Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 04, April -2018

Neural Network based Recognition of an Emotion using PCA of Leg Postures of a Human Being

Dr. Mukesh Patel [1]

[1]*Department of Mathematics, Uka Tarsadia University, Bardoli*

Abstract — *This paper presents the recognition of an emotion of a human being based on Leg Postures created by an individual human. Various leg postures related to an emotion are generated by different positions of legs viz. Sitting, Standing and Walking. The image set of all these postures is prepared and analyzed by certain mathematical techniques and image processing tools. Here, the images are classified into basic seven emotions Neutral, Happy, Sad, Fear, Anger, Surprise, and Disgust by Neural Network (NN) which is applied on the eigen features of these images, obtained by Principle Component Analysis (PCA).*

Keywords- *Emotion, Leg Postures, PCA, Neural Network, Multi-Layer Feed Forward, Back Propagation.*

I. INTRODUCTION

In the different situations a human formed different posture which is consciously or subconsciously commanded by the neurons regarding the emotion felt by the human being. It is an immediate physical reflection of a hidden virtual emotion by human body [1]. Actually, in such situation whole body reacts accordingly using its different flexible parts with which it forms a certain posture [1], [2]. The flexible parts of a human body includes Head, Hands and Legs from which we are considering only the leg postures of a human body during a particular emotion[2],[3]. Legs form multiple postures in different positions viz. Sitting Position, Standing Position and Walking Condition. In all the situations, how the legs are positioned, whether they are crossed, they are folded or remained straight, how much distance is kept etc. are the mention the inherent nature of behavior or an attitude of human being [4]. Ultimately, it is a physical reaction against the emotion felt by a human. All together or individually reflects the emotion by the meaningful expressions [5]. These, leg expressions related to an emotion are captured and stored into the database which further be classified into two categories Training set and Testing set. Mathematical methods and Image processing techniques are involved to process the data set for classification over basic seven emotions mentioned earlier. The analysis process for recognition of an emotion through leg postures is described in the following section in details.

II. MATHEMATICAL SYSTEM TO RECOGNIZE EMOTION BASED ON LEG POSTURE

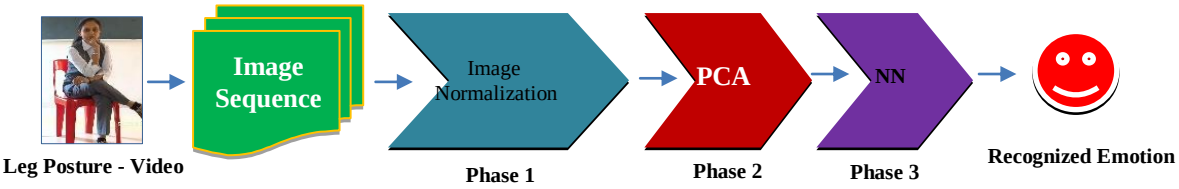


Figure 1. Mathematical System to Recognize Emotion based on Leg Posture

Figure 1 shows the Mathematical system to recognize an emotion using leg gesture of an individual. The whole system works in three different phases (i) Image Normalization (ii) Feature Extraction and (iii) Classification. Prior, a video of leg gesture is converted into the sequence of image frames and the significantly vary images are entered into the system. The mathematical system will follow the algorithm for classification is described as follows;

1.1 Step 1 - Image Normalization [6] [10]

Step 1.1 : Load video (.avi RGB Format) of leg posture related to an emotion of an individual into the system, processing under MATLAB interface.

Video_Happy = VideoReader (Leg Posture-Video);

Step 1.2 : Capture sequence of frames (.jpeg) from loaded video,

Consider only significantly variant images with desired specific region.

Image_frame = imcrop (Video_Happy, [Xmin Ymin Width Height]);

Step1.3 : *Gamma Correction:* Balance Darkness and Lightness of RGB input image $X_{New} = X^{Gamma}$

Where, Gamma $= \frac{dlog(I_{out})}{dlog(I_{in})}$, represents the slope of Gamma curve

I_{in} = Image input and I_{out} = Image output.

Image_Gamma = vision.GammaCorrector(Image_frame, *Value*)

Step 1.4 : *Dimension Reduction:* Reduce RGB Image Pixel Dimensions by converting RGB to Gray Scale image method.

$$I_{gray} = 0.2989 * R + 0.5870 * G + 0.01140 * B$$

Iamge_Gray = rgb2gray(Image_Gamma);

Step 1.5 : *Histogram Equalization:* Adjust contrast of gray image using Histogram Equalization method. Histogram equalization transformation:

$$s_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n} ,$$

For k = 1, 2, -----, L, where s_k is the intensity value in the output image corresponding to value r_k in the input image. L is total possible intensity levels in the range [0, G],

n_k is the number of pixels in the image whose intensity level is r_k,

r_k is the k^{th} intensity level in the interval $[0, G]$,

$p_r(r_j), j = 1, 2, \dots, L$, denote the histogram associated with the intensity levels.

Image_Histogram = histeq(Image_Gray, hgram);

Where, (hgram) *is intensity values in the appropriate range: [0, 1] for images of class double, [0, 255] for images of class uint8, and [0, 65535] for images of class uint16.*

Step 1.6 : *Image Resizing:* Resize the Histogram Equalized image to convert in same dimension of image matrix.

The Image matrix dimension

[Height , Width] = [numrows , numcols] = [30 , 27]

Image_Resize = imresize(Image_Histogram,[numrows numcols]);

1.2 Step 2 - Feature Extraction: Principle Component Analysis (PCA) [7], [8], [10]

Step 2.1: Restore M normalized expression images into column vectors α_i of image matrix A.
A_store = imread(Image_Resize);

Step 2.2: Compute the mean μ of M column vector α_i's.

$$\mu = \frac{1}{M}\sum_{n=1}^{M} \alpha_n$$

A_store = mean(A_store');

Step 2.3: Find normalized vectors σ_i by taking deviation of column vectors from their mean vector.

$$\sigma_i = \alpha_i - \mu$$

Normal_vector = A_store -(ones(P,1)* A_store)';

Step 2.4: Find the M orthonormal vectors δ_n , such that k^{th} vector δ_k is

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M} \left(\delta_k{}^T \sigma_n\right)^2$$

Step 2.5: Obtain the covariance matrix $C = AA^T$, where $A = [\sigma_1, \sigma_2, \sigma_3, \dots . \sigma_M]$, Such that

$$\lambda_k \delta_k = C \delta_k$$

Where, vectors δ_k and scalars$\lambda_k$are the eigenvectors and Eigen values, respectively, of the covariance matrix $C = AA^T$.

Step 2.6: Define a matrix $C' = A^T A$, to reduce the complexity of the computation,
Such that

$$AA^T AV_i = \eta_i AV_i$$

Here, AV_i is the eigenvector of $C = AA^T$.

Step 2.7: Derive p (p < nm) eigenvectors (V_i) of the matrix C′

[Vi_eigenvector,Vi_eigenvalue] = eig (1/(P-1)*(A_store'* A_store));

Step 2.8: Derive the p eigenvectors of covariance matrix C by multiplying A to V_i.
C_eigenvector = A_store * Vi_eigenvector;

Step 2.9: Choose number of eigenvectors which are significantly variant by arranging the eigenvectors in descending order of their respective eigen values.
Descending ordered Eigen values:
evalues = fliplr(Vi_eigenvalue);

howmany = m (number of eigenvectors)

Descending ordered Eigen vectors:
Selected_C_eigenvector = C_eigenvector (:,1:howmany);

Step 2.10: Transform expression image into feature space by projection operation

$$x_i = \delta_i{}^T(\alpha - \mu),\ i = 1, 2, 3, \text{------- p.}$$

Where x_i denotes the projection of the expression image α projected onto the i^{th} Eigen expression image component δ_i.
Projection = C_eigenvector'* Normal_vector;

Step 2.11: Find projection vector, $X^T = [x_1, x_2, \dots, x_p]$.
Treat the projection vector *X* as the Eigen Features of input normalized expression images.
Eigen_Fetures = Projection';

1.3 Step 3 – Classification – Neural Network [9]

Step 3.1: *Neural Network Architecture: Multi-Layer Feed Forward*

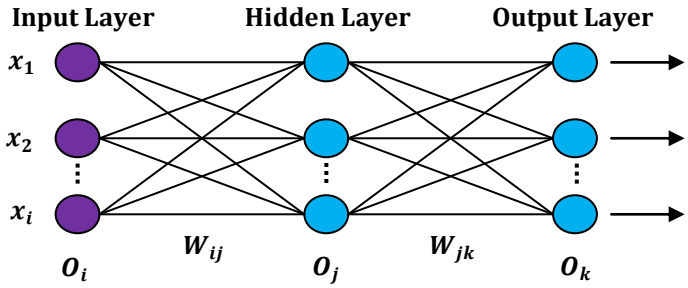


Figure 1. Fully Connected, Feed-Forward MLP network

Neural Network with Multi-layer Perceptron (MLP) has three different layers, Input, Hidden and Output Layer. A training sample, X = $x_1, x_2, \dots, x_i$ is enter at input layer, weighted connections Wij exits denotes the weight form a unit j in one layer to a unit i in the previous layer and k nodes are available at Output Layer. . Here, output layer is trained to respond +1 for matching and 0 for others. But, in real practice outputs are vary between 0 and +1.

Step 3.2: *Transfer function:* Log- sigmoid function is ideal for MLPs, the output range of this function is set to [0, 1] as NN is respond to 1 for classification and 0 for not. But, in practice it is always vary in between them.

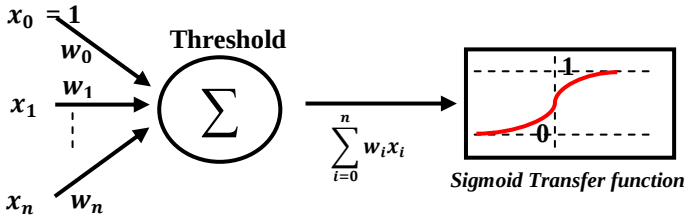


Figure 2. Sigmoid Function for MLPs

Step 3.3: *Back Propagation Algorithm:* [12]

Step – 3.1.1 : Initialize all network weights to small random numbers
Until the termination condition do:
Step – 3.1.2 : Propagate the input forward to the network and compute the observed outputs.
Step – 3.1.3 : Propagate the errors backward as follows:
(i) For each network output unit k, calculate its error term

$$\delta_k = O_k(1 - O_k)(t_k - O_k),$$

Where t_k is target value and O_k is output value.

(ii) For each hidden unit calculate its error term

$$\delta_h = O_h(1 - O_h) \sum_{k \atop outputs} W_{kh}\delta_k$$

Step – 3.1.4 : Update each weight

$$w_{ji} = w_{ji} + \Delta w_{ji}, \text{ where } \Delta w_{ji} = -\eta . \delta_j x_{ji}$$

(Here, 'ji' means from unit 'i' to 'j')

Step 3.4: *Set Target Matrix for input Eigen Images :*

P11	P12	---	P1n
P21	P22	---	P2n
---	---	---	---
Pm1	Pm2	---	Pmn

Expression Eigen Image

Neutral	Happy	Angry	Sad	Disgust	Fear	Surprise
0.9	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.9	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.9	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.9	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.9	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.9

Target Matrix

Figure 3. Target Matrix

Every eigen image related to an emotion is targeted to a matrix is seen in Figure 3. Where, 0.9 respond to recognition and 0.1 respond to no recognition.

Step 3.5: Neural Network parameters and other important values; [10], [11], [12]

Step 3.5.1: *Error Surface* : The standard gradient descent error of Backpropagation is Sum of square of error;

$$E(\overline{w}) = \sum_{n\ pattern} \sum_{k\ outputs} (t_{n,k} - o_{n,k})^2$$

Where, E is a function of the network's weight vector, t is targeted value and o is an output value.

Step 3.5.2: *Size of Hidden Layer:* Usually, for pattern recognition one hidden layer is sufficient for classification.

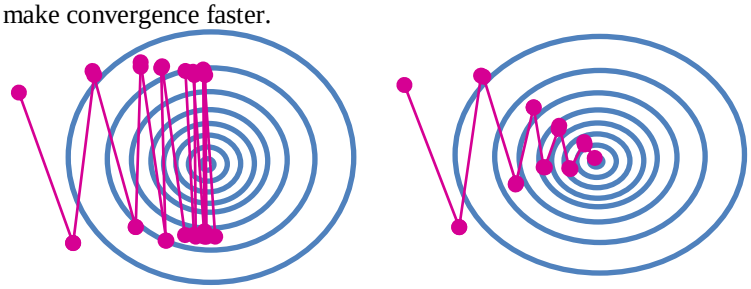
Step 3.5.3: *Number of Neurons at hidden Layer:* It is to be decided based on trial and error, also it is dependent on the values of other parameter, still an issue of research.

Step 3.5.4: *Learning rate*: In backpropagation algorithm the weight is updated with rule

$$\Delta w_{ji} = -\eta . \delta_j x_{ji}$$

High learning rate oscillates the algorithm and too small learning rate takes too long convergence. Thus, it is to be set by experiencing the training algorithm.

Step 3.5.5: *Momentum Term*: By adding this term to the formula of the final step in Table II, the update rule will be: $\Delta W_{ji} = \eta \delta_j x_{ji} + \alpha \Delta W_{ji}(n-1)$. Therefore, the update in iteration is affected by the update in n^{th} iteration multiplied by a factor 'α', called momentum. It helps to make convergence faster.



Training with large Learning Rate

Training with large Learning Rate Plus Momentum

Figure 4. Convergence by Learning Rate and/or Momentum

Step 3.5.6: *Input Standardization and Weights Initialization*: In the MLP set the weight 0 at hidden layer nodes and random at output layer nodes which gives better appearance of leg posture than the other settings.

Step 3.5.7: *Number of input eigen images:* For the classification in NN such number of eigen images

(vector) are chosen which are significantly variant in the nature that represent all the characteristics of data set while, similar eigen images are not taken into consideration.

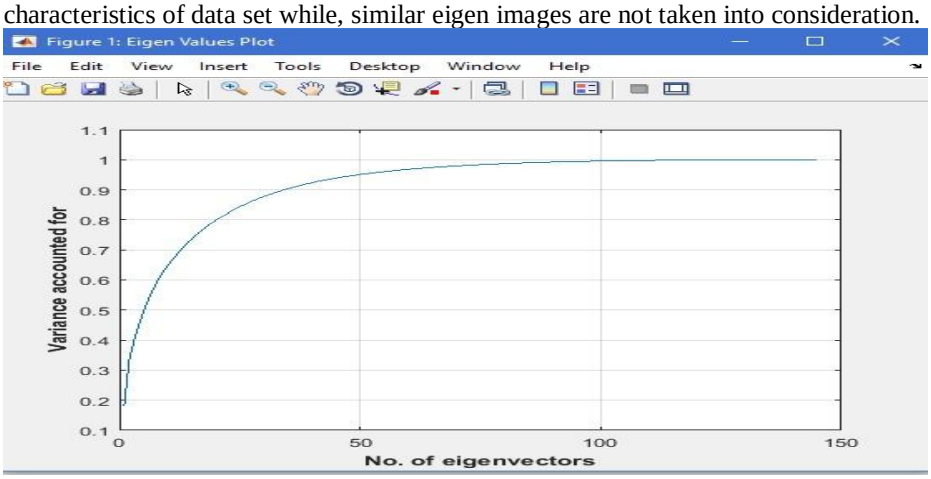


Figure 5. Number of eigen images

Figure 5 shows that in the data set 40-50 eigen images are significantly vary from each other which represent almost all characteristics of data set. Thus, 40 eigen image are taken into consideration for classification purpose for NN.

Step 3.5.8: *Training Stopping Criteria*: It may be Fixed number of iterations (Epochs) to be performed, Use particular threshold for error, Use threshold for error gradient, Early stopping with validation error.

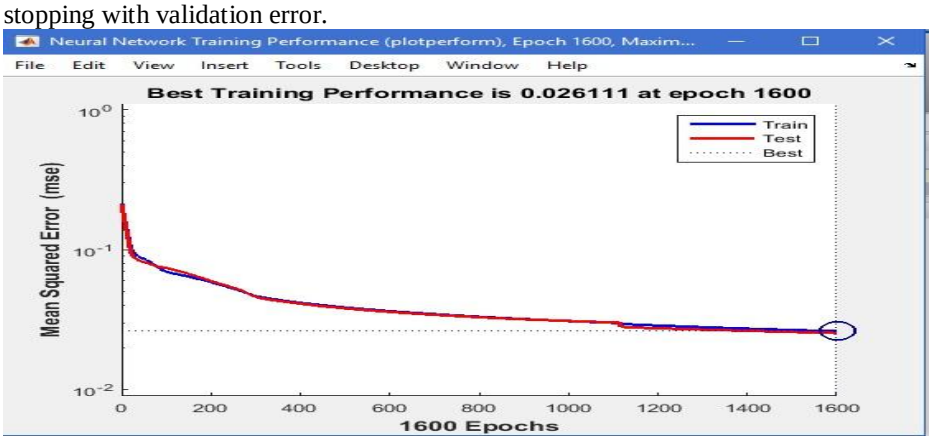


Figure 6. Number of Epoch V/S Mean Square Error

Figure 6 shows that, the training stopping criteria is set to 1600 epoch, where Mean Square Error (MSE) is minimum.

Step 3.6: *Optimum Topology for Neural Network:*[9]

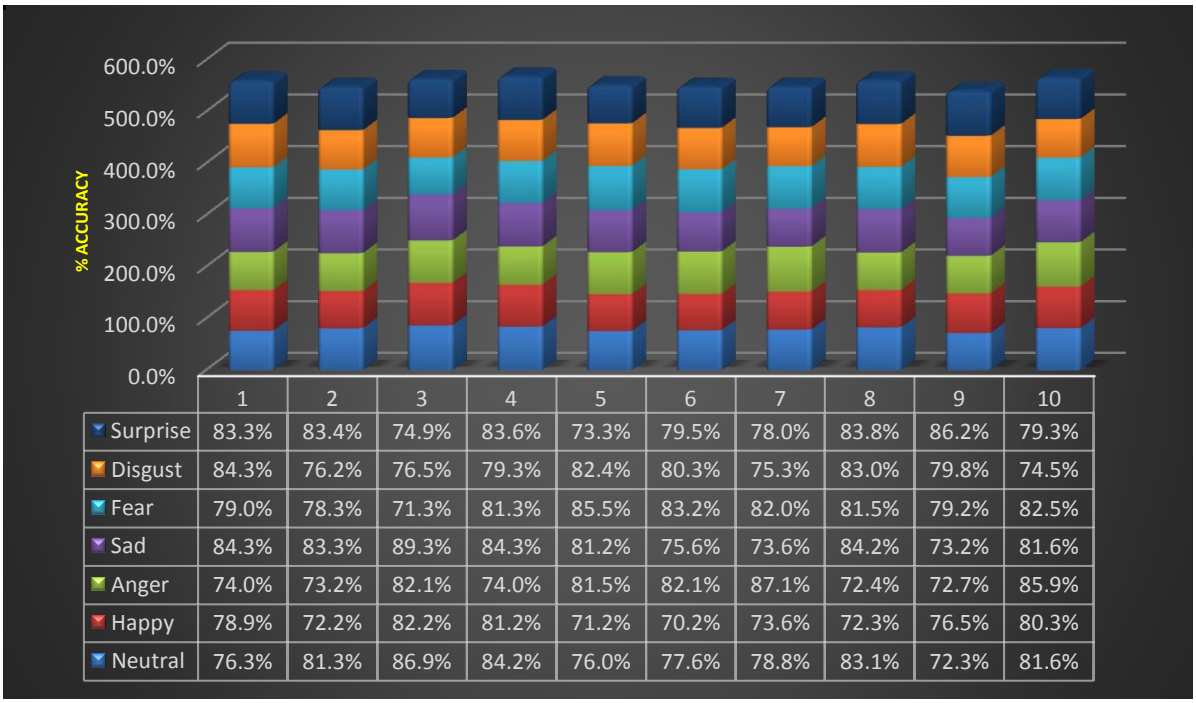
On the basis of previous step of algorithm the optimum topology for NN is set as follows.

Parameter	Values
Data set	Expression blocks 27 x 30 pixels
Training set	90 %
Test set	10 %
Hidden Layers	01
Input Standardization	PCA40
Weight Initialization	Hidden Layer = 0, Output Layer = random
Training Algorithm	Gradient Descent with Momentum
Transfer Function	Both layer use log-sigmoid
Neurons in Hidden Layer	22 neurons
Learning rate	0.9
Momentum term	0.6
Maximum Epochs	1600

Table 1. Optimum Topology For NN

III. EXPERIMENTAL RESULT

In the recognition of an emotion of a person using NN to classify PCA of leg posture, total 10 individual persons are taken as sample. In the demonstration under well-defined environment the leg posture based expression video is captured for all above 10 individual with respect to emotions. The captured expression videos are stored into database and the algorithm discussed in above section - II is simulated over the database of individual. The experimental results includes Training and Testing percentage accuracy is measured along with the respective Mean Square Errors (MSEs) displayed graphically as follows.



	1	2	3	4	5	6	7	8	9	10
Surprise	83.3%	83.4%	74.9%	83.6%	73.3%	79.5%	78.0%	83.8%	86.2%	79.3%
Disgust	84.3%	76.2%	76.5%	79.3%	82.4%	80.3%	75.3%	83.0%	79.8%	74.5%
Fear	79.0%	78.3%	71.3%	81.3%	85.5%	83.2%	82.0%	81.5%	79.2%	82.5%
Sad	84.3%	83.3%	89.3%	84.3%	81.2%	75.6%	73.6%	84.2%	73.2%	81.6%
Anger	74.0%	73.2%	82.1%	74.0%	81.5%	82.1%	87.1%	72.4%	72.7%	85.9%
Happy	78.9%	72.2%	82.2%	81.2%	71.2%	70.2%	73.6%	72.3%	76.5%	80.3%
Neutral	76.3%	81.3%	86.9%	84.2%	76.0%	77.6%	78.8%	83.1%	72.3%	81.6%

Figure 7. Percentage Train Accuracy NN with PCA Leg Posture

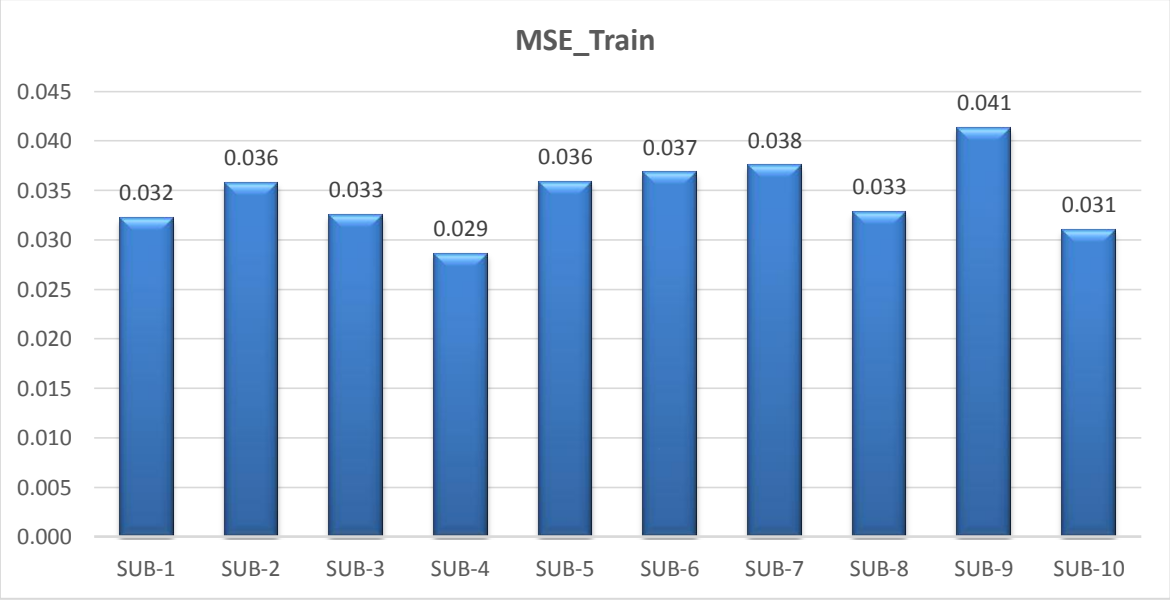
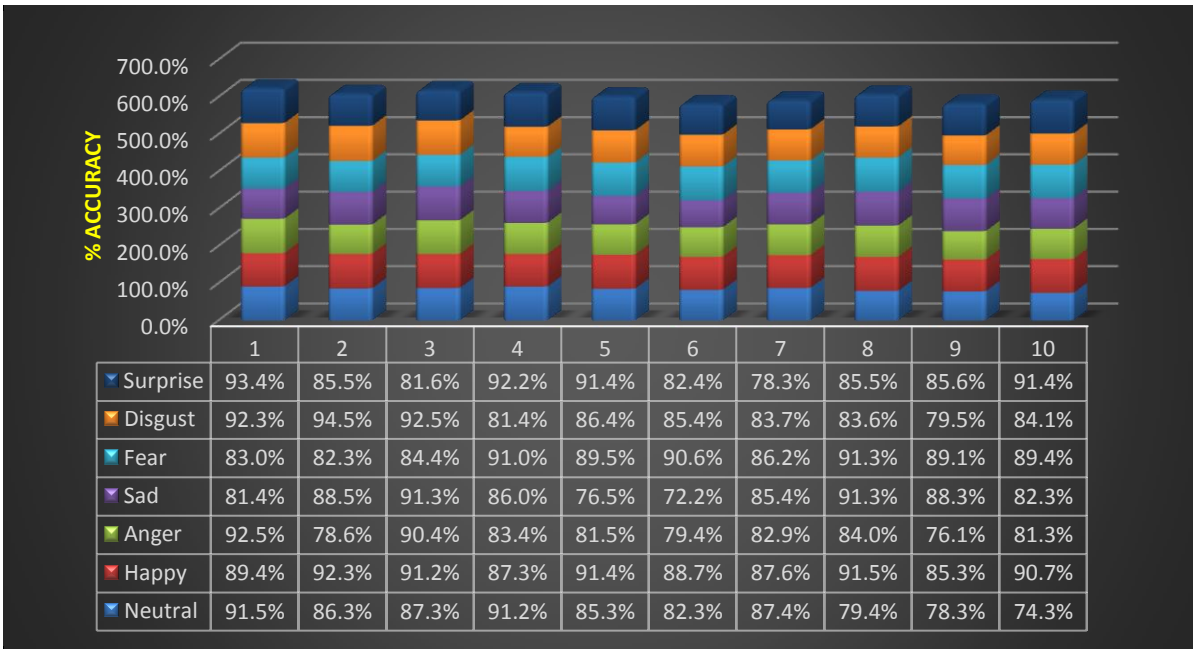


Figure 8. MSEs for Training Set

Figure 7 shows that the training accuracy obtained by NN classification of PCA of Leg postures of 10 individuals is overall 79.30 % which is good as per the complexity in the images of various leg posture. The measurement of accuracy indicates that it has scope of improvement can be improved by updating the training set with more realistic and accurate

images of respective postures. Also Figure 8 indicates the overall MSE of classification of training set of all 10 individuals can be reduced by taking more precise images.



	1	2	3	4	5	6	7	8	9	10
Surprise	93.4%	85.5%	81.6%	92.2%	91.4%	82.4%	78.3%	85.5%	85.6%	91.4%
Disgust	92.3%	94.5%	92.5%	81.4%	86.4%	85.4%	83.7%	83.6%	79.5%	84.1%
Fear	83.0%	82.3%	84.4%	91.0%	89.5%	90.6%	86.2%	91.3%	89.1%	89.4%
Sad	81.4%	88.5%	91.3%	86.0%	76.5%	72.2%	85.4%	91.3%	88.3%	82.3%
Anger	92.5%	78.6%	90.4%	83.4%	81.5%	79.4%	82.9%	84.0%	76.1%	81.3%
Happy	89.4%	92.3%	91.2%	87.3%	91.4%	88.7%	87.6%	91.5%	85.3%	90.7%
Neutral	91.5%	86.3%	87.3%	91.2%	85.3%	82.3%	87.4%	79.4%	78.3%	74.3%

Figure 9. Percentage Test Accuracy NN with PCA Leg Posture

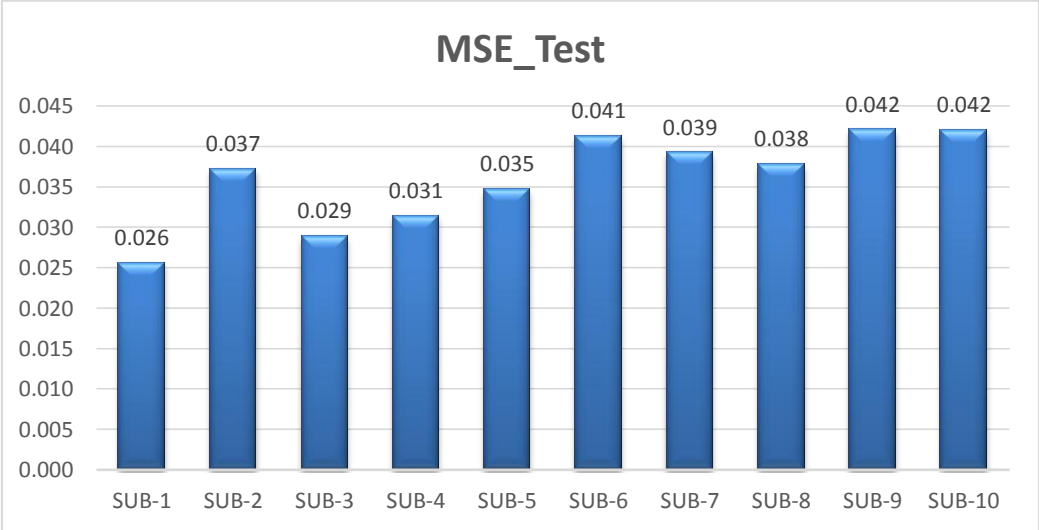


Figure 10. Percentage Test Accuracy NN with PCA Leg Posture

Figure 9 shows test accuracy level of classification of test set over the respective train set of 10 individual. The overall accuracy in this case is obtained 86% which is far good. But, it should be remember that it is because of the experiment is done under well-defined environment. For realistic approach it may be reduced but, can be managed by eliminating surrounding from the images and more precise images are taken in consideration. Also Figure 10 shows the test accuracy level which is again a good sign of the accuracy as it is expectable lower.

IV. CONCLUSION

In the recognition of an emotion of an individual human being using PCA of Leg posture, the accuracy level obtained by NN is expected to be satisfactory and it is. Also the MSEs of both the set training and testing set lower as per expectation. This recognition of emotion through leg gesture of a human improves the understating level of human nature, which is an essence to deal with human during any kind of communication.

REFERENCES

[1] Allan and Barbara, "The Definitive Book of Body Language, How to Read other's Thought by their Gesture", *Pease International*, ISBN 1-9208160-7-0, Australia, 2004.

[2] Konrad Schindler, Luc Van Gool, Beatrice de Gelder, "Recognizing Emotions Expressed by Body Pose: A Biologically Inspired Neural Model", *Neural Networks Journal* 21, pp. 1238–1246, 2008.

[3] David B. Givens, "The Nonverbal Dictionary of Gestures, Signs & Body Language Cues", *Center for Nonverbal Studies Press*, Spokane, Washington, 2002.

[4] Nele Dael, Marcello Mortillaro, and Klaus R. Scherer, "Emotion Expression in Body Action and Posture", *American Psychological Association*, DOI: 10.1037/a0025737, Vol. 12, No. 5, 1085–1101, 2012.

[5] Ekaterina P. Volkova, Betty J. Mohler, Trevor J. Dodds, Joachim Tesch, and Heinrich H. Bülthoff, "Emotion Categorization of Body Expressions in Narrative Scenarios", *Frontiers in Psychology*, *Emotion Science*, Volume 5, DOI: 10.3389, 2014.

[6] Gonzalez, R. C. and Woods, R. E., "Digital Image Processing", Prentice Hall, 3rd Ed. 2009

[7] Daw-Tunglin, Taiwan, "Facial Expression Classification Using PCA and Hierarchical Radial Basis Function Network" Journal of Information Science And Engineering 22, pp. 1033-1046, 2006.

[8] Jawad Nagi, Syed Khaleel Ahmed , "A MATLAB based Face Recognition System using Image Processing and Neural Networks", 4th International Colloquium on Signal Processing and its Applications, pp. 83-88, March 7-9, 2008.

[9] Rajasekaran, S. and VijayalakshmiPai, G.A.: "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", Prentice Hall of India, 2003.

[10] https://in.mathworks.com

