Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Emotion recognition from geometric facial features using self-organizing map

Anima Majumder*, Laxmidhar Behera, Venkatesh K. Subramanian

Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India

ARTICLE INFO

Article history:
Received 4 February 2013
Received in revised form
17 July 2013
Accepted 6 October 2013

Keywords:
Facial expression analysis
Geometric facial features
Self-organizing map
Features extraction
System identification
Radial basis function
Multi-layer perceptron
Support vector machine

ABSTRACT

This paper presents a novel emotion recognition model using the system identification approach. A comprehensive data driven model using an extended Kohonen self-organizing map (KSOM) has been developed whose input is a 26 dimensional facial geometric feature vector comprising eye, lip and eyebrow feature points. The analytical face model using this 26 dimensional geometric feature vector has been effectively used to describe the facial changes due to different expressions. This paper thus includes an automated generation scheme of this geometric facial feature vector. The proposed non-heuristic model has been developed using training data from MMI facial expression database. The emotion recognition accuracy of the proposed scheme has been compared with radial basis function network, multi-layered perceptron model and support vector machine based recognition schemes. The experimental results show that the proposed model is very efficient in recognizing six basic emotions while ensuring significant increase in average classification accuracy over radial basis function and multi-layered perceptron. It also shows that the average recognition rate of the proposed method is comparatively better than multi-class support vector machine.

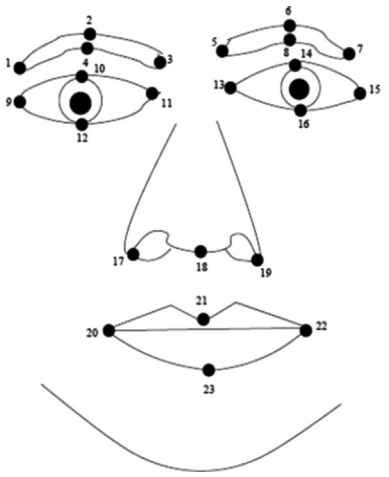
© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

In our day-to-day life, communication plays a very important role. With the growing interest in human–computer interaction, automation of emotion recognition became an increasingly crucial area to work on. Facial expressions are a kind of nonverbal communication. They are considered to be one of the most powerful and immediate means of recognizing one's emotion, intentions and opinion about each other. Mehrabian [16] found that when people are communicating feelings and attitudes, 55% of the message is conveyed through facial expression alone, vocal cues provide 38% and the remaining 7% is via verbal cues. Ekman and Friesen [3] did a rigorous study on facial expression and came to conclusion that facial expressions are universal and innate. They also stated that there are six basic expressions, these include happiness, sadness, disgust, anger, surprise and fear. Much efforts have gone towards the study of facial expression and emotion recognition, initially by cognitive scientists and later by computer vision researchers [27]. The Facial Action Coding System (FACS) [3] is a human observer based system, developed to detect the changes in facial features or facial muscles movements using 44 anatomically based action units. Determining FACS from images is a very laborious work, and thus, during the last few decades a lot of attention is given towards automating it. Automatic analysis of facial features requires extraction of relevant facial features from either static images or video sequences, which can either be further classified into different action units (AUs) or can be applied directly to the classifiers to give the respective emotion. Efficient extraction of facial features from faces of different persons is a crucial step towards accurate facial expression recognition. Generally, two common types of features are used for facial expression recognition: geometric features data and appearance features data. Geometric features give clues about shape and position of the feature, whereas appearance based features contain information about the wrinkles, bulges, furrows, etc. Appearance features contain micro-patterns which provide important information about the facial expressions. But one major drawback with appearance based methods is that it is difficult to generalize appearance features across different persons. Although geometric based features are sensitive to noise and the tracking of those features is rather difficult, geometric features alone can provide sufficient information to have accurate facial expression recognition [28]. We humans have a very extraordinary ability to recognize expressions. Even if we are given a cartoon image having only some contours, we can easily recognize the expression [5]. In many cases, it is observed that features obtained from facial

* Corresponding author. Tel.: +91 512 259 7854; fax: +91 512 259 0063.
E-mail addresses: animam@iitk.ac.in (A. Majumder), lbehera@iitk.ac.in (L. Behera), venkats@iitk.ac.in (V.K. Subramanian).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.10.010



Geometric features	Number of features	Description
Eyebrow	4x2	Two extreme corners, upper and lower mid points
Eyes	4x2	Two extreme corners, upper and lower mid points
Nose	3	Two nostrils and nose tip
Lip	4	Two extreme corners, upper and lower mid points

Fig. 1. Facial points of the frontal image.

contours alone can convey adequate information to recognize various expressions on the face.

The goal of this work is to introduce a completely automatic method of facial expression recognition using geometric facial features alone. The features extracted from the region of the eyes, eyebrows, lips, etc. play a significant role in providing sufficient information to recognize the presence of any of those six basic expressions. To remove planar head motion effects and scaling issues in subsequent image frames, all the feature parameters are calculated as the ratio of current values to those of the reference frame. This includes methodologies for detection of different facial features, such as eyebrow contours, state of eyes, lip contour and key points detection for each of the features. We also introduce methodologies to make the features rotation and illumination invariant. In order to come up with very accurate facial expression recognition results, a good classifier is extremely desirable. We propose a classification method using Kohonen Self-Organizing Map (KSOM) [31,7] to classify the features data into six basic facial expressions. KSOM has an extra-ordinary ability to arrange the data in an order that maintains the topology of the input data. The features data are first clustered using KSOM, then the cluster centers are used to train the data for recognition of the basic different emotions. To evaluate the performance of the proposed classification method, we compare the proposed approach with three widely used classifiers: radial basis function network (RBFN), 3 layered multilayer perceptron (MLP3) and support vector machine (SVM).

The rest of the paper is organized as follows. Section 3 presents segmentation and key features extraction techniques of the most important geometric features. Section 4 describes the architecture of SOM and the methodologies involved in applying 26 dimensional data to the SOM network for clustering the features data into basic six emotion zones. The section is followed by system identification using self-organizing map that creates a model by solving least square error of a supervised training system. Experimental results are given in Section 5 and finally in Section 6, conclusions are drawn.

2. Related works

Facial expression analysis approaches can be broadly classified into three basic stages: face detection, facial features extraction, facial expression classification. For decades, researchers are working on human facial expression analysis and features extraction. Substantial efforts were made during this period [26,27,30,23]. Major challenge was the automatic detection of facial features. Representation of visual information in order to reveal the subtle movement of facial muscles due to changes in expression is one of the vital issues. Several attempts were made to represent the visual informations accurately. Some of them are: optical flow analysis [14], local binary patterns (LBPs) [22], level set [23], active appearance model (AAM) [14], geometric analysis of facial features [32]. The major drawback with model based methods like AAMs and ASM is that they need prior information about the shape features. Generally, during the training phase of AAM and ASM, the shape features are marked manually [11]. Moore et al. found appearance based features by dividing the face image into sub-blocks. They used LBPs and variations of LBPs as texture descriptors [17]. Gu et al. [5] used contours of the face and its components with a radial encoding strategy to recognize facial expansions. They applied self-organizing map to check the homogeneity of the encoded contours. Kobayashi and Hara [8] modeled local facial features using geometric facial points. Zang et al. [32] used geometric components of facial points along with multi-scale and multi-orientation Gabor wavelet coefficients computed from every pixel of facial images.

Many techniques have been proposed for classification of facial expressions, such as multilayer perceptron (MLP) [33], radial basis function network (RBFN) [21,13], support vector machine (SVM) [1] and rule based classifiers [27].

3. Automatic facial features extraction techniques

The first and most crucial aspect of automatic facial expression recognition is the accurate detection of the face and prominent facial features, such as eyes, nose, eyebrows and lips. We present an analytical model shown in Fig. 1, consisting of 23 facial points which can describe all six basic facial expressions in frontal face images. The details of the 23 facial points are given in Fig. 1. We extract 26 dimensional geometric facial features using the concept of the analytical face model. The 26 dimensional geometric features are consisting of displacement of 8 eyebrow points, 4 lip points along x- and y-direction and projection ratios of two eyes. The displacement or movement of facial features is calculated using the neutral expression as reference where nose tip also plays the role in calculating the features displacement. Explanation of this part is given in Section 3.6.

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

3.1. Face detection

Face detection is considered as one of the most complex and challenging problems in the field of computer vision, because of the large intra-class variations caused by the changes in facial appearance, pose, lighting, and expression. The first and most significant step of facial expression recognition is the automatic and accurate detection of the face. We use Paul Viola and Michael Jones' face detection algorithm [29] to extract the face region. The face detection is 15 times quicker than any technique so far with 95% accuracy at around 17 fps. They use simple rectangular (Haar-like) features which are equivalent to intensity difference readings and are quite easy to compute.

3.2. Eye detection and eye features extraction

Accurate detection of eyes is desirable since eyes' centers play a vital role in face alignment and location estimation of other facial features [18], like lips, eyebrows, nose, etc. After the face is detected, we first estimate the expected region of eyes using facial geometry. In frontal face images the eyes are located in the upper part of the face. Removing the top (1/5)th part of the face region we take the first (1/3)rd vertical part as the expected region of eyes. We use Haar-like cascaded features and the Viola–Jones' object detection algorithm to detect the eyes.

The key challenge in eye state detection is due to the presence of eyelashes, shadows between eyes and eyebrows, too little gap between eyes and eyebrows. Moreover, the eye corners are situated in the skin region and do not have any distinct gray scale characteristics. To overcome these problems, we propose an effective eye states' detection technique using horizontal and vertical projections applied over the threshold image of eye's non-skin region. It can be assumed that the extend of opening of the eye is directly proportional to the maximum horizontal projection. To threshold this transformed image, an adaptive thresholding algorithm is proposed here, which is based on Niblack's [19] thresholding method, generally used to segment document images for optical character recognition. Niblack's method gives the threshold value within a local window. It calculates the threshold value for every pixel using local mean and local standard deviation. It yields effective results for document image segmentation but its performance is very poor in our case. With slight modifications the algorithm in its present form [15] gives good segmentation results. Niblack's adaptive thresholding algorithm is given as

$$T_{local}(x,y) = \mu_{local}(x,y) + k \times \sigma_{local}(x,y) \tag{1}$$

$$\sigma^2_{local}(x,y) = \frac{1}{w^2}\left[\sum_{j=y-w/2}^{y+w/2}\sum_{i=x-w/2}^{x+w/2}((\mu_{local}(x,y) - f(i,j))^2\right] \tag{2}$$

$$\mu_{local}(x,y) = \frac{1}{w^2}\left[\sum_{j=y-w/2}^{y+w/2}\sum_{i=x-w/2}^{x+w/2} f(i,j)\right] \tag{3}$$

and the proposed algorithm is given as

$$T_{local}(x,y) = \mu_{global}(x,y) + k \times \sigma_{local}(x,y) \tag{4}$$

$$\mu_{global}(x,y) = \frac{1}{M \times N}\left[\sum_{j=0}^{N}\sum_{i=0}^{M} f(i,j)\right] \tag{5}$$

where $T_{local(x,y)}$ is the threshold value of the pixel located at (x,y) coordinate computed within a window (w) of size 7. $\sigma_{local}(x,y)$ is the standard deviation and is obtained from local variance, μ_{local} is the local mean and μ_{global} is the global mean. $M \times N$ is the size of eye ROI image, k is taken as 5 and window w size is 7×7. The projection ratio is taken, rather than individual projection values, to remove scaling issues. According to Peer [10], one of the simpler methods for skin classification is as shown in the steps given below. It can be observed that the skin region is mainly dominated by the red color component compared to green and blue color. Red, green and blue components $(R(x,y), G(x,y), B(x,y))$ are extracted from the eye region. Since the red color component dominates the skin region, the normalized red component is obtained as follows. Normalization is necessary to eliminate the effect of brightness variation:

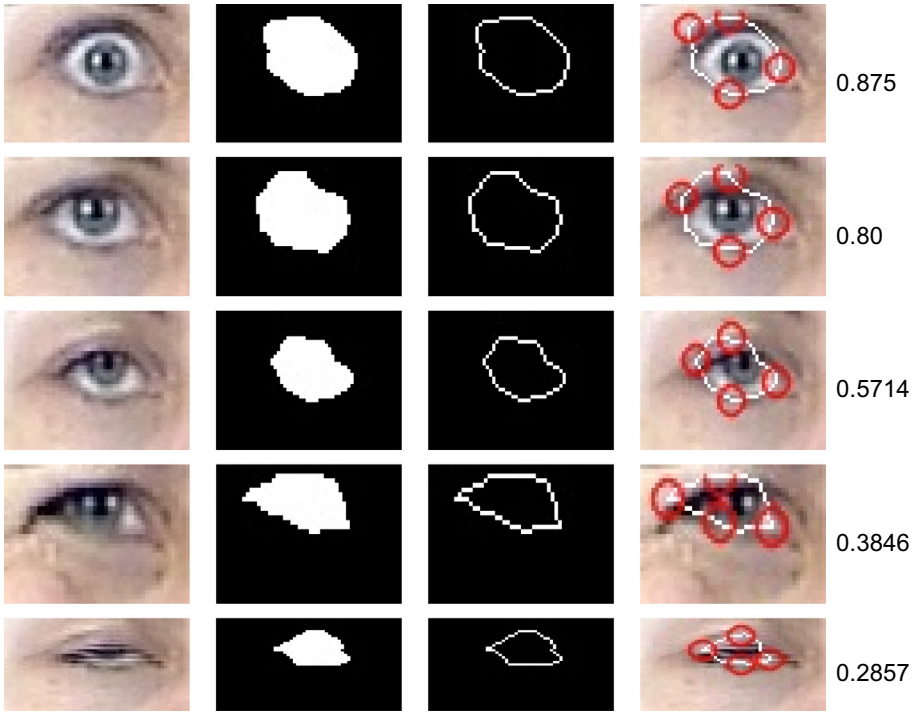


Fig. 2. Examples of eye segmentation, key feature points detection and projection ratios.

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

$$h_r(x,y) = \frac{255 \times R(x,y)}{R(x,y)+G(x,y)+B(x,y)} \tag{6}$$

$$h(x,y) = \frac{h_r(x,y) - \min_{x,y} h_r(x,y)}{\max_{x,y} h_r(x,y) - \min_{x,y} h_y(x,y)} \tag{7}$$

$$P_r = \frac{\max \sum_{j=1}^{N} I(x_i,y_j)}{\max \sum_{i=1}^{M} I(x_i,y_j)} \tag{8}$$

where $h_r(x,y)$ is the normalized red component. The eye region is extracted from the skin region by using a transformation given by $h(x,y)$. After segmenting the non-skin region of eyes, horizontal and vertical projections method is used over the threshold image to obtain maximum vertical projection and maximum horizontal projection. Assuming $I(x_i,y_j)$ as a threshold value of a point with (x_i,y_j)th coordinate in the eye region. The projection ratio is calculated using Eq. (8), where $\sum_{j=1}^{N} I(x_i,y_j)$ is the horizontal projection, $\sum_{i=1}^{M} I(x_i,y_j)$ is the vertical projection for the image of size $M \times N$ and P_r is the projection ratio. Before applying the method to find projection ratio, the largest connected region is extracted from the threshold image. This is done to eliminate the effect of noises. Fig. 2 shows a few examples of eye features detection. The first column shows eye images starting from widely open to nearly closed, the second column gives the threshold result, the third column gives the largest contour detected, the fourth column shows the detection results of key feature points and last column demonstrates the corresponding projection ratios. Fig. 3 depicts the plot of projection ratios for the sample images given in Fig. 2. The plot demonstrates how the eye projection ratio is modulated by changes in the state of the eye opening.

Algorithm 1. Steps for skin classification.

1: (R,G,B) is classified as skin if $R > 95$,
2: and $G > 45$ and $B > 20$ and $\max\{R,G,B\} - \min\{R,G,B\} > 15$,
3: and $|R-G| > 15$ and $R > G$ and $R > B$.

Algorithm 2. Eye feature points detection technique.

1: Using contour detection algorithm [24] gather all the contours from the threshold image.
2: Retrieve the largest contour and save the contour's data into an array.
3: Find the two extreme x-coordinate values of the largest contour, i.e., largest and smallest x coordinate values. Get the corresponding y-coordinates. The obtained points as left and right extreme feature points.
4: To detect upper and lower mid points of eyes, get the expected x-coordinate as $x = (x_1 + x_2)/2$, where x_1, x_2 are two extreme points. Then, find the nearest x-coordinate values to the expected x-coordinate value. Set a constraint within the search region for both x-direction and y-direction to keep the search within the ROI.
5: Among the two points, consider the lower mid point as the point with larger y-coordinate value and upper mid point as the point with smaller y-coordinate value.

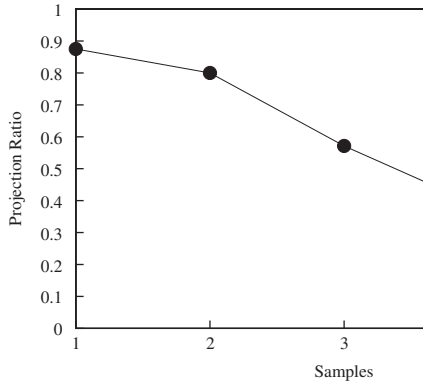


Fig. 3. The plot shows the projection ratios for the sequence of progressively closing eye images given above. The x-axis shows the image number, and the y-axis computes the corresponding projection ratio.

3.3. Eyebrow features extraction

The steps involved in this section include: eyebrow location estimation, pseudo-hue plane extraction, segmentation, contour extraction and, finally, key points detection. The objective of this process is to obtain a set of key points (a vector) which can adequately describe the characteristics of the eyebrow and can be further used to recognize facial expression.

Eyebrow location is estimated using basic facial geometry. As we are using frontal or nearly frontal face images, the eyebrow region will be found slightly above the eye region. Taking each eye region as a reference, we estimate the expected eyebrow region (which will take into account the possible movements of eyebrow in successive frames). Height of the eyebrow ROI is estimated as 1.5 times the eye ROI height.

3.3.1. Eyebrow pseudo-hue plane extraction

We now introduce a new eyebrow segmentation method based on color that we find to be a significant improvement over other reported methods [2,12]. It is well known that eyebrow hair consists of two types of pigments called eumelanin and pheomelanin. Pheomelanin is found to be there in all human beings and comprises red color information. We extract a pseudo-hue plane of the eyebrow region, based on this fact which tells us to expect that the eyebrow hairs has more of red color information than green. Fig. 4 shows an example of pseudo-hue images obtained after applying the algorithm. A clear distinction between eyebrow and non-eyebrow regions can be observed in the pseudo-hue images obtained.

Algorithm 3. Method for extracting pseudo-hue plane of eyebrow region.

1: Get the eyebrow ROI.
2: Split the RGB image of eyebrow ROI into HSI component planes. Enhance the contrast of the region by applying histogram equalization over the intensity plane. Marge back all the planes.
3: Extract the red, green and blue components of the image obtained from the above step.
4: Obtain the pseudo-hue plane of eyebrow as $h = r/(g+b)$ for all the pixels. where r, g and b are red, green and blue components of each pixel.

Fig. 4. Eyebrow features' detection steps: top row left image shows the pseudo-hue image obtained from a still image, top right image shows the thresholded image of the plane, bottom left image gives the largest eyebrow contour, and bottom right shows four key points extracted.

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

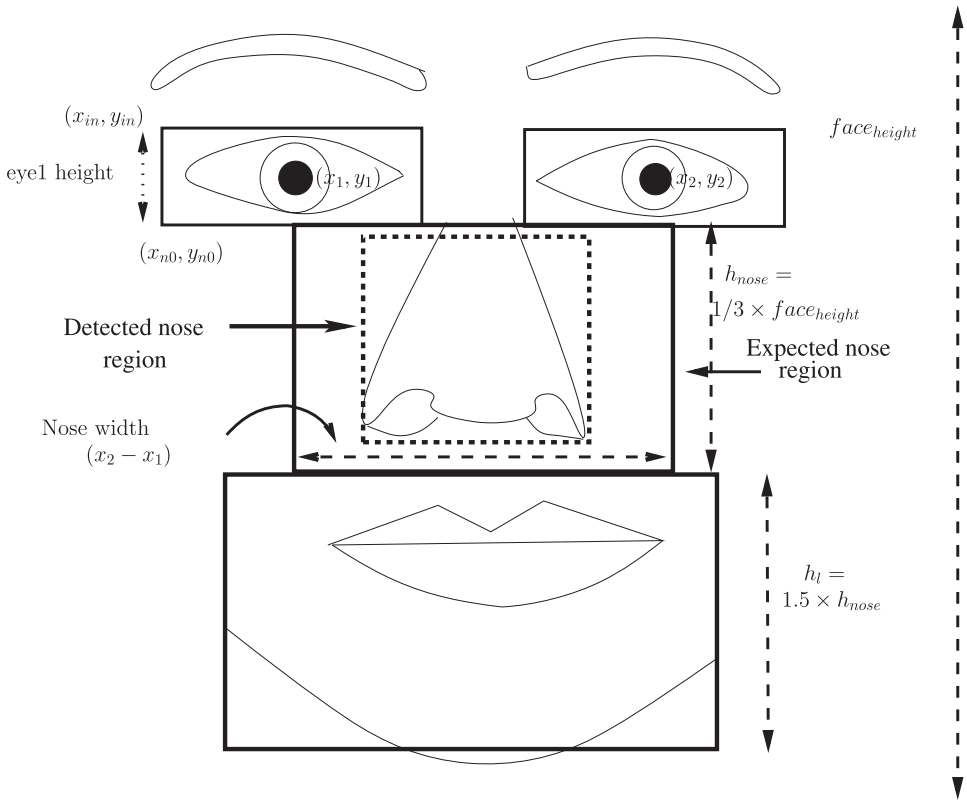


Fig. 5. Estimated location for nose and lip.

5: For an image of size $M \times N$
6: **for** i=0 **to** $M-1$ **do**
7: **for** j=0 **to** $N-1$ **do**
8: The pseudo-hue is normalized as follows: $h_{norm}(i,j) = (h(i,j) - \min(h))/(\max(h) - \min(h))$ where h_{norm} is the normalized (0 - 1) pseudo-hue, $\min(h)$ and $\max(h)$ are the minimum and maximum of the pseudo-hue value obtained over ROI in step 3. The pseudo-hue plane is scaled to an 8 bit image representation by multiplying h_{norm} with 255.
9: **end for**
10: **end for**

3.3.2. Eyebrow segmentation, contour extraction and key-points detection

The pseudo-hue plane extracted in Section 3.3.1 shows a clear distinction between eyebrow and skin regions. The plane is normalized to eliminate the effect of intensity variation. The normalization method is explained in Algorithm 3. The adaptive thresholding algorithm described in Section 3.3 is now applied to the pseudo-hue plane. A window of size 7×7 is taken to calculate the threshold iteratively. The thresholding method uses summation of global mean and constant k times local standard deviation to calculate the threshold. k is chosen as 0.5.

Morphological operations, erosion followed by dilation are applied on the thresholded image for 2–3 iterations to remove classification-induced near the eye region and boundary region (due to the presence of hair and eye lids near the boundary region). A contour detection method is used on the thresholded image to extract all the contours within the eyebrow region. The eyebrow feature points are detected by a process similar to the one described in Section 3.2. Fig. 4 shows an example of the eyebrow pseudo-hue plane, threshold image of the plane, contour extracted from the threshold image and four key points extracted from the largest contour.

3.4. Nose features detection

For a frontal face image, the nose lies below the eyes. Fig. 5 shows a pictorial description of its approximate nose position. Using this information of facial geometry, we estimate the nose position.

It is observed, generally the nostrils are relatively darker than the surrounding nose regions even under a wide range of lighting conditions. We apply a simple thresholding method on the gray image of nose ROI followed by conventional morphological operations that remove noises and thus, have a clear distinction between two nostrils. The contour detection method [24] is applied to locate two nostrils contours. The centers of these two contours are considered as the two nostrils.

3.5. Lip features extraction

Algorithm 4 lists the steps of lip region extraction technique. Next step is to detect the lip contour from the estimated lip region. A color based transformation method is used to extract lip from the expected region. The method was originally proposed by Hulbert and Poggio [4] and it can be given as follows:

$$h(x,y) = \frac{R(x,y)}{G(x,y) + R(x,y)} \tag{9}$$

where $h(x,y)$ is the pseudo-hue plane obtained after transformation. The lip segmentation result obtained after applying the above equation gives a clear distinction between red and green components within lip region and non-lip region. The obtained transformed plane is normalized to make it robust to change in

Fig. 6. The initial four images show lip contour detection results of a still image after applying snake algorithm with same initial parameters, but slightly different initial contours. The image shows the result of our proposed method obtained without any prior information. (a) Initial contour, (b) snake result, (c) initial contour, (d) snake result and (e) our result

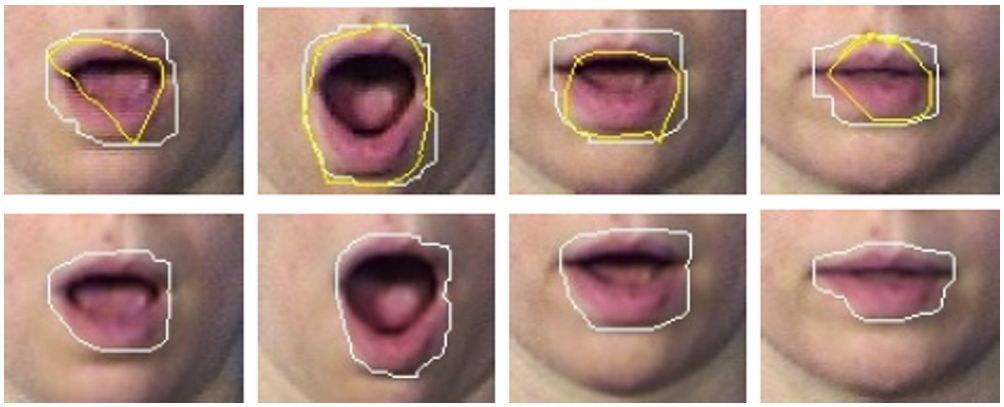


Fig. 7. The first row in the figure shows few examples of the lip contours detection results from video sequences found after applying snake algorithm. The initial parameters used for snake are $\alpha = 0.01$, $\beta = 1.0$ and $\gamma = 0.1$. The outer contour is the initial contour given to snake algorithm and the inner contour is the snake result. The second row shows the results of those image sequences using our proposed algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

intensity. The normalization method is given as

$$k(x,y) = \frac{h(x,y) - \min(h)}{\max(h) - \min(h)} \tag{10}$$

where k is the normalized (0–1) pseudo-hue, $\min(h)$ and $\max(h)$ are the minimum and maximum of the obtained pseudo-hue respectively. The pseudo-hue plane is scaled to 8 bit image representation by multiplying k with 255. An adaptive thresholding method is applied over the normalized pseudo-hue plane to segment the lip region.

Algorithm 4. Steps to estimate lip region.

1: Get the eye centers (x_1, y_1) and (x_2, y_2) after detecting face and eye using Haar-like cascaded features [29].
2: Detect nose using Haar-like cascaded features within the estimated nose region. Denote the height of the nose as n_{height}.
3: Estimate mouth region as follows:
   1. The mouth rectangular region can be given as $rect(x_l, y_l, h_l, w_l)$, where x_l and y_l are the x and y coordinates of left upper corner point, h_l is the height and w_l is the width of the rectangle.
   2. h_l is taken as 1.5 times to that of the height of the nose n_{height} taking into consideration that the expected lip movements will be covered within the region.
   3. Width w_l is taken as $(x_2 - x_1)$, i.e., distance between two eye's centers along x-axis. The x_l and w_l are increased with certain values so that it will cover the area when the person smiles or for any kind of mouth expansion.

3.5.1. Comparison of proposed approach with snake algorithm

The snake algorithm introduced by [6] is a well established method. But in practice, it is very difficult to fine tune its parameters and as a result it often gets converged to a wrong lip contour. Preservation of the lip corners is also difficult with snake algorithm. Beyond all these drawbacks, use of snake algorithm needs proper initialization of the starting contour (i.e., an initial contour must be set closer to the actual lip shape which is in reality often unknown to us). Moreover, it is highly computationally expensive as it may need many iterations to actually converge to the lip contour. Fig. 6 shows an example of snake applied over a still image taken from the FEI database [25]. The parameters are chosen as $\alpha = 0.01$, $\beta = 1.0$ and $\gamma = 0.1$ for both (a) and (c) with initial contours taken slightly different from each other. The results of the snake are shown in (c) and (d). The parameters are chosen after several trail and error. The result shows how the accuracy of snake depends on the choice of initial contour. In the first row of Fig. 7 we show some of the snake results obtained after applying the snake algorithm on a video (taken from MMI database [20]). The white colored contour is the initial contour given to the snake algorithm and the yellow colored contour is the resultant lip contour. The second row of the figure shows the lip contour found by using our proposed lip contour detection algorithm.

The result shows the improved accuracy of our algorithm compared to the snake algorithm. The frames are given the same initial parameters ($\alpha = 0.01$, $\beta = 1.0$ and $\gamma = 0.1$) and with initial contours very close to the actual lip contour (shown by the white line). The yellow (darker) line shows the corresponding snake results obtained. The results could have been improved by changing the parameters, but in general, when we are tracking lip movements in a video clip, we cannot change the parameter, as the nature of the outcome is unknown to us in each video frame. With the use of our proposed lip contour detection method, such problems are entirely eliminated and we get reasonably accurate lip contours without depending on any kind of initial parameter inputs or contour initialization.

3.6. Lip mid-points and corner-points detection technique

Lip key-points, i.e., two lips corners and upper and lower mid points of the lip are extracted using a similar method to that used

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

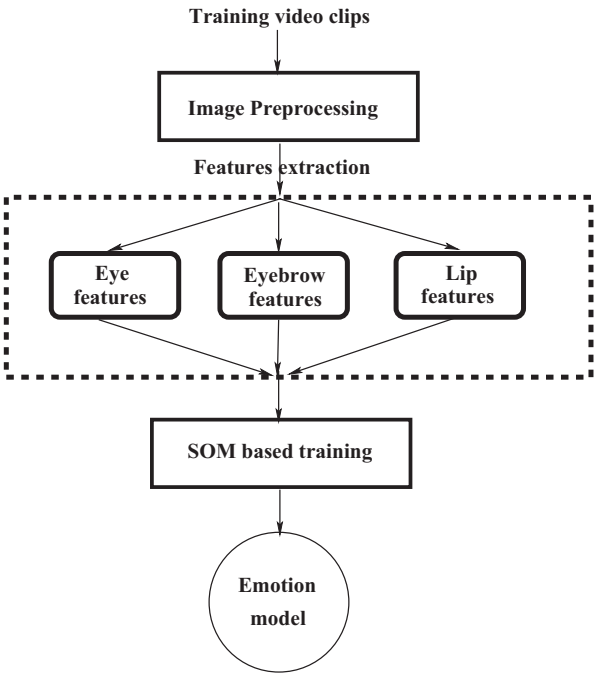


Fig. 8. System diagram of the proposed training approach.

for eyebrow key-points extraction given in Section 3.3.2. The displacement of each of the feature points with respect to its location in the neutral frame is considered as displacement data. This displacement data contains information about facial muscle movements which will in turn indicate the facial expression. The extended KSOM uses this displacement data as an input vector to train the network to classify different facial expressions. The first frame of video clip (taken from MMI database) is considered as the neutral frame as it is observed that usually, the clip starts with a neutral expression.

Following steps explain calculation of displacement data at each feature point:

1. A reference along y-axis taken as $(x=(x_1+x_2)/2, y)$ to measure movement of eyebrow feature points along horizontal direction. Two references along x-axis are taken as y_1 and y_2 to measure vertical movement of left and right points respectively, where (x_1, y_1) and (x_2, y_2) are the two eye's centers.
2. Horizontal distances of the neutral frame's eyebrow feature points are calculated from the references. $(x-browpt_x)$ and $(browpt_x-x)$ for left eyebrow features and right eyebrow features respectively. Similarly, vertical distances are calculated as $(y1-browpt_y)$ and $(y2-browpt_y)$, where $(browpt_x, browpt_y)$ are coordinates of each eyebrow feature points.
3. Using the similar method given in step 2 the horizontal (h_{dist}) and vertical (v_{dist}) distances of feature points in subsequent frames are calculated. Finally, the relative displacement of the feature points are measured as the difference between neutral frame's distance to the successive frames' distance from the reference.
4. The displacement data are multiplied with a scaling factor (x_{scale}/y_{scale}) where x_{scale} is given as standard x-scale divided by distance between two eye's centers $(x_{standard}/(x_2-x_1))$. And y_{scale} is given as $(y_{standard}/(nose_h))$, where $nose_h$ is the height of the nose which is given as y-coordinate of nose tip subtracted from the average of two eye's y-coordinates. $x_{standard}$ and $y_{standard}$ are chosen as 72 and 46 respectively.
5. Considering the nose tip as a reference point, the above procedure is followed to measure the displacement of lip feature points in both vertical and horizontal directions.

4. SOM based facial expression recognition

Kohonen self-organizing map (KSOM) [9] has an extra ordinary capability of clustering the data in an order that maintains the topology of input data. Because of this property of KSOM, the features data of similar facial expressions (small changes in features) get clustered into closer zones. This in turn makes the classification much better. This property of KSOM motivates us to use it for classifying the features data into six basic expressions. From the ontological prospective, the emotion space may not be topologically related. But in feature space there might exist topological relationship. Our present experimental results suggest this. Fig. 8 shows the flow diagram of the proposed SOM based facial expression recognition system. The normalized feature vector $\mathbf{X} \in \mathbb{R}^{26}$ is used to train KSOM network for classifying data into six basic emotion classes. A pictorial description of KSOM is shown in Fig. 9. KSOM discretizes the input and output spaces into several small zones, which also creates a linear mapping between input and output space. Since we want the output space to be discrete in nature, a logistic sigmoid function has been introduced after network output. The output of sigmoid function is further thresholded to yield either -1 or 1. For a given input vector $\mathbf{x}$, say if the desired output is for happiness data, we set the desired output as $\{1\ -1\ -1\ -1\ -1\ -1\}$. It means, the first bit that represents happiness is true and others are false.

4.1. SOM based system identification for emotion classification

The KSOM maps a high-dimensional space $\mathbb{R}^k$ to much lower dimensional space, usually one or two dimensions. The informations get compressed yet preserve matrix relationships of basic data, hence produces some kind of abstractions of informations. Fig. 9 shows an example of a two dimensional KSOM. The intension is to derive an extended KSOM based mathematical model using the sets of experimental features data and desired output vector. A 2D KSOM lattice network of size 10×8 is used to train feature vector $\mathbf{x}=[x_1, \ldots, x_m]^T \in \mathbb{R}^{26}$. For each node j in the 2D lattice structure, 3 parameters: weight vector $\mathbf{W}_j=[w_{j,1}, w_{j,2}, \ldots, w_{j,26}]^T \in \mathbb{R}^{26}$, matrix $\mathbf{A} \in \mathbb{R}^{6\times 26}$ and $\mathbf{b}_j \in \mathbb{R}^6$ a bias vector parameter are assigned. During training, each input vector $\mathbf{X}$ is compared against all the $\mathbf{W}_j$ to find the location of close match. The winning node named Best Matching Unit (BMU) denoted as i is given as follows:

$$i = \arg\min_j \|\mathbf{x}(n)-\mathbf{w}_j\| \tag{11}$$

For a feature vector $\mathbf{x}$, the network output vector $\mathbf{z} \in \mathbb{R}^6$ is obtained as the weighted average of over all neurons' output within the neighborhood function. The neuron closer to the BMU is given higher weightage $(h_{j,i})$ than those neurons which are far away, where the Euclidean distance $d_{j,i}$ between the BMU and each neuron is given in Eq. (17), σ is initially taken to be very large and gradually decreased at each iteration. The output vector $\mathbf{z}_j(n) \in \mathbb{R}^6$ at node j and entire network output vector $\mathbf{z}(n) \in \mathbb{R}^6$ for the nth iteration are given as follows:

$$\mathbf{z}_j(n) = \mathbf{b}_j(n) + \mathbf{A}_j(n)[\mathbf{x}-\mathbf{w}_j(n)] \tag{12}$$

$$\mathbf{z}(n) = \frac{\sum_{j=1}^{M\times N} h_{j,i}(n)\mathbf{z}_j(n)}{\sum_{j=1}^{M\times N} h_{j,i}(n)} \tag{13}$$

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

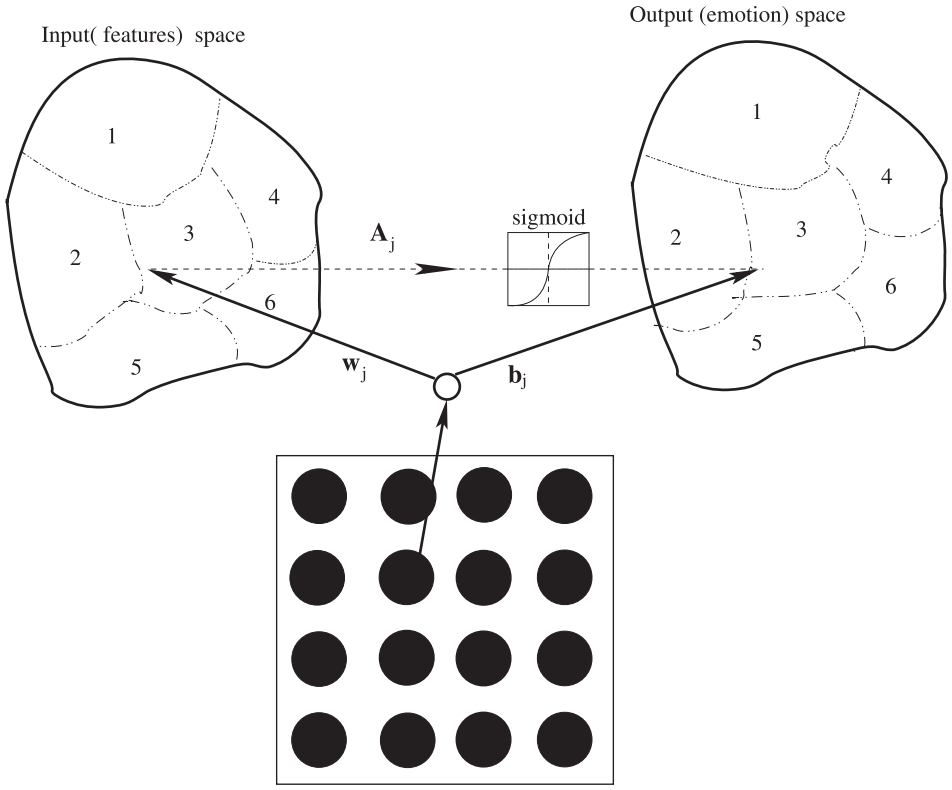


Fig. 9. Pictorial description of a 2D KSOM network for input vector **x**.

$$h_{j,i}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \tag{14}$$

where $h_{j,i}(n)$ is the neighborhood function, $\Delta\mathbf{x} = \mathbf{x} - \mathbf{w}_j$ for input vector $\mathbf{x} \in \mathbb{R}^{26}$ and corresponding weight vector $\mathbf{w}_j \in \mathbb{R}^{26}$ at the node j. The weight vector of each node j within the neighborhood is updated to make them more like input vectors. The closer the node is to the BMU i, the more its parameters get altered. The weight update rule at node j and iteration n is given as

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \alpha(n)h_{j,i}(n)[\mathbf{x}(n) - \mathbf{w}_j(n)] \tag{15}$$

$$\alpha(n) = \alpha_0 \times (\alpha_f/\alpha_0)^{n/NOD} \tag{16}$$

$$d_{j,i}^2 = \| \mathbf{r}_j - \mathbf{r}_i \|^2 \tag{17}$$

$$\sigma(n) = \sigma_0 \times (\sigma_f/\sigma_0)^{n/NOD} \tag{18}$$

where $\alpha(n)$ is the learning rate at iteration n, α_0 is the initial learning rate, α_f is the final learning rate and *NOD* is the total number of iteration needed for training. The Euclidean distance between BMU i and node j is given by $d_{j,i}$, $\sigma(n)$ is the radius of the neighborhood at iteration n. α_0 and α_f are chosen to be 0.95 and 0.005 respectively. σ_0 is the initial radius, σ_f is the final radius, The learning rate $\alpha(n)$ and the neighborhood radius $\sigma(n)$ gradually diminishes with each iteration. Around 20,000 input displacement data are used to train KSOM after randomizing and normalizing the data. Randomization of data is very essential for training each zones of the lattice uniformly. The network output is passed through a sigmoid function and each dimension of the final output vector **y** is checked against each dimension of the network output vector **z**. Result of final output node k which is y_k is given as

$$y_k = 1 \quad \text{if} \quad \frac{1}{1+e^{-z_k}} \geq 0.5 \tag{19}$$

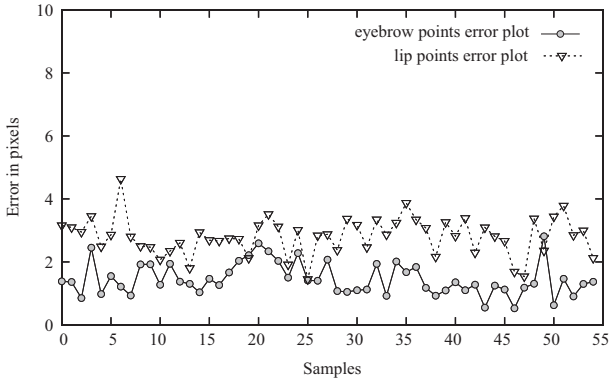


Fig. 10. Error in pixels for eyebrow and lip feature points detection results when compared with ground truth.

$$y_k = -1 \quad \text{if} \quad \frac{1}{1+e^{-z_k}} < 0.5 \tag{20}$$

where $k \in \{0, \ldots, 5\}$. The training of the parameters $\mathbf{b}_j$ and $\mathbf{A}_j$ is done simultaneously along with weight vector $\mathbf{W}_j$. The parameters are updated using error correction learning, which can be implemented by applying gradient descent method over the cost function E. Since output depends on the logical operations and output y_k is either -1 or 1, we need to use different objective functions or cost function instead of squared error norm, for the optimization problem. Here we consider perceptron criterion for cost function E which can be defined as

$$E = \frac{1}{M}\sum_{k=1}^{M} \max(0, -y_k^d z_k) \tag{21}$$

$$E_k = \max(0, -y_k^d z_k) \tag{22}$$

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

where E_k is the error at output node k. The term $\max(0, -y_k^d z_k)$ is zero if output at node k, y_k is predicted correctly. Otherwise, it is same as the confidence in the miss prediction. The parameters will be updated only if there is a loss, i.e., when the node at output does not match with the desired output component. Stochastic gradient descent approach is applied to update the parameters. The update rules of the parameters $\mathbf{b}_j$ and $\mathbf{A}_j$ are as shown in Algorithm 5.

Algorithm 5. Parameters update rules.

1: if $z_k > 0$: resultant output $y_k = 1$

$$\max(0, -y_k^d z_k) = \begin{cases} 0 & \text{if } y_k^d = 1 \\ z_k & \text{if } y_k^d = -1 \end{cases}$$

2: if $z_k < 0$: resultant output $y_k = -1$

$$\max(0, -y_k^d z_k) = \begin{cases} z_k & \text{if } y_k^d = 1 \\ 0 & \text{if } y_k^d = -1 \end{cases}$$

3: The derivatives of the cost function with respect to the parameters at each output node k are given as

$$\frac{\delta E_k}{\delta b_{j,k}} = -\frac{(y_k^d - y_k)h_{j,i}}{\sum_{j=1}^{M\times N} h_{j,i}} \tag{23}$$

$$\frac{\delta E_k}{\delta \mathbf{a}_{k,j}} = -\frac{(y_k^d - y_k)\times h_{j,i}(\mathbf{x} - \mathbf{w}_j)}{\sum_{j=1}^{M\times N} h_{j,i}} \tag{24}$$

where $\mathbf{a}_{k,j}$ is the kth row of the matrix $\mathbf{A}$ at the input node (node in SOM network) j.

4: When $z_k = 0$, sigmoid function output shown in Eq. (19) is 0.5 and since y_k is set to 1 for this case, we add a small positive value to z_k so that it follows the first update rule.

5: The updates of parameters $\mathbf{b}_j$ and $\mathbf{A}_j$ are thus given as follows:

$$\mathbf{b}_j(n+1) = \mathbf{b}_j(n) - \eta \frac{\delta E}{\delta \mathbf{b}_j(n)} \tag{25}$$

$$\mathbf{A}_j(n+1) = \mathbf{A}_j(n) - \eta \frac{\delta E}{\delta \mathbf{A}_j(n)} \tag{26}$$

$$\eta = \eta_I \times \left(\frac{\eta_F}{\eta_I}\right)^{\frac{n}{NOD}} \tag{27}$$

where η is the learning rate and η_I is the initial learning rate set to 0.9 , η_F is the final learning rate set to 0.005, n is the nth iteration and NOD is the total number of data.

5. Experimental results and discussion

This section presents the results of features detection and classification of facial expressions into six basic emotions (happiness (H), sadness (Sa), disgust (D), anger (A), surprise (Sur), fear (F)) demonstrating the accuracy of the proposed methodologies. We used the publicly available and well-known MMI database [20] for our research purpose. Some examples of the facial features detection results are displayed in Fig. 11. In our experiments, we used 81

Fig. 11. Examples of facial features detection results using proposed methods.

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

Table 1
Details of parameters setting.

SOM	RBFN	MLP3	SVM
2D lattice network of size 10×8	50 radial centers	10 neurons in hidden layer	15 SVMs for 6 class classification
Logistic sigmoid at output node	Gaussian radial function at centers	Sigmoid function at hidden layer	Radial basis function (RBF) as kernel
Neighborhood radius σ initial = 3.5, final = 0.001	Each center's sigma updated at each iteration		Penalty weight $C = 1$, RBF radius $\gamma = 0.10$
5 generations	10 generations	10 generations	100 generations

Table 2
Confusion matrix of emotion recognition for the 26 dimensional geometric features data using KSOM. The emotion classified with maximum percentage is shown to be the emotion detected.

	H	Sa	D	A	Sur	F
H	**90.5**	3.32	2.44	3.66	0	0
Sa	0	**88.7**	1.89	0	5.66	3.77
D	0	2.38	**95.2**	2.38	0	0
A	0	1.67	0	**98.3**	0	0
Sur	0	0	0	0	**98.1**	1.85
F	0	0	4.88	0	4.88	**90.2**

Table 3
Confusion matrix of emotion recognition for the 26 dimensional geometric features data using RBFN. The emotion classified with maximum percentage is shown to be the emotion detected.

	H	Sa	D	A	Sur	F
H	**72.6**	21.2	6.3	0	0	0
Sa	1.4	**56.6**	38.5	4.0	0	0
D	0	7.5	**47.9**	27.8	16.8	0
A	0	2.01	15.85	**75.9**	6.2	0
Sur	0	0	0	11.3	**82.4**	6.37
F	0	0	1.2	8.2	26.6	**63.9**

different video clips from the MMI database. The selected video clips fall into one of the six basic emotions. There are in total 12 different subjects. Each subject is showing all the six basic expressions. The directional displacement along x- and y-coordinate of each facial points are used as input feature for training the KSOM. First the KSOM is clustered to ordered zones. The clustered weights are further used to model the six basic emotions as a function of 26 directional displacement features data. The performance of the detection results is evaluated by comparing them against the ground truth (manually marked feature points). Fig. 10 shows the average detection error of the 4 lips feature points and 4 eyebrow feature points, in terms of pixels distance. The data are taken from one video clip. The lip height and width in a neutral expression in that video clip are 33 and 65 respectively. Also, the eyebrow height and width in neutral face are 1 and 55 pixels respectively. The error is calculated using average Euclidean distance for all the four lip/eyebrow points against the manually detected lip/eyebrow points. The average of the total errors for eyebrow and lips features points are 1.45 and 2.81 pixels respectively which can be considered as very less.

The second part of the experimental results focuses on the determination of classification accuracy obtained on using proposed KSOM based classification method. Table 2 demonstrates the classification accuracy when it is tested with new displacements data. The KSOM classification accuracy is found to be highest (98.33%) for anger and least (88.7%) for sadness with average recognition rate as 93.53%.

The classification accuracy of KSOM based method is compared against three widely used classifiers: RBFN [21,13], MLP3 [33] and multi-class SVM [23]. A parametric comparison is shown in Table 1. Parameters are set after several hit and trial method. RBFN uses 50 hidden layer. The σ value for each dimension in each of the hidden layer is updated during the training process. MLP3 uses 10 neurons in the hidden layer and sigmoid function at each hidden layer. The standard library libSVM is used for SVM based classification. It uses one-against one method that needs 15 SVMs for 6 class classification problem. Radial basis function (RBF) kernel which is observed to be giving best accuracy is used in SVM based training. Average recognition rate using RBFN is found to be 66.54% along with highest recognition rate as 82.36% for the facial expression showing surprise and lowest recognition rate as 47.9% for disgust. The average recognition rate for Multilayer perceptron having one hidden layer (MLP3) is 72.15% with highest recognition rate 97.3% for happiness and lowest 54.61% for disgust. Recognition accuracy of the proposed method is observed to be comparable to widely used multi-class SVM. The average recognition rate of multi-class SVM is 92.53% with highest recognition rate 98.33% and lowest 88.10%. Tables 3–5 show the confusion matrices for RBFN, MLP3 and SVM based classification methods respectively.

Table 4
Confusion matrix of emotion recognition for the 26 dimensional geometric features data using MLP3. The emotion classified with maximum percentage is shown to be the emotion detected.

	H	Sa	D	A	Sur	F
H	**97.3**	2.7	0	0	0	0
Sa	5.27	**55.2**	39.5	0	0	0
D	3.65	9.01	**54.6**	23.8	8.92	0
A	2.6	0	16.2	**79.3**	0.8	0.1
Sur	0	0	6.0	21.66	**71.4**	0.9
F	0	0	3.6	11.6	9.91	**75.1**

Table 5
Confusion matrix of emotion recognition for the 26 dimensional geometric features data using SVM. The emotion classified with maximum percentage is shown to be the emotion detected.

	H	Sa	D	A	Sur	F
H	**91.5**	3.66	2.44	2.44	0	0
Sa	1.88	**86.4**	2.24	3.77	5.66	0
D	0	2.38	**90.5**	7.14	0	0
A	1.67	0	0	**98.3**	0	0
Sur	1.88	0	0	0	**98.1**	0
F	4.76	2.43	2.43	0	0	**90.4**

6. Conclusions

A completely automated system for facial geometric features detection and facial expression classification is proposed. We introduce different techniques to detect eyebrow features, nose features, state of eyes and lip features. The proposed eye state detection method gives a clear distinction between different states

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

of eye opening. The detection results for eyebrow feature points and lip feature points are compared against the ground truth. It is observed that for a neutral face having lips with height and width 33 and 65 pixels respectively, the average detection error is only 2.81 pixels. And for eyebrows with height and width 15 and 55 respectively, the average error is 1.45 pixels, which can be considered as very less.

A new mechanism is introduced based on 2D KSOM network to recognize facial expression that uses only a 26 dimensional geometric feature vector, containing directional displacement information about each feature point. The KSOM network parameters are updated simultaneously to train the model for six basic emotions as a function of 26 directional displacement data. Experimentation is carried out over 81 video clips taken from MMI database. An average recognition rate of 93.53% is achieved using the proposed KSOM based recognition method, with the highest recognition rate as 98.33%. The performance of the proposed method is compared with three widely used classifiers: RBFN, MLP3 and multi-class SVM. The average recognition accuracy using RBFN is found as 66.54%, which is much lower than the recognition rate found using our proposed method. MLP3 gives comparatively better performance than RBFN with average recognition rate 72.15%. However, MLP3 achieved highest recognition rate for happiness data (97.3%) among all the four classification methods. On the other hand, the KSOM based recognition method gives much better performance on average than RBFN and MLP3. The performance of KSOM is increased by 1% as compared to the multi-class SVM which is known to be the state of the art in classification. The relative increase of recognition accuracy using KSOM over MLP3 is 21.39% and over RBFN is 26.99% which is a significant improvement. Thus, the extensive experiment illustrates the effectiveness and accuracy of KSOM based facial expression recognition system using only geometric features.

Conflict of interest statement

None declared.

References

[1] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, J. Movellan, Recognizing facial expression: machine learning and application to spontaneous behavior, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 2, June 2005, pp. 568–573.
[2] Q. Chen, W. Cham, K. Lee, Extracting eyebrow contour and chin contour for face recognition, Pattern Recognit. 40 (8) (2007) 2292–2300.
[3] P. Ekman, W.V. Friesen, J.C. Hager, Facial Action Coding System, A Human Face, Salt Lake City, 2002.
[4] N. Eveno, A. Caplier, P. Coulon, A parametric model for realistic lip segmentation, in: 7th International Conference on Control, Automation, Robotics and Vision, ICARCV, vol. 3, IEEE, 2002, pp. 1426–1431.
[5] W. Gu, Y. Venkatesh, C. Xiang, A novel application of self-organizing network for facial expression recognition from radial encoded contours, Soft Comput. Fusion Found. Methodol. Appl. 14 (2) (2010) 113–122.
[6] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vis. 1 (4) (1988) 321–331.
[7] M.H. Khosravi, R. Safabakhsh, Human eye sclera detection and tracking using a modified time-adaptive self-organizing map, Pattern Recognit. 41 (August (8)) (2008) 2571–2593.
[8] H. Kobayashi, F. Hara, Facial interaction between animated 3d face robot and human beings, in: IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 4, IEEE, 1997, pp. 3732–3737.
[9] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[10] J. Kovac, P. Peer, F. Solina, Human Skin Color Clustering for Face Detection, vol. 2, IEEE, 2003.
[11] A. Lanitis, C. Taylor, T. Cootes, Automatic interpretation and coding of face images using flexible models, IEEE Trans. Pattern Anal. Mach. Intell. 19 (July (7)) (1997) 743–756.
[12] C. Lee, J. Kim, K. Park, Automatic human face location in a complex background using motion and color information, Pattern Recognit. 29 (11) (1996) 1877–1889.
[13] D. Lin, Facial expression classification using PCA and hierarchical radial basis function network, J. Inf. Sci. Eng. 22 (5) (2006) 1033–1046.
[14] R. Luo, C. Huang, P. Lin, Alignment and tracking of facial features with component-based active appearance models and optical flow, in: International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, July 2011, pp. 1058 –1063.
[15] A. Majumder, L. Behera, K.S. Venkatesh, Novel techniques for robust lip segmentations, automatic features initialization and tracking, in: Signal and Image Processing, ACTA Press, 2011.
[16] A. Mehrabian, Nonverbal communication, Aldine De Gruyter, 2007.
[17] S. Moore, R. Bowden, Local binary patterns for multi-view facial expression recognition, Comput. Vis. Image Underst. 115 (4) (2011) 541–558.
[18] V.H. Nguyen, T.H.B. Nguyen, H. Kim, Reliable detection of eye features and eyes in color facial images using ternary eye-verifier, Pattern Recognit. (2012).
[19] W. Niblack, An Introduction to Digital Image Processing, Strandberg Publishing Company, Birkeroed, Denmark, 1985.
[20] M. Pantic, M.F. Valstar, R. Rademaker, L. Maat, Web-based database for facial expression analysis, in: Proceedings of IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, July 2005, pp. 317–321.
[21] M. Rosenblum, Y. Yacoob, L. Davis, Human expression recognition from motion using a radial basis function network architecture, IEEE Trans. Neural Netw. 7 (5) (1996) 1121–1138.
[22] C. Shan, S. Gong, P. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image Vis. Comput. 27 (6) (2009) 803–816.
[23] A.S.M. Sohail, P. Bhattacharya, Classifying facial expressions using level set method based lip contour detection and multi-class support vector machines, Int. J. Pattern Recognit. Artif. Intell. 25 (06) (2011) 835–862.
[24] S. Suzuki, et al., Topological structural analysis of digitized binary images by border following, Comput. Vis. Graph. Image Process. 30 (1) (1985) 32–46.
[25] C. Thomaz, G. Giraldi, A new ranking method for principal components analysis and its application to face image analysis, Image Vis. Comput. 28 (6) (2010) 902–913.
[26] Y.-I. Tian, T. Kanade, J. Cohn, Recognizing action units for facial expression analysis, IEEE Trans. Pattern Anal. Mach. Intell. 23 (February (2)) (2001) 97–115.
[27] F. Tsalakanidou, S. Malassiotis, Real-time 2d+3d facial action and expression recognition, Pattern Recognit. 43 (5) (2010) 1763–1775.
[28] M. Valstar, I. Patras, M. Pantic, Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR Workshops, June 2005, p. 76.
[29] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[30] T.-H. Wang, J.-J.J. Lien, Facial expression recognition system based on rigid and non-rigid motion separation and 3d pose estimation, Pattern Recognit. 42 (5) (2009) 962–977.
[31] S. Wu, T.W. Chow, Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density, Pattern Recognit. 37 (2) (2004) 175–188.
[32] Z. Zhang, M. Lyons, M. Schuster, S. Akamatsu, Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron, in: Proceedings of 3rd International Conference on Automatic Face and Gesture Recognition, IEEE, 1998, pp. 454–459.
[33] Z. Zhang, Feature-based facial expression recognition: sensitivity analysis and experiments with a multilayer perceptron, Int. J. Pattern Recognit. Artif. Intell. 13 (1999) 893–911.

Anima Majumder received her B.Tech. degree from North Eastern Regional Institute of Science and Technology (NERIST), Itanagar, India in 2005 and M.Tech. degree from Shri Guru Govind Singhji Institute of Engineering and Technology (SGGS), Nanded, India in 2007. She worked as a senior software engineer in Robert Bosch Engineering and Business Solutions Ltd., Bangalore, India. Currently she is a Ph.D. scholar in the Department of Electrical Engineering, Indian Institute of Technology Kanpur (IITK), Kanpur, India. Her research interests include computer vision, machine learning and image processing.

Laxmidhar Behera received the B.Sc. and M.Sc. degrees in engineering from National Institute of Technology Rourkela, Rourkela, India, in 1988 and 1990, respectively, and the Ph.D. degree from the Indian Institute of Technology (IIT) Delhi, New Delhi, India. He pursued his postdoctoral studies in the German National Research Center for Information Technology (GMD), Sank Augustin, Germany, in 2000–2001. He was an Assistant Professor with the Birla Institute of Technology and Science, Pilani, India, in

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010

ARTICLE IN PRESS

1995–1999. He has also worked as a Reader with the University of Ulster, Londonderry, UK, and as a Visiting Researcher at Fraunhofer-Gesellschaft, Sankt Augustin, Bonn, and at Eidgenoessische Technische Hochschule Zurich, Zurich, Switzerland. He is currently a Professor with the Department of Electrical Engineering, Indian Institute of Technology Kanpur (IITK), Kanpur, India. He has more than 125 papers to his credit published in refereed journals and presented in conference proceedings. His research interests include intelligent control, robotics, neural networks, and cognitive modeling.

Venkatesh K. Subramanian received his B.E. degree from Bangalore University, India in the year 1987. He completed his M.Tech. and Ph.D. degrees from Indian Institute of Technology Kanpur (IITK), Kanpur, India in 1989 and 1995 respectively. Currently he is an Associate Professor in the Department of Electrical Engineering, Indian Institute of Technology Kanpur. His research interests include signal processing, image and video processing, signal and system theory and computer vision with applications in robotics.

Please cite this article as: A. Majumder, et al., Emotion recognition from geometric facial features using self-organizing map, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.10.010