Expert Systems with Applications 41 (2014) 3383–3390

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A neural-AdaBoost based facial expression recognition system

Ebenezer Owusu *, Yongzhao Zhan *, Qi Rong Mao

School of Computer Science and Communication Engineering, Jiangsu University, 301 Xuefu Road, 212301 Zhenjiang, Jiangsu, China

ARTICLE INFO

Keywords:
Facial expression recognition
Bessel transform
Gabor feature
AdaBoost
MFFNN

ABSTRACT

This study improves the recognition accuracy and execution time of facial expression recognition system. Various techniques were utilized to achieve this. The face detection component is implemented by the adoption of Viola–Jones descriptor. The detected face is down-sampled by Bessel transform to reduce the feature extraction space to improve processing time then. Gabor feature extraction techniques were employed to extract thousands of facial features which represent various facial deformation patterns. An AdaBoost-based hypothesis is formulated to select a few hundreds of the numerous extracted features to speed up classification. The selected features were fed into a well designed 3-layer neural network classifier that is trained by a back-propagation algorithm. The system is trained and tested with datasets from JAFFE and Yale facial expression databases. An average recognition rate of 96.83% and 92.22% are registered in JAFFE and Yale databases, respectively. The execution time for a 100 × 100 pixel size is 14.5 ms. The general results of the proposed techniques are very encouraging when compared with others.

1. Introduction

Facial expression is the explicit transformation of the human face due to the automatic responses to the emotional instability. In most situations it is spontaneous and uncontrollable. The automatic facial expression involves the application of an artificial intelligent system to recognize the expressions of the face under any circumstance. Today, the studies of facial expressions have gained keen interest in pattern recognition, computer vision and its related fields. Mainly, such facial expressions are the seven prototypical ones, namely; anger, fear, surprise, sad, disgust, happy, and neutral.

Research into automatic facial expression recognition is very important in this modern society of technological age. For instance, the technology is applied in a wide variety of contexts, including robotics, digital signs, mobile applications, and medicine. It is reported that "some robots can operate by first recognizing expressions" of humans (Bruce, 1993). The AIBO robot for instance is a biologically-inspired robot that can show its emotions via an array of LEDs located in the frontal part of the head (Breazeal & Scassellati, 2002). In addition to this, the robot can also display 'happiness' feeling when it detects a face. In behavioral sciences and medicine for instance, expression recognition is effectively applied for intensive care monitoring (Morik, Brockhausen, & Joachims, 1999). Currently, there are developing systems that are capable of making routine examinations of facial behavior during pain in clinical settings. In infants the Neonatal Facial Coding System (NFCS) has been employed for real-time assessment within 32 to 33 week post-conceptional age infants who are undergoing a heel lance. The technology is being used in more advanced settings to reduce accidents through the implementation of automated detection of driver drowsiness in public transports. This system relays information about the drivers' emotional states to observers for effective surveillance leading to necessary awareness.

The hallmark of every facial expression system is accuracy and to some extent the speed of execution. However most of the existing systems produce poor performances in terms of accuracy; as for execution speed, most of the systems are even silent to give a hint. Some few examples; Franco and Treves (2001) proposed a neural based facial expression recognition system that used principal component analysis (PCA) to reduce the feature vectors. The features were fed into a feed-forward neural network that was trained by a back-propagation network. In this system an average recognition of 84.5% was reported on the Yale facial expression database – an achievement which is not very encouraging. Kumbhar, Jadhav, and Patil (2012) described a neural network classification facial expression recognition system that employs Gabor feature extraction and feature reduction by PCA to distinguish 7-class facial expression recognition on the JAFFE database. In this system they specified 20 inputs, 40 to 60 hidden layers and seven output feed-forward neural networks. Again, the 60–70% recognition accuracy they obtained by their procedure is not encouraging

* Corresponding authors. Tel.: +86 18914557945 (E. Owusu), tel.: +13852914090 (Y. Zhan).

E-mail addresses: kayowusueb@yahoo.com (E. Owusu), yzzhan@ujs.edu.cn (Y. Zhan), mao_qr@ujs.edu.cn (Q.R. Mao).

0957-4174
http://dx.doi.org/10.1016/j.eswa.2013.11.041

to befit the expectations of a real-time system. Recently, Londhe and Pawar (2012) extracted features of the face using Affine Moment Invariants and performed the classification using feed-forward neural network. The expression recognition obtained was 93.8% on the JAFFE database. Tai and Chung (2007) extracted the facial features using a Sobel filter. In their experiment they reserved the maximum connected component to reduce the wrinkles and noises and conducted 7-class classification on JAFFE facial expression database through the application of Elman network with two hidden layers, each layer containing fifteen neurons. With this approach the average accuracy of automatic facial expression recognition is 84.7%. Zhang and Tjondronegoro (2009) extracted the expressive face by using Gabor filters, feature reduction by PCA and expression classification by neural network. In this method an average facial recognition of 93.4% was recorded in the JAFFE facial expression database. Dailey and Cottrell (1999) also extracted facial features by Gabor techniques and reduced the features by PCA. The expression classifier was neural network and the average expression recognition was 94.5% ± 0.7 on the seven prototypical facial expressions, however the facial expression database was not mentioned.

Most of these studies advocate the use of Neural Network as the expression classifier and extracted the facial features by Gabor filters and reduced the features via PCA. The displeasing thing is that all the results were not very encouraging.

This study persists in exploring the potentials of neural networks to execute this kind of assignment, trying to esteem some biological constraints, utilizing the capabilities of modular systems.

Though many techniques have been used to extract the facial features, Gabor feature extraction remains a high-quality choice; there are other alternatives but they are not very promising. Just a few examples: Satiyan and Nagarajan (2010) utilized the Haar technique to extract facial features which were used as input to the neural network for classifying 8 facial expressions. The Haar wavelet extraction is very fast (Satiyan & Nagarajan, 2010; Van, 2008), however the wavelets are too huge to result to effective classification when used as input to classifiers in facial expression recognition (Cemre, 2008); in other words it is a potential cause to misclassifications and poor performance. Distance-based feature extraction methods are also one of the largely applied techniques used for feature extraction in both 2D and 3D static faces. The idea behind these procedures is that the muscle deformations which are the major causes of changes in facial expression from normal expression results in variations of the Euclidean distances between facial landmarks or points. These points, as well as their distances, have been widely employed for static facial expression analysis (Sha, Song, Bu, Chen, & Tao, 2011; Soyel & Demirel, 2007; Tang & Huang, 2008). Among the most successful ones is feature extraction based on the Bhattacharyya distance (Choi & Lee, 2003). However, despite some advantages of this method, the degree of computational complexity is unacceptably high. The matching of even a small model shape with a normal image can take half an hour on an eight-processor Sun SPARCServer 1000 (Rucklidge, 1997; Zhang & Lu, 2004). The Patch based feature extraction method is another alternative widely exploited for facial expression biometrics. Maalej, Amor, Daoudi, Srivastava, and Berretti (2010) for instance represented extracted patches from facial surfaces by sets of closed curves and then applied a Riemannian framework to obtain the shape analysis of the extracted patches. However, the patch-based features also have numerous drawbacks. First, particular representations cannot be applied to other solutions without major modifications: the majority of the techniques have only been utilized to a single class. Also, most methods do not exploit the large amounts of available training data (Aghajanian et al., 2009).

Thus on the basis of these we still considered Gabor features as the best approach, not because it does not have drawbacks, but the drawbacks can be easily managed. The Gabor filter is a superior model of simple cell receptive fields in cat striate cortex (Jones & Palmer, 1987), and it grants exceptional basis for object recognition and face recognition (Lades et al., 1993; Wiskott, Fellous, Kruger, & vonderMalsburg, 1997). Again, the Gabor methods are superior to all the above-mentioned methods because it extracts the maximum information from local image regions (Deng, Jin, Zhen, & Huang, 2005), and it is invariant against, translation and rotations (Al Daoud, 2009).

In this work, the data were reduced in dimensions by Bessel transform (Ganga, Prakash, & Gangashetty, 2011) and then after extraction of the face by Gabor methods, the features were further reduced via an AdaBoost-based (Freund & Schapire, 1995) feature reduction technique. The selected features which represented the facial deformation patterns were then fed into a 3-layer feed-forward neural network that is trained by a back-propagation algorithm. It is interesting to note that Bessel down-sampling techniques have never been adopted for facial expression recognitions. Again, the combinations of Bessel down-sampling and the formulated AdaBoost-based algorithm is an innovation that reduces the expression dataset to enhance accuracy and speed. Finally, the construction of the feed-forward neural network is influential to bring about successful results.

The rest of the work is arranged as follows. Section 2 discusses face detection and image down-sampling. Section 3 discusses Gabor feature extraction. Section 4 discusses feature selection. Section 5 discusses the multilayer feed-forward neural network (MFFNN). Results and analysis are presented in Section 6. The final conclusions of the work are drawn in Section 7.

2. Face detection and down-sampling

The face detection component was implemented by the adoption of Viola and Jones system (Viola & Jones, 2004). Fig. 1 shows sample face detection by the Viola–Jones classifier.

The size of the image is rescaled to a window of size 20×20 pixels by the use of Bessel down-sampling. Methods like bilinear interpolations have been utilized by several authors for this task in particular but interpolations are prone to aliasing problems (Munoz, Blu, & Unser, 2001). Bessel down-sampling reduces the size of the image and preserves the details and perceptual quality of the original image (Ganga et al., 2011). The down-sampling image signal $x_d(t_1, t_2)$ is expressed as:

$$x_d(t_1,t_2) = \sum_{n_1=1}^{p}\sum_{n_2=1}^{q} c(n_1,n_2) J_0\left(\frac{\alpha_{n_1}}{p-r}t_1\right) J_0\left(\frac{\alpha_{n_2}}{q-s}t_2\right) \quad (1)$$

where p and q refer to the respective image size, $p - r$ and $q - s$ are the required reduced size of the image; r and s are positive integers that represent the reduction values, n is the number of low-frequency DCT coefficients, $J_0(\alpha_{n_1})$ and $J_0(\alpha_{n_2})$ are zero order Bessel

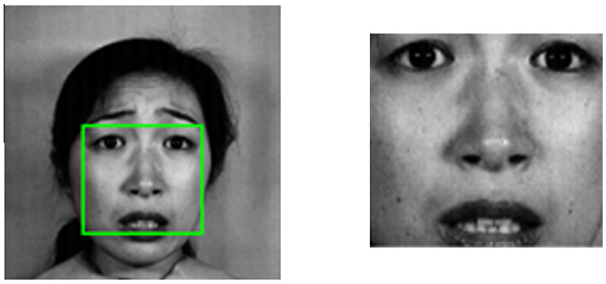


Fig. 1. Sample face detection images (left), cropped detected face (right).

functions, $c(n_1, n_2)$ are Bessel coefficients computed from the first order Bessel function, t_1 and t_2 are chosen such that $0 \leqslant t_1 \leqslant p - r$ and $0 \leqslant t_2 \leqslant q - s$. Interested readers are referred to (Al Daoud, 2009).

3. Gabor feature extraction

The 2-D Gabor filters are spatial sinusoids localized by Gaussian window, and they can be created to be selective for orientation, localization, and frequency as well. It is very flexible to demonstrate images by Gabor wavelets because the details about their spatial relations are preserved in the process.

$$G(x, y, \theta, u, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} \exp\{2\pi i(R_1 + R_2\} \tag{2}$$

where i is a complex number representing the square root of -1.

$R_1 = ux \cos\theta$ and $R_2 = uy \sin\theta$, u is the spatial frequency of the band pass, θ is the spatial orientation of the function G, (x, y) specify the position of light impulse in the visual field, σ is the standard deviation of 2-D Gaussian envelop. In this Gabor family, we chose eight orientations $\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \ldots, \frac{7\pi}{8}\}$ and five scales $\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$. In order to give added robustness to illumination we turned the Gabor filter to zero DC (direct current) by the expression

$$\tilde{G}(x, y, \theta, u, \sigma) = G(x, y, \theta, u, \sigma) - \frac{1}{q}\left[\sum_{i=-n}^{n}\sum_{j=-n}^{n} G(x, y, \theta, u, \sigma)\right] \tag{3}$$

where, q is the size of the filter, given by $q = (2n + 1)^2$. Fig. 2 shows the Gabor filter image.

The sample points of the filtered image is coded to two bits, real bit x_1 and imaginary bit x_2 such that,

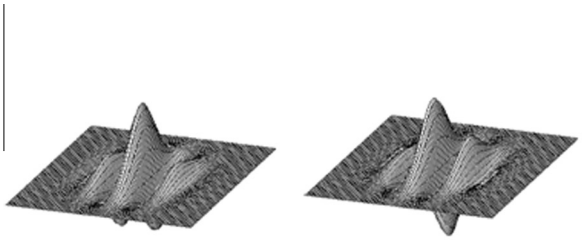
$$G_1 = \begin{cases} x_1 = 1, & \text{if } \left\{\Re\left[\tilde{G}(x, y, \theta, u, \sigma)\right] * I\right\} \geqslant 0 \\ x_1 = 0, & \text{if } \left\{\Re\left[\tilde{G}(x, y, \theta, u, \sigma)\right] * I\right\} < 0 \end{cases} \tag{4}$$

$$G_2 = \begin{cases} x_2 = 1, & \text{if } \left\{\Im\left[\tilde{G}(x, y, \theta, u, \sigma)\right] * I\right\} \geqslant 0 \\ x_2 = 0, & \text{if } \left\{\Im\left[\tilde{G}(x, y, \theta, u, \sigma)\right] * I\right\} < 0 \end{cases} \tag{5}$$

where I is subimage of the expressional face, $\Re$ and $\Im$ are the real and the imaginary parts of each Gabor kernel, $*$ is the convolution operator. With this coding, only the phase information in the facial expressions image is stored in the feature vector of size 256 bytes. The final magnitude response which is used to represent the feature vectors is computed by

$$G = \sqrt{G_1^2 + G_2^2} \tag{6}$$

Fig. 3 shows the magnitude response of a template image.



4. Feature selection

Due to the large size of the Gabor wavelets, it is not practically possible to use all the wavelets as input to our classifier, for fear of misclassification and possible system crash. The AdaBoost feature reduction algorithms have special speed advantage in increasing classification process (Shen & Bai, 2004). Thus we formulated an AdaBoost-based algorithm to select a few deserving portions of the wavelets.

Assuming the extracted Gabor features are represented by a total of $i \in (1, 2, \ldots, N)$ appearance features. Then the image I is represented as $\Phi_i = \{(x_n, y_n)\}_{n=\pm 1}^{N}$ configured by the parameters z, μ, v. The positive sets $\phi^{(+)}$ and the negative sets $\phi^{(-)}$ are denoted by $\phi^{(+)} = \{(x_n, y_n)\}_{n=1}^{N} \subset R^J \times (\pm 1)$ and $\phi^{(-)} = \{(x_n, y_n)\}_{n=-1}^{N} \subset R^J \times (\pm 1)$ respectively, where x_n is the nth data sample containing J features, and y_n is its corresponding class label. To train the vectors $||G||$, which is denoted by $\phi_{(u,v,z)}$ over a distribution D, we simply determined the weights of all the feature vectors $\Phi_i = \{(x_n, y_n)\}_{n=\pm 1}^{N} = \phi^{(+)} + \phi^{(-)}$. This gives us a threshold λ which indicated the decision hyperplane. λ is computed as:

$$\lambda = \frac{\sum_{\forall i \in \phi^{(+)}} D(i).\phi_{(\mu,v,z)}}{||\sum_{\forall i \in \phi^{(+)}} D(i).\phi_{(\mu,v,z)}||} + \frac{\sum_{\forall i \in \phi^{(-)}} D(i).\phi_{(\mu,v,z)}}{||\sum_{\forall i \in \phi^{(-)}} D(i).\phi_{(\mu,v,z)}||} \tag{7}$$

A sample is positive or client if it is located at the positive half of λ (which is the majority decision), otherwise it is a negative or an imposter. The status is reversed if the minority of the positive instances is rather located in the positive half space. Let c be denoted by clients and p be the imposters. For a given training dataset containing both positive and negative samples, where each sample is (S_i, y_i); $y \in \{\pm 1\}$ represents the corresponding class label, the feature selection algorithm is formulated as follows:

- Initialize sample distribution D_0 by weighting every training sample equally such that the initial weight $w_{1,i} = 1/2c, 1/2p$ for $y = 1$ and -1, respectively.
- For the iteration $t = 1, 2, \ldots, T$, where T is the final iteration, do:
  - (i) Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$, where w_t is a probability distribution and N is the total number of features.
  - (ii) Train a weak classifier h_t for feature j, which uses a single feature. The training error ξ_t is estimated with respect to w_t such that:

$$\xi_t = \sum_t w_{t,i} |h_t(x_i) - y_i|^2 \tag{8}$$

  - (iii) Select the hypothesis h_t^1 with the most discriminating information, that is to say, the hypothesis with the least classification error ξ_t^1, on the weighted samples.
  - (iv) Compute the weight ω_t that weights h_t^1 by its classification performance as:

$$\omega_t = \frac{1}{2} \ln\left[\frac{1}{\xi_t^1} - 1\right] \tag{9}$$

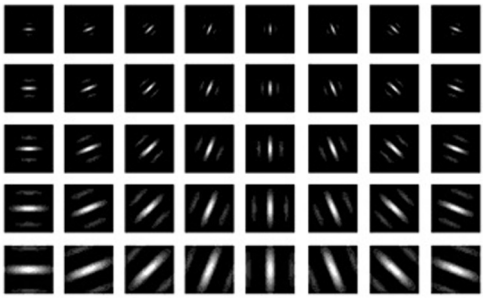


Fig. 2. Gabor filtered image: real (left) and imaginary (middle) parts of Gabor filter in 3D, and the real part of the Gabor kernels (2D) in the spatial and frequency domain.

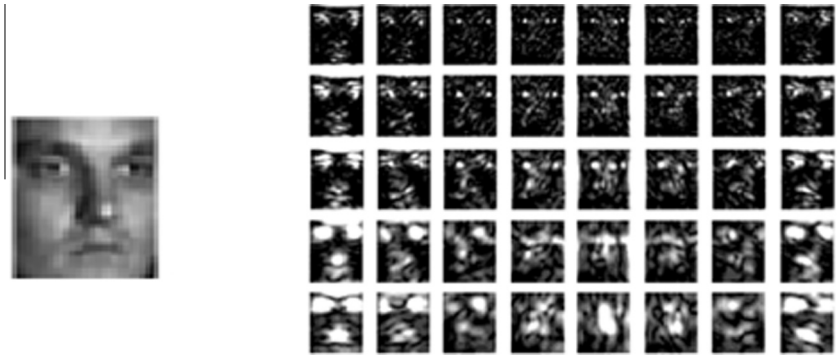


Fig. 3. A Gabor magnitude response of face image: a sample image (left), the magnitude response image of the whole Gabor filter bank of 40 Gabor filters (right).

(v) The weight distribution is then updated and normalized by:

$$w_{t+1,i} \approx w_{t,i}.e^{-\omega_t y_i h_t^1(S_t^1)} \tag{10}$$

- The final feature selection hypothesis $H(S)$ which is a function of the selected features is denoted by:

$$H(S) = \operatorname{sgn}\left[\sum_{t=1}^{T} \omega_t h_t^1(S_t^1)\right] \tag{11}$$

The selected features represent samples of the facial deformation patterns of the expressive face. The datasets which were images from the JAFFE and Yale databases were partitioned into training and testing by leave-one-out cross validation (Wu, Brubaker, Mullin, & Rehg, 2008).

5. Multilayer feed-forward neural network (MFFNN) classifier

The selected features are fed into the constructed neural network to train it to identify the seven universal facial expressions. The architecture is a 3-layer feed-forward neural network and trained by a back-propagation algorithm (Bouzalmat, Belghini, Zarghili, Kharroubi, & Majda, 2011; Londhe & Pawar, 2012). The back propagation algorithm basically replicates its input to its output via a narrow conduit of hidden units. The hidden units extract regularities from the inputs because they are completely connected to the inputs. Every network was trained to give the maximum value of 1 for exact facial expression and 0 for all other expressions. The construction is shown in Fig. 4. The input layer has 7 nodes, each for each facial expression whiles the hidden layer had 49 neurons; each expression for 7 neurons. We chose 7 neurons to compensate for the target output of seven facial expressions. This was the case for the seven prototypical facial expressions, which was validated by the use of the JAFFE facial expression database. Since the experiment was also validated using the Yale database, where four expressions were used there was a slight modification in the construction of the network for this application. Here, the hidden layer neurons were settled at 16 each facial expression dedicated for 4 neurons and 4 neurons in the output layer.

The input vectors of the network represented by $X = [x_1, x_2,..., x_l]^T$. The output layers are denoted by $Y = [y_1, y_2,..., y_k]^T$. The optimization model is formulated as $X: h \to Y$. The output dataset of each layer of the network is denoted by $\left(y_1^j, ..., y_{k-1}^j, y_k^j\right)$, $j = 1, 2, ..., k-1, k$, where $k-1$ corresponds to the total hidden layers and k represents the total output layers. We denote the target datasets and its additive white noise by $(t^1, t^2,..., t^K)$ and $\eta = (e^1, e^2,..., e^K)$, respectively. The variable K represents the total

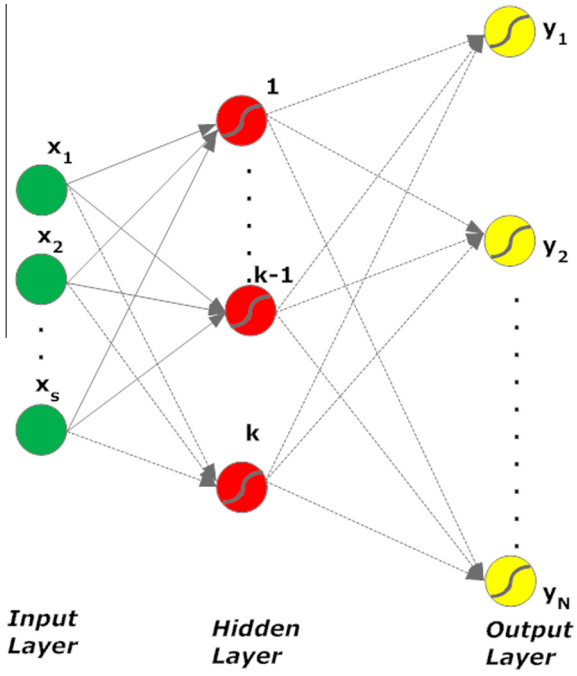


Fig. 4. A 3-layer feed-forward neural network.

patterns of the network. The corresponding vectors of the hidden units are denoted by $V = (v_1, v_2,..., v_{k-1})$.

The sigmoid activation function $h = (h_1, h_2,..., h_k)$ of each layer is $h_1, h_2,..., h_{k-1}$. The weights of the network are updated by $w_1, w_2,..., w_k$. The training epochs are 1000 and the target of error is 0.001. The training algorithm is modeled as:

$$\min_{h_1,h_2,v_1,w_1,w_2} \sum_{j=1}^{K} (t^j - y_2^j)^2 \tag{12}$$

Subject to the constraints

$$\left.\begin{aligned} y_1^j &= h_1(w_1 x^j),\ w_1 \in \Re^{v_1 \times M},\ y_1^j \in \Re^{v_1},\ x^j \in \Re^M \\ y_2^j &= h_2(w_2, y_1^j),\ w_2 \in \Re^{1 \times v_1},\ y_2^j \in \Re^1 \end{aligned}\right\} \tag{13}$$

The process of training involves weight initialization, calculation of the activation unit, adjustment, weight adaptation, and testing for convergence of the network. Assuming v_{ji} represents the weight between the jth hidden unit and ith input unit; and w_{kj} represents the weight between the kth output and the jth hidden unit. The activation unit is then calculated sequentially, starting from the input layer. The activation of hidden and output unit is calculated as:

$$y_j^{(p)} = h_{yj}^{(p)}\left(\sum_{i=1}^{I} v_{ji} z_i - v_{jo}\right) \tag{14}$$

$$o_k^{(p)} = h_{ok}^{(p)}\left(\sum_{j=1}^{J} w_{kj} y_j - w_{ko}\right) \tag{15}$$

where $y_j^{(p)}$ is the activation of the jth hidden unit and $o_k^{(p)}$ is the activation of the kth output unit for the pattern, p. h is a sigmoid function. k is the total number of output units, I is the total number of input units and J is the total number of hidden units. v_{jo} is the weight connected to the bias unit in the hidden layer, $z_o = -1$ and $y_o = -1$. We adjusted the weights, starting at the output units and recursively propagated error signals to the input layer. The detected output $o_k^{(p)}$ is compared with the corresponding target value $t_k^{(p)}$ which is a facial image, over the entire training set using the sigmoid function to express the approximation error in the network's target functions.

$$E^{(p)} = \frac{1}{2}\sum_{k=1}^{K}\left(t_k^{(p)} - o_k^{(p)}\right)^2 \tag{16}$$

The minimization of the error $E^{(p)}$, requires the partial derivative of $E^{(p)}$ with respect to each weight in the network to be computed. The change in weight is proportional to the corresponding derivative.

$$\Delta v_{ji}(t+1) = -\eta\frac{\partial E^{(p)}}{\partial v_{ji}} + \alpha\Delta v_{ji}(t) \tag{17}$$

$$\Delta w_{kj}(t+1) = -\eta\frac{\partial E^{(p)}}{\partial w_{kj}} + \alpha\Delta w_{kj}(t) \tag{18}$$

where, η is the learning rate, normally between 0 and 1, we set it to 0.9. The function α is also set to 0.9. The last term is a function of the previous weight change.

$$\frac{\partial E}{\partial v_{ji}} = \frac{\partial y_j}{\partial v_{ji}}\sum_{k=1}^{K} -(t_k - o_k)o_k(1-o_k)y_j w_{kj} \tag{19}$$

$\frac{\partial y_j}{\partial v_{ji}} = y_j(1-y_j)z_i$. Therefore,

$$\Delta v_{ji} = \eta\sum_{k=1}^{K}(t_k - o_k)o_k(1-o_k)y_j w_{kj} y_j(1-y_j)z_i \tag{20}$$

The weights are updated by,

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t+1) \tag{21}$$

$$v_{ji}(t+1) = v_{ji}(t) + \Delta v_{ji}(t+1) \tag{22}$$

where, t is equal to the current time step. Δv_{ji} and Δw_{kj} are the weight adjustments. We repeated the process once from the equation (14) in order to achieve the desired output.

6. Results and analysis

The facial expression recognition was validated with the JAFFE and Yale facial expression databases.

The JAFFE database contains 213 images of 10 female Japanese persons. Each respondent in the database posed three or four examples of each of the seven facial expression prototypes (happy (*ha*), sad (*sa*), anger (*an*), disgust (*di*), fear (*fe*), surprise (*su*), and neutral (*ne*)). 2 images of each individual from each class of expression are randomly selected for training, leading to a total of 140 images corresponding to 65.7%, the rest was preserved for testing. The trial was performed using tenfold cross-validation to obtain the average recognition rate. In order to create distinct datasets for cross-validation, none of the sets in the training folder appear in any of the remaining folders.

The Yale facial expression database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject. Each subject exhibited one of the six facial expressions; *ha*, *ne*, *sa*, sleepy (*sl*), surprise (*su*) and wink (*wi*). In this database we manually extracted 130 images corresponding to *ha*, *ne*, *sa*, and *su*. The datasets in this database were also partitioned into training and testing by using the same method described for the JAFFE. In due course about 77% of the images were used for training and the remaining, for testing.

We recorded an average recognition rate of 96.83% in JAFFE and 92.22% in Yale on Intel(R) Core(TM) 2 Duo CPU P8400 @ 2.26 GHz (2 CPUs) – 2.3 GHZ and 2.0 GHz RAM computer running on Windows 7 Ultimate 64-bit (6.1, Build 7601) (see Tables 1 and 2 for the confusion matrices).

The best recognitions were detected in *ha*, *su* and *di*, where we obtained almost 100% in the JAFFE database. We saw that facial images with extreme exhibited expressions recorded the best results. Generally, the performance of *ne* was the weakest; about 92.23% in JAFFE and 86.16 in Yale. The results show that the deformations of the muscles around the mouth and the eyes are the most reliable determinants for automatic facial expressions. This accounts why recognition in the neutral face is poor. Thus the increase in these muscle deformations increases the accuracies of automatic recognitions. The execution time for a pixel of size 100×100 is 14.5 ms. Fig. 5 shows the comparative performance of execution time with other neural network classifiers.

Table 2
Confusion matrix of 4-class facial expression recognition on Yale.

	Neural (%)	Sad (%)	Happy (%)	Surprise (%)
Neutral	86.16	9.81	0	4.03
Sad	8.35	86.79	0	4.86
Happy	1.7	0	97.60	0.7
Surprise	0.95	0.72	0	98.33
Average Recognition = 92.22%				

Table 1
Confusion matrix of 7-class facial expression recognition on JAFFE.

	Neutral (%)	Sad (%)	Fear (%)	Anger (%)	Disgust (%)	Happy (%)	Surprise (%)
Neutral	92.23	4.31	1.11	2.35	0	0	0
Sad	3.85	93.9	1.28	0.97	0	0	0
Fear	0	0.95	96.08	0	1.83	0	1.14
Anger	2.8	0	1.1	96.10	0	0	0
Disgust	0	0	0	0.61	99.91	0	0.29
Happy	0.21	0	0.07	0	0	99.72	0
Surprise	0	0	0.1	0	0.03	0	99.87
Average Recognition = 96.83%							

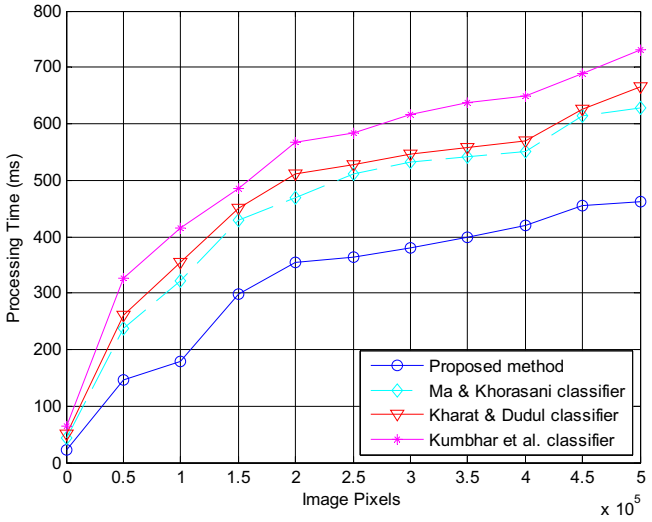


Fig. 5. Comparing execution time of different classifiers.

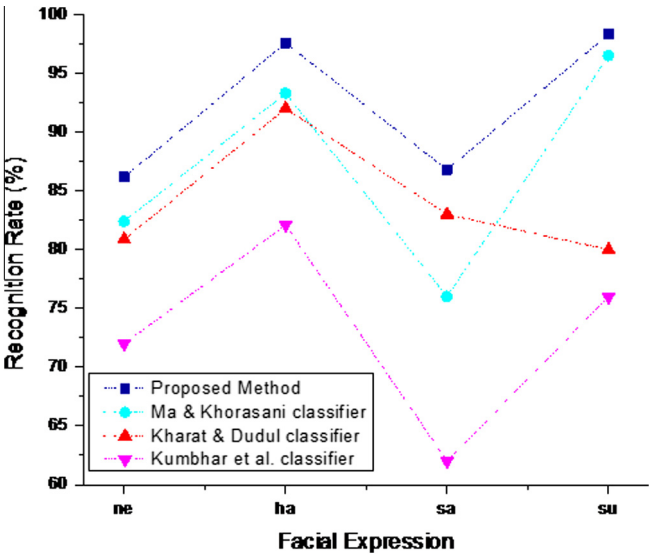


Fig. 7. Comparing recognition rates of different methods in JAFFE database.

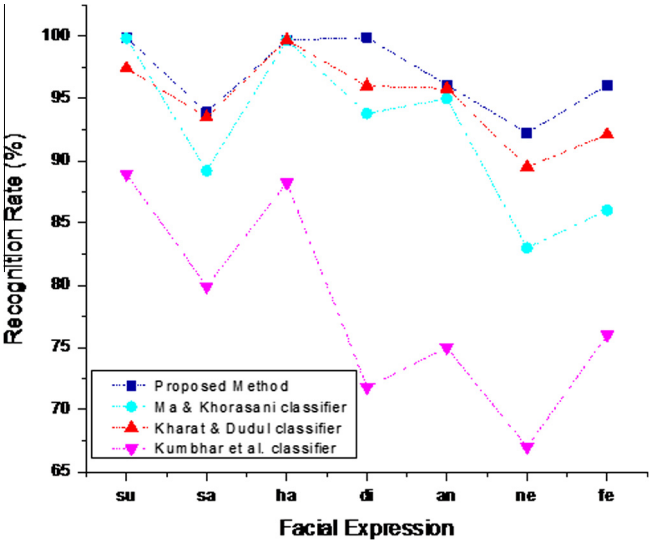


Fig. 6. Comparing recognition rates of different methods in JAFFE database.

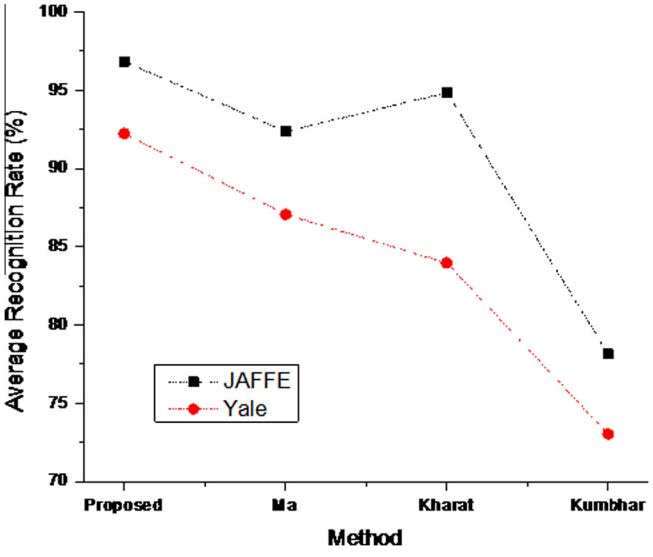


Fig. 8. Comparing the average recognition rates of different methods in JAFFE and Yale databases.

The proposed method was compared with three different classifiers to access its performance in terms of recognition accuracy and execution speed. The system was also tested with some real life images from the World Wide Web. The results indicate that the proposed method is statistically better ($p < 0.05$) both in accuracy and speed. The three methods are described as follows:

Method I (same as Ma and Khorasani (2004) method): The feature detector is by a discrete cosine transform (DCT), pruning technique is used to reduce the input size of the network, the training algorithm is back-propagation procedure, the expression classifier is a feed-forward neural network with one hidden layer.

Method II (same as Kumbhar et al. (2012) method): The feature detector is Gabor method, the feature dimensionality is principal component analysis (PCA) and the expression classifier is a feed-forward neural network.

Method III (same as Kharat and Dudul (2009)): The feature detector is discrete cosine transform, the feature reduction is by principal component analysis (PCA), and the classifier is a feed-forward neural network.

All the classifiers were trained and tested with the same datasets used for the proposed method. Figs. 6 and 7 shows the average recognition rates of various expressions in JAFFE and Yale respectively; the intent is not to compare the performances of the two databases but to investigate for the robustness of the system in diverse databases. Fig. 8 also shows the comparison of the overall average recognition rates of various descriptors in JAFFE and Yale. Fig. 9 shows sample real-time expression recognitions by the system. The average recognition rates are also compared with other methods that employed the same datasets in their experiment to give a general idea of the performance of the proposed method (see Tables 3 and 4 for details). However this does not signify a direct comparison because the experiments were not conducted under the same environment. The results show that the proposed method is very encouraging. Though performance in the Yale facial expression database is reduced as compared to that in the JAFFE facial expression database, it is far better than all the performances in Yale we compared with.

Fig. 9. Sample facial expression recognitions: happy (left), anger (middle), disgust (right).

Table 3
Comparative performance of recognition rates in different methods on JAFFE database.

Author	Classifier/ method	Database	Rate (%)
Lekshmi and Sasikumar (2009)	SVM	JAFFE	86.9
Kumbhar et al. (2012)	Image feature	JAFFE	60–70
Zhi and Ruan (2008)	2D-DLPP	JAFFE	95.91
Zhao, Zhuang, and Xu (2008)	PCA and neural network	JAFFE	93.72
Lee, Huang, and Shih (2010)	RDA	JAFFE	96.7
Proposed method	MFFNN	JAFFE	96.81

Table 4
Comparitive performance of recognition rates in different methods on the Yale database.

Author	Classifier/method	Database	Rate (%)
Lekshmi and Sasikumar (2009)	SVM	Yale	89.5
Franco and Treves (2001)	Neural network	Yale	84.5
Proposed method	MFFNN	Yale	91.52

7. Conclusions

This study employs many advanced techniques to improve the recognition rate and execution time of facial expression recognition system. Face detection was carried out by the application of Viola–Jones descriptor. Detected faces were down-sampled by the Bessel transform. This approach reduced the image dimensions and preserved the perceptual quality of the original image. An AdaBoost based algorithm was formulated to select a few hundreds of Gabor wavelets from the several thousands of the extracted features to reduce the computational cost and to avoid misclassification as well. The selected features were fed into a well-designed multilayer feed-forward neural network classifier. The network is thus trained with sample datasets from both JAFFE and Yale facial expression databases. The remaining datasets from the two databases and some images from the World Wide Web were used to test for the system. The execution time for a pixel of size 100×100 is 14.5 ms; the average recognition rate in JAFFE database is 96.83% and that in Yale is 92.22%. The proposed method is compared with several methods and the performance is outstanding. The results of the study also show that automatic expression recognitions are very accurate in surprise, disgusts and happy; about 100%. Mild expressions like sad, fear and neutral have lower accuracies. However fear can be very accurate when it is at the peak because accuracies in recognitions largely depend on the magnitude of facial deformations around the mouth and eyes. To advance towards 100% efficiency we believe the development of natural databases would be of more help since many artificial databases have many confused scenarios among facial expressions in sad, neutral and mild anger. Again future improvements of recognition accuracies will look at the possibility of increasing the number of hidden neurons to expressions that recorded lower values.

Acknowledgments

This paper is supported by the National Nature Science Foundation of China (Nos. 61272211 and 61170126), the Natural Science Foundation of Jiangsu Province (No. BK2011521), and the Research Foundation for Talented Scholars of Jiangsu University (No. 10JDG065)

References

Aghajanian, J., Warrell, J., Prince, S. J., Li, P., Rohn, J. L., & Baum, B. (2009). Patch-based within-object classification. In: *IEEE 12th International Conference on Computer Vision, 2009* (pp. 1125–1132).

Al Daoud, J. E. (2009). Enhancement of the face recognition using a modified Fourier–Gabor filter. *International Journal of Advanced Software Computer Applications*, 1.

Bouzalmat, A., Belghini, N., Zarghili, A., Kharroubi, J., & Majda, A. (2011). Face recognition using neural network based Fourier Gabor filters & random projection. *International Journal of Computer Science and Security, 5*, 376–386.

Breazeal, C., & Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences, 6*, 481–487.

Bruce, V. (1993). What the human face tells the human mind: Some challenges for the robot–human interface. *Advanced Robotics, 8*, 341–355.

Cemre, Z. (2008) Facial Expression Recognition. MSc Thesis, University of Surrey.

Choi, E., & Lee, C. (2003). Feature extraction based on the Bhattacharyya distance. *Pattern Recognition, 36*, 1703–1709.

Dailey, M. N., & Cottrell, G. W. (1999). PCA = Gabor for expression recognition. UCSD CSE TR CS-629.

Deng, H. B., Jin, L. W., Zhen, L. X., & Huang, J. C. (2005). A new facial expression recognition method based on local Gabor filter bank and PCA plus LDA. *International Journal of Information Technology, 11*, 86–96.

Franco, L., & Treves, A. (2001). A neural network facial expression recognition system using unsupervised local processing. In: *IEEE Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis* (pp. 628–632).

Freund, Y., & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory* (pp. 23–37). Berlin, Heidelberg: Springer.

Ganga, M. P., Prakash, C., Gangashetty, S. V. (2011). Bessel transform for image resizing. In: *IEEE 18th International Conference on Systems, Signals and Image Processing*, (pp. 1–4).

Jones, J. P., & Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology, 58*, 1233–1258.

Kharat, G. U., & Dudul, SV. (2009). *Emotion recognition from facial expression using neural networks*. *Human–Computer Systems Interaction*. Berlin, Heidelberg: Springer (pp. 207–219). Berlin, Heidelberg: Springer.

Kumbhar, M., Jadhav, A., & Patil, M. (2012). Facial expression recognition based on image feature. *International Journal of Computer and Communication Engineering, 1*, 117–119.

Lades, M., Vorbruggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Wurtz, R. P., et al. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers, 42*, 300–311.

Lee, C. C., Huang, S. S., & Shih, C. Y. (2010). Facial affect recognition using regularized discriminant analysis based algorithms. *EURASIP Journal on Advances in Signal Processing*, 1.

Lekshmi, V. P., & Sasikumar, M. (2009). Analysis of facial expression using Gabor and SVM. *International Journal of Recent Trends in Engineering*, 2.

Londhe, R., & Pawar, V. (2012). Facial expression recognition based on Affine Moment Invariants. *International Journal of Computer Science Issues*, 9.

Ma, L., & Khorasani, K. (2004). Facial expression recognition using constructive feed-forward neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34*, 1588–1595.

Maalej, A., Amor, B., Daoudi, M., Srivastava, A., Berretti, S. (2010). Local 3D shape analysis for facial expression recognition. In: *IEEE 20th International Conference on Pattern Recognition* (pp. 4129–4132).

Morik, K., Brockhausen, P., & Joachims, T. (1999). Combining statistical learning with a knowledge-based approach – A case study in intensive care monitoring. In: *ICML* (pp. 268–277).

Munoz, A., Blu, T., & Unser, M. (2001). Least-squares image resizing using finite differences. *IEEE Transactions on Image Processing, 10*, 1365–1378.

Rucklidge, W. J. (1997). Efficient locating objects using Hausdorff distance. *International Journal of Computer Vision, 24*, 251–270.

Satiyan, M., & Nagarajan, R. (2010). Recognition of facial expression using Haar-like feature extraction method. In: *IEEE International Conference on Intelligent and Advanced Systems* (pp. 1–4).

Sha, T., Song, M., Bu, J., Chen, C., & Tao, D. (2011). Feature level analysis for 3D facial expression recognition. *Neurocomputing, 74*, 2135–2141.

Shen, L., & Bai, L. (2004). AdaBoost Gabor feature selection for classification. In: *Proceeding of Image and Vision Computing*, (pp. 77–83), New Zealand.

Soyel, H., & Demirel, H. (2007). *Facial expression recognition using 3D facial feature distances*. *Image Analysis and Recognition*. Berlin, Heidelberg: Springer (pp. 831–838). Berlin, Heidelberg: Springer.

Tai, S. C., & Chung, K. C. (2007). Automatic facial expression recognition system using Neural Networks. In: *TENCON IEEE Region 10 Conference* (pp. 1–4).

Tang, H., & Huang, T. (2008). 3D facial expression recognition based on properties of line segments connecting facial feature points. In: *8th IEEE International Conference on Automatic Face & Gesture Recognition* (pp. 1–6).

Van, P. F. (2008). *Discrete Wavelet Transformations: An Elementary Approach with Applications*. Hoboken, NJ: Wiley-Interscience.

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision, 57*, 137–154.

Wiskott, L., Fellous, J. M., Kruger, N., & vonderMalsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, 775–779.

Wu, J. X., Brubaker, S. C., Mullin, M. D., & Rehg, J. M. (2008). Fast asymmetric learning for cascade face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*, 369–382.

Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition, 37*, 1–19.

Zhang, L., & Tjondronegoro, D. (2009). *Selecting, optimizing, and fusing 'salient' Gabor features for facial expression recognition*. *Neural Information Processing*. Berlin, Heidelberg: Springer (pp. 724–732). Berlin, Heidelberg: Springer.

Zhao, L., Zhuang, G., & Xu, X. (2008). Facial expression recognition based on PCA and NMF. In: *IEEE 7th World Congress on Intelligent Control and Automation* (pp. 6826–6829).

Zhi, R., & Ruan, Q. (2008). Facial expression recognition based on two-dimensional discriminant locality preserving projections. *Neurocomputing, 71*, 1730–1734.