Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OFFLINE HANDWRITTEN SIGNATURE VERIFICATION USING BACK PROPAGATION ARTIFICIAL NEURAL NETWORK MATCHING TECHNIQUE

[1]ANWAR YAHY EBRAHIM, [2] GHAZALI SULONG
[1]PHD student, Department of Computing, Universiti Tecknologi Malaysia (UTM), 81310 Johor Bahru, Malaysia
[2] Professor, Department of Computing,Universiti Tecknologi Malaysia (UTM),81310 Johor Bahru, Malaysia
E-mail: [1]anwaralawady@gmail.com, [2]ghazali@utmspace.edu.my

ABSTRACT

Handwriting is a skill that is highly personal to individuals and consists of graphical marks on the surface in relation to a particular language. Signatures of the same person can vary with time and state of mind. Several studies have come up with several methods on how to detect forgeries in signatures given to the security implication of signatures to daily business and personal transactions. This paper illustrates the proposed methodology for an offline handwritten signature identification and verification system which extracts certain dynamic features derived from velocity and acceleration of the pen together with other global parameters like total time taken and number of pen-ups in order to distinguish between forged signatures and genuine signatures signed under duress. Adaptive Window Positioning technique was employed for feature extraction, which focuses on not just the meaning of the handwritten signature but also on the individuality of the writer by dividing the handwritten signatures into 13 small windows of size nxn (13x13) such that it is large enough to contain ample information about the style of the author and small enough to ensure a good identification performance. Then, a signer specific codebook approach was used to generate a separate codebook of patterns for each individual signer such that the number of classes in each codebook varies as a function of the writing sample (signer), and a 3-layered Backward Propagation Artificial Neural Network (BPANN) method was used to produce a maximal matching and preserve the efficiency of the network. The proposed method was validated using a trained GPDS data set of 2400 original signatures of 100 different signers and comparing the results with those of two different known techniques of offline handwritten signature verification systems. The findings indicate that the proposed technique had the lowest ERR value of 7.23, indicating a more improved performance when compared against the two known techniques respectively thus proving to be a more efficient and superior method for offline handwritten signature identification and verification.

Keywords: *Offline Handwritten Signature, Identification, Verification, Adaptive Window Positioning, Signer Specific Codebook, Backward Propagation Artificial Neural Network*

1. INTRODUCTION

The use of handwritten signatures for authenticating documents or personal identification dates back to to ancient times and has since been increasingly used for numerous financial and business transactions [1]. The use of handwritten signature for authentication plays an important role in the everyday life of society and is applied in almost every sphere of human activity due to its relative ease of use, especially offline handwritten signatures [2]. According to [3], most financial institutions give preference to the use of the offline handwritten signature despite the fact that the online signatures have proved to be more reliable but however require more complex processing and high-tech gadgets which the offline signatures do not require. Offline signatures can be signed on a piece of paper, which as at today plays a very vital role in documentation despite the ongoing e-revolution. Online signatures on the other hand, require special hardware such as digitizers and pressure tablets necessary to acquire the dynamic information such as pressure and speed of the signer, besides the static image of the signature. The problem however is that the offline signature can be easily imitated or forged which could lead to false representation or fraud [4]. Therefore, there is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a need for adequate protection of personal signatures. Verification decision of offline handwritten signatures usually undergoes a series of processes which include pre-processing (where the local and global features of the handwritten signature is extracted), identification and verification of the extracted features against a standardized database. A good verification result can be derived by matching the robust features of the sample signature against the signature of the user through appropriate techniques or classifiers [5]. However, most studies in this domain have often overlooked the impact of external influences such as duress and mind state of the signer when signing their signatures [6][7][8]. This has made it impossible to distinguish between a forged signature and a genuine signature signed under duress, for example. With the advent of highly advanced computers with unimaginable processing prowess, there is need for the development of a new technique and algorithm that can take into account some of these external factors when investigating or verifying a signature. This paper therefore attempts to address this challenge by proposing a new method for offline handwritten signature identification and verification through the use of a combination of techniques and methods such as adaptive window positioning technique for signature feature extraction, signer specific codebook for clustering and BPANN for matching.

The rest of the paper is divided as follows: Section II looks at the related literature in this field, Section III discusses the methodology adopted for this study, Section IV presents and discusses the results of the study and Section V concludes the study.

2. BACKGROUND OF THE STUDY

The challenge in creating a system with the ability to recognize handwritten offline signatures and verify its authenticity has been a problem particularly in the use computer to identify forgeries beyond the convention of writing an algorithmic process [5]. This challenge involves making the computer solve the problem using a series of new steps. Past studies, in an attempt to bridge this gap have employed several techniques and methods in addressing the challenge. The issue has been complicated with the increasing role signatures play in daily financial, legal and commercial transactions, thus requiring a more secured authentication. Handwritten signatures are the most widely accepted means of personal identity authentication, which can be captured and processed as an image by digital computers [3]. However, an individual's signature features may tend to vary depending on the individual's state of mind, and the time and circumstances of signing, thus making some of the signature features unreadable by conventional signature identification and verification systems. The advent of modern computers with its high processing power has prompted the need for the development of faster signature recognition algorithms [9] so as to enable the analysis of the differences and intrapersonal variations of an individual's signature rather than analyzing them as a complete image. One such technique that has been developed to facilitate this is the Artificial Neural Network (ANN). ANN simulates the operation of the biological nervous systems in its information processing. It is made up of a large network of highly interconnected processing elements (neurons) that function collectively to address particular issues. Like in humans, ANN learns its way through a problem by examples, adjusting to the synapses between the interconnected neurons. These characteristics make neural networks a very useful candidate when it comes to pattern extraction and trend detection from imprecise or complicated data thereby giving meaning to it, which other conventional computer techniques fall short in [10]. The main reason for the widespread usage of neural networks (NNs) in pattern recognition lies in their power and ease of use. A simple approach is to firstly extract a feature set representing the signature (details like length, height, duration, etc.), with several samples from different signers. This phase generates for the NN to learn the relationship between a signature and its class (either "genuine" or "forgery"). Once this relationship has been learnt, the network can then be presented with test signatures that can be classified as belonging to a particular signer. NNs therefore are highly suited to modeling global aspects of handwritten signatures, using structure features from the signatures contour, modified direction feature and additional features in which a signature is divided into two halves and for each half a position of the center of gravity is calculated in reference to the horizontal axis [10].Although several other approaches exist and several hundreds of NNs have been proposed by previous studies, a small category of "classic" networks is widely recognized as the basis on which most other NNs are built on, such as the BPANN. Most researchers have described BPANN as a standard for NN. BPANN is not actually the network per say, but learning or training algorithm that learns by example, with the ability of yielding the desired output for a specific input for mapping and simple

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pattern recognition tasks [10].Thus, this paper presents a methodology that can be used for identifying and verifying offline handwritten signatures through a series of logical steps beginning from the use of Adaptive Window Positioning technique for signature feature extraction, signer specific codebook approach for generating separate codebook patterns for each individual signer, and a 3-layered BPANN method for maximal matching and to preserve the efficiency of the network. Table 1 presents a list of different offline handwritten signature identification and verification approaches highlighting their characteristics, advantages and disadvantages.

3. PROPOSED METHOD

This section illustrates the proposed methodology used for the offline handwritten signature identification system based on BPANN matching method. The methodology has been divided into three phases namely: the pre-processing phase, codebook generation phase, and the matching phase. Details of each of these phases, together with the logical steps undergone, beginning from acquiring the signature images to preprocessing, clustering, identification and verification are presented in a flowchart diagram as shown in Figure 1 and discussed below.

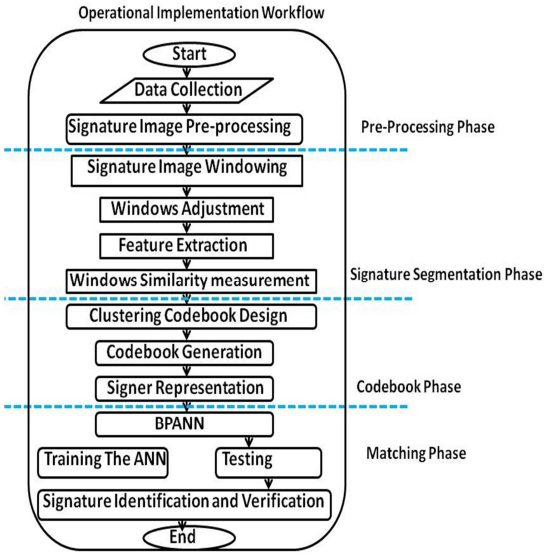


Figure 1: Methodology Flowchart

3.1 Preprocessing Phase

In this phase, the data is acquired and the signature image preprocessed. GPDS database was used for collecting data because it is a standardized database of offline English signature images with a dataset of 4870 signatures from 90 different writers. Also, it enjoys wide acceptability and usage among researchers in the domain of offline handwritten signature identification and verification applications. The collected signature images were preprocessed in order to improve the image quality for noise-free feature extraction thus improving identification performance. A global threshold of Otsu's algorithm was employed for the conversion into binary images of the selected signature.

Adaptive window positioning technique was then used to segment the signature images into small fragments or sub images enabling the elimination of redundancy and facilitating a meaningful comparison of the segmented fragments. A 13x13 window size was applied to the signature images for optimum output. The main objective of this technique is to achieve optimal window positioning by tracing the ink of the handwritten signature image in the context of the drawing. Next, we extract the set of features (shape measures) from the patterns and represent the images in a feature space. Since a typical data analysis problem involves many observations as well as a good number of respective features, it is important to organize such data in a sensible way before it can be presented and analyzed by humans or machines. The aim of shape characterization is to obtain shape measures to be used as features for classification in patterns. After representing the sub-images from a set of features, we need to do a similarity measure of all the windows by comparing the sub-images (fragments) in pairs. The sub-images are compared with the following correlation similarity measurement:

$$S(X,Y)=\frac{n_{11}n_{00}-n_{10}n_{01}}{[(n_{11}+n_{10})(n_{01}+n_{00})(n_{11}+n_{01})(n_{10}+n_{00})]^{1/2}} \quad (1)$$

Where, Nij is the number of pixels of the two sub-images X and Y, with values i and j respectively, at the corresponding pixel positions. This measure will be close to 1 if the two compared sub-images are similar and in extreme case it will have a value equal to 1 indicating that the two shapes are exactly the same.

3.2 Codebook Generation Phase

In signer specific codebook approach, a separate codebook of patterns is generated for each individual signer. The number of classes in each codebook therefore varies as a function of writing sample (signer). The features discussed in this section have been used to represent each fragment, which is later used for grouping similar fragmented patterns to form the codebook. For each individual

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

signer, once the primary codebook has been generated, the next step is to represent the codebook with the set of features that allow comparing between two signature samples. Suppose for a signature handwritten D, let P^D be the number of classes of primary codebook such that the primary codebook of signature D with P^D classes are given by:

$$CD^P = \{C_J \; 1 \leq J \leq P^D\} \quad (2)$$

Where, C^{D^P} represents the primary codebook:

$$C_J = \{ f_{1,j}, f_{2,j}, f_{3,j}, \ldots\ldots\ldots f_{K,J} \} \quad (3)$$

Where, K=card (C_J), card, is the cardinality of C_J and f representing the set of feature of the pattern in the classes. To represent individual codebook, the proposed technique considers the codebook as a probability distribution. For each class of the codebook, it counts the number of elements and then computes the probability of occurrence of each pattern in the codebook. In other words, a histogram is created where each bin represents a pattern in the codebook and the frequency of the bin represents the number of elements in the class of that pattern. The histogram is normalized to convert it into a probability distribution. This is carried out for primary codebooks and hence each signature handwritten sample is represented by two distributions. In relation to that, the pseudo-code of codebook representation for each individual signer with features is given by the following algorithm:

1. Let number Of Patterns be the total number of elements in each codebook of individual signature.
2. Let class Elements be the number of elements in an individual class.
3. Let Prob be the probability of occurrence of current feature.
4. Let covMatrix be the covariance matrix in which each row is an observation, and each column is a variable. The covariance between two random variables is defined by:

$$\text{Covariance}(x,y) = \frac{\sum_{i=1}^{n}(x_i - m_1)(y_i - m_2)}{n-1} \quad (4)$$

Where, μ_1 and μ_2 denote the means of X and Y, respectively.

5. Let identity be a matrix which is computed by the covMatrix, and the size of the identity matrix is the same as covMatrix.
6. Let determinant be the size of square matrix, which is defied by:

$$\text{determinant}(A) = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} A_{i,\sigma_i} \quad (5)$$

where, A represents n×n matrix, sign is the sign function of permutation group which returns +1 for even and -1 for odd matrix respectively.

3.3 Matching Phase

In a neural network approach to the clustering, the neurons in the network are connected with a 1-D or 2-D structure, and they correspond to the codevectors. The feature vectors are feed to the network by finding the nearest codevector for each input vector. The best matched codevector and its neighboring vectors (according to the network structure) are updated by moving it towards the input vector. After processing the training set by a predefined number of times, the neighborhood size is shrunk and the entire process is repeated until the neighborhood shrinks to zero. This phase generates is for the NN to learn the relationship between a signature and its class (either "genuine" or "forgery"). Once this relationship has been learned, the network can be presented with test signatures that can be classified as belonging to a particular signer. In this work there is a challenge of creating a system with the ability to recognize and differentiate hand written signature and verify its authenticity not just from forgeries but also from those signed under duress. This poses a problem because we are trying to get the computer to solve a problem with a method of solution that goes outside the convention of writing an algorithmic process. The challenge involves making the computer solve the problem using a series of new steps. After a lengthy research, the only feasible solution required is using the concept of the Neurons in human brain, which is familiar with medical practitioners. Thus, the Back Propagation Artificial Neural Network algorithm has been chosen because it is probably the easiest to implement, while preserving efficiency of the network. A 3-layered BPANN consisting of an Input Layer (which holds input for the network), Output Layer (that holds the output data and also serves as an identifier for the input), and Hidden Layer (that acts as an interface between the input and output layers and serves as a propagation point for sending data from the previous layer to the next layer) was employed for this study. Training was carried out iteratively in two phases until the sum of square of the output errors reached an acceptable value. The trained data was tested against a GPDS dataset of 5400 samples. Signature verification was carried out by computing the equal error rate (ERR) and validated by

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparing it with the ERR values of other offline handwritten signature verification methods. ERR provides a quick comparison method for determining the accuracy of a verification system such that the lower the ERR the more accurate the system.

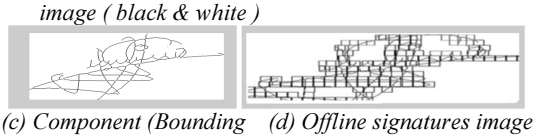
4. DISCUSSION OF RESULT

This section discusses the results of the proposed method as illustrated in the procedural methodology presented in section 3. Matlab and Delphi programming language was used in a Microsoft Windows operating system environment. Details of the results illustrated with diagrams and screen shots are discussed below. The validation of the proposed method's performance is also discussed at the tail end of this section.

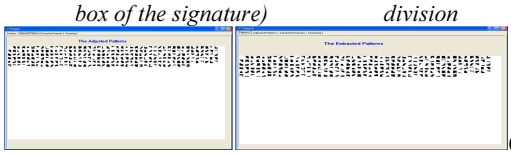
4.1. Pre Processing Phase

In this phase, the original signature image was scanned and converted from grayscale to binary image (Figure 2a) and then thinned out (Figure 2b) so as to remove the background noise for improved identification process. The thinned image was then bounded in a box for easy component identification (Figure 2c) after which Adaptive Window Positioning technique was used to segment the image into small window fragments of 13x13 window sizes (Figure 2d). The main objective of using the windowing technique is to trace the image trajectory by ensuring that no overlap exists. This is achieved by placing the windows over each of the identified component vertically (from top to bottom) and horizontally (from left to right), in an onward direction of the image trajectory. Next, we extract the patterns each signature image in each window (Figure 2e) and then adjust the patterns in each window by moving it to the upper left corner (Figure 2f). The pattern adjustment helps improve the accuracy of the feature extraction process which is a very vital input to the performance of the entire signature identification and verification process and makes computation much easier and with minimal error. Also, the higher the number of related features extracted, the better categorized the generated codebooks will be, implying a better identification result. The result of the feature extraction is shown in Figure 2g based on Eqn 1. as stated in Section 3 sub section A. The values shown in Figure 2g represents the frequency of patterns extracted from each window. The higher values suggest that there is a more specific pattern with the original signature in the data set, which implies that the similarity between the data set signature and the test signature is high.

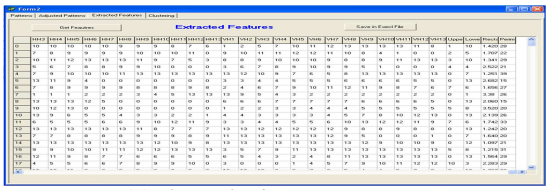
(a)Original image to Binary image (black & white) *(b) Thinning image*



(c) Component (Bounding box of the signature) *(d) Offline signatures image division*



(e) *Shows the extracted patterns* *(f) Shows the extracted patterns after making the adjustment*



(g) Shows the feature extraction

Figure 2: The procedural stages of the preprocessing phase

4.2. Codebook Generation Phase

In this phase, classification of the extracted features are grouped into classes based on their similarity characteristics, using the similarity function stated in Eqn. (2) and (3) in Section 3 sub section B. This helps in determining the range of values for a variety of features by classifying them based on a specific threshold magnitude. Hierarchical clustering was used for the classification of the extracted features such that each class length represents the authorship of signer and their respective differences. This clustering approach starts with each window as a single class and merges windows into the classes until all windows are in one cluster. The proposed technique needs to define a distance (or similarity) measure allowing comparison of two classes. Based on this study, a total of 47 classes were clustered from the many variety of features extracted as shown in Figure 3.

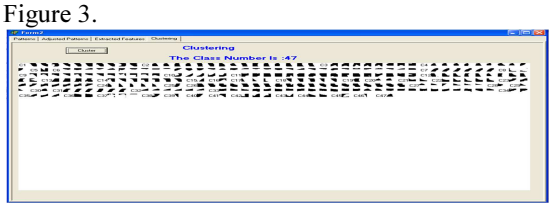


Figure 3: A cluster of 47 classes

In a signer specific codebook approach, a separate codebook of patterns is generated for each

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

individual signer. The number of classes in each codebook therefore varies as a function of writing sample (signer). The framework using signer specific approach is shown in Figure 4, which is a continuity of the main methodology discussed in Section 3 (Figure 1).

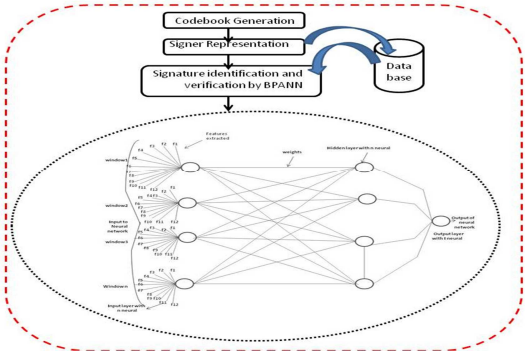


Figure 4: Framework For Writer Specific Approach

In this approach, a specific codebook of fragmented patterns is generated for each individual signature sample. The numbers of classes in each relevant codebook (primary codebook) are not known because it varies from sample to sample. In addition, since the number of classes for each signer is not known, this study uses the distance criterion to represent the number of clusters. For each signer the proposed technique generates the clusters from the main as well as adjacent windows. In this research, main and adjacent clusters are also termed as primary codebook. In relation to that, the pseudo-code of windows clustering to generate codebook is given in the algorithm presented in Section 3, sub section B. Once the codebook for main and adjacent windows are generated, the proposed technique sorts the classes according to the cardinality and keeps only those classes which have sufficient number of windows. Classes in each primary codebook correspond to the frequent stroke patterns occurring in the main and adjacent windows. In primary codebook, the number of patterns per class naturally depends upon the size of the signature sample but interestingly the number of classes is a signer-dependent parameter. This is illustrated in Figure 5 where the numbers of classes in primary codebooks are shown. The codebooks are generated from 30 different signers having two samples each. It can be seen that the curves representing numbers of classes in the two samples of a particular signer is more or less similar for both codebooks. This supports the idea that the number of classes is a signer-dependent attribute.

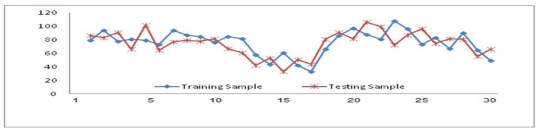


Figure 5: Number Of Primary Classes For Two Samples Of 30 Writers

As an example, primary codebook produced from the main and adjacent fragmented windows are illustrated in Figure 6. The codebook contains a number of different classes. Each class contains relatively homogeneous groups of similar patterns, which are dissimilar to elements in the other classes. These classes are separated by a class number in the codebook as illustrated in Figure 6.

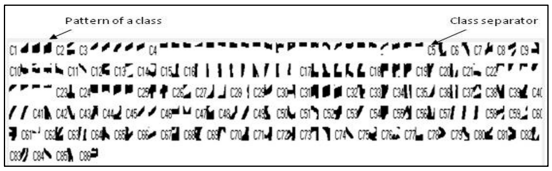


Figure 6 : Signer-Specific Primary Codebook Obtained From The Main Fragmented Windows On A Signature Sample

Once the primary codebooks are generated for each signer, the next step is to determine how to use this information to represent the codebook with the set of features that allow comparing between two signature samples. To represent each individual codebook, the proposed technique considers the codebook as a probability distribution. For each class of the codebook, it counts the number of elements and then computes the probability of occurrence of each pattern in the codebook. In other words, a histogram is created where each bin represents a pattern in the codebook and the frequency of the bin represents the number of elements in the class of that pattern. The histogram is normalized to convert it into a probability distribution. This is carried out for primary codebooks and hence each handwritten signature sample is represented by two distributions.

4.3 Matching Phase

Using a 3-layered BPANN approach consisting of an Input Layer (which holds input for the network), Output Layer (that holds the output data and also serves as an identifier for the input), and Hidden Layer (that acts as an interface between the input and output layers and serves as a propagation point for sending data from the previous layer to the next layer) (details of this approach is discussed in Section 3, sub section C), we take the black nodes (on the extreme left of Figure 6) as the initial inputs. Training such a network involves two phases. In the first phase, the

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

inputs are propagated forward to compute the outputs for each output node. Then, each of these outputs is subtracted from its desired output, causing an error [an error for each output node]. In the second phase, each of these output errors is passed backward and the weights are fixed. These two phases are continued until the sum of square of output errors reaches an acceptable value. Each neuron is composed of two units. The First unit adds products of weights coefficients and input signals while the second unit realizes nonlinear function, called neuron activation function. Signal E is adder output signal and Y=F E is output signal of nonlinear element. Signal Y is also the neuron's output signal. To train the neural network, we need dataset. The training data set consists of input signals X1 and X2 assigned with corresponding target (desired output) Y. The network training is an iterative process. In each iteration, the weight coefficients of nodes are modified using new data from the training dataset. Each training step starts with forcing both input signals X1 and X2 from the training dataset to the neural network. After this stage we can determine the output signal values for each neuron in each network layer. The entire process is repeated until the neighborhood size is shrunk to zero. This result generated by this phase helps the NN to learn the relationship between a signature and its class (either "genuine" or "forgery"). Once this relationship has been learned, the network can be presented with test signatures that can be classified as belonging to a particular signer. This method yields a better maximal matching result. The screen shots of the above discussed processes are illustrated in Figure 7. Figure 7a shows the menu page of the proposed signature identification and verification system based on BPANN method, while Figure 7b shows the neural network training processes for 29 iterations (epochs) with a performance of 1.79 and a gradient of 4.86. The performance of the system is measured by the Mean Squared Error (MSE) while the neural network training is measured by the gradient. Figure 7c shows that as the number of iterations (epochs) increased to 69, the neighborhood size came to near zero, recording a MSE value of 0.00094675, which is the best training performance value derived by the system, with a gradient of 0.013439 and a learning rate of 0.28978 (Figure 7).

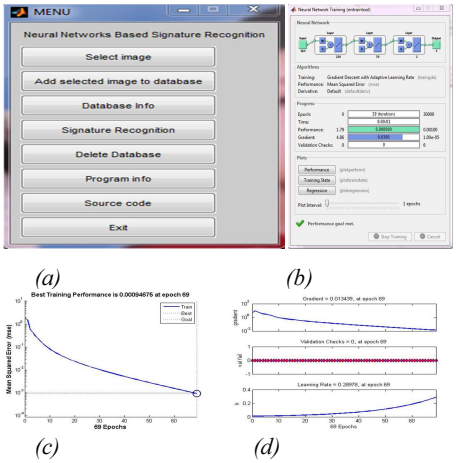


Figure 7: The Matching Phase

However, the objective is to create a system that can identify and distinguish between the handwritten signatures and verify its authenticity not just from forgeries but also from those signed under duress. Thus, by using all 24 original and 30 forged signatures from the GPDS dataset of 5400 signature samples, we were able to determine the average deviation of feature properties for both the forged (False Acceptance Rate - FAR) and the original (False Rejection Rate - FRR) with respect to the threshold as shown in Figure 8. The FAR and FRR values obtained for the proposed method are shown in Table 2. Based on the graphical analysis (Figure 8), it was observed that a similar pattern exists for both FRR and FAR of each signer when referenced to the threshold implying that a forger cannot reproduce every aspect of original signature with same accuracy as the original signer. Also, it indicates that even though the same systematic error may be repetitively made by a particular forger, when weighed against other possible forgeries, the deviations tend to cancel themselves out. As such the properties of a feature reproduced by a forger will always vary around a given mean as the means always tend to overlap. However, for a genuine signer who may tend to produce a signature under duress, there are certain unconscious features that will always remain stable irrespective of the influencing human factors, which are almost impossible to forge.

5. VALIDATION OF RESULT

The validation of the performance of the proposed system was carried out using three standard evaluation criteria. Firstly, the False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate (EER) were computed and compared against two other widely accepted

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

signature identification and verification methods (Table 2). FAR was used to account for all the skilled forgeries, FRR for only genuine signatures, and EER for verification. The method with the lowest EER can be considered as the most accurate technique. As shown in Table 2, with a GPDS dataset containing 5400 signature samples from 100 different signers, the EER value for the proposed was found to be 7.23, thus proving to be a more superior method than the others. Also, most studies did not compute FAR and FRR but based their judgments on only EER. EER is considered very useful for describing the general performance of a verification system but unable to provide information on how the systems performs when tuned to make stricter or more tolerant decisions. The wide margin between FAR and FRR (+-2.87) obtained for this study indicate that both FAR and FRR cannot be used to admissibly approximate EER, implying its ability to be used for describing the real world application where the focus is biased on keeping the FAR or FRR low [13].

ACKNOWLEDGMENT

We wish to acknowledge the Faculty of Computing, Universiti Tecknologi Malaysia (UTM) for their support to this research.

6. CONCLUSION

Signature signing is an act consciously carried out by the signer and as such should be associated with human behavior. Just like other areas of human activity and behavior, signatures are also influenced by a host of human factors such as the writing position, fatigue, and type of pen used, the state of mind of the signer, etc. and which may influence the final output of the signature. Thus, there is need to consider signature the human behavioral factor when authenticating signatures and in developing signature verification systems. The major assumption of this study is that firstly, the proposed signature verification algorithm was based on the assumption that certain unconscious features of the signer always remains stable irrespective of the above mentioned factors that may tend to produce variations in the signature of the same person. Secondly, the study assumed that these stable features are impossible to forge. Thus, the algorithm used for the proposed system has been developed to help identify, extract and efficiently compare these features so as to distinguish between forged and genuine signatures signed under duress. This research can be extended to include handwritten words or letters written by an individual though for the purpose of this study. The major limitation of the study is in the use of GPDS dataset for collecting signature samples used in this study. Although GPDS database has a very wide range of signers, its binary images are very low in quality and this may affect the accuracy of the feature extraction process which is a very vital input to the performance of the entire signature identification and verification process and can also influence the minimal error accuracy. Another limitation is that the proposed algorithm is for only offline handwritten signatures.

REFERENCES:

[1] B. Miroslav, K. Petra, F. Tomislav, "Basic on-line handwritten signature features for personal biometric authentication", *MIPRO, 2011 Proceedings of the 34th International Convention ,Opatija*, May 2011, pp. 1458-1463.

[2] S. Arora, D. Bhattacharjee, M. Nasipuri , L. Malik , M. Kundu and D. K. Basu, "Performance comparison of SVM and ANN for handwritten Devnagari character recognition", *International Journal of Computer Science Issues, vol. 7 (3),* May 2010, pp. 1-10.

[3] E. Alattas, and S. Meshoul, "An effective feature selection method for on-line signature based authentication", *Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Shanghai, vol. 3, July 2011, pp.1431-1436.

[4] H. B. Kekre, V. A. Bharadi, "Gabor filter based feature vector for dynamic signature recognition", *International Journal of Computer Application,* vol. 2 (3), May 2010, pp 74-80.

[5] Y. M. Al-Omari, S. H. S. Abdullah, and K. Omar, "State-of-the-art in offline signature verification system", *International Conference on Pattern Analysis and Intelligent Robotics,* June 2011, Putrajaya, Malaysia, 2011, pp. 59-64.

[6] S. A. Daramola, T. S. Ibiyemi , "Offline signature recognition using Hidden Markov Model (HMM)", *International Journal of Computer Applications,* vol. 10 (2), November 2010, pp. 17-22.

[7] S. K. Shrivastava, S. Gharde, "Support vector machine for handwritten Devanagari numeral recognition" *International Journal of Computer Applications* (0975 – 8887), vol. 7 (11), October 2010.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[8] D. Samuel, and I.Samuel , "Novel feature extraction technique for off-line signature verification system", *International Journal of Engineering Science and Technology* ,vol. 2 (7), 2010, pp. 3137-3143.

[9] Kumar, Pradeep, et al. "Hand Written Signature Recognition & Verification Using Neural Network." *International Journal of Advanced Research in Computer Science and Software Engineering* 3.3 (2013).

[10] Deka, Manoj Kumar. "Offline Signature Verification System Using Artificial Neural Network." *International Journal of Engineering Research and Technology*. Vol. 2. No. 4 (April-2013). ESRSA Publications, 2013.

[11] J. F. Vargas, M. A. Ferrera, C. M. Travieso, and J. B. Alonso, "Off-line signature verification based on grey level information using texture features," Pattern Recognition, vol. 44, no. 2, p. 375–385, Feb. 2011.

[12] L. Batista, E. Granger, and R. Sabourin, "Dynamic selection of generative–discriminative ensembles for off-line signature verification," Pattern Recognition, vol. 45, no. 4, p. 1326–1340, Apr. 2012.

[13] Kovari, Bence Andras, and Hassan Charaf. "Models and Algorithms in Off-Line, Feature-Based,HandwrittenSignature Verification." *A PhD Thesis Submitted to the Faculty of*Automation and Applied Informatics ,University of Technologyand Economics(Unpublished),Budapest,2013.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Different Offline Handwritten Signature Identification and Verification Methods

S/N	Approach	Characteristics	Advantages	Disadvantages
1.	Template Matching approach	- Employs pattern comparison process	- Suitable for detecting genuine signatures via rigid matching	- Not appropriate for detecting skilled forgeries
2.	Neural Networks (NN) approach	- Learns by example thus good for learning the underlying structure of data set. - Can be used to model complex functions - Highly suitable for modeling global features of handwritten signatures	- Widely accepted classifiers for pattern recognition problems - Has very low FAR and FRR results	- Not very suitable for modeling statistical and geometric features - Requires a highly representative data set
3.	Hidden Markov Models (HMM) approach	- Best suited for sequence analysis in signature verification - Uses stochastic matching (model and signature) to extract variability between patterns and their similarities - Has various topologies and adopts probability density function modeling in its design for the verification task	- Can easily detect simple and random forgeries in signature verification	- Very poor in detecting skilled forgery
4.	Statistical approach	- Employs statistical method to determine the relationship, deviation, etc between two or more data items - Uses the concept of Correlation Coefficients	- Good at identifying random and simple forgeries Its graphometry-based approach avails so many usable features for signature verification, e.g., calibration, proportion, guideline and base behaviours	- Its use of static features limits it from detecting skilled forgery
5.	Structural and Syntactic approach	- Uses symbolic data (e.g. signatures) structures such as strings, graphs, and trees to represent recognition patterns - Employs the use of Modified Direction Feature (MDF) to extract transition locations	- Appropriate for detecting genuine signatures and targeted forged signatures	- Very exhaustive method as it requires large computational efforts and training sets
6.	Wavelet-based approach	- It is a multi-resolution transform that can decompose a signal into lowpass and highpass information - Wavelet theory is employed in decomposing a curvature-based signature into a multi-resolution signal	- Can be applied in both offline and online signature verification - Can decompose a curvature-based signature into a multi-resolution format - Can be applied to symbolic languages such as Chinese and Japanese besides English	

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: An Evaluation Table Comparing the Proposed Method with other previously known Methods

NO	Author (Year)	FAR	FRR	EER (rand)	Processing	Classification	Classes	Signers	Original signatures	Skilled forgery (simple forgery)	Data bases
1	The proposal method	9.02	6.15	7.23	Preprocessing relation windows and using cluster algorithm	BPANN	one	100	2400	3000	GPDS
2	J. F. Vargas, M. A. Ferrera, C. M. Travieso, and J. B. Alonso .2011	-	-	9	LBP+GLCM features	LS-SVM	one	100	2400	3000	GPDS
3	L. Batista, E. Granger, and R. Sabourin . 2012	-	-	20.6	Dynamic selection of generative–discriminative ensembles	HMM+SVM	one	160	3840	4800	GPDS

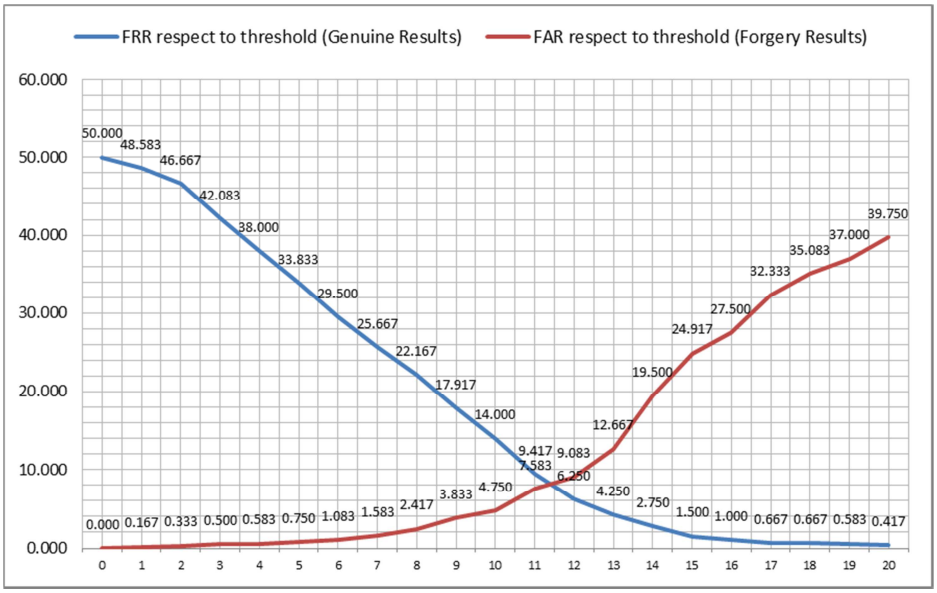


Figure 8: A graphical analysis of the FAR, FRR and ERR