



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Application of Improved AAM and Probabilistic Neural network to Facial Expression Recognition

N. Neggaz, M. Besnassi and A. Benyettou

Department of Computer Science, Signal-Image-Parole Laboratory,
University of Sciences and Technology of Oran-Mohamed Boudiaf, Algeria,
B.P.1505 Oran El Mnaouer 31000 Oran, Algeria

Abstract: Automatic facial expression analysis is an interesting and challenging problem and impacts important applications in many areas such as human-computer interaction. This study discusses the application of improved Active Appearance Model (AAM) based on evolutionary feature extraction in combination with Probabilistic Neural Network (PNN) for recognition of six different facial expressions from still pictures of the human face. Experimental results demonstrate an average expression recognition accuracy of 96% on the JAFFE database, which outperforms the rate of all other reported methods on the same database. The present study, therefore, proves the feasibility of computer vision based on facial expression recognition for practical applications like surveillance and human computer interaction.

Key words: Improved active appearance model, evolutionary algorithms, facial expression recognition, JAFFE, probabilistic neural network

INTRODUCTION

A facial expression is a visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person; it plays a communicative role in interpersonal relations. Facial expressions and other gestures, convey non-verbal communication cues in face-to-face interactions. These cues may also complement speech by helping the listener to elicit the intended meaning of spoken words. With the availability of low cost imaging and computational devices, automatic facial recognition systems now have a potential to be useful in several day-to-day application environments like operator fatigue detection in industries, user mood detection in Human Computer Interaction (HCI) and possibly in identifying suspicious persons in airports, railway stations and other places with higher threat of terrorism attacks. Early work in this area are due mainly to Darwin proposed an evolutionary theory of facial expressions (Ekman *et al.*, 1969).

In the 19th century, Guillaume Duchenne was the first individual to locate the various facial muscles by electrical activation. He was the first to deliver a set of photographs showing the activation of different facial muscles. Paul Ekman, a psychologist, was interested in the mid 60's to expressions and human emotions (Yingli *et al.*, 2001),

proposed a system for automatic analysis of facial expressions by units action recognition. The classification was done by neuron networks. Pantic and Rothkrantz (2004) proposed a system of units action recognition from static pictures of faces (front and profile), the analysis is performed in part by a contouring of the face on the picture profile.

Viola used Adaboost method to solve computer vision problems such as image retrieval and face detection (Viola and Jones, 2001), which can select features in the learning phase using a greedy strategy. AdaBoost method does not perform well in a small sample case (Guo and Dyer, 2003), which is the case in our experiments. Yacoob and Davis (1994) used the inter-frame motion of edges extracted in the area of the mouth, nose, eyes and eyebrows. Bartlett *et al.* (1996), used the combination of optical flow and principal components obtained from image differences. Hoey and Little (2000) approximated the flow of each frame with a low dimensional vector based on a set of orthogonal Zernike polynomials and applied their method to the recognition of facial expressions with hidden Markov models (HMMs), Lyons *et al.* (1998, 1999), Zhang (1999) and Zhang *et al.* (1998) used Gabor wavelet coefficients to code face expressions. In their work, they first extracted a set of geometric facial points and then use multi-scale and

multi-orientation Gabor wavelets filters to extract the Gabor wavelet coefficients at the chosen facial points. Similarly, Wiskott *et al.* (1997) used a labeled graph, based on the Gabor wavelet transform, to represent facial expression images. They performed face recognition through elastic graph matching.

Lyons *et al.* (1999) proposed a method for classifying facial images automatically based on labeled elastic graph matching, 2D Gabor wavelet representation and Linear Discriminant Analysis (LDA). The recognition rate was 92%. Gao *et al.* (2003) used the structural and geometrical features of a user-sketched expression model to match the Line-Edge Map (LEM) descriptor of an input face image. The Active Appearance Model (AAM) proved to be effective for interpreting images of deformable objects (Cootes *et al.*, 1992; Cootes and Taylor, 2004). Davoine *et al.* (2004), used an AAM for facial expression recognition and synthesis, which can normalize the facial expression of a given face and artificially synthesize new expressions on the same face.

DETECTION AND ANALYSIS OF FACIAL FEATURES

Facial feature extraction attempts to find the most appropriate representation of face images for recognition and is for instance the key step in facial expression recognition. For this phase we use the Active Appearance Model (AAM) basic or improved by the Differential Evolution (DE).

Basic active appearance model: AAM (Cootes *et al.*, 1998) uses PCA to encode both shape and texture variation of the training dataset. The shape of an object can be represented by vector *s* and the texture (gray level) by vector *g*. We apply one PCA on shape and another PCA on texture to create the model, given by:

$$\begin{aligned} s_i &= \bar{s} + \varphi_s \times b_s \\ g_i &= \bar{g} + \varphi_g \times b_g \end{aligned} \tag{1}$$

where, *s_i* and *g_i* are shape and texture, \bar{s} and \bar{g} are mean shape and mean texture. φ_s and φ_g are vectors representing variations of orthogonal modes of shape and texture respectively. *b_s* and *b_g* are vectors representing parameters of shape and texture. *i* is the image dataset index. By applying a third PCA on:

$$b = \begin{bmatrix} b_s \\ b_g \end{bmatrix}$$

$$b = \varphi \times c \tag{2}$$

\times is a matrix of *c* eigenvectors obtained by PCA and *c* is the appearance vector.

The modifications of *c* parameters change both shape and texture of the object. Each object is defined by the appearance vector *c* and poses vector *t*:

$$t = [t_x \ t_y \ \theta \ S]^T \tag{3}$$

where, *t_x* and *t_y* are x and y axis translation, θ is the angle of orientation and *S* is the Scale.

AAM learns the linear regression models which gives us the predicted modifications of model parameters δ_c and δ_t :

$$\begin{aligned} \delta_c &= R_c G \\ \delta_t &= R_t G \end{aligned} \tag{4}$$

R_c and *R_t* are the appearances and pose regression matrix respectively. The model search is driven by the residual *G* of the search image and model reconstruction.

Required memory: column number of regression matrix is equal to number of model pixels.

Row number is product of number of experiment *q* with the number of parameter to be optimized: 4 for *R_t* (Eq. 3) and as much as parameter as eigenvector retained in matrix φ (Eq. 2) for *R_c*.

The search algorithm in new image is as follow:

- Generate texture *g_m* and form *s* according to *c* parameter (initially equal to 0)
- Calculate *g_i*, the image texture which is in the form *s*.
- Evaluate $\delta_{g_0} = g_i - g_m$ and $E_0 = |\delta_{g_0}|^2$
- Predict the modification $\delta_{c_0} = R_c \times \delta_{g_0}$ and $\delta_{t_0} = R_t \times \delta_{g_0}$, which has to be given to the model
- Find the first attenuation coefficient *k* (among [1 0.5 0.25]) generate an error $E_i < E_0$

With $E_i = |\delta_{g_1}|^2 = |g_m - g_{m1}|$, *g_m* is the texture created by $c_1 = c - k \times \delta_{c_0}$ and *g_{m1}* is the texture of image being in the form *s_{m1}* (form given by *c₁*).

- If error *E_i* is not stable, the difference *E_i-E_{i-1}* is higher than a threshold ξ defined previously, return to step 1 and replace *c* by *c₁*

When the convergence is reached, the searching form and texture face is generated with model given by *g_m* and *s* represented using *c₁*. The number of iterations required is function of error *E_i* stabilization.

Differential evolution: Differential algorithm is a new heuristic approach which has three basic advantages: It finds the true global minimum of classical benchmarks such as Sphere, Rosenbrock's and Rastrigin's functions optimization task regardless of the initial parameter values; it has a fast convergence rate and uses few control parameters (Karaboga and Akay, 2009). DE algorithm is a population based like the genetic algorithms that use similar operators, crossover, mutation and selection but they are some differences, for example at the level of selection, operator GA uses stochastic selection (RWS) and DE uses deterministic selection (Storn and Price, 1997). Furthermore, the structure of mutation in DE algorithm is different in comparison to genetic algorithm and the crossover at the level of genetic algorithm allows producing two new individuals.

The algorithm DE supports the mutation operator that is described below (Price *et al.*, 2005):

- **DE/rand/1:** This notation specifies that the vector to be perturbed is randomly chosen and that the perturbation consists of one difference vector

For each individual $x_{i,G}$ $i = 0, 1, \dots, NP-1$ a mutated individual is generated according to:

$$v_{i,G+1} = x_{r3,G} + F(x_{r1,G} - x_{r2,G}) \quad (5)$$

- **DE/best/1:** Unlike the previous strategy the individual of the next generation is generated by the best member of the population using the following formula:

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} - x_{r2,G}) \quad (6)$$

- **DE/best/2:** This strategy uses two difference vectors as a perturbation:

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} - x_{r2,G} + x_{r3,G} - x_{r4,G}) \quad (7)$$

- **DE/rand to best/1:** This strategy places the perturbation at a location between the randomly chosen population member and the best population member:

$$v_{i,G+1} = x_{i,G} + F(x_{r1,G} - x_{r2,G}) + F(x_{r1,G} - x_{r2,G}) \quad (8)$$

- **DE/rand/2:** This strategy is generated by:

$$v_{i,G+1} = x_{r5,G} + F(x_{r1,G} - x_{r2,G} + x_{r3,G} - x_{r4,G}) \quad (9)$$

Where:

$x_{i,G}$:ith : Individual of the current generation G
 x : Set of population
 F : Mutation constant $\epsilon \in [0,2]$

$x_{best,G}$, $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$, $x_{r4,G}$, $x_{r5,G}$ are the randomly selected population in the current generation. After mutation, crossover is applied to the individuals by the following rule:

$$U_{i,G+1} = x_{i,G} \cdot (1 - P_c) + v_{i,G+1} \cdot P_c \quad (10)$$

Where, p_c is a crossover constant.

All the solutions in the population can be selected as parents without depending on their fitness value and then the produced offspring after the mutation and crossover operations is evaluated. After that, the performance of the child vector and its parent are compared with the selection of the best one (Neggaz and Benyettou, 2009).

Finally, the parent is retained in the population if it still represents the best element.

The general pseudo code of DE is summarized in Algo 1 (Lampinen and Zelinka, 2000)

Algo 2: Pseudo-code for a general DE

```

Initialization
Evaluation
Repeat
Mutation
Recombination
Evaluation
Selection
Until (Termination criteria are met)
    
```

Improved active appearance model: We propose to adapt the Differential Evolution to optimize the AAM parameters in the segmentation phase. When we have to align an object (detecting characteristics points and texture) with AAM, we must find a vector v that minimizes the sum of quadratic errors E with:

$$E = \sum_1^M e_i^2 \quad (11)$$

$$v = \begin{bmatrix} c \\ t \end{bmatrix} \quad (12)$$

where, c is the appearance vector, M is the number of pixels of the model and e_i the error in the pixel i .

Differential evolution neither needs additional required memory space necessary to store the model nor information on the directions to a priori to minimize the error E . The size of the two matrices (of appearance R_c and of pose R_t) is equal to $M \times (d_c + d_t)$ Bytes, M being the

number of pixels contained in texture of the model (dc and dt are the c (Eq. 2) and t Size (Eq. 3)).

FACIAL EXPRESSIONS RECOGNITION

In this section we propose classification method: probabilistic neural networks PNN. Several architectures have been proposed to evaluate the performance of facial expression recognition using connectionist models, varying the input parameters (shape, texture, texture + shape).

The overall objective of our study is to construct a system of automatic recognition of facial expressions (joy, sadness, anger, disgust, fear, surprise) the JAFFE database (Press *et al.*, 1988).

Probabilistic neural networks

Mechanism of PNN: Neural networks are widely used in pattern classification since they do not need any information about the probability distribution and the a priori probabilities of different classes.

PNNs are basically pattern classifiers. It combines the well known Bayes decision strategy with the Parzen non-parametric estimator of the probability density functions (pdf) of different classes. PNNs have gained interest because they yield a probabilistic output and are easy to implement.

Taking a two categories situation as an example. We should decide the known state of nature θ to be either θ_A or θ_B . Suppose a set of measurements is obtained as p-dimensional vector $x = [x_1, \dots, x_p]$, the Bayes decision rule becomes:

$$d(x) = \begin{cases} \theta_A & \text{if } h_A I_A F_A(x) > h_B I_B f_B(x) \\ \theta_B & \text{if } h_A I_A F_A(x) < h_B I_B f_B(x) \end{cases} \quad (13)$$

Here, $f_A(x)$ and $f_B(x)$ are the PDF for categories A and B, respectively. I_A is the loss function associated with the wrong decision $d(x) = \theta_B$ when $\theta = \theta_A$, I_B is the loss function associated with the wrong decision $d(x) = \theta_A$ when $\theta = \theta_B$ and the losses associated with correct decisions are taken to be zero. h_A and h_B are the a priori probability of occurrence of patterns from category A and B, respectively.

In a simple case that assumes the loss function and a priori probability are equal, the Bayes rule classifies an input pattern to the class with a higher PDF. Therefore, the accuracy of the decision boundaries depends on what the underlying PDFs are estimated. Parzen's results can be extended to estimate in the special case where the

multivariate kernel is a product of univariate kernels. In the particular case of the Gaussian kernel, the multivariate estimates can be expressed as:

$$f_A(X) = \frac{1}{(2\pi)^{p/2} \sigma^p} \frac{1}{m} \sum_{i=1}^m \exp \left[-\frac{(x - x_{Ai})^T (x - x_{Ai})}{2\sigma^2} \right] \quad (14)$$

Here, m is the number of training vectors in category A, p is the dimensionality of the training vectors, x_{Ai} is the ith training vector for category A and σ is the smoothing parameter. It should be noted that $f_A(x)$ is the sum of small multivariate Gaussian distributions centered at each training sample, but the sum is not limited to being Gaussian (Specht, 1990).

PNN structure: Figure 1 shows the outline of a PNN. When an input is presented, the first layer computes distances from the input vector to the Input Weights (IW) and produces a vector whose elements indicate how close the input is to the IW. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a compet transfer function on the output of the second layer picks the maximum of these probabilities and produces a 1 for that class and a 0 for other classes (Gill and Sohal, 2005). Figure 1 shows a PNN (R, Q and K represent the number of elements in the input vector, the input/target pairs and classes of the input data, respectively. IW and LW represent the input weight and the layer weight, respectively.

The mathematical expression of the PNN can be expressed as:

$$a = \text{radbas} (\|IW-x\|b) \quad (15)$$

$$y = \text{compet} (LW\alpha) \quad (16)$$

In this study, the radbas is selected as:

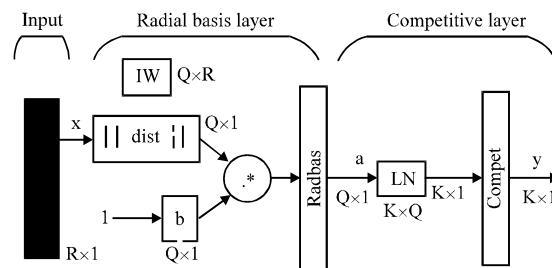


Fig. 1: Architecture of a Probabilistic Neural Network (PNN)

$$\text{radbas}(n) = \exp(-nr^2) \quad (17)$$

The compet function is defined as:

$$\text{Compet}(n) = e_i = [0 \ 0 \ \dots \ 0^i \ 0 \ \dots \ 0], n(I) = \max(n) \quad (18)$$

This type of setting can produce a network with zero errors on training vectors and obviously it does not need any training.

FEATURE REDUCTION

Excessive features increase computation times and storage memory. Furthermore, they sometimes make classification more complicated, which is called the curse of dimensionality. It is required to reduce the number of features.

Principal Component Analysis (PCA) is an efficient tool to reduce the dimension of a data set consisting of a large number of interrelated variables while retaining most of the variations. It is achieved by transforming the data set to a new set of ordered variables. This technique has three effects: it orthogonalizes the components of the input vectors so that uncorrelated with each other, it orders the resulting orthogonal components so that those with the largest variation come first and eliminates those components contributing the least to the variation in the data set.

It should be noted that the input vectors should be normalized to have zero mean and unity variance before performing PCA, which is shown in Fig. 2.

The normalization is a standard procedure. Details about PCA could be seen in Ref (Zhang *et al.*, 2009).

DATABASE

The database contains 213 picture of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models (Jaffe). The database has been planned and assembled by Miyuki Kamachi, Michael Lyons and Jiro GYOBA. The photos were taken in the department of psychology at the University of Kyushu (Lyons *et al.*, 1998, 1999). The

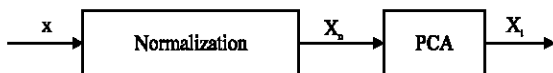


Fig. 2: Using normalization before PCA

learning base contains 110 pictures and test database contains 103 images with 6 facial expressions in different poses.

EXPERIMENTAL RESULTS

Results of the AAM: Figure 3 shows an original picture to the left, its corresponding shape in the middle and texture right.

Figure 4 shows the iterative process of the AAM.

Architecture and parameters: Our network of facial expression recognition is composed of two layers. An input layer of 84 neurons (84 items related to the number of principal components obtained by PCA on the results obtained by the active appearance model), an output layer of 6 neurons representing 6 expressions (joy, sadness, anger, disgust, fear, surprise). The input parameters are determined by the active appearance model and its variants. These parameters are represented by:

- Face texture: a square matrix [340 340] pixels
- Face shape: vector 68 points
- Texture face + face shape: concatenation between texture and shape vector

For PNN, we used the results of active appearance model:

- Basic Active Appearance Model
- Active appearance Model improved by Differential Evolution

Table 1 shows recognition rates for both types of neural networks.

According to results shown in Table 1, we note that the input architecture PNN, the texture of the face, gave good results compared to the other two architectures. And the Differential Evolution method has great advantage over the active appearance basic model.

The memory has been reduced and we obtained the shape and texture of faces in different poses, without the Matrix experience.

Table 1: The input features recognition rates vs. classification approaches

Classification approaches/ input features	Recognition rate (%)		
	Shape	Texture	Texture and shape
PNN/ basic AAM	72	96	96
PNN/ improved AAM using DE	84	96	94

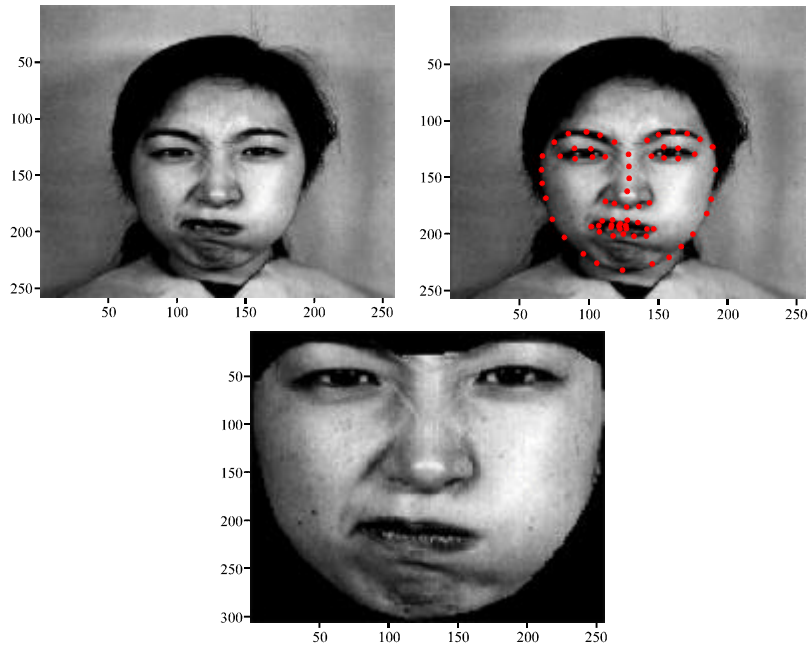


Fig. 3: Original left picture, its corresponding shape in the middle and right texture

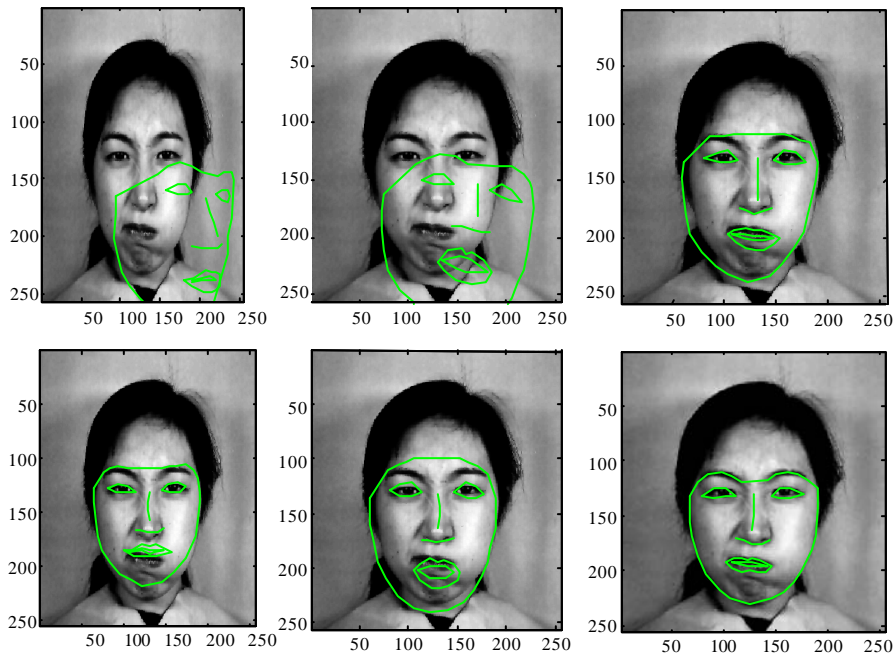


Fig. 4: Iterative process on the face of a sad expression

Table 2 shows the confusion matrix of the facial expression recognition when the PNN classifier is used. It shows that most of the happy, anger, disgust and

sadness expressions were classified properly but some of the fear and surprise expression are misclassified as the disgust / fear expression due to their similarity.

Table 2: Confusion matrix using PNN based on improved AAM using DE

State	Happy	Anger	Disgust	Fear	Sadness	Surprised
Happy	100%	0	0	0	0	0
Anger	0	100%	0	0	0	0
Disgust	0	0	100%	0	0	0
Fear	0	0	2.5%	87.5%	0	0
Sadness	0	0	0	0	100%	0
Surprised	0	0	0	8.02%	0	81.82%

Table 3: Comparison of recognition rate on JAFFE database

Classifier	Feature (%)			
	Contour	Image	Gabor wavelet coefficients	Texture
MLP	90.92	74.76	90.1	
KNN	83.38	79.52	-	
RBF	-	-	89.11	
SOM	30.02			
Our approach				96

Comparative study: Table 3 compares the facial expression recognition performances using different types of classifiers: the k-NN, MLP, RBF and SOM (Zhang, 1999; Praseeda and Sasikumar, 2008; Gu *et al.*, 2010). Table 3 shows that PNN based on texture is known as one of the best classifiers, with a recognition rate of 96%. The MLP classifier based on contour outperforms other classifiers like KNN and SOM, with a recognition rate of 90.29%.

CONCLUSION

The aim of this study is a robust classification of facial expressions. Various inputs are used the PNN network by results of the AAM (the shape of the face, the face texture, texture + shape of the face). After several experiments we found that the texture of the face improves the classification results. For the AAM we used the differential evolution optimization method to limit the memory space and obtain forms and textures of realistic face in different images.

The results are encouraging enough to explore real-life applications of facial expression recognition in fields like surveillance and user mood evaluation. Like future work, we are interested in applying artificial bee algorithm with probabilistic neural network for dynamic facial expression recognition.

REFERENCES

Bartlett, M., P. Viola, T. Sejnowski, L. Larsen, J. Hager and P. Ekman, 1996. Classifying Facial Action. In: *Advances in Neural Information Processing Systems*, Touretzky, D., M. Mozer, M. Hasselmo (Eds.). Vol. 8, MIT Press, Cambridge, Massachusetts, pp: 823-829.

Cootes, T. and C. Taylor, 2004. *Statistical Models of Appearance for Computer Vision*. Imaging Science and Biomedical Engineering, USA.

Cootes, T.F., D.H. Cooper, C.J. Taylor and J. Graham, 1992. A trainable method of parametric shape description. *Image Vision Comput.*, 10: 289-294.

Cootes, T.F., G.J. Edwards and C.J. Taylor, 1998. Active appearance models. *Proceedings of the European Conference on Computer Vision, 1998, Active Appearance Models*, pp: 484-498.

Davoine, F., B. Abboud and M. Dang, 2004. Analyse de visages et d'expressions faciales par modèle actif d'apparence. *Traitement du Signal*, 21: 179-193.

Ekman, P., E.S. Richard and V.F. Wallace, 1969. Pan-cultural elements in facial displays of emotion. *Science*, 164: 86-88.

Gao, Y., M.K.H. Leung, S.C. Hui and M.W. Tananda, 2003. Facial expression recognition from line-based caricatures. *IEEE Trans. Syst. Man Cybernet-Part A Syst. Human*, 33: 407-412.

Gill, G.S. and J.S. Sohal, 2005. Battlefield decision making: A neural network approach. *J. Theor. Applied Inform. Technol.*, 4: 697-699.

Gu, W.F., Y.V. Venkatesh and C. Xiang, 2010. A novel application of self-organizing network for facial expression recognition from radial encoded contours. *Soft Comput.*, 14: 113-122.

Guo, G.D. and C.R. Dyer, 2003. Simultaneous feature selection and classifier training via linear programming: A case study for face expression recognition. *IEEE Conf. Comput. Vision Pattern Recogn.*, 1: 346-352.

Hoey, J. and J.J. Little, 2000. Representation and recognition of complex human motion. *IEEE Conf. Comput. Vision Pattern Recogn.*, 1: 752-759.

Karaboga, D. and B. Akay, 2009. A comparative study of artificial bee colony algorithm. *Applied Mathe. Comput.*, 214: 108-132.

Lampinen, J. and Y. Zelinka, 2000. On stagnation of the differential evolution algorithm. *Proceedings of the MENDEL Conference, June 7-9, Brno, Czech Republic*, pp: 76-83.

Lyons, M., S. Akamatsu, M. Kamachi and J. Gyoba, 1998. Coding facial expressions with Gabor wavelets. *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, April 14-16, Nara, Japan, pp: 200-205.

Lyons, M.J., J. Budynek and S. Akamatsu, 1999. Automatic classification of single facial images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21: 1357-1362.

Neggaz, N. and A. Benyettou, 2009. Hybrid models based on biological approaches for speech recognition. *Artificial Intel. Rev.*, 32: 45-57.

- Pantic, M. and L.J.M. Rothkrantz, 2004. Facial action recognition for facial expression analysis from static face images. *IEEE Trans. Syst. Man Cybernet. Part B*, 34: 3-3.
- Praseeda, L.V. and M. Sasikumar, 2008. A neural network based facial expression analysis using gabor wavelets. *World Acad. Sci. Eng. Technol.*, 42: 563-568.
- Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vettering, 1988. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK.
- Price, K.V., R.M. Storn and J.A. Lampinen, 2005. *Differential Evolution: A Practical Approach to Global Optimization*. 1st Edn., Springer-Verlag, New York, ISBN-10: 3540209506, pp: 538.
- Specht, D.F., 1990. Probabilistic neural networks. *Neural Network*, 3: 109-118.
- Storn, R. and K. Price, 1997. Differential evolution A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimiz.*, 11: 341-359.
- Viola, P. and M. Jones, 2001. Rapid object detection using a boosted cascade of simple feature. *IEEE Conf. Comput. Vision Pattern Recogn.*, 1: 511-518.
- Wiskott, L., J.M. Fellous, N. Kruger and M.C. Vonder, 1997. Face recognition by bunch graph matching. *IEEE Trans. Pattern Anal. Machine Intell.*, 19: 775-779.
- Yacoob, Y. and L. Davis, 1994. Recognizing facial expressions by spatiotemporal analysis. *Proc. Int. Conf. Pattern Recog.*, 1: 747-749.
- Yingli, T., K. Takeo and F.C. Jeffrey, 2001. Recognizing action units for facial expressions analysis. *IEEE Trans. Pattern Analysis Machine Intell.*, 23: 97-115.
- Zhang, Y., L. Wu, N. Neggaz, S. Wang and G. Wei, 2009. Remote-sensing image classification based on an improved probabilistic neural network. *Neural Networks Sensors*, 9: 7516-7539.
- Zhang, Z., M. Lyons, M. Schuster and S. Akamatsu, 1998. Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron. *Proceedings of the International Conference Automatic Face and Gesture Recognition*, IEEE Computer Society Washington, DC., USA., April 14-16, pp: 454-459.
- Zhang, Z., 1999. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multi-layer perceptron. *J. Pattern Recogn. Artif. Intell.*, 13: 893-911.