Facial Expression Recognition from Video Using Eigen Values

Manish Trivedi, **Ruchi Chaurasia, Manoj Tolani**

Abstract: ***Facial expressions recognition is now a main area of interest within various fields such as computer science, medicine, and psychology. Facial expressions commit authorized information about emotions of any person. Understanding facial expressions accurately is one of the difficult tasks for interpersonal relationships. Now a days automatic detection of emotions is very popular. We have to build such a efficient approach for human emotion recognition. To improve the human-computer interaction (HCI) to be as good as human-human interaction, building an efficient approach for human emotion recognition is required. These emotions could be blended from several modalities such as facial expression, hand gesture, acoustic data, and biophysiological data. This paper proposes an approach to solve this limitation using 'salient' distance features, which are obtained by extracting patch-based Eigen value, selecting the 'salient' patches, and performing patch matching operations. The experimental results demonstrate high correct recognition rate (CRR), tracking of face and detect the expressions.***

Keywords: Facial expression analysis, feature evaluation and selection, computer vision, neural network, Eigen values. Eigen faces, Eigen faces, Eigen Vectors, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA).

1. Introduction

Face plays a vital role in human communication because human face is a natural & complex physical object that tends not to have automatically identified features and edges. That"s it is difficult to develop a mathematical model of the faces that can be used as prior information when analyzing a particular image. Facial expressions and gestures incorporate nonverbal information which contributes to human communication. By identification and recognizing the facial expressions from facial images, a number of applications in the field of human computer interaction vision and interaction can be alleviated. Last two decades, the developments, as well as the prospects in the field of multimedia signal processing have attracted the attention of many computer vision researchers to concentrate in the problems of the facial expression recognition. According to Ekman studies, the basic facial expressions are those representing disgust, anger, happiness, sadness, fear, surprise and neutral. Facial Action Coding System (FACS) was developed by Ekman and Friesen to code facial expressions in which the movements on the face are described by action units. The work on the video and images are analyzed by many researchers, where they categorized different facial expression by adopting of tracking of facial features and measuring the amount of facial movements. Now a day"s work on analysis and recognition of facial expression by used these five basic expressions as their basis for the introduced systems. Most of the methods developed by using 2D distribution of facial features as inputs into a classification system, and the outcome are one of the facial expression classes. On the basis of selected facial features these are different. Information extracted from 3D face models are rarely used in the analysis of the facial expression recognition. This paper considers the techniques using the information extracted from 3D space for the analysis of facial images and video for the recognition of facial express ions. Our objective is to implement the model for a particular face and distinguish it from a large number of stored faces with some real-time variations as well.

The method is based on an information theory approach that decomposes face images into a small piece of characteristic feature images called „eigenfaces", which are actually the principal components of the initial training set of face images. Facial expression recognition method is performed by projecting a new image into the subspace spanned by the eigenfaces („face space") and after that we compare its position in the face space with the position of the known individuals for classifying the faces .

.

Mathematical models of faces have been an active area of research since late 1980s, for they are contributing in theoretical areas as well as practical applications, such as criminal identification, security systems, image and film processing, and human-computer interaction, etc.

Generally, there are three stages for face recognition such as-
i) face representation
ii) face detection
iii) face identification

<u>Face representation</u> is the first job, that is, how to model a face. The method to represent a face determines the successive algorithms of identification and detection. For the entry-level recognition, a face category should be characterized by generic properties of all faces; and for the subordinate-level recognition, detailed features of eyes, lips, chin eyebrow, mouth , jaw, nose, and forehead have to be assigned to each individual face. For face representation we have a variety of approaches, which can be classified as: template-based, feature-based, and appearance-based.

The simplest approach is *template-matching* approach which represents an entire face using a single template, i.e., a 2-D array of intensity, which is usually an edge map of the original face image. In a broad way of template-matching, multiple templates may be used for each face to account for recognition from different viewpoints. Although it requires a large memory but it is simple. Inefficient matching is also drawback of this method.

Features based approach-It has smaller memory requirement and a higher recognition speed than template based. In this technique a face is represented by extracting geometric features, such as position and width of eyes, nose and mouth, eyebrow‟s thickness and arches face breath or invariant moments. This method is particularly useful for 3D head model-based pose estimation and face scale normalization. However, perfect extraction of features is shown to be difficult in implementation [5].

The *appearance-based* approaches-in this approach the face images are to be projected onto a linear subspace of low dimensions. This subspace is firstly constructed by principal component analysis on a set of training images, with eigenfaces as its eigenvectors. Later, the concept of eigen feature has come from the eigenfaces such as eigen mouth, eigeneyes etc fro detection of facial features [6]. More recently, fisherface space [7] and illumination subspace [8] have been proposed for dealing with recognition under varying illumination.

Face detection is to spot a face in a given images and to separate it from the remaining scene. Many approaches have been planned to accomplish the task. One of them is to utilize the elliptical structure of human head [9]. This method locates the head outline by the Canny's edge finder and then fits an ellipse to mark the boundary between the head region and the background. However, this method is applicable only to frontal views, the detection of non-frontal views needs to be investigated.

A second approach is useful manipulates the images for face detection in “face space” [1]. When projection into the face has been come then Images of faces do not change entirely, while projections of nonface images appear quite different. This basic idea is used to detect the presence of faces in a arena: at every location in the image, calculate the distance between the local subimage and face space. This measured distance from face space can be used as a measure of “faceness”, so the result of calculating the distance from face space at every point in the image is a “face map”. Low values (white color) or we can say the short distances from face space, in the face map indicate the presence of a face.

Face identification is performed at the secondary-level. At this level, we compared a new face to face models stored in a database and then classified to a known individual if a correspondence is found. The affected parameters of performance of face identification are: disguise, pose, illumination scale and facial expression.

We handled scale of faces by rescaling process. In the appraoch of eigenface the scaling factor can be determined by multiple trials. The idea is to use multiscale eigenfaces, in which, at every number of scale a test face image is compared with eigenfaces. In this case, the image will appear to be near face space of only the closest scaled eigenfaces. Equivalently, we can scale the test image to multiple sizes and use the scaling factor that results in the smallest distance to face space.

We can change pose result Varying *poses* result by changing of view points and head orientation. Different identification algorithms provide different sensitivities to pose variation.

The challenging task for face recognition is to identify faces in different *illuminance* conditions. The lighting condition changes dramatically of same person, with the same facial expression, and seen from the same viewpoint.

Different from the effect of pose, scale and illumination, *facial expression* can greatly change the geometry of a face. Even attempts have been made in computer graphics to model the facial expressions from a muscular point of view [10]. *Disguise* is another problem encountered by face recognition in practice. Glasses, hairstyle, and makeup all change the appearance of a face. Most research work so far has only addressed the problem of glasses [7][1].

In this paper, a method has been introduced to design an Eigenvector based facial expression recognition system for images as well as video. Eigenvector based features are extracted from the images. In the training phase, Eigenvectors specific to the expressions are stored and a set of images for each basic expression is processed. In the testing phase, the Eigenvector of the testing image is computed and the Euclidean distance of the Eigenvector of the testing image and all the stored Eigenvector is computed. The testing image is classified as a particular facial expression if the Euclidean distance between the Eigenvectors of that expression and the Eigenvectors of the testing image is obtained minimum compared to the Eigenvectors of the other expressions. To make the system more efficient instead of the whole image being considered, segments of the image is processed.

We propose a method for recognizing the emotions through facial expressions displayed in a video sequence. Our method is similar with the one proposed by Sebe, et al. [11] in the sense that we use the same features and we classify each frame of the video to a facial expression based on these features computed for that time frame. The novelty of this work is in going beyond the NB assumption. While in the NB assumption the features are considered to be independent, here we show that is beneficial to use the inherent dependencies that are present between the facial features.

The first part of the work introduces the methods of extracting information from 3D models for facial expression recognition. The 3D distributions of the facial feature points provide the estimation of characteristic distances in order to represent the facial expressions are explained by using a rich collection of illustrations with the help of face images. The second part of the chapter introduces 3D distance-vector based facial expression recognition. Facial movement features, which include shape changes and feature position, are generally caused by the movements of facial elements and muscles during the course of emotional expression. Facial elements may change their expressing emotion as we change the position. As a result, the same feature in different images usually has different positions, as shown in Fig. 1a. In some cases, the shape of the feature may also be destroyed due to the subtle facial muscle movements. For example, the mouth in the third images in Fig. 1 has different shapes from the other images. Therefore, for any feature showing a certain emotion, the geometry-based position and appearance based shape normally change from one image to another image in image databases, as well as in videos. This type of movement features represents a rich field of both static and dynamic characteristics of expressions, which play a critical role for FER.

The productivity of facial movement features into account [1].Already some efforts have been made in capturing and utilizing Facial movement features. These efforts may be try to adopt one of the technique-i) geometric features of the tracked facial points (e.g., shape vectors [2], facial animation parameters [3], distance and angular [4], and trajectories [5]), ii) appearance difference between holistic facial regions in consequent frames (e.g., optical flow [6], and differential-AAM [7]). iii) Motion and texture changes in local facial regions (e.g., surface deformation [8], motion units [9], spatiotemporal descriptors [10], animation units [11], and pixel differences [12]). Although they achieved promising results, these approaches often require accurate location and tracking of facial points, which remains problematic [11]. We have another alternative way is image-based FER techniques which is used to recognize emotions based on appearance-based features in a single image, and are important for the situation where only several images are available for training and testing. However, to the best of our knowledge, no research has been reported on image-based FER that considers facial movement features. In this paper, we aim for improving the performance of FER by automatically capturing facial movement features in static images based on distance features. They also show a great advantage over the commonly used fiducially point-based.

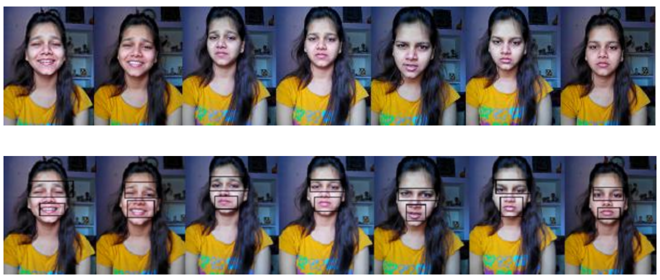


Figure 1: Facial movement features (with different kind of expressions)

2. Proposed Framework

Fig.2 represents the proposed framework, which is consists of face detection cropping, preprocessing, training, and test stages. Firstly we take images and then perform cropping. At the preprocessing stage, we have any area of face for example nose, nose as the center and keeping main facial components inclusive. Further no processing is imparted to imitate the results of real face detectors. During the training stage, a whole set of patches is extracted by moving a series of patches with different sizes across the training images. After this the patch matching method has been started to extracted patches to distance features. To capture facial movement features, the matching area and matching scale are defined to increase the matching space, whereas the minimum rule is used to find the best matching feature in this space. Based on the converted distance features, a set of "salient" patches is selected by Adaboost. At the test stage, the same patch matching operation is performed on a new image using the "salient" patches. The resulting distance features are fed into a multiclass support vector machine (SVM) to recognize six basic emotions, including anger (AN), disgust (DI), neutral (NE), happiness (HA),and sadness (SA).

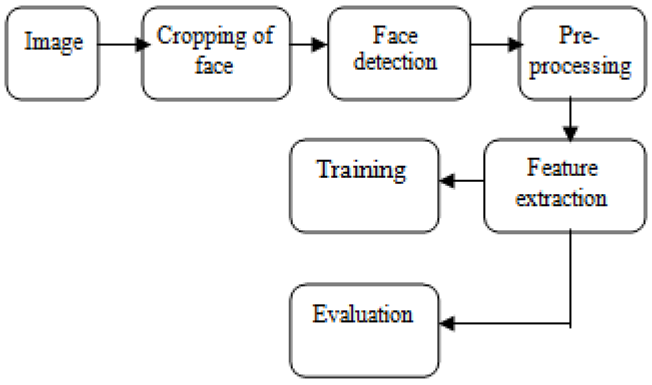


Figure 2: A technique for facial expressions recognition

2.1 Eigenfaces for Recognition

In the area of information theory, the aim is to extract the relevant information in a face image, encode this information as efficiently as possible, and compare one face encoding with a database of models encoded in the same way. We adopt simple approach to extract the information contained in a face image is to somehow capture the variation in a collection of face images, independent of any judgement of features, and use this information to compare individual face images and encode. As already we discussed we find out the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. These eigenvectors may be as a set of features which 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 called an *eigenface*. Some of these faces are shown in Figure 1.

In the training set each faces can be represented exactly in terms of a linear combination of the eigenfaces. The number of face images in the training set is equal to the number of possible eigenfaces. Those faces have the largest eigenvalues can be approximated as a "best" eigenfaces The primary reason for using fewer eigenfaces is computational efficiency. The most meaningful M eigenfaces span an M-dimensional subspace—"face space"—of all possible images. The eigenfaces are essentially the basis vectors of the eigenface decomposition.

The concept of using eigenfaces had been come by a technique for efficiently representing pictures of faces using principal component analysis. It is discussed that a collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard pictures. Therefore, if a multitude of face images can be reconstructed by weighted sum of a small collection of characteristic images, then an efficient way to learn and recognize faces might be to build the characteristic features from known face images and to recognize particular faces by comparing the feature weights needed to (approximately) reconstruct them with the weights associated with the known individuals.

The following initialization operations involves for face recognition in eigenfaces approach

1) Desried a set of training images.
2) Calculate the eigenfaces from the training set, keeping only the best *M* images with the highest eigenvalues. These *M* images define the "face space". As new faces are experienced, the eigenfaces can be updated.
3) Calculate the corresponding distribution in *M*-dimensional weight space for each known individual (training image), by projecting their face images onto the face space.

Having initialized the system, the following steps are used to recognize new face images:

1) Given an image to be recognized, calculate a set of weights of the *M* eigenfaces by projecting the it onto each of the eigenfaces.
2) Determine if the image is a face at all by checking to see if the image is sufficiently close to the face space.
3) If it is a face, classify the weight pattern as eigher a known person or as unknown.
4) (Optional) Update the eigenfaces and/or weight patterns.
5) (Optional) Calculate the characteristic weight pattern of the new face image, and incorporate into the known faces.

2.3 Calculating Eigenfaces

Let a face image Γ(x,y) be a two-dimensional *N* by *N* array of intensity values. An image may also be considered as a vector of dimension N^2, so that a typical image of size 256 by 256 becomes a vector of dimension 65,536, or equivalently, a point in 65,536-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis is to find the vector that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call "face space". Each vector is of length N^2, describes an *N* by *N* image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, they are referred to as "eigenfaces".

Let the training set of face images be Γ_1, Γ_2, Γ_3, …, Γ_M. The average face of the set if defined by-

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n .$$

Each face differs from the average by the vector $\Phi_n = \Gamma_n - \Psi$. An example training set is shown in Figure 1a, with the average face Ψ shown in Figure 1b. This set of very large vectors is then subject to principal component analysis, which seeks a set of *M* orthonormal vectors, μ_n, which best describes the distribution of the data. The *k*th vector, μ_k is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(\mu_k^T\Phi_n)^2 \tag{1}$$

is a maximum, subject to

$$\mu_l^T\mu_k = \begin{cases}1, l = k\\ 0, otherwise\end{cases} \tag{2}$$

The vectors μ_k and scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \tag{3}$$

where the matrix $A = [\Phi_1\Phi_2...\Phi_M]$. The matrix *C*, however, is N^2 by N^2, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. A computationally feasible method is needed to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space ($M < N^2$), there will be only $M-1$, rather than N^2, meaningful eigenvectors (the remaining eigenvectors will have associated eigenvalues of zero). Fortunately, we can solve for the N^2-dimensional eigenvectors in this case by first solving for the eigenvectors of and *M* by *M* matrix—e.g., solving a 16 x 16 matrix rather than a 16,384 x 16,384 matrix—and then taking appropriate linear combinations of the face images Φ_n. Consider the eigenvectors v_n of A^TA such that

$$A^TAv_n = \lambda_n v_n \tag{4}$$

Premultiplying both sides by *A*, we have

$$AA^TAv_n = \lambda_n Av_n \tag{5}$$

from which we see that Av_n are the eigenvectors of $C = AA^T$.

Following this analysis, we construct the *M* by *M* matrix $L = A^TA$, where $L_{mn} = \Phi_m^T\Phi_n$, and find the *M* eigenvectors v_n of *L*. These vectors determine linear combinations of the *M* training set face images to form the eigenfaces μ_n:

$$\mu_n = \sum_{k=1}^{M}v_{nk}\Phi_k = Av_n, n = 1,......, M \tag{6}$$

With this analysis the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (*M*). In practice, the training set of face images will be relatively small ($M < N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizeing the variation among the images.

2.4 Using Eigenfaces to Classify a Face Image

The eigenface images calculated from the eigenvectors of *L* span a basis set with which to describe face images. As mentioned before, the usefulness of eigenvectors varies according their associated eigenvalues. This suggests we pick up only the most meaningful eigenvectors and ignore the rest, in other words, the number of basis functions is further reduced from *M* to *M'* (*M'*<*M*) and the computation is reduced as a consequence. Experiments have shown that the RMS pixel-by-pixel errors in representing cropped versions of face images are about 2% with *M*=115 and *M'*=40 [11].

In practice, a smaller *M'* is sufficient for identification, since accurate reconstruction of the image is not a requirement. In this framework, identification becomes a pattern recognition task. The eigenfaces span an *M'* dimensional subspace of the original N^2 image space. The *M'* most significant eigenvectors of the *L* matrix are chosen as those with the largest associated eigenvalues.

A new face image Γ is transformed into its eigenface components (projected onto "face space") by a simple operation

$$\omega_n = \mu_n(\Gamma - \Psi) \tag{7}$$

for *n*=1,……,*M'*. This describes a set of point-by-point image multiplications and summations.

The weights form a vector $\Omega^T = [\omega_1, \omega_2, ..., \omega_{M'}]$ that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The vector may then be used in a standard pattern recognition algorithm to find which of a number of predefined face classes, if any, best describes the face. The simplest method for determining which face class provides the best description of an input face image is to find the face class *k* that minimizes the Euclidian distance

$$\varepsilon_k^2 = \left\|(\Omega - \Omega_k)^2\right\| \tag{8}$$

where Ω_k is a vector describing the *k*th face class. The face classes Ω_k are calculated by averaging the results of the eigenface representation over a small number of face images (as few as one) of each individual. A face is classified as "unknown", and optionally used to created a new face class.

Because creating the vector of weights is equivalent to projecting the original face image onto to low-dimensional face space, many images (most of them looking nothing like a face) will project onto a given pattern vector. This is not a problem for the system, however, since the distance ε between the image and the face space is simply the squared distance between the mean-adjusted input image $\Phi = \Gamma - \Psi$ and $\Phi_f = \sum_{i=1}^{M'} \omega_i \mu_i$, its projection onto face space:

$$\varepsilon^2 = \left\|\Phi - \Phi_f\right\|^2 \tag{9}$$

Thus there are four possibilities for an input image and its pattern vector: (1) near face space and near a face class; (2) near face space but not near a known face class; (3) distant from face space and near a face class; (4) distant from face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is present. The last two cases indicate that the image is not a face image. Case three typically shows up as a false positive in most recognition systems; in this framework, however, the false recognition may be detected because of the significant distance between the image and the subspace of expected face images.

3. Summary of Eigenface Recognition Procedure

The eigenfaces approach for face recognition is summarized as follows:

1) Collect a set of characteristic face images of the known individuals. This set should include a number of images for each person, with some variation in expression and in the lighting (say four images of ten people, so *M*=40).
2) Calculate the (40 x 40) matrix *L*, find its eigenvectors and eigenvalues, and choose the *M'* eigenvectors with the highest associated eigenvalues (let *M'*=10 in this example).
3) Combine the normalized training set of images according to Eq. (6) to produce the (*M'*=10) eigenfaces $\mu_k, k = 1, \ldots\ldots, M'$.
4) For each known individual, calculate the class vector Ω_k by averaging the eigenface pattern vectors Ω [from Eq. (8)] calculated from the original (four) images of the individual. Choose a threshold θ_ε that defines the maximum allowable distance from any face class, and a threshold θ that defines the maximum allowable distance from face space [according to Eq. (9)].
5) For each new face image to be identified, calculate its pattern vector Ω, the distance ε_k to each known class, and the distance ε to face space. If the minimum distance $\varepsilon_k < \theta_\varepsilon$ and the distance $\varepsilon < \theta$, classify the input face as the individual associated with class vector Ω_k. If the minimum distance $\varepsilon_k > \theta_\varepsilon$ but $\varepsilon < \theta$, then the image may be classified as "unknown", and optionally used to begin a new face class.
6) If the new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated (steps 1-4). This gives the opportunity to modify the face space as the system encounters more instances of known faces.

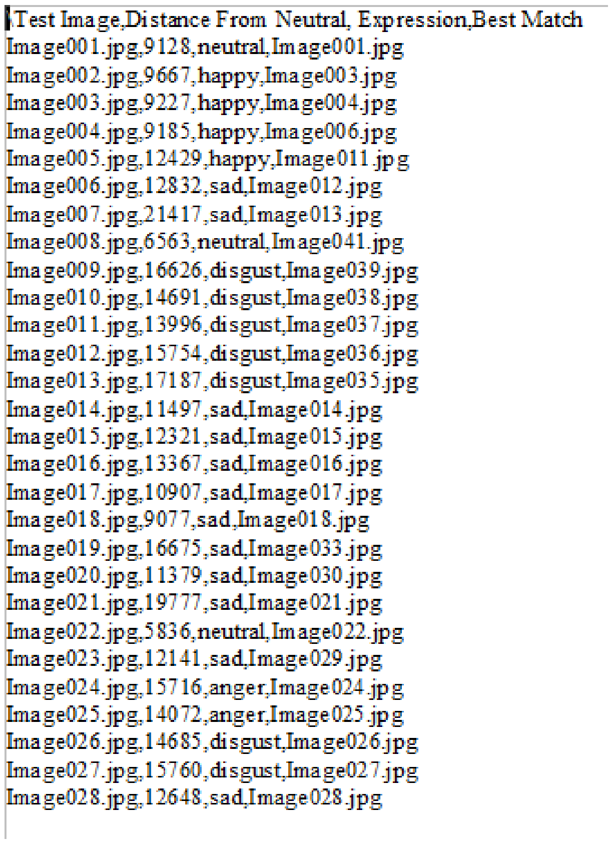
Implementation Issues

The entire program consists of four functional blocks, namely „Load Images", „Construct Eigen faces", „Classify New face", and „undo Update Eigen faces". There is also a „main" function, which calls „Construct Eigen faces" and „Classify New face" functions to complete the face recognition task.

System Structure
The structure of the system is shown in Figure 1. In the figure, the square shape indicates functions, and the parallogram represents files. An arrow pointing out from a file to a function means the function loads the file; an arrow pointing in the other direction indicates that the function creates or updates the file; a bidirectional arrow means the file is first read by the function, and later modified or updated by it. These files help the „Construct Eigen faces" and „Classify New face" functions communicate with each other in a well organized way.

4. Conclusion and Future Work

This paper explores the issue of facial expression recognition for video as well as image. The effectiveness of the proposed approach is testified by the recognition performance. The experimental results also demonstrate significant performance improvements due to the consideration of facial movement features and promising performance under face registration errors. The results indicate that eigen face based features show a better performance over point-based in terms of extracting regional features, keeping the position information, achieving a better recognition performance, and requiring a less number. Different emotions have different "salient" areas; however, the majority of these areas are distributed around mouth and eyes. In addition, these "salient" areas for each emotion seem to not be influenced by the choice of using point-based or using patch-basedfeatures. The "salient" patches are distributed across all scales with an emphasis on the higher scales. For both the JAFFE and CK databases, DL2 performs the best among four distances. As for emotion, anger contributes most to the misrecognition. The JAFFE database requires larger sizes of patches than the CK database to keep useful information. The proposed approach can be potentially applied into many applications, such as patient state detection, driver fatigue monitoring, and intelligent tutoring system. In our future work, we will extend our approach to a video-based FER system by combining patch-based Gabor features with motion information in multiframes. Recent progress on action recognition [7] and face recognition [8] has laid a foundation for using both appearance and motion feature.



```
Test Image,Distance From Neutral, Expression,Best Match
Image001.jpg,9128,neutral,Image001.jpg
Image002.jpg,9667,happy,Image003.jpg
Image003.jpg,9227,happy,Image004.jpg
Image004.jpg,9185,happy,Image006.jpg
Image005.jpg,12429,happy,Image011.jpg
Image006.jpg,12832,sad,Image012.jpg
Image007.jpg,21417,sad,Image013.jpg
Image008.jpg,6563,neutral,Image041.jpg
Image009.jpg,16626,disgust,Image039.jpg
Image010.jpg,14691,disgust,Image038.jpg
Image011.jpg,13996,disgust,Image037.jpg
Image012.jpg,15754,disgust,Image036.jpg
Image013.jpg,17187,disgust,Image035.jpg
Image014.jpg,11497,sad,Image014.jpg
Image015.jpg,12321,sad,Image015.jpg
Image016.jpg,13367,sad,Image016.jpg
Image017.jpg,10907,sad,Image017.jpg
Image018.jpg,9077,sad,Image018.jpg
Image019.jpg,16675,sad,Image033.jpg
Image020.jpg,11379,sad,Image030.jpg
Image021.jpg,19777,sad,Image021.jpg
Image022.jpg,5836,neutral,Image022.jpg
Image023.jpg,12141,sad,Image029.jpg
Image024.jpg,15716,anger,Image024.jpg
Image025.jpg,14072,anger,Image025.jpg
Image026.jpg,14685,disgust,Image026.jpg
Image027.jpg,15760,disgust,Image027.jpg
Image028.jpg,12648,sad,Image028.jpg
```

References

[1] http://www.asprs.org/a/publications/pers/97journal/august/1997_aug_975-983.pdf
[2] http://lmb.informatik.unifreiburg.de/papers/download/te_dagm06.pdf
[3] http://freeproject.co.in/source/Facial-Expression-Recognition Using-Facial-Movement-Features2011.aspx?pf=.Net&t=ieee
[4] http://www.denniscodd.com/dotnetieee/Facial/Expression/RecognitionUsing/Facial.pdf
[5] http://www.academia.edu/2696024/Facial_expression_recognition_using_3D_facial_feature_distances
[6] http://www.slideshare.net/Projucti/facial-expression-recognition-using-facial-movement-features-28159320
[7] http://www.ijarcsse.com/docs/papers/Volume_4/4_April2014/V4I4-0235.pdf
[8] http://ibug.doc.ic.ac.uk/media/uploads/documents/EncycBiometrics-Pantic-FacExpRec-PROOF.pdf
[9] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/CHIBELUSHI1/CCC_FB_FacExprRecCVonline.pdf
[10] http://research.microsoft.com/enus/um/people/zhang/papers/ijprai.pdf
[11] http://www.eecs.qmul.ac.uk/~sgg/papers/ShanGongMcOwan_IVC09.pdf
[12] http://arxiv.org/ftp/arxiv/papers/1203/1203.6722.pdf