International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-6 Issue-4, April 2017

Gender Classification with Weighted Principal Component (wPC) using BPN

Janaki Sivakumar, K. Thangavel

Abstract*: Gender Classification in the field of forensic Science becomes essential in the case of criminal investigation. Automated tools can help forensic experts by reducing their manual efforts. Soft computing techniques like Fuzzy Computing, Neural Networks and Genetic Algorithm are all helpful to develop automated tools for human identification. Lateral Cephalogram plays a vital role in Gender Classification from skeletal remains. Principal component analysis (PCA) is a technique that is useful for the compression and classification of data. This study proposes Weighted Principal Components of lateral Cephalogram landmarks as an ideal measure .Also this study recommends BPN as an optimal classifier for Gender Classification from lateral Cephalogram.

Index Terms*: Lateral Cephalogram, Forensic Anthropology, Cephalofacial Landmarks, Linear Measurements, GLCM Features, Principal Components, Weighted Principal Components, Feature Extraction and Back Propagation Neural Network

I. INTRODUCTION

Forensic Anthropology is the application of physical anthropology to assist medico legal procedure. The need of anthropometry arises under several circumstances as natural, intentional and accidental to identify human body [1]. It also serves an important role in human identification [2].Soft tissues are commonly no longer present, due to carbonization and decomposition. Gender Classification is ideal from the skeletal remains like pelvis, ribcage, shoulders, sacrum bone, teeth and skull. Among all, Skull is probably the major region to classify the gender and it is playing a vital role as it resists adverse environmental conditions, preserved part of the skeleton over the period of time after death. In many cases skull is the only available part for forensic examination. Lateral Cephalogram could reveal morphological and architectural details of a skull on single radiograph and thereby providing additional characteristics and multiple points for comparison [3].This study focuses on Back Propagation Neural Network and weighted Principal Component's to achieve better results rather than simple BPN or simple PCA. In this study, initially from the 51 extracted landmarks, Principal Component Matrix has been extracted to reduce the dimension .Secondly by adding weights with the Principal components a new weighted PC matrix has been derived . Thirdly Back Propagation Neural Network applied as classifiers for training and testing the samples from the below sets of extracted landmarks of lateral Cephalogram.

a) 51 extracted landmarks
b) GLCM features
c) 12 linear derived features
d) Principal Component matrix and
e) Weighted Principal Component matrix

II. MATERIALS USED

A total of 140 digitized lateral Cephalogram images in which 69 female classes and 71 male classes were selected from AAOF craniofacial growth legacy collection with class I within the ages limit of 15 years to 54 years. The Fig.1 shows the lateral Cephalogram with basic anthropometry landmarks and following Table 1describes the landmarks.

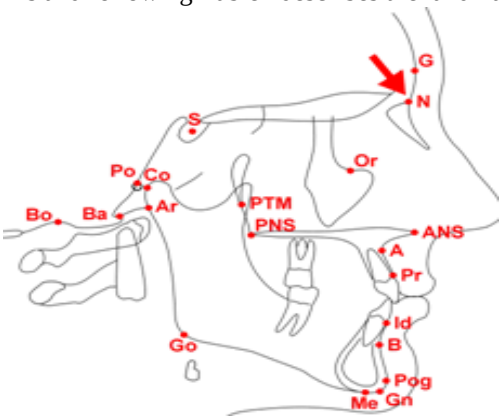

Fig.1. Lateral Skull View with basic Landmarks

TABLE I. CEPHALOGRAM LANDMARKS

Cephalometric Landmarks	
A-point (Point A, Subspinale, ss)	Menton (Me)
Anterior nasal spine (ANS)	Nasion (N, Na)
Articulare (Ar)	Posterior nasal spine (PNS)
B-point (Point B, Supramentale, sm)	Porion (Po)
Glabella (G)	Orbitale (Or)
Gonion (Go)	Sella (S)

As an initial stage, the image set with known male and female classes were preprocessed: wiener filtered, resized, cropped and ROI extraction was done. Secondly, a total of 51 cephalometric landmarks were identified by using single fixed view active appearance mode as in Fig.2 .Thirdly twenty one GLCM features were extracted based on three major sub regions Sella, Nasion and Mouth. Finally 12 linear measurements were calculated as derived features between extracted landmarks as in Fig. 3.

Manuscript published on 30 April 2017.
***** Correspondence Author (s)
Janaki Sivakumar, Research Scholar, PRIST University, Thanjavur (Tamil Nadu)-613403, India, E-mail: pjanaki78@gmail.com
Dr. K. Thangavel, Professor and Head, Department of Computer Science, Periyar University, Salem, Tamil Nadu, India, E-mail: drktvelu@yahoo.com

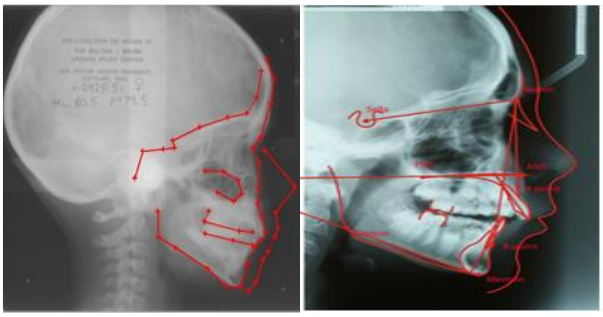

Fig.2(a) Extracted Landmarks Fig. 2(b). Linear Distances

III. PROPOSED METHODS

This paper focuses on classification of gender from cephalometric landmarks using Principal Components (PC) with weighted approach for dimensionality reduction .Weighted principal Components (*w*PC) are fed into BPN for classification. This paper proposes Weighted Principal Component (*w*PC) with BPN classification. It proposes more accuracy comparatively with straightforward simple BPN application from extracted landmarks.

IV. PRINCIPAL COMPONENT ANALYSIS

PCA is the most widely used, well known multivariate method, invented by Pearson(1901).The objective of PCA is to take a data matrix of **N** objects and **P** variables ,which are correlated and summarizes it by uncorrelated axes ,that are linear combination of original P variables[4]. Centroid of the points is defined by the mean of each variable. The variance of each variable is the average squared deviation of its N values around the mean of that variable. Degree to which the variables are linearly correlated is represented by their covariance. Objective of PCA is to rigidly rotate the axes of this p-dimensional space to new positions (Principal Axes) that have the following properties:

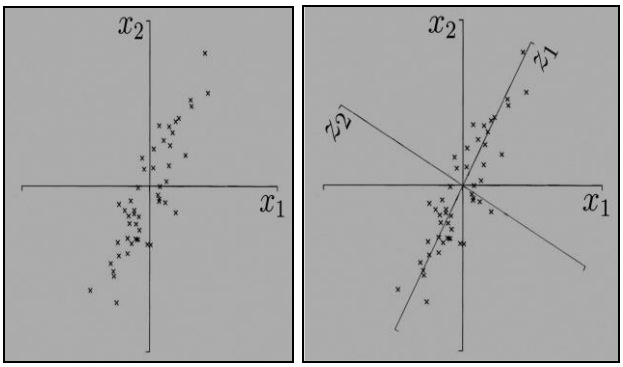

Fig.3(a) N observations in a 2D plane Fig. 3(b). PC's in New Coordinate System

Ordered such that principal axis 1 has the highest variance, axis 2 has the next highest variance… and axis p has the lowest variance. Covariance among each pair of the principal axes is zero, hence the principal axes are uncorrelated.

Fig. 3(a), Shows the Geometric Picture of N observations on 2D plane. Principal Component Z_1 is a minimum distance fit to a line .Principal Component Z_2 is a minimum distance fit to a line orthogonal to Z_1 in Fig.3 (b).

Principal Components are a series of Linear Least Squares (LLS) fit to a sample plane each perpendicular to all previous. The purpose is to reduce the dimensionality of a data set (sample) by finding a new set of variables, smaller than the original set of variables that retains most of the sample's information.

V. WEIGHTED PRINCIPAL COMPONENTS

Classical PCA is a mature tool whose performance in dimensionality reduction and classification has been assessed for a long time. Usual PCA is very sensitive to the presence of noise in the data, given that it is based on Pearson's correlation coefficient. The inherent consequence is that the classical PCA implementations made no difference between variance coming from a genuine underlying features and variance coming from measurement noise. At the present time, some methods are able to deal with weight matrices having the same size as the corresponding data set [5].

As per statistical rule on sample sizes, large samples should get smaller weights and smaller samples should get larger weights [6]. Based on this rule, the below weights shown in Equations 1 and 2 have been assigned to PCA matrix depending upon their sample and class size

$$Weight(D_F) = \frac{1}{[N * Size(F)]} \quad (1)$$

where D_F is the data on female samples, N is the number of classes and Size(F) is the number of female samples.
Similarly

$$Weight(D_M) = \frac{1}{N * Size(M)} \quad (2)$$

where D_M is the data on male samples, N is the number of classes and Size (M) is the number of male samples.

The algorithm Weighted PCA shown in Fig.4 is expressing the steps involved in complete Weighted Principal Components Analysis Approach.

Algorithm: **Weighted PC –BPN Classifier**
Input: Information Matrix X
Decision Matrix ***D***
Output: Weighted PC Matrix (**wPC**) and Classification Results
Step 1: Find S ←**COV**(***X***)
Step 2: Find Eigen Value ***(λ)*** and Eigen Vector (***E*** $_{\lambda}$)
Step 3: PC ← the Eigen vectors with Largest Eigen Values
Step 4: Compute Co efficient Matrix ***B*** using PC's
Step 5: Compute Standard Score Matrix ***Z*** ***∀X***
Subtracting each element of ***X*** with corresponding Column Mean ($\bar{x}$) and Divide by Standard Deviation (***σ***) as

$$Z_i \leftarrow \frac{X_i - \bar{x}}{\sigma}$$

Step 6: Compute Matrix ***PC ←Z * B***
Step 7: For Each unit $x_i \in PC$
Step 8: For each decision $d_j \in D$
Step 9:If $x_i \in d_j$ then

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Find $W_i \leftarrow \frac{1}{n(D) * n(X_j)}$, Where

$n(D) \leftarrow$ the No. of Decision classes &

$n(X_j) \leftarrow$ The No. of X_j classified as d_j

End

End

Step 10: Compute wPC $\forall$ element Y_i as

$Y_i \leftarrow PC_i * W_i$

End

Step 11: Call $BPN(wPC, D)$

Fig. 4 Procedure w PCA-BPN

VI. BACK PROPAGATION NEURAL NETWORK CLASSIFICATION

Back propagation neural network is a multilayered, feed forward neural network and is by far the most extensively used. It is also considered as one of the simplest and most general methods used for supervised training of multilayered neural networks[7]. Back propagation works by approximating the non-linear relationship between the input and the output by adjusting the weight values internally. It can further be generalized for the input that is not included in the training patterns.

BPN is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Developing a neural network involves first training the network to carry out the desired computations.

Generally, the Back propagation network has two stages: training and testing. During the training phase, the network is "shown" sample inputs and the correct classifications. For example, the input might be an encoded picture of a face and the output could be represented by a code that corresponds to the name of the person.

A further note on encoding information - a neural network, as most learning algorithms, needs to have the inputs and outputs encoded according to an arbitrary user defined scheme. The scheme will define the network architecture so that once a network is trained; the scheme cannot be changed without creating a totally new net. Similarly there are many forms of encoding the network response.

Fig. 5 shows the topology of the Back propagation neural network that includes and input layer, one hidden layer and an output layer. It should be noted that Back propagation neural networks can have more than one hidden layer.

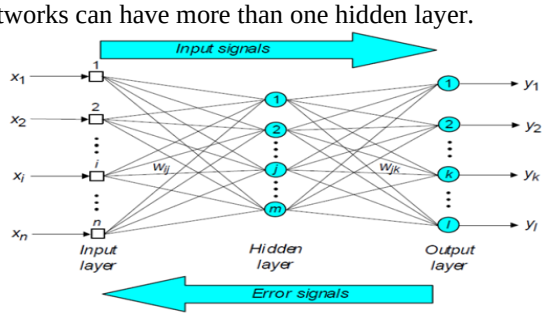

Fig .5.Back Propagation Neural Network with Hidden Layer

A) Feed Forward Neural Network

BPN is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements working in unison to solve specific problems. Developing a neural network involves first training the network to carry out the desired computations.

The feed-forward neural network architecture is commonly used for supervised learning. Feed-forward neural networks contain a set of layered nodes and weighted connections between nodes in adjacent layers. Feed-forward networks are often trained using a back propagation-learning scheme. Back propagation learning works by making modifications in weight values starting at the output layer then moving backward through the hidden layers of the network. Neural networks have been criticized for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learned weights. Advantages of neural networks, however, include their high tolerance to noisy data as their ability to classify patterns on which they have not been trained [8-11].

B)Description

The operations of the Back propagation neural networks can be divided into two steps: Feed forward and Back propagation. In the feed forward step, an input pattern is applied to the input layer and its effect propagates, layer by layer, through the network until an output is produced. The network's actual output value is then compared to the expected output and an error signal is computed for each of the output nodes. Since all the hidden nodes have, to some degree, contribute to the errors evident in the output layer, the output error signals are transmitted backwards from the output layer to each node in the hidden layer that immediately contribute to the output layer. This process is then repeated, layer by layer, until each node in the network has received an error signal that describes its relative contribution to the overall error.

Once the error signal for each node has been determined, the errors are then used by the nodes to update the values for each connection weights until the network converges to a state that allows all the training patterns to be encoded. The Back propagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent[12]. The weights that minimize the error function are then considered to be a solution to the learning problem.

The network behavior is analogous to a human that is shown a set of data and is asked to classify them into predefined classes. Like a human, it will come up with "theories" about how the samples fit into the classes. These are then tested against the correct outputs to see how accurate the guesses of the network are. Radical changes in the latest theory are indicated by large changes in the weights, and small changes may be seen as minor adjustments to the theory.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

There are also issues regarding generalizing a neural network. Issues to consider are problems associated with under-training and over-training data. Under-training can occur when the neural network is not complex enough to detect a pattern in a complicated data set. This is usually the result of networks with a few hidden nodes that it cannot accurately represent the solution, therefore under-fitting the data (Fig.6). On the other hand, over-training can result in a network that is too complex, resulting in predictions that are far beyond the range of the training data. Networks with too many hidden nodes will tend to over-fit the solution (Fig.6). The aim is to create a neural network with the "right" number of hidden nodes that will lead to a good solution to the problem (Fig.6) [13].

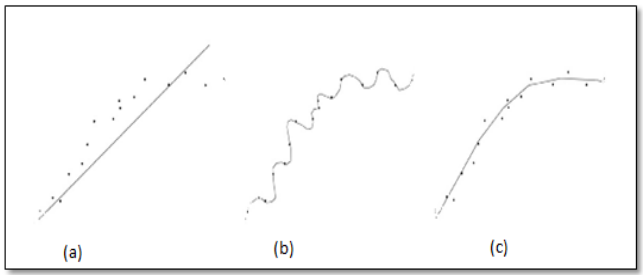

Fig.6.(a) Under-fit (b) Over-fit (c) Correct fit

C) BPN- Weight Optimization Process

Back propagation is a common method for training a neural network, the goal of back propagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs.

Back propagation method contains the following steps:

Step 1: Initialization; Set all the weights and threshold levels of the network to random numbers uniformly distributed inside a range

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i}\right) \tag{3}$$

where Fi is the total number of inputs in the network.

Step 2: Activation; Activate the back-propagation neural network by applying inputs **x1**(**p**),**x**2(**p**),… ,**xn**(**p**) and desired outputs **yd**,**1**(**p**),**yd**,**2**(**p**), … ,**yd**,**n**(**p**)

Calculate the actual outputs of the neurons in the hidden layer

$$Y_j(P) = sigmoid\left[\sum_{i=1}^{n} X_i(P).W_{ij}(P) - \theta_j\right] \tag{4}$$

where n is the number of inputs of neuron j in the hidden layer.

- Calculate the actual outputs of the neurons in the output layer

$$Y_k(P) = sigmoid\left[\sum_{j=1}^{m} X_{jk}(P).W_{jk}(P) - \theta_k\right] \tag{5}$$

where m is the number of inputs of neuron k in the output layer

Step 3: Weight; training (back-propagate)

Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

- Calculate the error gradient for the neurons in the output layer

$$\delta_k(P) = Y_k(P).[1 - Y_k(P)].e_k(P) \tag{6}$$

Where

$$e_k(P) = Y_{d,k}(P) - Y_k(P) \tag{7}$$

- Calculate the weight corrections

$$\Delta W_{jk}(P) = \alpha.Y_j(P).\delta_k(P) \tag{8}$$

- Update the weights at the output neurons

$$W_{jk}(P+1) = W_{jk}(P) + \Delta W_{jk}(P) \tag{9}$$

- Calculate the error gradient for the neurons in the hidden layer

$$\delta_j(P) = Y_j(P).[1 - Y_j(P)].\sum_{k=1}^{l} \delta_k(P).W_{jk}(P) \tag{10}$$

- Calculate the weight corrections

$$\Delta W_{ij}(P) = \alpha.X_i(P).\delta_j(P) \tag{11}$$

- Update the weights at the hidden neurons

$$W_{ij}(P+1) = W_{ij}(P) + \Delta W_{ij}(P) \tag{12}$$

Step 4: Iteration; Increase iteration p by one, go back to Step 2 and repeat the process until the selected error criterion is satisfied. Ideally, the error function should have a value of zero when the neural network has been correctly trained. This, however, is numerically unrealistic.

D) BPN-Pseudo Code

```
Assign all network inputs and output
Initialize all weights with random numbers,
typically between  -1 and 1
Repeat
   For every pattern in the training set
      Present the pattern to the network
// propagated the input forward through the network:
         For each layer in the network
            For every node in the layer
               i. Calculate the weight sum of the inputs to
                  the node
              ii. Add the threshold to the sum
              iii. Calculate the activation for the node
            End
         End
 // propagate the errors backward through the network
            For every node in the output layer
```


Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

```
 Calculate the error signal
        End
        For all hidden layers
           For every node in the layer
              i. Calculate the node's signal error
              ii. Update each node's weight in the network
           End
End
//      Calculate Global Error
        Calculate the Error Function
   End
While((maximum number of iterations<than specified)
 AND  (Error Function is > than specified))
```

E) BPN Architecture Used In This Research

Practically, it is very difficult to determine a good network topology just from the number of inputs and outputs. It depends critically on the number of training examples and the complexity of the classification trying to learn. Every classifier is associated with some parameters, which has to be tuned properly while training to get the optimal performance. This is true fact that by taking suitable number of hidden layers and the number of neurons in each hidden layer, better results can be obtained. Many researchers develop approaches to estimate the number of neurons and hidden layers requirement for a neural network but the approximation also gets dependable on the type of the database samples for which the network is designed. Also, unnecessary increment in the neurons or layer will lead to over fitting problem. So it is quite essential that before designing the neural network, training database samples must be analyzed so that approximation of number of neurons and hidden layers can be guessed properly.

In this research, data set of 140 samples with classification group of two classes (Male and Female) has been used. Since the class is only two, Back Propagation Neural network has been designed as 1-1-1 architecture. Number of Neurons in input layer varies in each feature set. Number of neurons in hidden layer is decided as per the general theory as

$$\text{Number of Hidden neurons} = 2/3 * (\text{Number of Input Neurons}) \quad (13)$$

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance. The default performance function for feed forward networks is Mean Square Error (MSE). There are two different ways in which training can be implemented: Incremental Mode and Batch Mode. In Incremental Mode, the gradient is computed and the weights are updated after each input is applied to the network. In batch mode, all the inputs in the training set are applied to the network before the weights are updated.

Batch training is significantly faster and produces smaller errors than incremental training. For training multilayer feed forward networks, any standard numerical optimization algorithm can be used to optimize the performance function. In this research, Gradient Descendant Training algorithm has been used to train the network. This optimization method uses gradient of the network performance with respect to the network weights. The gradient is calculated using a technique called the *back propagation* algorithm, which involves performing computations backward through the network.

The stopping criterion is checked at the end of each epoch: The error, at the end of an epoch is below a threshold, all training examples are propagated and the mean error is calculated. Network training will stop:

- If the error fails to improve (has reached a minimum)
- If the rate of improvement drops below a certain level
- If the error reaches an acceptable level
- When a certain number of epochs have passed

At the end of the epoch, check the stopping criteria are satisfied. If yes, stop training, if not, continue training. Maximum number of epochs set by this algorithm is 1000 epochs.

VII. K FOLD CROSS VALIDATION

Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the training algorithm will do when it is asked to make new predictions for data it has not already seen.

One way to overcome this problem is not to use the entire data set when training. Some of the data are removed before the commencement of training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on ``new'' data. This is the basic idea for a whole class of model evaluation method called cross validation. K-fold cross validation is one way to improve training for classification. The data set is divided into K subsets, and the holdout method is repeated K times.

Each time, one of the K subsets is used as the test set and the remaining K-1 subsets are put together to form a training set. Then the average error across all K trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set K-1 times. Extensive experiments have shown that 10- fold cross validation is the best choice to get an accurate estimate. Total data set will be divided into 10 subsets of equal size. Then 9 sets will be used for training the network and one - set has been used for testing the network performance. Ten-fold cross-validation is repeated ten times and the results are averaged. Cross-validation avoids overlapping test sets.

VIII. EXPERIMENTAL RESULTS AND ANALYSIS

Back Propagation Neural network with ten-fold cross validation has been applied to all feature sets which include the following:

(a). SFVAM Feature Extraction - 51 Landmarks
(b). Linear Measurements Feature set - 12 Features
(c). Principal Component Matrix - 4 PC's
(d). Weighted Principal Components - 4 weighted PC's

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

As a whole, the overall performance on classification accuracy has been summarized in Table 2. From the Table, one can conclude the following that Weighted Principal component Analysis (**d**) is producing better performance than The direct conventional Principal component Analysis (**c**).

TABLE I. OVERALL PERFORMANCE ANALYSES ON 10-FOLD BPN CLASSIFICATIONS

K	Training	Testing	Accuracy rate using BPN (in %)			
			(a)	(b)	(c)	(d)
1	15-140	Jan-14	85.71	85.71	78.57	85.71
2	1-14, 29-140	15-28	85.71	78.57	85.71	85.71
3	1-28, 43-140	29-42	78.57	78.57	85.71	64.29
4	1-42, 57-140	43-56	85.71	78.57	64.29	100
5	1-56, 71-140	57-70	64.29	85.71	85.71	100
6	1-70, 85-140	71-84	78.57	78.57	78.57	85.71
7	1-84, 99-140	85-98	78.57	100	100	100
8	1-98, 113-140	99-112	85.71	78.57	85.71	100
9	1-112, 127-140	113-126	100	78.57	100	85.71
10	1-126	127-140	92.86	85.71	85.71	78.57
Average Accuracy			**83.57**	**84.28**	**84.99**	**88.57**

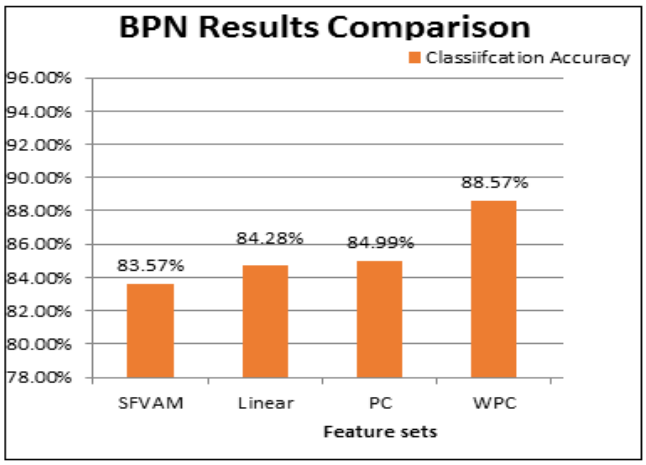

Fig.7 Performance Analyses from BPN Accuracy

Gender classification using Lateral Cephalogram using soft computing models; one can achieve a maximum of 88.57 % accuracy on gender classification using Back Propagation Neural Networks. Fig.7 shows that, out of the four feature sets implemented in BPN ten-fold cross validation, **wPCA on linear measures** promisingly gives more accuracy than other approaches.

IX. CONCLUSION

In this research work, AI based technique BPN was developed for gender identification with an objective to achieve high accuracy. The result of this technique is also compared with the conventional techniques discriminant function analysis. BPN classifier tested on a sufficiently large database of 140 benchmarked images of both the classes. Four feature sets have been given as input to BPN to test the classification accuracy. From the classification results it is found that Weighted PCA approach has given more accuracy in gender classification.

REFERENCES

1. Patil KR, Mody RN. Determination of sex by discriminant function analysis and stature by regression analysis: a lateral cephalometric study. Forensic Sci. Int. 2005; 147:175-80.2.
2. Graw M, Wahl J, Ahlbrecht M. Course of the meatus acusticus internus as criterion for sex differentiation Forensic Sci. Int. 2005; 147:113-7
3. Almas Binal and BK yashoda devi,"Identification of sex using Lateral Cephalogram:Role of Cephalofacial Parameters", JIAOMR, Oct-Dec, 2012, 24(4):280-283
4. Jonathon Shlens, "A Tutorial on Principal Component Analysis..
5. Delchambre L, Weighted principal component analysis: a weighted covariance eigen decomposition approach, Monthly Notices of Royal Astronomical Society, 446(4), December, 2014.
6. . Debbie l, A primer for using and understanding weights with national datasets, The Journal of Experimental Education, 73(3), pp: 221–248, 2005
7. Wei Lu, Neural Network Model for Distortion Buckling Behaviour of Cold-Formed Steel Compression Members, Helsinki University of Technology Laboratory of Steel Structures Publications 16, 2000
8. Haykin S, Neural Networks - A comprehensive foundation, Macmillan Press, New York,1994
9. Rinehart D.E, Hinton G.E and Williams R. J., learning internal representations by error propagation, Parallel Distributed Processing, Cambridge, MA: MIIT Press,1994
10. Lu H, Setiono R and Liu H, Effective data mining using neural networks, IEEE Transaction On Knowledge and Data Engineering,1996
11. Roy, Artificial neural networks – a science in trouble, SIGKDD Explorations, 2000
12. Raul Rojas , Neural Networks - A Systematic Introduction,The Back propagation Algorithm, 1996.
13. Robert Barry, Artificial Neural Network Prediction of Wavelet Sub-bands for Audio Compression, Thesis for Bachelor of Science, Dept. of Electrical & Computer Engineering, 2000

Janaki Sivakumar received M.C.A degree in 2002 from Bharathidasan University and M.Phil (C.Sc) from Mother Teresa University Kodaikanal in 2004. Currently she is a research scholar in Computer Science, PRIST University, Tanjore, India. Her main area of interest is in Image and Signal Processing. She has published papers in 5 International Journals & attended 3 International Conferences.

Dr. K. Thangavel He has received Ph. D degree in the area of Optimization Algorithms from the Gandhigram Rural Institute-Deemed University, Gandhigram, Tamilnadu, India in 1999. Currently he is working as Professor and Head in the Department of Computer Science, Periyar University, Salem, Tamilnadu, India. He has published more than 200 research publications in various National, Inter National Journals and conferences . He has edited three books published by Narosa Publishers NewDelhi, India. He has successfully guided for 16 Ph. D and 14 M. Phil scholars. More than 11 Ph. D scholars are pursuing under his research supervision. His research interest includes Data Mining, Digital Medical Image Processing, Soft Computing, Mobile Computing and Bio-informatics. He is a life member of Operational Research Society of India and member of the research group in Rough set Society. He has organized five National Conferences, three National Seminars, five Research workshops and two 21 day UGC refresher progrmmes. He is reviewer for leading publishers such as Elseviewer, Springer, Taylor and Francis.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)