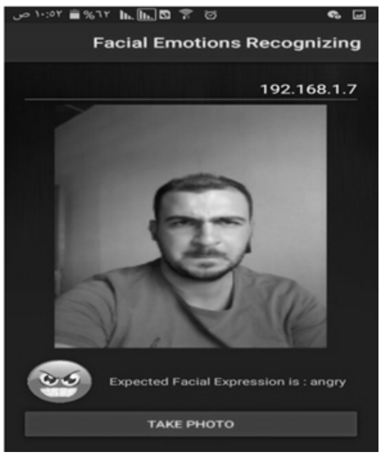

Figure 23: Angry emotion using Android application.

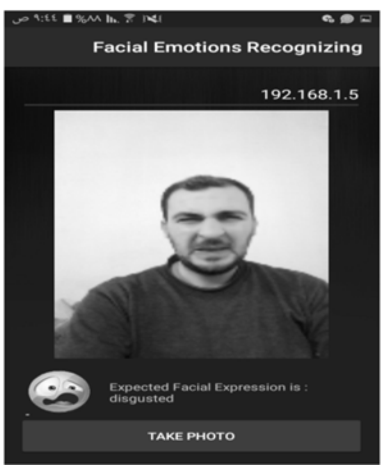

Figure 24: Disgusted emotion using Android application.

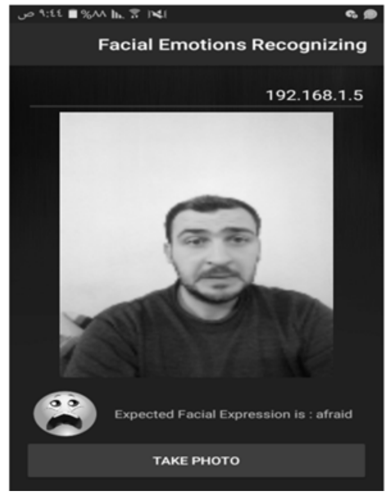

Figure 25: Afraid emotion using Android application.

10. EVALUATION AND COMPARISON:

We will evaluate and compare the results of our system with the results of several previously implemented systems to identify emotions. Table (1) presents the results of emotion recognition for seven emotions and for several algorithms such as (PCA) Principal Component Analysis algorithm, Gabor algorithm, (HLAC) higher order local autocorrelation algorithm, (LBP) Local Binary Pattern algorithm and (GFSSIM) Gabor filters structural similarity algorithm, in addition to our system

The emotion	**PCA%**	**Gabor%**	**HLAC%**	**LBP%**	**GFSSIM%**	**Our system%**
Natural	61.4	72.3	78.4	73.5	79.3	90.4
Happy	87.6	96.3	96.2	94.9	96.9	100
Sad	59.4	85.2	82.5	81.3	81.2	87.2
Surprised	90.5	100	98.9	98.1	98.2	89.8
Angry	66.1	83.3	86.7	82.2	88.1	85.3
Disgusted	61.4	89.8	90.6	87.6	91.2	96.4
Afraid	59	85.8	85.2	86.5	87.6	90.5

Table 1: The recognition rate of different state of the art methods.

Note that our system gave better results than the rest of the systems to identify the seven emotions mentioned above, so in our system we used several algorithms to make the identification process faster and more accurate. In addition, In our system, we used the Fisherfaces algorithm, which includes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PCA algorithm and LDA algorithm, which have been implemented through the OpenCv Library.

11. RECOMMENDATION AND FUTURE WORK:

Despite the promising results, the presented approaches in this thesis are limited to acted facial expressions. In practice, spontaneous and subtle facial expressions can more reveal the real emotional state of human beings. The proposed methods may suffer from the subtle changes and irregular motion variation of facial expression in spontaneous behavior. In many applications of human-computer interaction (HCI), it is important to be able to detect the emotional states of the person in a natural situation. Measuring the intensity of spontaneous facial expressions is, of course, more difficult than measuring acted facial expressions due to the complexity, subtlety and variability of natural expressions. Acted facial expressions may differ in appearance and timing from spontaneously occurring expressions. Hence, there is still room for improvement and extension to spontaneous facial expressions in order to make a dynamic facial descriptor sufficiently generalized, stable, efficient and accurate.

Though the proposed approaches seem practical and robust to these effects, most of the experiments are based on artificial databases. As far as we know, the light, facial occlusion and view changes are still difficult problems, not only for facial expression recognition, but also for face recognition and micro-expression analysis. Also, how to automatically recognize facial expression when three cases occur at the same time remains a challenging issue. Up to now, there are no complete databases, including all three conditions of illumination, occlusion and pose, especially in videos. Thus, it would be interesting to study whether a combination of the proposed methods would be working at more difficult environments.

The main problem in our system is the time in the training phase so that when the database grows bigger it takes more time in training, but in contrast when the database is large the results become better and more accurate so we can put a hidden camera in a public place and take pictures of people spontaneously to get a database that contains all the required conditions, but the problem here is that we cannot get a lot of facial expressions. We may not get a lot of the following expressions: angry or afraid or disgusted, unless we ask and take the consent of people to do these expressions.

12. CONCLUSION:

The main objective of this research is to design, implement and evaluate the system of knowledge of emotions through the analysis of facial expressions using different techniques.

My work has made facial tracking techniques work better under varying illumination and pose.

The main contributions of this research are as follows. Firstly, using Viola and Jones algorithm that based on the famous OpenCV library for facial tracking and detection in difficult conditions. Secondly, applying histogram equalization and Gaussian blur for per-processing facial images. Thirdly, extract features from facial images such as eyes and mouth. Finally, classify emotions using Fisherface system that based on PCA and LDA algorithms.

In this research, we identified seven types of emotions (neutral- happy- sad- surprised- afraid- angry- disgusted). Additionally, emotional images were obtained from a Karolinska database that trained using OpenCv library.

In the end, We conducted a thorough investigation into the efficacy of the Fisherface model in predicting emotional expressions as depicted in facial images. We built and tested a variety of different Fisherface models. We considered the impact of pixel dimensions and problem complexity (i.e. number of classes), and evaluated stand-alone models. Our results suggest that the Fisherface model can be successful in recognizing human emotions in facial images.

REFERENCES:

[1] Cohn-Kanade AU-Coded Expression Database. Available online: http://www.pitt.edu/~emotion/ck-spread.htm
[2] Viola and M.J.Jones .(2001) .*Rapid Object Detection using a Boosted Cascade of Simple Features* .Computer Vision and Pattern Recognition.
[3] M.J. Black and Y. Yacoob, “Tracking and Recognizing Rigid and Non-Rigid Facial Motions Using Local Parametric Models of Image Motions,” *Proc. Int’l Conf. Computer Vision,* pp. 374-381, 1985.
[4] Histogram Equalization. Available online: www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf
[5] C. Darwin, “The Expression of the Emotions in Man and Animals,” *J. Murray,* London, 1872.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] I. Cohen, N. Sebe, A. Garg, L.S. Chen, and T.S Huang, "Facial Expression Recognition From Video Sequences Temporal and Static Modeling," *Computer Vision and Image Understanding,* vol. 91, pp. 160-187, 2003.

[7] L. Sirovich, and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. Opt. Soc. Am. A.,* vol. 4, no. 3, 1987.

[8] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence.* vol. 19, no. 7, pp. 711-720, 1997.

[9] OpenCv. http://opencv.org. *Open Source Computer Vision Library.* [Online] 2013.

[10] P. Viola, and M.J. Jones. *Robust Real - Time Object Detection.* s.l. : Int. J. Computer Vision, 2004. no.2, pp. 137-154.

[11] P. Belhumeur, J. Hespanha, and D. Kriegman. *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection.* s.l. : IEEE Trans, 1997. pp. 711-720.