



# An Improved Intelligent Multi Biometric Authentication System

Benson-Emerike Mercy E.  
Dept. of Computer Science  
Abia State Polytechnic, Aba

Nwachukwu E.O.  
Dept. of Computer Science  
University of Port Harcourt  
Port Harcourt

## ABSTRACT

The beginning of the 21st century was marked with events that focused on the world's attention to public security. Increase in technological advancement gave people possibilities of information transfer and ease of physical mobility unseen before. With those possibilities comes risk of fraud, theft of personal data, or even theft of identity. One of the ways to prevent this is through biometric authentication system. Unibiometric systems rely on the evidence of a single source of information whereas multibiometric systems consolidate multiple sources of biometric evidences. Multibiometric systems, if designed properly, are able to enhance the matching performance. In this paper, Intelligent Multi-Biometric Authentication System, face and fingerprint biometric traits are used. When the images are captured, preprocessing in face and fingerprint images is done using Enhanced Extracted Face (EEF) method and Plainarized Region Of Interest (PROI) method respectively. These are fed into a Cascaded Link Feed Forward Neural Network (CLFFNN) which is a classifier trained with back-propagation algorithm. CLFFNN comprises of CLFFNN(1) used for training and CLFFNN(2) used as the main classifier. They are arranged in cascades. Afterwards, both outputs from face and fingerprints are combined using AND operation.

## Keywords

Multibiometric, Authentication, Enhanced Extracted Face (EEF), Plainarized Region of Interest (PROI), Preprocessing, Recognition speed, Cascaded Link Feed Forward Neural Network (CLFFNN), Back propagation.

## 1. INTRODUCTION

Biometrics is a field of science that uses computer technology to identify people based on their physical or behavioral characteristics, such as fingerprints or face biometrics. "Biometric" comes from the Greek words "bio" (life) and "metric" (to measure). "Bio" in the name refers to the physiological traits that are measured, while "metrics" refers to the quantitative analysis that provides a positive identification of a unique individual. Even though, the biometric identification systems out-perform peer technologies, there is not one single biometric technology that would be ideal for all applications. Each technology has its own benefits and weaknesses in terms of accuracy, cost, ease of use, intrusiveness, ease of deployment. Hence, the unimodal biometric systems have to contend with a variety of problems, namely, false rejection, noisy data, intra-class variations, restricted degrees of freedom, non-universality and spoof attacks. To overcome these difficulties multi-biometric systems are used [11], [42], [25]. Many of these limitations can be addressed by deploying multi-modal biometric systems that integrate the evidences presented by multiple sources of information [16], [27], [42], [44], [49].

This paper is arranged in 5 sections. First section deals with the introduction, second section deals with related work, third section deals with face detection using Artificial Neural Network, fourth section deals with fingerprint capture and detection using ANN while the fifth section deals with multibiometric fusion using CLFFNN.

## 2. RELATED WORK

Muhammad Imran Razzak et al, (4 September, 2010) in their work, "Multimodal face and finger veins biometric Authentication" introduced an automatic face and finger vein detection. They used client specific linear discriminant analysis (CSLDA) which combines Linear Discriminant Analysis [LDA] and Principal Component Analysis (PCA). For finger veins, they used Gabor filter for feature enhancement and then applied thinning. Structural, topological and statistical features from thinned finger vein image were extracted. They used the nearest cosine classifier for recognition purpose. In their experiment, web camera was used for face image. Six images of each user were taken from 35 persons for face training while 3 images of first finger, from finger veins, were extracted. Using the eye point, face was extracted and normalized to 60x60. Fusion of face was performed using CSLDA and finger veins recognition is performed using statistical and topological features as seen in figure 2.22. Their system yielded FAR of 0.05 and GAR of 91.4.

Yan Y. and Zang Y. (2011) used a correlation Filter bank based fusion multimodal biometric system using face and palmprint. The differences between face and palmprint modalities were considered and useful information in each modality were duly exploited. They used the designed correlation filter bank (CFB) and PCA for fusion and to reduce the dimensionality of feature set respectively. Their system's recognition rate was in the range of 0.0765 to 0.9964.

Shubhangi D. C. et al, (July, 2012) in their work, "Artificial Multi-Biometric Approaches to Face and Fingerprint Biometrics" employed two methodological stages: feature extraction using PCA and recognition using feed forward back propagation Neural Network (FFBPNN). They used face-finger recognition sequence. The input pixels are grouped by the trained ANN into different clusters which provide the results. Their fingerprint authentication was based on core and minutiae detections. Denoising was done using a bank of Gabor filters. During recognition, the group number which the input fingerprint image shows and the Euclidian distance are compared with the fingerprint image stored in the database. If the Euclidian distance equals zero, then the input image matches exactly. They used information theory approach of coding and decoding for face image recognition.

Tran Binh Long and Le Hoang Thai (2012), designed a



multimodal[involving fingerprint and face] and a multi-algorithmic system. They used the following algorithms: Zernike moment (ZM), Pseudo Zernike Moment (PZM), Polar Cosine Transform (PCT) to extract both face and fingerprint features. A Radial Basis Function (RBF) neural network was used as a classifier in the face and fingerprint recognition system in which the inputs to the neural network are the feature vectors derived from the proposed feature extraction technique. In this stage, the majority method was used for authentication decision strategy. The decisions were at last fused with AND rule at decision level. The AND rule requires a positive decision from all verification modules, so it will not only lead to low false authentication rates, but also result in high false rejection rates. For the recognition performance evaluation, a False Acceptance Rate (FAR) and a False Rejection Rate (FRR) test were performed. These two measurements yield another performance measure, namely Total Success Rate (TSR). The system performance was evaluated by Equal Error Rate (EER) where FAR=FRR. A threshold value was obtained, based on Equal Error Rate criteria where FAR=FRR. Threshold value of 0.2954 was gained for ZM-PZM-PCT- RBF as a measure of dissimilarity.

Nayak P.K. and Narayan D. (2013) in their work “Multimodal Biometric Face and Fingerprint Recognition Using Adaptive Principal Component Analysis and Multilayer Perception” combined ridge-based and eigenface approach for parallel execution. They employed PCA as a neural algorithm called APEX which was developed by kung and Diamantaras (1990). They used normalized Hebbian learning rule to train all the synaptic weights of the model. They employed weight orthogonalization which allows for the growth and shrinkness of the size of the structure so that there will not be need to retrain the old units. They used MLP with back propagation learning technique and combined various fusion levels. They also used a Neural Network classifier to differentiate face from non-face and employed matching score level fusion. They utilized parallel execution of both face and fingerprint. Their major shortcoming was on the speed of face recognition system. They had a low recognition rate.

Prasad K.N et al (2013) proposed a multimodal biometric authentication model called Context-Sensitive Exponent Associative Memory Model (CSEAM). Their model was in succession ranging from acquisition of biometric traits of face and fingerprint, preprocessing, feature extraction as vectors and representation of features in matrix form. After feature acquisition, SVD factorization was applied to generate the keys which transformed to CSEAM model as Associative Memory. Exponential kronecker product was used for encoding and verification in the model. Evaluation was done based on performance metrics such as mean square error (MSE). MSE threshold was fixed with the value  $d=0.001$ .

Long B. Tran and Thai H. Le (May, 2015) in their work, “Person Authentication using Relevance Vector Machine (RVM) for Face and Fingerprint” extracted both face and fingerprint features using Zernike Moment. This moment is invariant to rotation, scaling and noise image, while authentication is done using RVM, which is a sparse linearly parameterized model. Preprocessing involves illuminating fingerprint and face and by reducing or eliminating some of their variations. Image normalization, noise elimination, illumination normalization are some of the preprocessing steps employed using histogram equalization and wavelet

transform. Face and fingerprint features are extracted in a parallel structure and recognition is based on the global information or statistical method.

### 3. OUR PROPOSED METHOD

The proposed method is an Intelligent Multi-biometric Authentication System using a Cascaded Link Feed-Forward Neural Network [CLFFNN] as a classifier. The preprocessing methods used for fingerprint and face are Plainarized Region of Interest Method (PROI) and Enhanced Extracted Face Method (EEF) respectively.

#### 3.1 Intelligent Face Preprocessing and Detection

Face detection can be regarded as the fundamental part of face recognition systems according to its ability to focus computational resources on the part of an image containing a face [Omaira N. A. AL-Allaf, (2014)]. It is a computer technology that is based on learning algorithms to allocate human faces in digital images [Mohammad A. et al, (2013)].

In the past few years, face recognition has received a significant attention and regarded as one of the most successful applications in the field of image analysis [Zhao W. et al, (2000)]. A face is represented as an array of pixel intensity values suitably pre-processed in appearance based approaches (texture). This array is then compared with a face template using a suitable metric [Bouchra A. et al, (2004)]. Face detection in images is really a complex process because of variability present across human faces such as: pose; expression; position and orientation; skin color; presence of glasses or facial hair; differences in camera gain; lighting conditions; and image resolution [Phil B., (2002)].

Face detection is performed before recognition and this is done to extract relevant information for face and facial expression analysis. Artificial neural networks (ANN) were used largely in the recent years in the fields of image processing (compression, recognition and encryption) and pattern recognition. ANN can be used in face detection and recognition because these models can simulate the way neurons work in the human brain. This is the main reason for its role in face recognition. The development of an intelligent face recognition system requires providing sufficient information and meaningful data during machine learning of a face. ANN have become very attractive for many applications due to its abilities such as Learning, generalization, less data requirement, fast computation, ease of implementation of software and hardware availability features.

Neural Nets can be used to construct systems that are able to classify data into a given set or class, in the case of face detection, a set of images containing one or more face, and a set of images that contains no faces. Neural Networks consist of parallel interconnections of simple neural processors. Figure 1 shows an example of a single neural processor, or neuron. Neurons have many weighted inputs, that is to say each input ( $x_1, x_2, x_3, \dots, x_n$ ) has a related weighting ( $w_{1j}, w_{2j}, w_{3j}, \dots, w_{nj}$ ) according to its importance. Each of these inputs is a scalar, representing the data. In the case of face detection, the shade of GRAY of each pixel could be presented to the neuron in parallel (thus for a 10x10 pixel image, there would be 100 input lines  $p_1$  to  $p_{100}$ , with respective weightings  $w_1$  to  $w_{100}$ , corresponding to the 100 pixels in the input image).

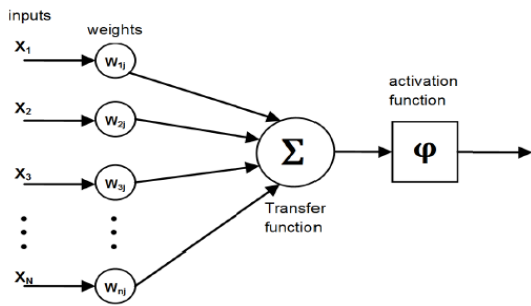


Fig 1: A Neuron

### 3.2 Enhanced Extracted Face Method

Input image is acquired by taking photographs using the webcam. Preprocessing is the process of screening the input image so as to obtain a better-quality output image before passing it on to the next stage of the authentication system. [Benson-Emenike M.E. and Nwachukwu E.O., (2015)]. Preprocessing is an important phase in face recognition that makes the quality of face image standard and ready for feature extraction. Feature Extraction is achieved with the help of an Enhanced Extracted Face Method (EEF) in order to detect face in the image (see figure 2). This method is useful for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull, etc. The Enhanced Extracted Face Method is a morphological shape detector that can be used to look for particular patterns of foreground and background pixels on an image.

### 3.3 Problems in Face Detection

Most commonly used databases for developing face recognition systems rely on images of human faces captured and processed in preparation for implementing the recognition system. The variety of information in these face images makes face detection difficult. For example, according to Yang et al.,(2002), some of the conditions that should be accounted for, when detecting faces are: Pose (Out-of Plane Rotation), Presence or absence of structural components such as beards, mustaches and glasses, Facial expression, Occlusion (i.e. some parts of human faces are unobserved, especially the facial features), Orientation (In Plane Rotation i.e. face appearance directly varies for different rotations about the camera's optical axis), Imaging conditions such as lighting, camera characteristics and resolution, Face localization, Facial feature extraction, Face recognition (identification), Facial expression recognition, Human pose estimation and tracking, etc.

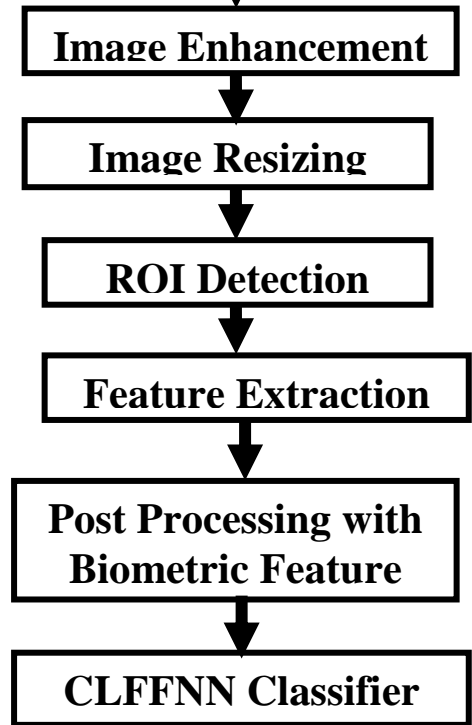
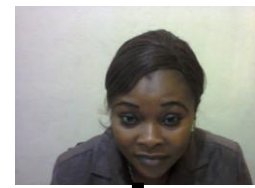


Figure 2: Enhanced Extracted Face Method (EEF) Of Face Image Preprocessing

### 3.4 Face Recognition

Face recognition is an interesting and successful application of Pattern recognition and Image analysis. Facial images are essential for intelligent vision-based human computer interaction. Face processing is based on the fact that the information about a user's identity can be extracted from the images and the computers can act accordingly. To build fully automated systems, robust and efficient face detection algorithms are required. The face is detected once a person's face comes into a view. Once a face is detected, the face region is cropped from the image to be used as "Probe" into the knowledge to check for possible matches. The face image is pre- processed for factors such as image size and illumination and to detect particular features. The data from the image is then matched against the knowledge. The matching algorithm will produce a similarity measure for the match of the probe face into the knowledge base.

## 4. FINGERPRINTS

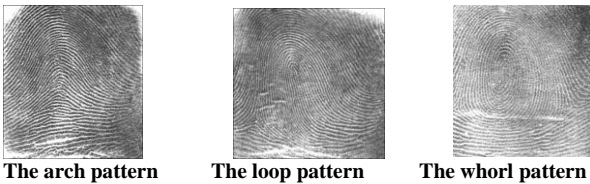
A fingerprint is the feature pattern of one finger. A fingerprint pattern is characterized by a set of ridgelines that often flow in parallel, but intersect and terminate at some points. The uniqueness of a fingerprint is determined by the local ridge characteristics and their relationships [Hong, L., et al (1998), Lee, H.C. & Gaensslen, R.E. (1991)]. It is believed with strong evidences that each fingerprint is unique.

### 4.1 Fingerprint Features

The prominent local ridge is referred to as minutiae. This characteristic of the finger print image is used to compare an individual’s finger image with the others stored finger images. The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These include Patterns and Minutia features.

(a) **Patterns:** Patterns are the aggregate characteristics of ridges. The three basic patterns of fingerprint ridges are the arch, loop and whorl as seen in figure 3.

- ✚ **Arch:** An arch is a pattern where the ridges enter from one side of the finger, rise in the center forming an arc, and then exit the other side of the finger.
- ✚ **Loop:** The loop is a pattern where the ridges enter from one side of a finger, form a curve, and tend to exit from the same side they enter.
- ✚ **Whorl:** In the whorl pattern, ridges form circularly around a central point on the finger. Scientists have found that family members often share the same general fingerprint patterns, leading to the belief that these patterns are inherited.



**Figure 3: The Three Basic Patterns of Fingerprint Ridges (Langenburg G., 2005)**

(b) **Minutia Features:** Minutiae are the unique features found within the patterns. The three major Minutia features of fingerprint ridges are:

- ✚ **Ridge ending:** This is the abrupt end of ridge or the point at which a ridge terminates.
- ✚ **Ridge Bifurcations:** These are points at which a single ridge splits into two ridges.
- ✚ **Short ridges (or dots)/independent ridge:** These are ridges which are significantly shorter than the average ridge length on the fingerprint.



**Figure 4: The three major Minutia features of fingerprint ridges**

Other forms as shown in figure 5 include:

- **Island:** This is a single small ridge inside a short ridge or ridge ending that is not connected to all other ridges
- **Ridge enclosure:** This is a single ridge that bifurcates and reunites shortly afterward to continue as a single ridge
- **Spur:** This is a bifurcation with a short ridge

branching off a longer ridge.

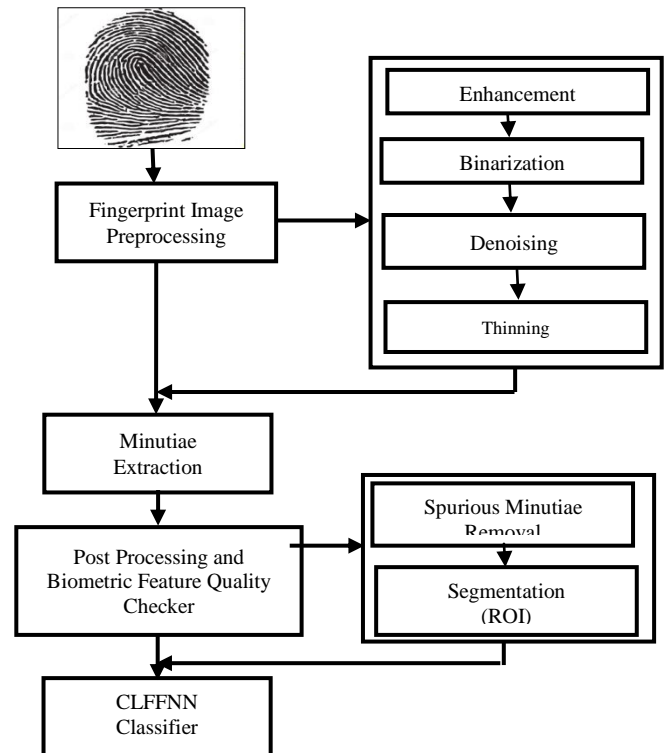
- **Crossover or bridge:** This is a short ridge that runs between two parallel ridges.
- **Delta:** This is a Y-shaped ridge meeting.
- **Core:** This is a U-turn in the ridge pattern.



**Figure 5: A Fingerprint [Reetu Awasthi and Ingolikar R.A. (2013)]**

### 4.2 Fingerprint Preprocessing

Noises (unwanted images) are inevitably present during the scanning process of the fingerprint. Preprocessing refers to the process of preparing the input fingerprint image to be ready for the next step of the system by removing noises and enhancing the fingerprint rigid patterns. This produces a good enough quality of output fingerprint image. Most techniques used in the preprocessing stage of this work are developed by other researchers but they have been modified through various tests. In this work, we use a preprocessing method called Plainarized Region of Interest Method (PROI) as shown in figure 6.



**Figure 6: Plainarized Region of Interest Method (PROI) of Fingerprint Image Preprocessing**

## 5. CASCADED LINK FEED-FORWARD NEURAL NETWORK [CLFFNN]

After preprocessing and extracting the features from the given face or fingerprint image using Enhanced Extracted Face (EEF) method and Plainarized Region of Interest (PROI) method respectively, a recognizer is needed to recognize the face or fingerprint image from the stored database[5]. Cascaded Link Feed-Forward Neural Network [CLFFNN] is a hierarchical cascade of neural network ensembles to enhance detection accuracy and efficiency. It uses a two neural network classifiers -CLFFNN(1) and CLFFNN(2) to form a neural network ensemble. Each classifier is specialized in a sub region in the finger and face-pattern space. they are organized in a pruning cascade to reduce the total computation cost of face detection. Our work proposes a verification method, which uses two networks: CLFFNN(1) and CLFFNN(2) as seen in figure 7.

### 5.1 CLFFNN(1)

CLFFNN(1) is a neural network that uses Back Propagation algorithm. Back propagation can train multilayer feed-forward networks with differentiable transfer functions to perform function approximation, pattern association, and pattern classification. The CLFFNN(1) is designed with one input layer, one hidden layer and one output layer. The input layer consists of neurons with a sigmoid activation function. The inputs to this network are feature vectors derived from the enhanced extracted face (EEF) feature extraction method. The classifier CLFFNN(1) is used to explore face candidate locations and try to filter as many non-face patterns as possible before passing hard patterns to the final stage classifier; it means false rejection must be avoided at any cost. Here, a majority of non-face patterns in image backgrounds are rejected by this network by improving the overall detection efficiency while maintaining detection accuracy.

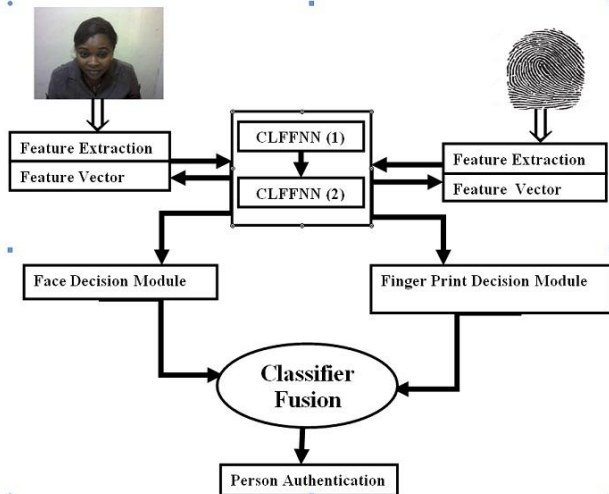


Figure 7: Proposed Intelligent Multibiometric System

The CLFFNN(1) has two cascaded stages as seen in figure 8. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face; or whether a finger scan is a life scan. When a sub-window is classified to be a non-face or not a life scan by a given stage, it is immediately discarded. Conversely a sub-window classified as a maybe-face or maybe-live scan is passed on to the next stage in the cascade. This sub window passed through each

section of image and minimizes the false positive rate (i.e. non faces for example hand) and detects the face. This is repeated until the face/finger is found. In this method the false positive rate is very low as compared to others. After applying these steps and obtaining the desired feature, the height and width of image is calculated. The time required to detect the fingerprints and faces in facial images from dataset are also computed for finding time complexity of fingerprint and face detection using sequential algorithm.

Because of the property of CLFFNN, we can avoid false rejection as long as we give CLFFNN the enough training data during the learning stage. In other words, the training set must be ideally and identically distributed in the sample space. The network is trained using the samples, see figure 10.

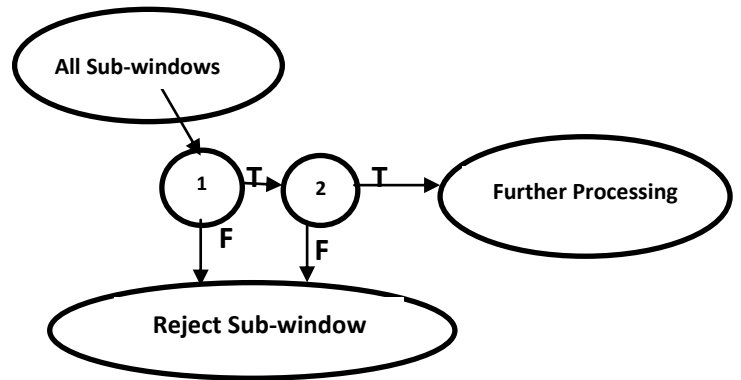


Figure 8: Two Cascaded Stages of CLFFNN(1)

The output layer contains one neuron. The simulated result obtained from the output layer is given as the input to the CLFFNN(2) for the function approximation.

### 5.2 CLFFNN(2)

CLFFNN(2) uses the gaussian function for approximation. The output of CLFFNN(1) is approximated since it is connected with CLFFNN(2). CLFFNN(2) are universal approximators and have a very fast learning speed due to their locally tuned neurons. They have a very compact topology. CLFFNN(2) is used as recognizer in face recognition system and the inputs to this network are the results obtained from the CLFFNN(1). It has a feed forward architecture with an input layer, a hidden layer and an output layer. The output from the CLFFNN(2) is considered as the recognition result.

During training, the partially-trained system is applied to images of scenery which do not contain faces. Any region in the image detected as faces are errors, which can be added into the set of negative training examples.

## 6. THE BACK PROPAGATION ALGORITHM AS USED IN CLFFNN

Back propagation is a feed forward supervised learning network. The general idea with the backpropagation algorithm is to use gradient descent to update the weights to minimize the squared error between the network output values and the target output values. The update rules are derived by taking the partial derivative of the error function with respect to the weights to determine each weight's contribution to the error. Then, each weight is adjusted. This process occurs iteratively for each layer of the network, starting with the last set of weights, and working back towards the input layer, hence the name "backpropagation". The network is trained to perform





its ability to respond correctly to the input patterns that are used for training and to provide good response to input that are similar.

Initialization of the network is the first stage in Back Propagation Algorithm. This is done by setting up all its weights to be small random numbers – say between  $-1$  and  $+1$ . Next, the input pattern is applied and the output calculated. This is called the *forward pass*. Normally, the calculation at the first instance will always give an output which is completely different from the expected i.e. the Target, because of the random weights. Then the Error of each neuron is calculated as;

**Error = Target – Actual Output**

This error is then used mathematically to change the weights in such a way that the error will get smaller i.e. the Output of each neuron gets closer to its Target. This is called the *reverse pass*. The process is repeated severally until the error is minimal.

The connection we are interested in is between neuron X (a hidden layer neuron) and neuron Y (an output neuron) and has the weight  $W_{XY}$ .

The Back propagation training takes place in four stages:

a) **Feed forward of input training pattern:** Here, we apply the inputs to the network and work out the output. Each input neuron receives an input value and broadcasts it to each hidden neuron, which in turn computes the activation and passes it on to each output unit, which again computes the activation to obtain the net output. During training, the net output is compared with the target value and the appropriate error is calculated.

b) **Back propagation of the associated error:** The error factor is obtained which is used to distribute the error back to the hidden layer. Here, we work out the error for neuron Y. The error is *What you want – What you actually get*.

**Error<sub>Y</sub> = Output<sub>Y</sub> (1-Output<sub>Y</sub>)(Target<sub>Y</sub> – Output<sub>Y</sub>)**

The “*Output(1-Output)*” term is necessary in the equation because of the Sigmoid Function.

c) **Weight adjustment:** A weight in a neural network is a segment of the information about the input signal that has to be stored. Here, we change the weight. If  $W_{XY}(\text{new})$  is the new (trained) weight and  $W_{XY}$  is the initial weight, then

$$W_{XY}(\text{new}) = W_{XY} + (\text{Error}_Y \times \text{Output}_X)$$

Normally, we use the output of the connecting neuron, neuron X not Y. We update all the weights in the output layer in this way.

d) **Calculate the Errors for the hidden layer neurons:** This cannot be calculated directly unlike the output layer (because we don't have a Target), so we *Back Propagate* them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron X is connected to Y and Z then we take the errors from Y and Z to generate an error for X.

**Error<sub>X</sub> = Output<sub>X</sub> (1 - Output<sub>X</sub>)(Error<sub>Y</sub>  $W_{XY}$  + Error<sub>Z</sub>  $W_{XZ}$ )**

Having obtained the Error for the hidden layer neurons now proceed as in stage three to change the hidden layer weights.

By repeating this method we can train a network of any number of layers. The network keeps training all the patterns repeatedly until the total error falls to some pre-determined low target value and then it stops. The final error is calculated as the sum of all the individual neuron errors for each pattern and once the network has been trained, it should be able to recognize not just the perfect patterns, but also corrupted or noisy versions.

The use of Validation Set is a better way of working out when to stop network training. This prevents network overtraining (i.e. network becoming too accurate, which can lessen its performance). This is done by having a second set of patterns which are noisy versions of the training set (but aren't used for training themselves). Each time after the network has trained; this set (called the Validation Set) is used to calculate an error. When the error becomes low the network stops. When the network has fully trained, the Validation Set error reaches a minimum. When the network is overtraining the validation set error starts rising. If the network overtrains, it won't be able to handle noisy data so well.

## 6.1 Back Propagation Algorithm in CLFFNN for Face Recognition

Face recognition involves two major tasks: finding face in an image and face recognition and verification. Face recognition problems can be in two versions: recognition and verification. In Face verification involves testing the quality of match of an image against a single model while face recognition involves finding the best match for an unknown image against a database of face models or to determine whether it does not match any of them well. The practical importance of this distinction has to do with the speed required. Generally if there are N subjects in the database then the recognition process will be N times slower than the verification process. This may place practical limits on the algorithm used.

Unfortunately, developing a computational model of face recognition developing is a quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli. The approach has advantages over other face recognition schemes in its speed and simplicity, learning capacity, and insensitivity to small or gradual changes in the face image.

## 6.2 Face Learning and Training using CLFFNN

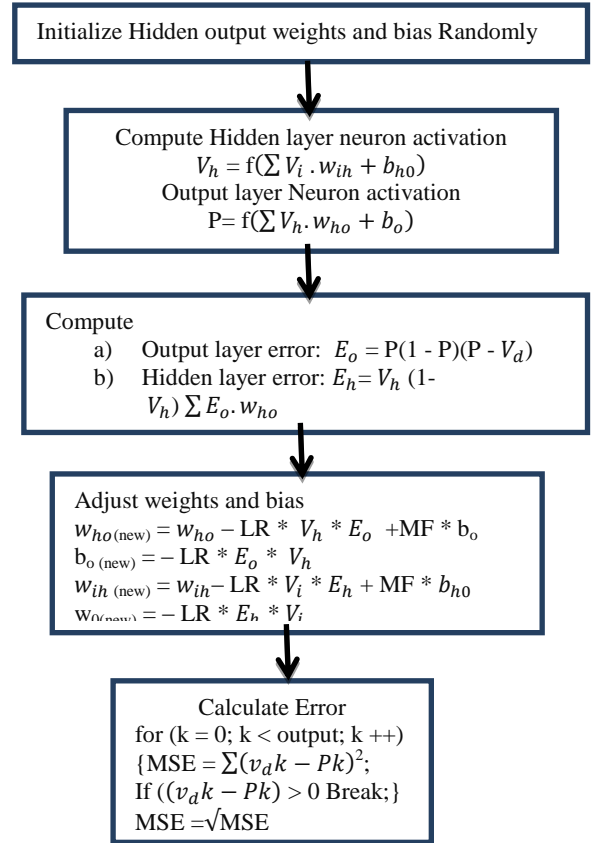
Two networks were used. The first network CLFFNN(1) has a single output, and like our system it is trained to produce a positive value for centered faces, and a negative value for non-faces. This network scans over the image to produce candidate face locations. The network must be applied at every pixel position, but it runs quickly because of its architecture: using facial connections and shared weights, much of the computation required for one application of the detector can be reused at the adjacent pixel position. This optimization requires the preprocessing to have a restricted form, such that it takes as input the entire image, and produces as output a new image.

When the samples are captured, they are divided into a disjointed Clutter class and the Target class. The main idea is to apply a linear projection followed by a threshold (in each iteration). The projection vector and the corresponding thresholds are chosen such that at each iteration the number of rejected Clutter samples are maximized. This means that after the first classification iteration, many of the Clutter samples

are already classified as such, and discarded from further consideration. The process is continued with the remaining Clutter samples, again searching for a linear projection vector and thresholds that maximizes the rejection of Clutter points from the remaining set. This process is repeated iteratively until convergence to zero or a small number of Clutter points. The remaining samples at the final stage are considered as targets. Having segmented every image into several segments and approximated every segment with a small number of representative pixels, we can exhaustively search for the best combination of segments that will reject the largest number of non-face images. We repeat this process until the improvement in rejection is negligible.

After preprocessing on the image, we pass it to the CLFFNN(1) network which is used to explore face candidate locations and try to filter as many non-face patterns as possible before passing hard patterns to the final stage classifier CLFFNN(2); it means false rejection must be avoided at any cost. Because of the property of CLFFNN, we can avoid false rejection as long as we give CLFFNN the enough training data in learning stage. In other words, the training set must be ideally and identically distributed in the sample space.

The employed multilayer feed forward neural network with back propagation algorithm consists of neurons with a sigmoid activation function. It is used in two modes. In classification mode, it is presented at the input layer and propagated forward through the network to compute the activation value for each output neuron. The second mode is called the training or learning mode. Learning in CLFFNN involves the adjustment of the weights in order to achieve the desired processing for a set of learning face samples. More specifically, the second mode includes feeding a neural network with a number of training pairs. Then the networks parameters are adjusted through a supervised training algorithm. Then the network is simulated using the features taken from the test set of images. The simulated result is given as the input to the CLFFNN(2) for the function approximation, as seen in figure 9. The output from the CLFFNN is considered as the recognition result.



**Fig 9: Face Learning and Training using CLFFNN**

Where

$V_i$  = The vector of input layer neurons

$V_h$  = The vector of the hidden layer neurons

$P$  = represents the output layer neurons

$w_{ih}$  = the weight matrix between the input and the hidden layer

$b_{h0}$  = the bias of the hidden layer neurons

$w_{ho}$  = weight matrix connecting the hidden and the output layers

$b_0$  = the bias of the output layer neurons

$E_o$  = the error vector for output neurons

$E_h$  = the error vector of each hidden layer neuron

$V_d$  = the desired output vector

LR = learning rate

MF = momentum factor

The sigmoid activation function is defined by

$$f(V_i) = \frac{1}{1 + \exp(-V_i)}$$

### 6.3 Face Learning Algorithm

The following is the algorithm used in Face Learning:

1. Create an initial set of non-face images by generating 30 random images. Apply the preprocessing steps to each of

these images.

2. Train a neural network to produce an output of 1 and -1 for face and non-face examples respectively. Randomly initialize the network's weights on the first iteration of this loop and after the first iteration, use the weights computed by training in the previous iteration as the starting point.
3. Run the system on an image of scenery which contains no faces. Collect sub images in which the network incorrectly identifies a face (an output activation > 0).
4. Select up to 50 of these sub images at random, apply the preprocessing steps, and add them into the training set as negative examples. Go to Step 2.
5. The neurons use the then activation function, which gives an output ranging from -1 to 1, hence the threshold of 0 for the detection of a face. Since we are not training with all the negative examples, the probabilistic arguments of the previous section do not apply for setting the detection threshold. Since the number of negative examples is much larger than the number of positive examples, uniformly sampled batches of training examples would often contain only negative examples, which would be inappropriate for neural network training. Instead, each batch of positive and negative examples is drawn randomly from the entire training sets, and passed to the back propagation algorithm as a batch. We choose the training batches such that they have 50% positive examples and 50% negative examples. This ensures that initially, when we have a much larger set of positive examples than negative examples, the network will actually learn something from both sets. Note that this changes the distribution of faces and non faces in the training sets compared with what the network will see at run

## 6.4 Fingerprint Image Matching

When the above process has been applied, we get the feature matrix of each fingerprint image. These feature matrixes are used as input data set to the neural network. Based on average feature value of fingerprint image, we fix target output. After that we train the network for each fingerprint image. A program is developed to train the neural network for fingerprint recognition system, which takes the input data from the input file and reduce the error between the accepted output and the actual output.

We stop the training process immediately the weight file is generated by the error and the accepted output for our fingerprint recognition system is gotten. The training process will generate the weight file, which contains the weights of the neurons of the network. The weight file will be needed for testing the fingerprint identification system. Then we simulate the network with the sample of fingerprint image and get a desired output. Error is calculated from the distance between target and desired output value. Then the weights are adjusted to reduce this error.

The back-propagation algorithm is sensible approach to dividing the contribution of each weight. As in the network-training algorithm, it tries to minimize the error between each target output and the output actually computed by network. The size of the training vector sets was expanded from its initial position. The minutiae positions used to make up the training and testing sets were randomly separated to test the

generalization properties of these back propagation networks. Training is performed with a variety of sub-sampling strategies. The generalization properties of these networks are then tested by applying the testing data sets. Finally selected high performance networks were fully tested by convolving them with the entire image.

## 6.5 Back Propagation and Network Training in Fingerprint Recognition System

The Cascaded Link Feed Forward Neural Network [CLFFNN] is a multi-layered, feed forward Neural Network that is fully interconnected by layers. Thus, there are no connections that bypass one layer to go directly to a later layer. The network is called a mapping network because it is able to compute some final relationship between its input. It is trained with a back propagation algorithm. Figure 10 shows a layer of this network.

The input vector is represented by  $V[x][a]$ , where  $x$  is the fingerprint Mr.  $V$  or any other person in question and  $a$  is the pattern matrices,  $16 \times 16$  array ( $a = 0, 1, 2, 3, \dots, 256$ ). The target output is represented by  $G[x][a]$ .

The learning of this network has been accomplished by back propagation algorithm. This learning rule trains the network as described below:

The weight vectors  $W_{ab}$  and  $W_{bc}$  and the threshold values for each PE in the network are initialized within random numbers. The network is provided with the input pattern and also the desired respective output patterns. The input patterns are connected to hidden (PEs) through the weights  $W_{ab}$ . In the hidden layer, each PE computed the weighted sum according to the equation which is given by

$$N_{x_b} = \sum W_{ab} I_{x_a} \quad (1)$$

where  $I_{x_a}$  is the input of unit  $a$  for pattern number  $x$ . The threshold of each PE was then added to its weighted sum to obtain the activation  $atv(b)$  of that PE therefore

$$atv_b = N_{x_b} + yf_b \quad (2)$$

When  $yf_b$  is the hidden threshold weight for  $b$ th PEs. This activation determined whether the output of the respective PE was either 1 or 0. [fires or not] by using a sigmoid function,

$$I_{x_b} = 1/(1 + e^{-c1*atvb}) \quad (3)$$

where  $C1$  is called the spread factors, these  $I_{x_b}$  were then served as the input to the output computation. Signal  $I_{x_b}$  were then fan out to the output layer according to the relation.

$$N_{x_c} = \sum W_{bc} I_{x_b} \quad (4)$$

and the output threshold weight  $yI_c$  for  $c$ th output PEs is added to it to find out the activation  $atvI_c$ .

$$atvI_c = N_{x_c} + yI_c \quad (5)$$

The actual output  $I_{x_c}$  is computed using the same sigmoid function, which is

$$I_{x_c} = 1/(1 + e^{-c2*atvIc}) \quad (6)$$

Here another spread factor  $C2$  is employed for the output units. In the second stage, after completing the feedforward propagation, an error is computed by comparing the outputs  $I_{x_c}$  with the respective target  $tx_c$ , i.e.

$$\delta x_c = tx_c - I_{x_c} \quad (7)$$



This error is then used to adjust the weight vector  $W_{bc}$  using the equation:

$$\Delta W_{bc} = \eta_2 C_2 \delta x_c I_x I_c (1 - I_x) \quad (8)$$

where  $\int(atv|c) = C_2 I_x (1 - I_x)$ , the derivation of sigmoid function. The weight vector  $W_{bc}$  is then adjusted to  $W_{bc} + \Delta W_{bc}$ . For the threshold weight of the output PE, similar equation is employed.

$$\Delta y I_c = \eta_2 C_2 \delta x_c I_x (1 - I_x) \quad (9)$$

and the new threshold weight equaled  $y I_c + \Delta y I_c$ .

In the next step, this error and the adjusted weight vector  $W_{bc}$  are feedback to the hidden layer to adjust the weight vector  $W_{ab}$  and threshold weight  $y f_b$ . In this layer, change in weight vector  $W_{ab}$  is computed by using equation,

$$\Delta W_{ab} = \eta_a C_a I_x I_b (1 - I_x) \sum \delta x_c W_{bc} \quad (10)$$

Where  $\int(atvf_b) = C_a I_x (1 - I_x)$ .

The weight vector  $W_{ab}$  is then adjusted to  $W_{ab} + \Delta W_{ab}$ . For the threshold weights of the hidden PEs, similar equation is employed.

$$\Delta y f_b = \eta_a C_a (1 - I_x) I \delta x_c W_{bc} \quad (11)$$

and new threshold weights are calculated.

$y f_b + \Delta y f_b$

The properties of sum-squared error equation dictate that as output approaches its maximum or minimum value, adjustments to individual weights become less pronounced. This is a testament to the stability of the back propagation algorithm. The significance of the training process is that, as the network trains, the nodes in the intermediate layers organized themselves such that different nodes learn to recognize different features of the total input space.

## 6.6 Back Propagation Algorithm in CLFFNN(1) For Fingerprint Recognition

**Input:** The inputs to the network include:

- i) the training samples -- samples;
- ii) the learning rate --  $\eta$ ;
- iii) a multilayer feed-forward network -- network.

**Output:** The output is a neural network trained to classify the samples.

**Method:**

Initialize all weights and biases in network;  
 while terminating condition is not satisfied {  
 for each training sample  $V$  in sample {  
 // Propagate the input forward:  
 for each hidden or output layer unit  $b$  {  
 $P_b = \sum a W_{ab} I_a + \Theta_b$ ; //compute the set input of unit  $b$   
 with respect to the previous layer,  $a$   
 $I_b = 1 / (1 + e^{-P_b})$ ; } // compute the output of each unit

$b$

// Backpropagate the errors:  
 for each unit  $b$  in the output layer  
 $Err_b = I_b (1 - I_b) (G_b - I_b)$ ; // compute the error  
 for each unit  $b$  in the hidden layers, from the last to the first hidden layer

$Err_b = I_b (1 - I_b)$  // compute the error with respect to the next higher layer,  $k$

for each weight  $W_{ab}$  in network {  
 $\Delta W_{ab} = (1) Err_b I_a$ ; // weight increment  
 $W_{ab} = W_{ab} + \Delta W_{ab}$ ; } // weight update  
 for each bias  $\Theta_b$  in network {  
 $\Delta \Theta_b = (1) Err_b$ ; // bias increment  
 $\Theta_b = \Theta_b + \Delta \Theta_b$ ; } // bias update  
 } }

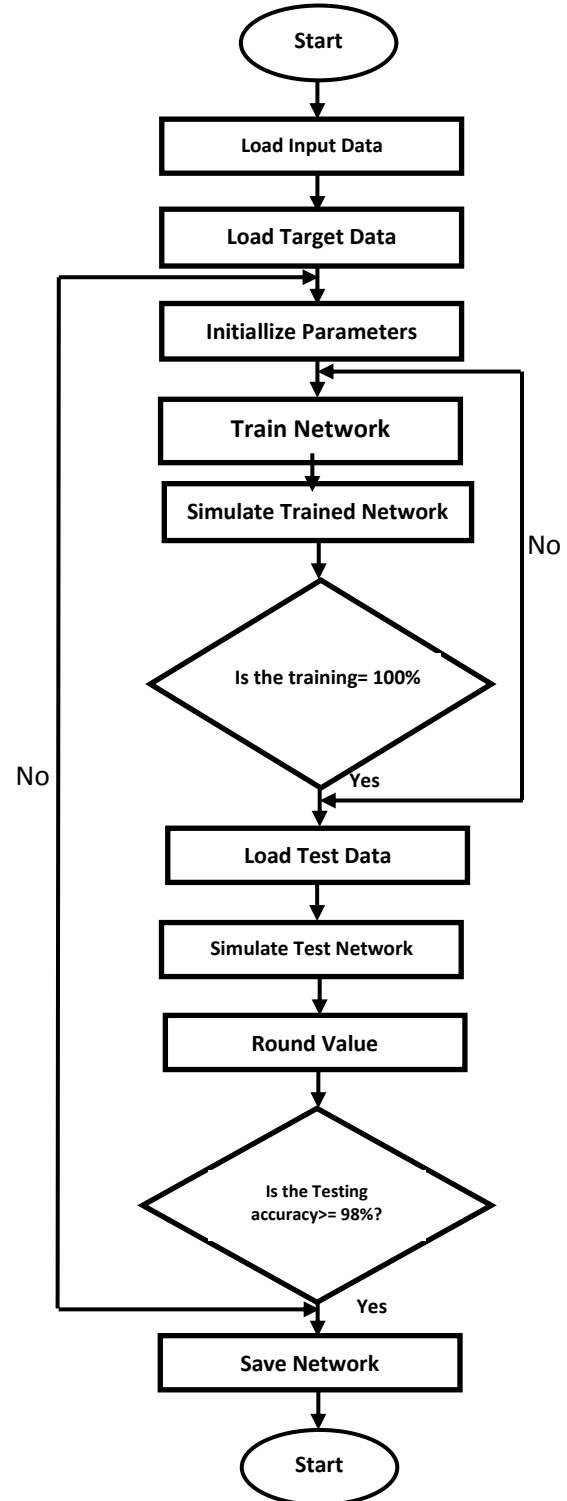


Figure 10: Flowchart of Training and Testing



## 7. IMPLEMENTATION AND VALUATION

The system is implemented using vb.net on a visual studio 2010 and MySQL as relational database. At first the neural network architecture was created and then trained with training set of faces and non-faces. In the evaluation process, a model of a newly acquired test subject is compared against all existing models in the database and the most closely corresponding model is determined. If these are sufficiently close, a recognition event is triggered.

## 8. TESTING

Different faces and fingerprints were used to assess the performance of the system. For each person, one fingerprint with acceptable quality was selected as the template. All but one face images were used to train the face recognition subsystem. The remaining face image was paired with the fingerprint to form a test sample. This process was repeated several times.

## 9. RESULTS

The identification accuracy of the integrated system as well as identification accuracies of face recognition and fingerprint identification is compiled in table 1. The response time of the integrated system for a typical identification is 3 seconds see table 2. In comparison, identification using only fingerprints on a database of 5 persons takes 9 seconds. From this, the system can achieve desirable identification accuracy with an acceptable response time. This was developed and run on a machine with following configuration; Intel core i5 2.5 GHZ processor with window 7 32bit operating system, 2 GB RAM.

**Table 1: Identification accuracy of face recognition and fingerprint identification**

| No. of fingerprints | No. of Faces | No. of faces detected at each compare | No. of fingerprints detected at each compare | Accuracy (%) |
|---------------------|--------------|---------------------------------------|--|--------------|
| 15                  | 15           | 1                                     | 1  | 99           |

**Table 2: Response time of the system**

| Face Location (Seconds) | Face Retrieval (Seconds) | Fingerprint Verification (Seconds) | Total (Seconds) |
|-------------------------|--------------------------|------------------------------------|-----------------|
| 0.5                     | 0.5                      | 2.0                                | 3.0             |

## 10. CONCLUSION

Multi-biometrics is a new and exciting area of information science research for accurate and reliable personal authentication. A good combination of multiple biometric traits can be used to overcome the limitations in unimodal biometrics. This system has successfully integrated and implemented face and fingerprint technologies for personal recognition. The system overcomes some limitations of face recognition and fingerprint verification systems. Experimental results demonstrate that the system performs well. It meets the response time as well as the accuracy required.

## 11. REFERENCES

[1] Avinash K., Raina J. P. S. (2010) “Face Detection using Neural Network & Gabor Wavelet Transform”, International Journal of Computer Science and

Technology (IJCT), Vol. 1, Issue.1, pp58-63, September 2010, ISSN : 0976 - 8491.

- [2] Avinash Pokhriyal and Sushma Lehri, (2010) “A new method of fingerprint authentication using 2D wavelets,” Journal of Theoretical and Applied Information Technology
- [3] Avinash Pokhriyal et. al., (2010) “MERIT: Minutiae Extraction using Rotation Invariant Thinning”, International Journal of Engineering Science and Technology Vol. 2(7), 2010, 3225-3235
- [4] Belghini N., Zarghili A., Kharroubi J. and Majda A.,(2011) Sparse Random Projection and Dimensionality Reduction Applied on Face Recognition, in The Proceedings of International Conference on Intelligent Systems & Data Processing, January 2011,pp.78-82.
- [5] Benson-Emenike M.E and Nwachukwu E.O.(2015) AN Efficient Image Preprocessing In An Improved Intelligent Multi Biometric Authentication System. International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 9 – No.6, September 2015 – www.ijais.org
- [6] Beszedes M. & Oravec M. (2014) “A System For Localization Of Human Faces In Images Using Neural Networks”, Journal Of Electrical Engineering, Vol. 56, No 7-8, pp195–199. The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.1 Facial expression recognition and synthesis based on an appearance model”, Signal Processing: Image Communication, Vol. 19, Issue 8, pp723-740.
- [7] Bouzalmat A., Zarghili A., Kharroubi J.,(2011) “Facial Face Recognition Method Using Fourier Transform Filters Gabor and R\_LDA”, IJCA Special Issue on Intelligent Systems and Data Processing, pp.18-24, 2011.
- [8] Chatterjee A., Mandal S., Atiqur Rahaman G. M., and Arif A. M.,(2010) “Fingerprint Identification and Verification System by Minutiae Extraction Using Artificial Neural Network”, ISSN 2078-5828 (PRINT), ISSN 2218-5224 (Online), Volume 01, Issue 01, Manuscript Code: 100703.,
- [9] Devika Chhachhiya, Amita Sharma and Manish Gupta (2013) Recapitulation on Transformations in Neural Network Back Propagation Algorithm International Journal of Information and Computation Technology ISSN 0974-2239 Volume 3, Number 4, pp. 323-328 © International Research Publications Househttp://www.irphouse.com/ijict.htm
- [10] Erik Hjelm and Boon Kee Low (2001) Face Detection: A Survey Computer Vision and Image Understanding 83, 236–274 (2001) doi:10.1006/cviu.2001.0921, available online at http://www.idealibrary.com
- [11] Farhat Anwar, et al. (2008), "Multibiometric Systems Based Verification Technique" Faculty of Engineering, Department of ECE International Islamic, University Malaysia.
- [12] Hazem M. El-Bakry (2002), Face Detection Using Neural Networks and Image Decomposition Lecture Notes in Computer Science Vol. 22, pp:205-215.



- [13] Henry Rowley, Baluja S. & Kanade T. (1999) “Neural Network-Based Face Detection, Computer Vision and Pattern Recognition”, Neural Network-Based Face Detection, Pitts-burgh, Carnegie Mellon University, PhD thesis
- [14] Henry A. Rowley, Shumeet Baluja & Takeo Kanade (1997) Rotation Invariant Neural Network-Based Face Detection, December, CMU-CS-97-201
- [15] Iwasokun G. B. et al (2012) Fingerprint Image Enhancement: Segmentation to Thinning. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1
- [16] Krishna Prasad P. E. S. N., Pavan Kumar K, Ramakrishna M. V. and Prasad B. D. C. N. (2013) Fusion Based Multimodal Authentication In Biometrics Using Context-Sensitive Exponent Associative Memory Model : A Novel Approach Computer Science & Information Technology (CS & IT) Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA – 2013 pp. 81–90, 2013. © CS & IT-CSCP
- [17] Le Cun, V. et al., (2012). “Handwritten digit recognition with a back propagation network”, Neural Information Processing Systems, Vol. 2, pp. 396-404.
- [18] Lourde M.R. and Khosla D. (2010) Fingerprint Identification in Biometric Security Systems International Journal of Computer and Electrical Engineering, Vol. 2, No. 5, October, 2010 1793-8163
- [19] Mansaf M Elmansori & Khairuddin Omar (2011) “An Enhanced Face Detection Method Using Skin Color and Back-Propagation Neural Network”, European Journal of Scientific Research, ISSN 1450- 216X, Vol.55 No.1, pp80-86, <http://www.eurojournals.com/ejsr.html>
- [20] Marasco E., (2010) “Secure Multibiometric Systems” PhD thesis submitted to University of Naples Federico
- [21] Mohammad Abadi, et al, (2011) “Face Detection with the Help of Gabor Wavelets Characteristics and Neural Network Classifier”, American Journal of Scientific Research, Issue.36, pp67-76, ISSN 1450-223X, <http://www.eurojournals.com/ajsr.htm>
- [22] Mohammad Alia, Abdelfatah Tamimi and Omaira Al-Allaf, (2013) "Integrated System For Monitoring And Recognizing Students During Class Session", AIRCC's: International Journal Of Multimedia & Its Applications (IJMA), Vol.5, No.6, December, pp:45-52. [aircse.org/journal/ijma.html](http://aircse.org/journal/ijma.html)
- [23] Mohammadi S., Frajzadeh A. (2009) A Matching Algorithm of Minutiae for Real Time Fingerprint Identification System, World Academy of Science, Engineering and Technology 60 2009.
- [24] Mohammed S Khalil et al. ( 2010) “Fingerprint Verification Based on Statistical Analysis”, IEEE,.
- [25] Muhammad Imran Razzak, Rubiyah Yusof and Marzuki Khalid (2010): “Multimodal face and finger veins biometric authentication of Scientific Research and Essays “, Vol. 5(17), pp. 2529-2534, 4 September, 2010.
- [26] Nayak P.K. and Narayan D. (2013) Multimodal Biometric Face and Fingerprint Recognition Using Adaptive principal Component Analysis and Multilayer Perception. International Journal of Research in Computer and Communication Technology, Vol 2, Issue 6, June-2013.
- [27] Rashmi S. & Payal J., (2012), “Multi Biometric System: Secure Security System “IJREAS Volume 2, Issue 2 ISSN: 2249-3905 International Journal of Research in Engineering & Applied Sciences 182 <http://www.euroasiapub.org>
- [28] National Biometric Security Project, (2008), "Biometric Technology Application Manual" Volume One: Biometric Basics, Updated Summer.
- [29] Phil Brimblecombe (2002) “Face Detection using Neural Networks”, H615 – Meng Electronic Engineering, School of Electronics and Physical Sciences, URN: 1046063
- [30] Prasad K.N. (2013) “Face Detection using Neural Network”, International Journal of Computer Applications (0975 – 8887), Vol.1, No.14, pp36-39.
- [31] Raj A., Bincy G., Mathu T., (2012)” Survey on Common Data Mining Classification techniques”, International Journal of Wisdom Based Computing, Vol. 2, Issue 1, April.
- [32] Ravi Kumar R.L., Kumar S.S., Prasad J. R., Rao S.B.V., Prakash P.R.(2012) “ Fingerprint Minutia Match Using Bifurcation Technique”, S Sai Kumar et al , International Journal of Computer Science & Communication Networks, Vol 2(4), 478-486, Sep. 2012.
- [33] Reetu Awasthi and Ingolikar R.A.(2013) A Study Of Biometrics Security System International Journal Of Innovative Research & Development April, 2013 Vol 2 Issue 4
- [34] Ritu, Matish Garg(2014) A Review on Fingerprint-Based Identification System International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 3, March 2014 Copyright to IJARCC [www.ijarcc.com](http://www.ijarcc.com) 5849
- [35] Sahoolzadeh, Sarikhanimoghadam and Dehghani (2008) “Face Detection using Gabor Wavelets and Neural Networks”, World Academy of Science, Engineering and Technology, Vol. 45, pp552- 554.
- [36] Salim Lahmiri, (2011) “A Comparative Study Of Backpropagation Algorithms In Financial Prediction”, International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.4
- [37] Sangita K Chaudhari, (2012) “An algorithm for fingerprint enhancement & matching”, National Conference on Emerging Trends in Engineering & Technology (VNCET-30 Mar'12)
- [38] Sasan Golabi, Saiid Saadat, Mohammad Sadegh Helfroush, and Ashkan Tashk, (2012) “A Novel Thinning Algorithm with Fingerprint Minutiae Extraction Capability”, International Journal of Computer Theory and Engineering, Vol. 4, No. 4, August 2012
- [39] Sawarkar S.D., Shubhangi Vaikole, , Shila Hivrale, Taruna Sharma, (2009) “Minutiae Extraction from Fingerprint Images”, IEEE International Advance Computing Conference, pp. 691-696, 2009.



- [40] Shubhangi D C, Manohar Bali (2012): “Multi-Biometric Approaches to Face and Fingerprint Biometrics” International Journal of Engineering Research & Technology Vol. 1 Issue 5 ISSN: 2278-0181, July - 2012.30274, 2006.
- [41] Simao-Zortia D. et al.,( 2003) “Image Quality and position variability assessment in minutiae-based fingerprint verification”, IEEE proceedings , , vol. 150 pp 402-408.
- [42] Singh H. & Gayathri R., (2012), “Image Authentication Technique Using Fsim Algorithm” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 2, pp.1129-1133. www.ijera.com
- [43] Stefan W., Christian I. & Uwe H., (2004) “Evolutionary Optimization of Neural Networks for Face Detection”, Proceedings of the 12th European Symposium on Artificial Neural Networks, Evere, Belgium: d-side publications.
- [44] Tran Binh Long, Le Hoang Thai (2012) Hybrid Multi-Biometric Person Authentication System Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I WCECS 2012, October 24-26, 2012, San Francisco, USA.
- [45] Gualberto Aguilar, Gabriel Sanchez (2007) “Fingerprint Recognition”IEEE,.
- [46] Hudson, Hagan and Demuth, (2012) Neural Network Toolbox™ User’s Guide R2012a, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, , www.mathworks.com
- [47] Omaira N. A. AL-Allaf (2014) Review Of Face Detection Systems Based Artificial Neural Networks Algorithms The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.1, February 2014 DOI : 10.5121/ijma.2013.6101 1
- [48] Vijaya Sathiaraj (2012) “A Study on the Neural Network Model for Finger Print Recognition” International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5, Oct 2012
- [49] Yan Y. and Zang Y.(2011) “ Multimodal Biometrics Fusion Using Correlation Filter Bank”, IEEE, DOI-978-1-4244-2175-6
- [50] Zhang Q. and Zhang X.,(2010) “Research of Key Algorithm in the Technology of Fingerprint Identification,” Second IEEE International Conference on Computer Modeling and Simulation, pp. 282-284, 2010.
- [51] Zhao W. et al, (2000) “Face recognition: a literature survey”, Technical Report CAR-TR-948, University of Maryland, October 2000.
- [52] Zoran Bojkovic & Andreja Samcovic (2006) Face Detection Approach In Neural Network Based Method For Video Surveillance, 8th Seminar on Neural Network Applications in Electrical Engineering, Neurel, Faculty Of Electrical Eng., University Of Belgrade, Serbia, September 25-27.