ISSN:0975-9646
L. Ravi Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1618-1624

Genuine and Forged Offline Signature Verification Using Back Propagation Neural Networks

L. Ravi Kumar, A.Sudhir Babu
Department of Computer Science & Engineering, PVP Siddhartha Institute of Technology
Vijayawada, Andhra Pradesh, India

Abstract: **---The need to make sure that only the right people are authorized to access high-security systems has paved the way for the development of systems for automatic personal authentication. Handwritten signature verification has been identified as a main contender in the search for a secure personal verification system. Signatures in offline systems are based on the scanned image of the signature. A new approach for offline signature verification is proposed and implemented. The proposed signature authentication system functions based on global and texture features of a given signature sample. This method makes use of the global features pulled out from the skeleton of the signature. While legitimate signatures of the same person may show some differences over a period, the differences between a skilled forgery and an actual signature may be imperceptible. When a genuine sample is given for enrollment, the system will automatically train the network with statistics generated from the given samples. The Back propagation network used verifies the global features for validity. The result is a gray level co-occurrence matrix representation of the signature sample, which is obtained from the picture matrix of spatial or texture features extracted. Based on the values obtained the network will decide the appropriateness of the signature.**

***Keywords*--** **Preprocessing, Feature Extraction, Global Features, Texture Features, False Acceptance Rate, False Rejection Rate.**

I. INTRODUCTION

The need to make sure that only the right people are authorized to access high-security systems has paved the way for the development of systems for automatic personal authentication. Palm prints, Fingerprints, voice, and handwriting have all been used to verify the declared identity of an individual.

A. Problem Definition and Motivation

Signatures are a special case of hand writing in which special characters and flourishes are available. In many cases, the signature is not readable even by a human. Signature is a behavioral biometric. It is not based on physiological properties of the individual, such as fingerprint or face, but behavioral ones. As such one's signature may change over time and it is not nearly as unique or difficult to forge as iris patterns or fingerprints. However signature's widespread acceptance by the public makes it more suitable for certain lower-security authentication needs. Signature has a fundamental advantage in that it is the customary way of identifying an individual in daily operations such as automated banking transactions and electronic fund transfers. Signature analysis can only be applied when the person is/was conscious and disposed to write in the usual manner. To give a counter example, a person's fingerprint may also be used when the person is in an unconscious state of mind.

Signature verification [4] is split into two classes according to the available data in the input. *Offline (static)* signature [3], [14] verification takes the image of a signature as input and is useful in automatic verification of signatures that may be found on bank cheques and documents. *Online (dynamic) [3], [14]* signature verification uses signatures that are captured by pressure-sensitive tablets that extract dynamic properties of a signature. Signatures in offline systems usually may have noise, due to scanning hardware or paper background, and contain less discriminative information since only the image of the signature is the input to the system. While genuine signatures of the same person may slightly vary, the differences between a forgery and a genuine signature may be imperceptible, which make automatic offline signature verification a very challenging pattern recognition problem.

B. Classes of Signature Verification Systems

Depending on the input devices, two classes of systems exist for signature verification.

1) *Online (Dynamic) systems:* Online signature verification systems differ on various issues, such as data acquisition, pre-processing and dissimilarity calculation. Most commonly used online signature acquisition devices are pressure sensitive tablets with or without visual feedback. Online systems uses dynamic information such as pressure at pen tip, acceleration, and pen tilt, stroke sequence, number of strokes, direction of each stroke to be captured while a signature is being written or executed.

Features can be classified in two types: global and local. Global features are features related to the signature as a whole, for instance the signing speed, signature bounding box, and Fourier descriptors of the signature's trajectory. Local features correspond to a specific sample point along the trajectory of the signature. Examples of local features include distance and curvature change between successive points on the signature trajectory. Due to behavioral changes of a writer, two signatures signed by the same person may have different trajectory lengths (hence feature vectors of differing lengths). Therefore, straightforward methods, such as the Euclidean distance or autocorrelation, are not very useful in calculation of the dissimilarity value between two signatures. To overcome the problem, methods that can nonlinearly relate vectors of different length are commonly used. For instance, dynamic time warping algorithm with some sort of the Euclidean distance and Hidden Markov Models are commonly used in aligning two signatures.

2) *Offline (Static) systems:* The offline signature verification systems are based on the use of computer image processing and pattern recognition techniques to solve the different types of problems encountered in pre-processing,

feature extraction, and specimen comparison and performance evaluation. Other difficulties such as variation within genuine signatures, noise introduced by the scanning device or a difference in pen width make offline signature verification a challenging problem. It is worth to notice that, even professional forensic examiners perform at about 70% of correct classification rate as genuine or forged signature [4].

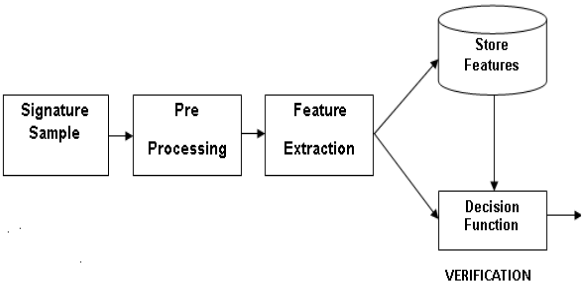

Fig. 1 Enrollment and Verification Process

C. Basic Overview of the System

Signature verification [4] and identification deal with different kinds of problems. The former verifies whether a given signature belongs to the person whose signature characteristics are known in advance. The output is a binary decision. The latter search for the identity of a given signature through a signature database and gives the class name as output.

Signature verification can be viewed as a subset of classification, where a decision function, having *a priori* knowledge of the test sample's class, has to evaluate the *belongingness* of a sample to a particular class. Conceptually this task is illustrated in Fig 1 where a decision function takes input from features extracted from a sample and an information source describing authentic samples. An absolute decision on the authenticity of a pattern is reached by thresholding the *belongingness* of the sample.

The signature sample is given as input to the system. Pre-processing is the necessary step as there might be some noise during the scanning process of the signature sample. In Feature extraction step, the necessary features are extracted from the sample. The features to be extracted are based on the application and vary from system to system. Specific and discriminate functions or parameters are computed from the filtered data and are used to represent a signature. During enrollment process the extracted features are stored in a reference database under a given class name, which can be used for verification process. For a given class, the enrollment process is to be done for sufficient number of samples of that class to generate the reference set.

During verification, the features are extracted for the test sample in the same way as done in the enrollment process. The class information of the test sample is used to extract the proper reference set from reference database. The features of the test signature just collected are then compared to the reference set of the given class. Finally a decision process evaluates the comparison algorithm with respect to a threshold and signature is accepted or rejected in a verification system. The proper action is executed by the output system according to the result.

III. PREPROCESSING

Pre-processing is method that usually concerned with the preparation of the related information. Generally in any image processing application pre-processing is required to eliminate noise, distortions etc., from the original image.

Based on the necessary features of thinning stage the pre-processing algorithm performs additional operations on the extracted image. Any normal scanner with sufficient resolution can be used as an image acquisition device for offline-operation. The hardware that is used for scanning may produce noise to a signature image. An additional source of noise may be speckled paper background on which the signature is signed. Noise on a signature image may prevent feature extraction process; therefore it needs to be removed. Hence preprocessing [14] methods should be chosen cautiously as they may eliminate signature properties peculiar to a signer.Pre-processing [10], [11], [13], [15] has lot of significance in offline signature verification. To separate signature from background, the two techniques that has been proposed are thresholding and slicing.

But they are normally insufficient in practice in a noisy background. A four step pre-processing operation proposed by *Ammar* [6] has been found to be successful in eliminating the overlapping in a signature. Since signature images are assumed to be on a white back ground, a three-step pre-processing is sufficient to extract the necessary features.--Thresholding, Smoothing, and Thinning

A. Thresholding

The process of thresholding begins with comparing the brightness value of each pixel to the threshold value. Also the pixel is assigned to the algorithm below based on whether the threshold is exceeded or not. This is usually a simple concept. In this, the brightness threshold called as parameter θ is selected and applied to the image a[m,n] as

If $a[m, n] \geq \theta$, $a[m, n] = object = 1$

Else a [m, n] = background = 0

Many thresholding algorithms have been proposed to select the value of threshold. But, no single algorithm is proved to be the best [5] . Fixed threshold, histogram-driven threshold, background symmetry algorithm, triangle algorithm and Nib lack's thresholding technique [7] are some examples. In the present system, fixed threshold technique is examined. Fixed threshold technique is obtaining satisfactory results.

Algorithm by Nib lack

This algorithm mainly depends on the local mean and local standard deviation. And also is designed to vary the threshold over the image. The threshold at pixel (x, y) is calculated as

$Th(x, y) = m(x, y) + k * Sd(x, y)$

Where $m(x, y)$ and $Sd(x, y)$ are the sample mean and standard deviation values, respectively, in a local neighborhood of (x, y). The size of the area should be little enough to store local details, but at the same time big enough to restrain noise. The k value is used to change how much of the total print object boundary is used as a part of the given object. The original gray scale image is shown in

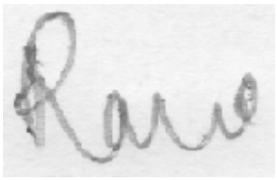

Fig3.1 Original Gray scale Image

Fig3.2 Image after Thresholding

It can be observed that the pixels that are greater than threshold are made black and pixels that are less than threshold are made white. There will not be much difference in the images that are obtained by fixed threshold and by nib lack's algorithm. In the fixed threshold method, the threshold value is fixed at 40.

B. Smoothing

The image smoothing [13],[16] which is known as noise reduction is a necessary step before going to the next stage of processing the image. The main necessary point in image smoothing is to store important features while eliminating noise from the image. Smoothing [2] create the problem of blurring sharp edges of an image. As a precautionary smoothing method which is edge preserving is used. When a new value is calculated for the pixel it is based on averaging of brightness values for it in some neighborhood.

The smoothing process is done by calculating the sum of object pixels in an 8-pixel neighborhood around each pixel in the image. This sum is used to verify whether the sum is greater than threshold or not. If it happens, the pixel is noticed as object pixel; else as background pixel. This entire process covers the rough regions of the noisy image, while thresholding out smaller speckles of noise

Fig 3.3 shows the smoothed image corresponding to fig 3.1. It can be noticed that the edges of the image are smoothened than previous.

Fig 3.3 Image after Smoothing

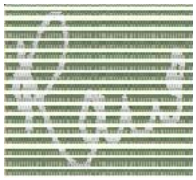

Fig 3.4Binary Matrix Generated

C. Thinning

Thinning is a process that deletes the dark points and transforms the pattern into a "thin" line drawing called skeleton. Thinning [12] [13] plays an important role in digital image processing and pattern recognition, since the outcome of thinning can largely determine the effectiveness and efficiency of extracting the distinctive features from the patterns. Image thinning reduces a connected region in the image to a smaller size and minimum cross-sectional width character [9]. The minimum cross-sectional width could be one character in which case it would be a stick figure representation. The thinned pattern must preserve the basic structure of the original pattern and the connectedness.

Algorithm by Zhang and Wang

A fast parallel algorithm proposed by *Zhang and Wang [8]* was used to thin each character to unitary thickness. In parallel picture processing the newly generated value given to a point at the n^{th} iteration depends on its own value in addition to those of its eight neighbours at the $(n-1)^{th}$ iteration in order that all points can be progressed simultaneously. Detailed explanation of the algorithm is described here. A pixel P1 under consideration and its eight neighbors are shown in

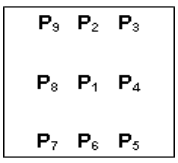

Fig3.5. Pixel P1 its 8 neighbor

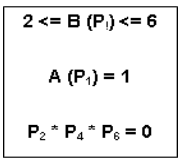

Fig3.6 Condition Set 1

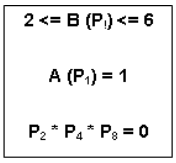

Fig3.7 Condition Set2

In the algorithm, each-iteration is divided into two sub iterations. In the first sub iteration, P1 which is the contour point is marked for deletion if it satisfies any of the conditions given figure 3.6 and in the next if it satisfies any of the conditions given in figure 3.7, where $A(P_1)$ is the number of 0 -1 transitions in the ordered sequence of P_2, P3...P_9 and $B(P_1)$ is the number of non-zero neighbors of P_1. Then all the points that are marked for deletion are deleted. This method is applied iteratively until no more points are deleted, at which point the algorithm concludes producing the skeleton of the image.

Some drawbacks found by *Lu and Wang [9]* were that sometimes noise is enlarged and some structures destroyed or removed totally. Changing the condition "2<= B (P_1) <= 6" to "3 <= B (P_1) <= 6" preserved the structures. The output of the algorithm is a single pixel width skeleton, which is connected, in the 4-neighbourhood sense. To reduce this into one connected in the 8-neighbourhood sense.

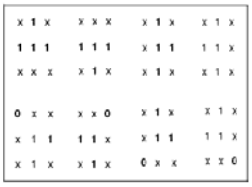

Fig 3.8 ***Templates to Delete*** *a Pixel Image 8-Connected*

Fig3.9 Image after Thinning

After applying the above procedure the thinned image, which is connected in the 8-neighbourhood, is obtained. The image after thinning is shown in Fig 3.9. The image after thinning has unit thickness from which many features can be extracted. The thinned image is the key to the feature extraction phase. From each of the described pre-processing, some global features can be obtained.

IV. FEATURE EXTRACTION

The selection of powerful set of features is vital in optical recognition systems. The features utilized must be mostly suitable for the application and also for the applied algorithm. In a signature verification system, feature extraction phase is carried out in the enrollment process, to build up the reference database of the system, and at run time for verifying the authenticity of a given signature. Feature extraction [10], [13], [14], plays a vital role in signature verification. It gives the information about the signature in terms of features and their locations. Once the feature set is

extracted from the signature, the signature itself can be represented by these features, which reduces the complexity in matching process. Almost all-dynamic information is lost in the Offline techniques. By the use of gray scale image it is possible to extract some of the dynamic information [25]. In this system, two groups of features are categorized as global features and texture features [14]. While global features give information about particular cases regarding the structure of the signature, texture features are projected to provide overall signature appearance information in different levels of detail. Few authors have suggested some ways to extract dynamic features from gray scale images such as temporal information, high-pressure regions and striations but they've not been considered now due to time constraints.

A) The Feature Set:

The matching process is done in stages for which the features are divided into two classes based on the stage they are being processed. The entire feature set is divided as global features [10] [13] and texture features [1].

1) Global Features

The values of the global features are crisp and are specific to the structure of the image. So they can be processed using a back propagation neural network [4]. The Global features include:

Slant,, Slant Direction, Density of Smoothed image Density of Thinned image, Width to Height Ratio

2) Extraction Of Global Features

A)Slant and Slant direction:

To estimate the slant of the signature the algorithm proposed by *Ammar [6]* was used. The algorithm makes use of the thinned image obtained during the pre-processing. A 3X3-sliding window is used for computation. The window is moved starting from the left-top pixel to the right-bottom pixel, one pixel at a time in a row major order. Let *P(i, j)* be a given non-zero (object) pixel in the thinned signature image, where I is row index and j is column index. Then the non-zero (object) pixel *P(i+1, j-1)* is called *negatively slanted*, the non-zero (object) pixel *P(i+1, j)* is called *vertically slanted* and the non-zero (object) pixel *P(i+1, j+1)* is called *positively slanted*. V. Karki in [10] also proposed another method by rotating the image in specific angles.Let the window contains 9 non-zero pixels named P_1, P_2…P_9 .From the above theory, P_7 is negatively slanted with respect to P_5, P_8 is vertically slanted with respect to P_5 and P_9 is positively slanted with respect to P_5. As the window proceeds in the thinned image, the total number of negatively slanted pixels N, vertically slanted pixels V, positively slanted pixels P with respect to any non-zero pixel are counted. The maximum value of N, V, and P corresponds to the *Slant Direction*. The *Slant* value can be calculated by using the following formula.

Slant = Max of (N, V, P)/total no of pixels in the thinned Image

B) Density of Smoothed Image: The total number of black pixels in the image can be treated as image area [1] . It represents the density of the image. As smoothing and thinning were considered in the pre-processing, the densities for both smoothed image and thinned image were considered as global features. The density of smoothed image can be calculated by the following formula after smoothing is performed.

Density of Smoothed image=No of non zero pixels/Total no of pixels in the thinned image

C) Density of thinned Image: The density of thinned image can be calculated after thinning is performed. It corresponds to the measure of density of signature traces. The density of smoothed image can be calculated by the following formula.

Density of thinned image = No of non zero pixels in the thinned image/Total no of pixels in the thinned image

D) Width to Height Ratio: The width and height of a signature can be calculated from the contour of the signature [17]. Here the width and height are considered in terms of number of pixels. The contour of the signature is outer line of the non-zero pixels. It can also be calculated from the smoothed image. The height of the signature image, after width normalization is performed, can be measured as a way of representing the height to width ratio. The minimum and maximum values of x and y coordinates of non-zero pixels in the smoothed image are sufficient to calculate the width to height ratio. Let minx and maxx be the minimum and maximum values of x coordinates of non-zero pixels and miny and maxy be the minimum and maximum values of y coordinates of non-zero pixels. Width to height ratio is the ratio of range of x coordinates to the range of y coordinates. The formula for calculating width to height ratio is given as Width to Height Ratio = Maxx-Minx/Maxy-Miny

3) Texture Features

The topological features are pixel positions in the image with respect to the property of the feature. These can be processed using a matcher that uses co-occurrence matrix of the picture image. The topological features includes: End points, Branch points, crossing points

4) Extraction of Texture Features

A) End Points, Branch Points, and Crossing Points: To extract end points, branch points, crossing points it is necessary to apply the preprocessing techniques like thresholding, smoothing and thinning on a gray scale signature image. End points are points where a signature stroke begins or ends. Branch points are points where on signature stroke bifurcates into two strokes. Crossing points are points where one signature stroke crosses another stroke. Fig 4.2 shows an example scenario [1]

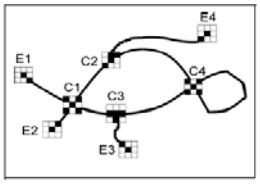

Fig 4.1 ***Example of End points,*** *Crossing points and Branch points*

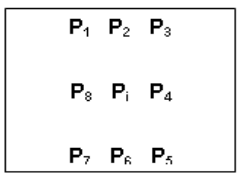

Fig 4.2 *Pixel* P_i and its eight *Neighbor*

In the fig 4.1, E1, E2, E3, E4 are four possible end points. C1, C4 are crossing points. C2, C3 are branch points. Each pixel in the thinned image (Skeleton) can be associated with a value represented by a transition function $T(P_i)$ for that pixel. Let Pi is the pixel under discussion and its eight neighbors are shown asP1,P2,P3,P4,P5,P6,P7,P8.

V.PROPOSED SYSTEM

The signature verification system should be surely insensitive to intra-personal variations, although sensitive to inter-personal variations. The system can be made sensitive to inter-personal variations by extracting features efficiently, which are sensitive for an individual.

A) System Design

A novel approach to solve the problems of intra-class differences between genuine signatures is incorporated. The system uses a limited number of samples for training the back propagation network so that the variations can be accommodated. The underlying basis of the approach is that forensic experts first locate the differences between the input signature and the stored genuine signatures by comparing local features, then analyze the stability of these features in the genuine samples to judge whether the differences are essential or accidental. To achieve this, a method for artificially generating forgeries from genuine signatures is provided. The basic design of the system is outlined in section1.4. The overall process of the system is shown in fig

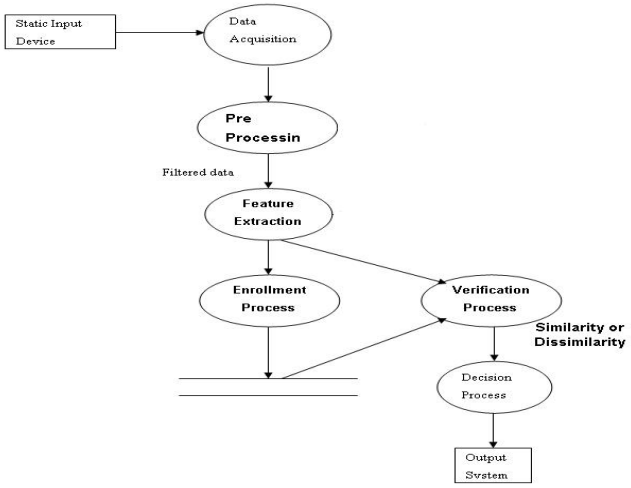

Fig 5.2 *Data Flow Diagram of the Proposed System*

The static signature image is given as input to the system. After pre-processing and feature extraction the system has two alternatives. If the sample is given for enrollment, control is given to forgery generation module. It will generate a forgery that is more or less same as the original sample. The feature values of given genuine input and system generated forgery are stored in the reference database under the given class name. If the sample is given for testing, the feature values of the corresponding class are obtained from the reference set and compared against the feature values of the test sample. As the verification process proceeds in two stages, the control is passed to the second stage once the first stage verification finishes. The decision system makes the decision based on the results obtained from the two verifiers. The output is submitted to the user specifying whether the given sample is genuine or forged one.

B) The Enrollment Process

In the enrollment phase, the feature values of different features are extracted from the signature and stored separately in different files. These values are helpful in comparing them with those generated for a sample to be verified.

1) Enrollment Of The Global Features

The global features extracted (slant, slant direction, density, width to height ratio, pressure factor) for a signature sample are stored in a file under the given unique class name. When a new sample of the same class is used for enrollment the feature values are added to the same database, otherwise they are added to a new file under its class name. Each time a new class (a new person) is added to the database, they system will be trained with the new sample signatures of that person. This is a simple task and does not require heavy processing. This scheme simplifies the addition of a new class to the database.

Back propagation neural network (BPN) with one-class-one-network scheme [14] is used. The structure of BPN used is assumed to have only one hidden layer. The numbers of units in each layer are 5x3x1. (5 input unit, 3 hidden units, and 1 output unit). The 5 input units correspond to slant, thin density, smooth density, width to height ratio, and pressure factor. The feature values of the given signature sample are used to train the network after normalizing. The threshold for total error rate is set appropriately. The network is trained and the weight matrices are stored separately under the given class name. So for each class that is enrolled to the system, 3 reference databases are created with respect to global features. The first one is to store the feature values, second is to store the normalized values, and third is to store the weight matrix of the particular class.

2) Enrollment of the Texture Features

The texture features that are extracted are the (pixel positions of) end points (E), branch points (B), crossing points (C). To store texture features of this type, special multidimensional co-occurrence matrices proposed by *V.Kovalev* are used. They are used for the description and representation of some basic image structures. The co-occurrence matrices are obtained from the picture matrix of features extracted from signatures. In the present work the symbolic picture (picture matrix), which is obtained from features and their locations was used in finding the multi dimensional co-occurrence matrix.

B) Back Propagation Network

The field of neural networks has provided the most excellent way of finding solution the problems that are most difficult to solve by traditional computational methods. Back propagation [18], [11], [12] is one such best algorithm which has hugely contributed to neural network. One of the most well known common applications of NNs is in image processing. Few examples would be identifying hand-written characters, voice recognition, and pattern recognition [4].

In back propagation network, whenever we are training a network we not only give it with the input but also with a value that we need the network to produce. The well known BPN learns by example, which means we ourselves must provide a learning set that consists of few input examples and some known-correct output for every case. Using these examples we show the network the type of behaviour we want, and BPN algorithm allows the network to adapt. The neural network approach provides a major advantage with a Neural Network solution, that there is no need to understand the solution of the problem. The

conventional techniques require one to understand the inputs, algorithms, outputs. With a Neural Network, we can simply demonstrate it saying: "this will be the exact output, for this input". Once trained, the network mimics the function that we are trying to show. And the best part of Neural Network about the inputs required is that, even if some of the inputs that are applied are irrelevant, the network learns to ignore such inputs that may not lead to the output. On the other hand, if some important inputs are left out, it's easy to find out as the network fails in converging at a solution. A typical BPN having 3 input unit, 3 hidden units, and 2 output unit is shown below:

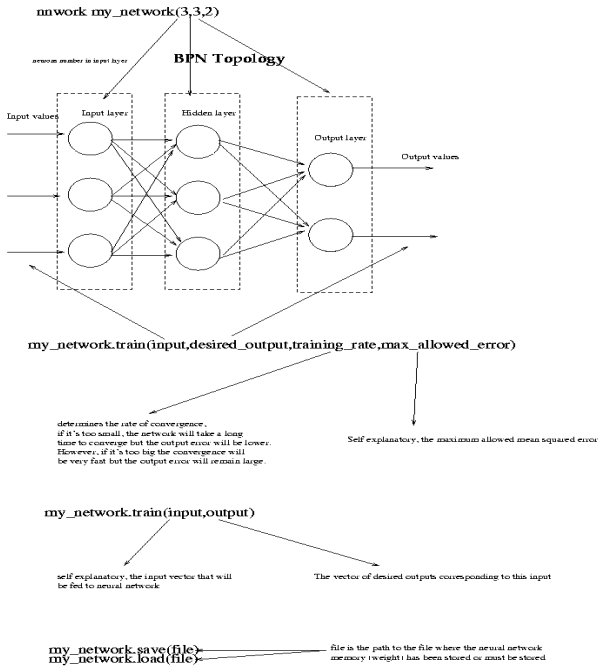

Fig 5.3 Typical Back Propagation Network

C) Verification Process

The verification process compares the signature that is to be tested with the reference signatures stored in the database. The comparison is based on the assumption that the values of the feature sets or structural description extracted from genuine signatures are more stable than the signatures that are forged. That is, the intra-personal variations are smaller than inter-personal variations. So, the given test signature may be accepted or rejected based on its similarity to the reference signature set.

The verification process proceeds in two levels. The final decision is based on the decisions of both stages. The test specimen is given as input to the system to check whether it is a genuine or forged. The feature values of the test specimen are extracted in the same way as *Verification* process of the signature samples is done in two levels. Each level takes a subset of the feature set that is obtained from the feature extraction phase as input and compares with the reference feature set. After extracting the feature values of the test sample, control is handed over to the first level for verification of Global Features and from there to the next level for the verification of Texture Features.

The first level evaluates the global features for validity, where as the second level is dedicated to texture features. Each stage takes its own decision about the authenticity of the given signature. These decisions will be sent to the decision function. The decision function decides whether the given signature is a genuine sample or a forged sample based on the individual results.

*1) Verification of Global Feature*s:

The first level of verification checks the global features for validity. The features extracted for test sample are normalized and these normalized values are used to test the network. The weight matrix of the corresponding class is taken from the database and is assigned to the network before testing. If the output of the network is less than specified threshold then the verification process marks the given sample as a genuine one. One optimization can be done before testing the sample with the network. That is, the feature values (global) of the test sample are checked whether they fall within the minimum and maximum values of that feature value that is stored in the database. The system sets a range of values for each feature extracted for the given class. The least value corresponds to minimum and the highest value corresponds to maximum of the feature of the given class. Then, it verifies that whether all these features of the given test signature lie within the minimum and maximum of the corresponding features of the given class. If they are in the range then the normalized values are tested using BPN. If not, the testing with BPN network can be skipped and report error. This is referred to as static data analysis.

2) Verification of Texture Features:

The multi dimensional co-occurrence matrix is obtained for each sample that is given for training and the values are stored in the database. During verification, the same has to be obtained for the test specimen. Once having obtained the multi dimensional co-occurrence matrix, each element of the multidimensional co-occurrence matrix of the test signature is compared with the corresponding element of that in the database .If the compared values in the range for a maximum number of elements then output is said to be genuine, otherwise forged.

The value of total error is plotted against the epochs and is shown in fig 6.1. Within an epoch the error rate keep on increasing but it is decreasing from epoch to epoch. The BPN was trained till the total error becomes zero value.A set of signatures consisting of 12 genuine signatures of 3 classes (persons), used as a reference database. To test the performance of the system 3 forged samples were used. The values of False Acceptance Rate (FAR) and False Rejection Rate (FRR) were observed. Table 6.4 shows the values of FAR and FRR. The value of FRR is the percentage of number of samples rejected falsely. In case 1 it is 2 / 12 * 100 = 16.67%. The value of FAR is also calculated similarly.

Total Genuine Samples Tested	12
Correctly Accepted	10
Falsely Rejected	2
False Rejection Rate (FRR)	16.67%
Total Forged Samples Tested	3
Correctly Rejected	3
Falsely Accepted	0
False Acceptance Rate (FAR)	0%

Table 6.4 shows the values of FAR and FRR

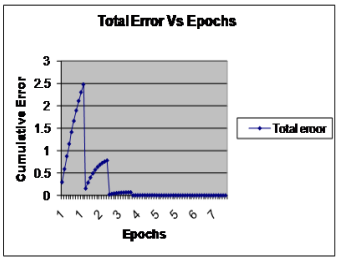

Fig 6.1 Total Error Rate Plot Against Epochs

CONCLUSIONS

Signatures in offline systems are based completely only on the image of the signature. It contains less discriminative information as it is often contaminated with noise either due to scanning hardware or paper background. A novel approach for off-line signature verification is proposed and implemented using a Back Propagation Neural Network algorithm. The system that is proposed based on global and texture features. Features exhibiting good performance are considered, and finally a near-optimal solution using blend of such features in terms of high verification and time efficiency is derived.

The system is tested against the genuine and the forged samples. The values of FAR and FRR are observed and these results look very promising. Although the operations used in obtaining the features are computationally expensive, they are adopted in order to get good results. The performance of the system is satisfactory. However the system has to be tested on many more samples (obtained from real-life data from a bank or any other similar organization).

REFERENCES

[1] H. Baltzakis, N. Papamarkos, "A new signature verification technique based on a two-stage classifier", *PERGAMON, Engineering Applications of artificial intelligence* 14 (2001) 95-103.

[2] Alisher Anatolyevich Kholmatov, "Biometric Identity Verification Using On-Line & Off-Line Signature Verification", *Master of Science thesis* Sabanci University Spring 2003.

[3] S. Lee and J. C. Pan, "Off-line tracing and Representation of Signatures", *IEEE Trans. on systems, man, cybernetics*, Vol. 22, No. 4, July/August 1992.

[4] Rasha Abbas," Backpropagation networks prototype for offline signature verification",Computer Science,RMIT,1994

[5] Trier and A. K. Jain, "Goal directed evaluation of binerization methods", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 12, 1995.

[6] M. Ammar, Y. Yoshida and T. Fukumura, "Description of Signature Images and its application to their Classification", Accepted for Publication, *Japan*, pp. 23-26, 1988

[7] W. Niblack, "An Introduction to Digital Image Processing", *Engle wood cliffs, N.J. Prentice Hall*, pp. 115-116, 1986

[8] Zhang and Wang, "A fast and flexible thinning algorithm", *IEEE Transactions on Computers*, Vol. 38, No 5 May 1989.

[9] T.Y.Zhang and C.Y.Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns", Communications of ACM,Volume 27,236-239,1984

[10] Maya V. Karki, K. Indira, Dr. S. Sethu Selvi;" Off-Line Signature Recognition and Verification using Neural Network" International Conference on Computational Intelligence and Multimedia Applications 2007 ; Pages 307 312.

[11] Alan McCabe, Jarrod Trevathan and Wayne Read," Neural Network-based Handwritten Signature Verification", JOURNAL OF COMPUTERS, VOL. 3, NO. 8, AUG 2008.

[12] Andrew. T. Wilson, "Off-line handwriting recognition using artificial neural networks",Univeristy of Minnesota,

[13] Emre Ozgunduz, Tulin Senturk, and Elif M. Karsligil. "off-line signature verification and Recognition by support vector machines." In Computer Analysis of Images and Patterns. 11th International Conference, CAIP 2005

[14] Jingbo Zhang , Xiaoyun Zeng, Yinghua Lu, Lei Zhang, Meng Li ," A Novel Off-line Signature Verification Based on One-class-one-network" Third International Conference on Natural Computation (ICNC 2007).

[15] Ramachandra A C, Pavithra K, Yashasvini K, K B Raja, Venugopal K R and L M Patnaik;" Offline Signature Authentication using Cross-validated Graph Matching" *Compute* 2009, Jan9,10, Bangalore,Karnataka,India.

[16] C.C Tappert, C.Y. Suen, and T. Wakahara, "The state of the art in on-line handwriting recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 787-808, 1990.

[17] Hwei-Jen ,Fu-Wen Yang, "Off-Line Verification for Chinese Signatures", International Journal of Computer Processing of Oriental Languages, Vol. 14, No. 1 (2001) 17-28

[18] Madasu Hanmandlu, Mohd Yusof, and Vamsi K. Madasu. Off-line signature verification and forgery detection using fuzzy modeling. Pattern Recognition, 38(3):341{356, March 2005.