International Journal of Modern Engineering Research (IJMER)

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1171-1175 ISSN: 2249-6645

Offline Signature Verification Using Neural Network

S.T. Kolhe, S. E. Pawar

(Dept. of Computer Engg, AVCOE, Sangamner, India)

(Dept. of I.T. Engg, AVCOE, Sangamner, India)

ABSTRACT

The concept of offline signature verification is challenging because of several reasons. Firstly, there exists great variation even between two signatures of the same person. They never start from the same position and neither do they terminate at the same position. Also, the angle of inclination of the signatures, the relative spacing between letters of the signatures, height of letters - all vary even for the same person. Hence it becomes a challenging task to compare between two signatures of the same person, an offline signature verification system is proposed in this paper take care of that. The proposed model has four stages: image pre-processing, feature extraction, neural network training, verification and recognition. The user introduces into the computer the scanned signature images, modifies their quality by image enhancement and noise reduction techniques, to be followed by feature extraction, training the neural network and finally verification and recognition.

Keywords- **Backpropagation, Image processing, neural network, Offline signature verification.**

I. INTRODUCTION

Signatures are composed of special characters and flourishes and therefore most of the time they can be unreadable. Also intrapersonal variations and interpersonal differences make it necessary to analyze them as complete images and not as letters and words put together [1]. As signatures are the primary mechanism both for authentication and authorization in legal transactions, the need for research in efficient automated solutions for signature recognition and verification has increased in recent years. Recognition is finding the identification of the signature owner. Verification is the decision about whether the signature is genuine or forgery. In this decision phase the forgery images can be classified in three groups: (i) random, (ii) simple, (iii) skilled [2]. Random forgeries are formed without any knowledge of the signer's name and signature's shape. Simple forgeries are produced knowing the name of the signer but without having an example of signer's signature. Skilled forgeries are produced by people looking at an original instance of the signature, attempting to imitate as closely as possible. The signatures can be distinguished in two different categories of verification systems depending on acquisition of signatures: online signature, for which the signature is captured during the writing process and making the dynamic information available, and offline, for which the signature is captured once the writing process is over and, thus, only static information is available. The objective of the signature verification system is to discriminate between two signature classes: the genuine and forgery signature. A lot of work has been done in the field of off-line signature verification [4], [5]. Although a large number of works is focused on random and simple forgery detection, more efforts are still needed to address the problem of skilled forgery detection [3]. No verification algorithms are proposed which might be dealt well with skilled forgeries.

The proposed offline signature system provides automated method of verification and recognition by extracting features that characterizes the signature. The approach starts by scanning images into the computer, then modifying their quality through image enhancement and noise reduction, followed by feature extraction and neural network training, and finally verifies whether a signature is original or fake.

II. SYSTEM OVERVIEW

2.1 Preprocessing

Generally in any image processing application pre-processing is required to eliminate noise, distortions etc., from the original image. Any normal scanner with sufficient resolution can be used as an image acquisition device for offline-operation

2.1.1 Scaling

If H_I and W_I are the height and width of the image then we can make the image at uniform 100x100 pixels by the simple equations:

$$X_{new} = X_{old} * S_x \qquad (1)$$

Where Sx=100/H_I; X is any horizontal unit and is the horizontal scaling factor.

$$Y_{new} = Y_{old} * S_y \qquad (2)$$

Where Sy=100/W_I; Y is any vertical unit and is the vertical scaling factor.

2.1.2 Conversion of color image to grayscale

The luminosity method is a more sophisticated version of the average method to convert the color image to grayscale. It also averages the values, but it forms a weighted average to account for human perception. The formula for luminosity is 0.21 R + 0.71 G + 0.07 B.

2.1.3 Thresholding

In computer vision and image processing, Otsu's method is used to automatically perform histogram shape-based image

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1171-1175 ISSN: 2249-6645

thresholding or the reduction of a gray level image to a binary image. The algorithm assumes that the image to be threshold contains two classes of pixels (e.g. Foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. In Otsu's method we exhaustively search for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_1(t)\,\sigma_1^2(t) + \omega_2(t)\,\sigma_2^2(t) \qquad (3)$$

Weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes. Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\,\omega_2(t)\,[\mu_1(t) - \mu_2(t)]^2 \qquad (4)$$

which is expressed in terms of class probabilities ω_i and μ_i class means which in turn can be updated iteratively.

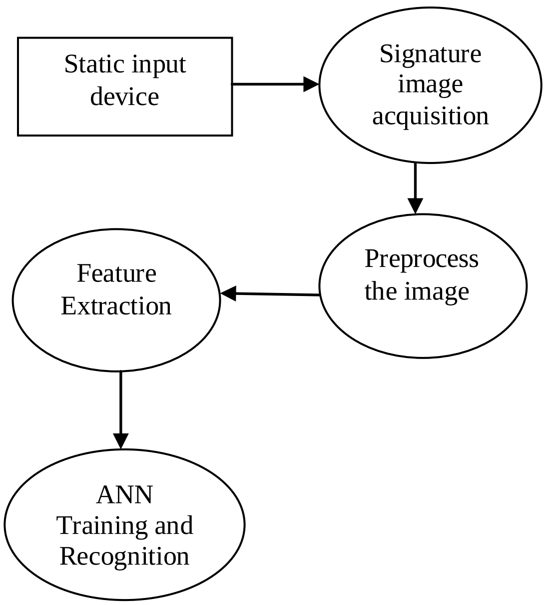

Fig 1: Flow Diagram of Offline Signature verification

2.1.4 Thinning

Thinning was introduced to describe the global properties of objects and to reduce the original image into a more compact representation. It uses a set of four templates to scan the image. Figure 2 shows these four templates.

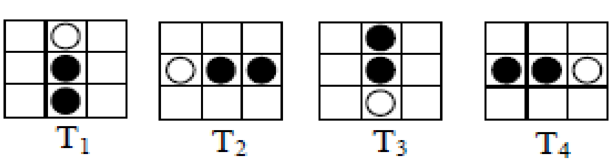

Fig 2: Templates to identify pixels to be eroded

The Stentiford Algorithm can be stated as following [6]:

1. Find a pixel location (i, j) where the pixels in the image match those in template T1. With this template all pixels along the top of the image are removed moving from left to right and from top to bottom.

2. If the central pixel is not an endpoint, and has connectivity number = 1, then mark this pixel for deletion. Endpoint pixel: A pixel is considered an endpoint if it is connected to just one other pixel. That is, if a black pixel has only one black neighbor out of the eight possible neighbors.

Connectivity number: It is a measure of how many objects are connected with a particular pixel.

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2}) \qquad (5)$$

where: N_k is the color of the eight neighbors of the pixel analyzed. N_0 is the center pixel. N_1 is the color value of the pixel to the right of the central pixel and the rest are numbered in counterclockwise order around the center.

$S = \{1, 3, 5, 7\}$

Figure 3 illustrates the connectivity number. Figure 3 a) represents connectivity number = 0. b) Represents connectivity number = 1, the central pixel might be deleted without affecting the connectivity between left and right. c) Represents connectivity number = 2, the deletion of the central pixel might disconnect both sides. d) Represents connectivity number = 3, and e) represents connectivity number = 4.

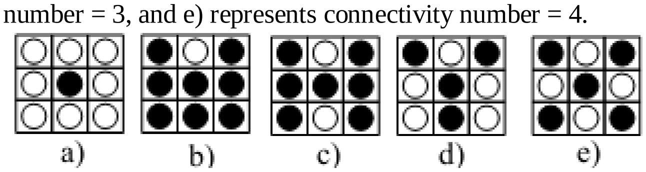

Fig 3: Connectivity Number

3. Repeat steps 1 and 2 for all pixel locations matching T1.

4. Repeat steps 1-3 for the rest of the templates: T2, T3, and T4. T2 will match pixels on the left side of the object, moving from bottom to top and from left to right. T3 will select pixels along the bottom of the image and move from right to left and from bottom to top. T4 locates pixels on the right side of the object, moving from top to bottom and right to left.

5. Set to white the pixels marked for deletion. Figure 7 shows the signature after the thinning process using the Stentiford Algorithm.

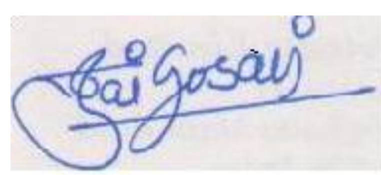

Fig 4: Original Signature image

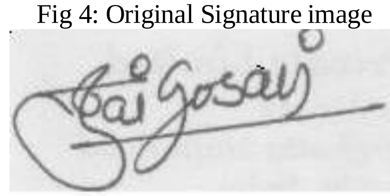

Fig 5: Grayscale signature image

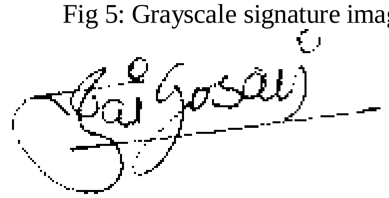

Fig 6: Signature after Thresholding

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1171-1175 ISSN: 2249-6645

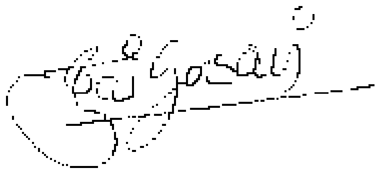

Fig 7: Signature after Thinning

III. FEATURE EXTRACTION

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 signature. The term feature here refers to a certain characteristic that can be measured using designed algorithms; which can then be retrieved by "extraction".

In this system, two groups of features are categorized as global features and texture features. 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.

3.1 Global features

3.1.1 Slant and Slant direction

To estimate the slant of the signature the algorithm proposed by Ammar 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.

3.1.2 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

3.1.3 Width to Height Ratio

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar: Let min_x and max_x be the minimum and maximum values of x coordinates of non-zero pixels and min_y and max_y 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:

$$\text{Width to Height Ratio}=(Max_x-Min_x/(Max_y-Min_y) \quad (6)$$

3.1.4 Skewness

Skewness is a measure of symmetry or more precisely, the lack of symmetry. The measurement of skewness allows us to determine how bowed are the lines in each segment of the signature

$$\text{Skewness} = \sum(Y_i-Y')s^3 \ / (N-1)\ s^3 \ \text{ for } i=1 \text{ to } N \quad (7)$$

Where *Y* is the mean, *S* is the standard deviation, and *N* is the number of data points. The measurement of skewness allows us to determine how bowed are the lines in each segment of the signature. The percentage of this torsion is then calculated and extracted. Furthermore, this percentage is compared to that extracted from the other image.

3.2 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.

3.2.1 Extraction of Texture Features

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

3.3 Enrollment process

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.

3.3.1 Enrollment of the Global Features

1. The global features extracted (slant and slant direction, density, width to height ratio, skewness) for a signature sample are stored in a file under the given unique class name.

2. 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.

3.3.2 Enrollment of the Texture Features

1. The texture features that are extracted are the (pixel positions of) end points (E), branch points (B), crossing points (C).

2. The co occurrence matrices are obtained from the picture matrix of features extracted from signatures. The symbolic picture (picture matrix), which is obtained from

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1171-1175 ISSN: 2249-6645

features and their locations will be used in finding the multi dimensional co-occurrence matrix.

IV. NEURAL PATTERN RECOGNTION

Back propagation neural network (BPN) with one-class one-network scheme 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, slant direction, thin density, width to height ratio, and skewness. 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, three 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.

4.1 Algorithm Backpropagation

It is neural network learning algorithm for classification or prediction, using the back propagation algorithm. *Input:* *D*, a data set consisting of the training tuples and their associated target values; *l*, the learning rate; *Network*, a multilayer feed-forward network; O*utput:* A trained neural network.

Method:

(1) Initialize all weights and biases in *network*;

(2) While terminating condition is not satisfied {

(3) for each training tuple *X* in *D* {

(4) // propagate the inputs forward:

(5) for each input layer unit *j* {

(6) $O_j = I_j$; // output of an input unit is its actual input value

(7) for each hidden or output layer unit *j* {

(8) $I_j = \sum_i w_{i\,j} O_i + \theta_j$; //compute the net input of unit *j* with respect to the previous layer, *i*

(9) $O_j = 1 / (1+e^{-I\,j})$; } //compute the output of each unit *j*

(10) // Backpropagate the errors:

(11) for each unit *j* in the output layer

(12) $Err_j = O_j\,(1\text{-}O_j)\,(T_j\,\text{-}O_j)$; // compute the error

(13) for each unit *j* in the hidden layers, from the last to the first hidden layer

(14) $Err_j = O_j\,(1\text{-}Oj) \sum_k Err_k\, w_{jk}$; // compute the error with respect to the next higher layer, *k*

(15) for each weight $w_{i\,j}$ in *network* {

(16) $\Delta w_{i\,j} = (l)\, Err_j\, O_i$; // weight increment

(17) $w_{i\,j} = w_{i\,j} + \Delta\, w_{i\,j}$;} // weight update

(18) for each bias θ_j in *network* {

(19) $\Delta\,\theta_j = (l)\, Err_j$; // bias increment

(20) $\theta_j = \theta_j + \Delta\,\theta_j$; } // bias update

(21) }}

V. 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.

5.1 Verification of Global Features

1. 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.
2. The weight matrix of the corresponding class is taken from the database and is assigned to the network before testing.
3. If the output of the network is less than specified threshold then the verification process marks the given sample as a genuine one.
4. 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.

5. 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.
6. 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.
7. 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.

5.2 Verification of Texture Features

1. The multi dimensional co-occurrence matrix is obtained for each sample that is given for training and the values are stored in the database.
2. 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

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1171-1175 ISSN: 2249-6645

signature is compared with the corresponding element of that in the database.

3. If the compared values in the range for a maximum number of elements then output is said to be genuine, otherwise forged.

VI. CONCLUSION

The main objective of this work is to present a robust system for off-line signature verification. The ultimate goal of signature verification is to have machines which can read any text with the same verification accuracy as humans but at a faster rate. 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. Although the operations used in obtaining the features are computationally expensive, they are adopted in order to get good results

REFERENCES

[1] Ramachandra, Jyothi Srinivasa Rao, K B Raja, K R Venugopla ,L M Patnaik, Robust Offline Signature Verification Based On Global Features, 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009

[2] Meenakshi S Arya, Vandana S Inamdar,A Preliminary Study on Various Off-line Hand Written Signature Verification Approaches 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 9

[3] V A Bharadi, H B Kekre, Off-Line Signature Recognition Systems, International Journal of Computer Applications (0975 - 8887)Volume 1 – No. 27

[4] Alessandro Zimmer, Lee Luan Ling, “Offline Signature Verification System Based on the Online Data” EURASIP Journal on Advances in Signal Processing Volume 2008, Article ID 492910

[5] H.N. Prakash, D. S. Guru, “Offline Signature Verification: An Approach Based on Score Level Fusion” International Journal of Computer Applications (0975 - 8887)Volume 1 – No. 18

[6] Sonka, M., Hlavac, V., Boyle, R., “Image Processing, Analysis, and Machine Vision”, 2nd Edition, Pws. Pub. Co.