

Signature Recognition Using Conjugate Gradient Neural Networks

Jamal Fathi Abu Hasna

Abstract—There are two common methodologies to verify signatures: the functional approach and the parametric approach. This paper presents a new approach for dynamic handwritten signature verification (HSV) using the Neural Network with verification by the Conjugate Gradient Neural Network (NN). It is yet another avenue in the approach to HSV that is found to produce excellent results when compared with other methods of dynamic. Experimental results show the system is insensitive to the order of base-classifiers and gets a high verification ratio.

Keywords—Signature Verification, MATLAB Software, Conjugate Gradient, Segmentation, Skilled Forgery, and Genuine.

I. INTRODUCTION

SIGNATURE verification is to evaluate whether a suspected signature is genuine or forgery. It's widely used in the fields of finance and security. Usually three kinds of forgery can happen in signature verification. Random forgery is taking the genuine signature of others for that of the current user. Skilled forgery is produced with close imitations. It is hard to be discriminated from the genuine one only by shape variations. Simple forgery is produced with the knowledge of content but without close imitations. For example, the forger signs out of his/her memory on the genuine signature.

Many systems for HSV have been proposed in the literature. Sabourin and Drouhard [1] presented a method based on directional probability density functions together with BP neural networks to detect random forgery. Qi and Hunt [2] used global and grid features with a simple Euclidean distance classifier.

In this paper, multiple classifiers integration using the Neural Network with verification by the Conjugate Gradient Neural Network (NN) algorithm is proposed. This system is designed to detect both random and simple forgeries. In the rest part of Bajaj and Chaudhury[3] proposed a system consisting of sub-classifiers that are based on three sets of global features. Sansone and Vento [4] proposed a sequential three-stage multi-expert system, in which the first expert eliminates random and simple forgeries, the second isolates skilled forgeries, and the third gives the final decision by combining decisions of the previous stages together with

reliability estimations. Baltzakis and Papamarkos [5] developed a two-stage neural network, in which the first stage gets the decisions from neural networks and Euclidean distance classifiers supplied by the global, grid and texture features, and the second combines the four decisions using a radial-base function (RBF) neural network.

This paper, section 2 discusses Quantifying Gradient Change in Images. Section 3 presents the SDM Analysis. Section 4 explains algorithm definitions. Section 5 mentions future work.

Finally, section 6 and 7 give our experimental results and conclude this paper.

II. QUANTIFYING GRADIENT CHANGE IN IMAGES

The majority of the generic shape recognition methods rely on using the topological features of an image. *SDM* quantifies the property of gradient change as the shape changes. The 16 level gray scale image obtained (in this study) is initially converted into a binary image by assigning every pixel value equal to or greater than 128 a value 1 and all others a value 0. The source image is skeletonized and divided into nine segments containing equal number of pixels. For each segment, its *SDM* feature is calculated. The set of rules employed can be summarized below.

Rule 1. The whole process proceeds in a horizontal manner, row by row. At any one time, two successive rows are under consideration.

Rule 2. The process continues unless all rows of the image pixels have been exhausted. Start with the first row.

Rule 3. Let's call the black pixel found in row i as B_i .

Rule 4. Find the distance between B_i and B_{i+1} . Lets say the distance is S_i .

Rule 5. Sum up all the distances for a total of k rows,

$$SD = \sum_{i=1}^k S_i; \quad SD \text{ is the String Distance for the whole}$$

segment. This is shown in Fig. 1 below with two examples illustrating positive and negative *SDs*.

Manuscript received June 26, 2006. This work was supported in part by the Near East University, Electrical & Electronics Engineering Department, KKTC.

Jamal Fathi Abu Hasna is with Near East University, Electrical & Electronics Engineering Department, North Cyprus, Turkey via Mersin-10, KKTC (phone: 00905338658472, e-mail: jamalfathi2004@yahoo.com).

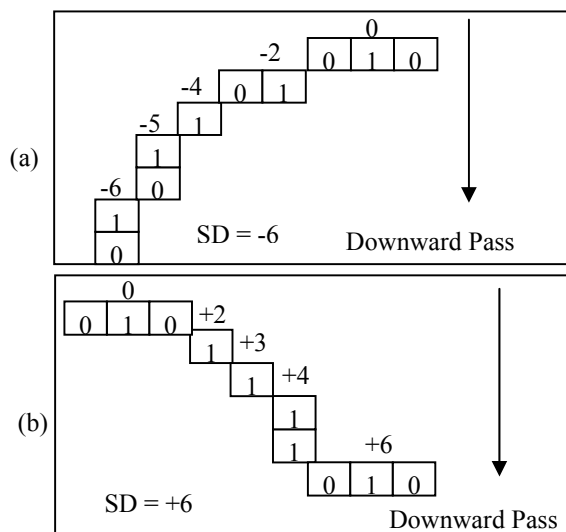


Fig. 1 SD Calculations (a) (Negative Values); and (b) (Negative Values)

SD represents the change in gradient as we traverse in downward direction which is summed up by considering any two adjacent rows at any one time. *SDs* are calculated for each of the nine segments. The overall procedure is described in the next section. From Fig. 1 (a) and (b), it is evident that *SDs* may have a zero value if the Fig. 1 (a) and (b) shapes coexist in the same segment. Hence multidirectional lines will contribute differently (positive and negative values) to the summed measurement.

Also single horizontal and vertical lines will have zero *SD* values. This may be shown below in Fig. 2 (a) and (b).

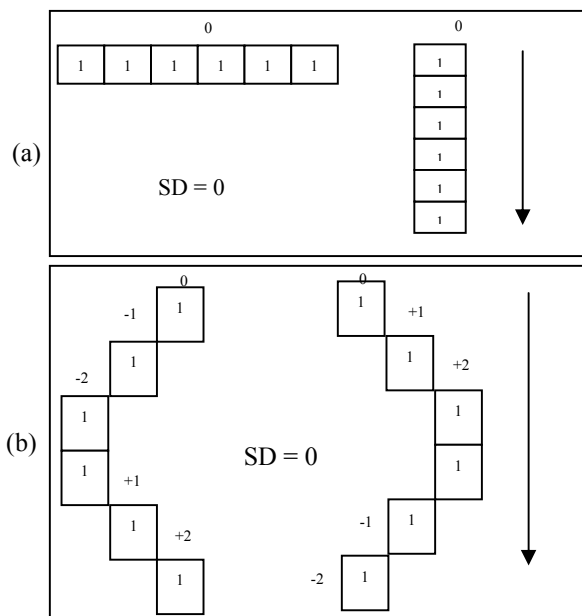


Fig. 2 SD Computations (Zero Values) for (a) and (b)

The *SD* computation can be summarized as: (when looking from the right hand side) Positive for convex shapes, Negative

for concave shapes, and Zero for shapes which are symmetrical around the *x* or *y* axis. However it is needed to include the extent of horizontal and vertical lines which are not properly measured by *SD* alone. Another parameter called String Line Measurement (*SLIM*) is therefore introduced. *SLIM* has a value 1 if a horizontal or vertical line is detected, otherwise it is 0. This *SLIM* value must be scaled to the *SD* measurement before it can be added to it. Also *SLIM* values must be modified to represent the contribution of longer lines as greater to the overall *SDM* value. This can be achieved by taking into consideration the total number of black pixels in the straight horizontal or vertical lines with respect to the total number of black pixels in the segment. The final equation used for the string distance measurement of segment *i* with a total of *n* V/H (vertical or horizontal) segments is in equation (1) below.

$$SDM_i : \alpha factor_i + \sum_{j=1}^n \beta factor_j \quad (1)$$

Where

$\alpha factor_i = SD_i$

$\beta_j factor_i = SD_i * (SLIM_j * Black\ Pixels\ within\ V/H\ segment_j) / (Black\ Count_j)$

Here *Black Count* represents the total number of black pixels within the individual segment (one of the nine) under consideration. Horizontal and Vertical scans were separately needed for each segment for identifying horizontal and vertical lines. If there is more than one black pixel in a row or a column, it indicates the presence of a horizontal or vertical line segment. Longer horizontal or vertical segments yield a higher value for the $\beta factor$ and increase the overall value of *SDM*.

III. SDM ANALYSIS

The overall analysis can be divided into three stages: image acquisition and preprocessing, feature extraction, and neural network analysis. For the draft as examining our system 77 persons were asked to sign using the pen for Casper Tablet PC Computer in Windows Journals Program, and for final different 5 persons also asked to sign as testing our proposed system as shown in Fig. 3 and used as database.

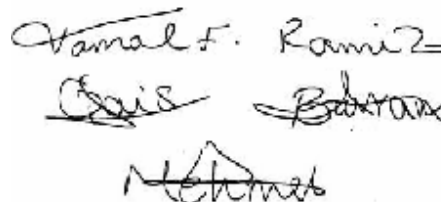


Fig. 3 Database

The image was first skeletonized and segmented as shown in Fig. 4. The obtained skeleton by making use of a conditional erosion algorithm which eroded the original image with four successive passes: from left, right, up and down, using the condition: "erode if (next pixel is black) and (one or more surrounding pixels are black)".

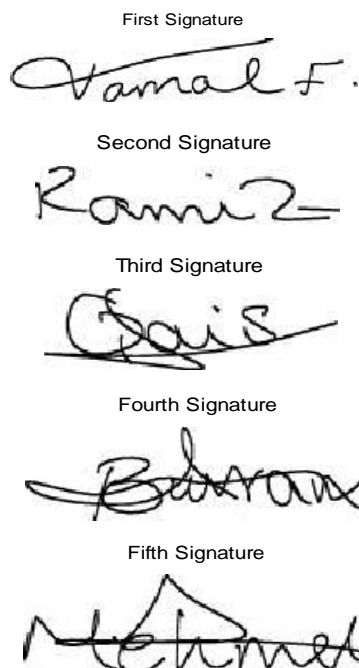


Fig. 4 Segmented Signatures, (Line Segmentation)

The process continued until the image could no longer be eroded. The resultant image was clipped so that the edges of the image were confined within a fixed boundary as shown in Fig. 5.



Fig. 5 Confined Edges within Boundary

The image was then segmented in 5 parts as in Fig. 5, each segment with its own binary address. The features (*SDM* values) were extracted for each segment and the complete image was described by vector $S = (SDM1, SDM2, \dots, SDMn)$. In practice although for two different images, a few of the segments may yield identical *SDM* values, it would be rare if all the five values were same for completely different shapes. The patterns obtained were used as inputs to the neural network for recognizing different signatures. The overall process can be shown in Fig. 6 with the following flowchart.

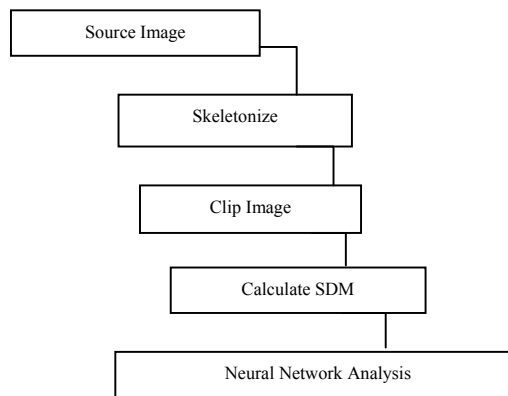


Fig. 6 Flowcharts for Methodology

The skeletonized image can be shown in Fig. 7 below.

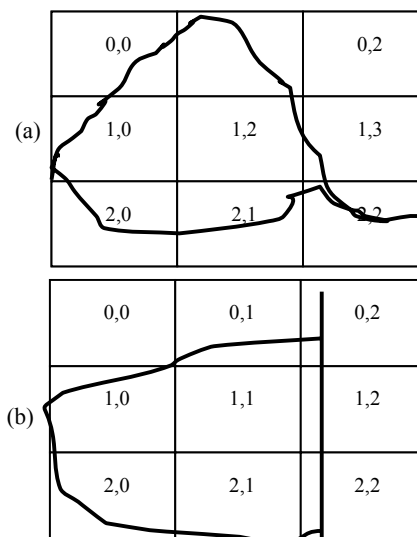


Fig. 7 Skeleton Image for (a) "a"; (b) "d"

Image processing software was developed which segments the image and analyzes the partitioned segments individually. It can be seen from Fig. 7 (a) and (b) that *SDM* values for different segments will be vary across different images, e.g. for Fig. 7(a) segment (0, 0) it has a negative value, but it is nearly zero for the same segment in Fig. 7(b), then applied to the Conjugate Gradient N.N.

IV. ALGORITHM DEFINITION TRAINING & TESTING

A Multi-layer Neural Network trained using Conjugate Gradient classifies the authentication attempt of a user as who is the signer. The neural network is initially trained against a set of 77 valid signatures given by the user when he or she is first introduced to the system as well as a set of *m* target signatures; the training process is shown in Fig. 8 with a goal of 0.001, the performance was 0.000999985, and 31 Epochs.

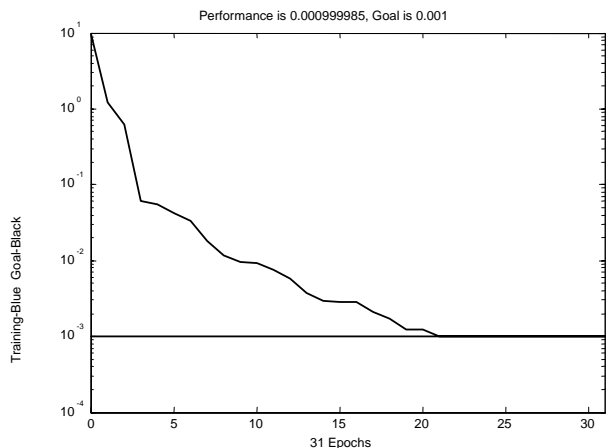
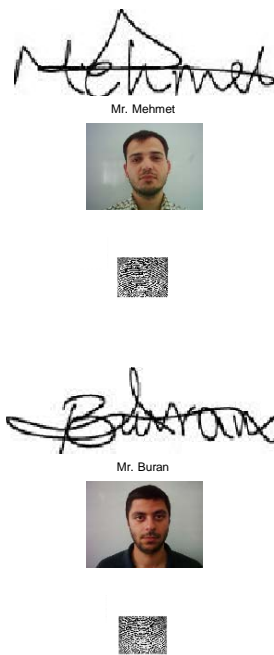


Fig. 8 N.N. Training Graph



For testing, data was collected from 5 participants. Each participant was asked to sign his name ten times as consistently as possible. Most participants have not used a tablet PC before, which may have affected the consistency of their signatures. The result of testing the program is shown in Fig. 9.

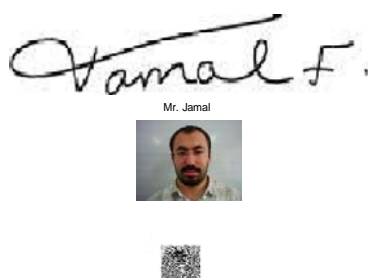


Fig. 9 Result contains Person's Signature, Name, Picture and his Fingerprint

V. SOME EXAMPLES



VI. COMPARISON TESTS

Compared with single feature set, our method reaches much better performance on random forgeries (refer to the last row in Table I). It may be due to the fact that simple forgeries have smaller distances to the reference mean than random forgeries. Their introduction will decrease the relevant threshold, which in turn increases FRR will increase a bit accordingly. Based on the tests, we finally include 5 simple forgeries in the training dataset.

TABLE I
 SYSTEM VERIFICATION RESULTS

Error Rate	Total Error (%)
Texture Features	2.83
Grid Gray Features	7.45
Ink Distribution Features	5.72
Global Features	11.97
Integrated Classifier	1.69
The present approach	1.6

The present approach is also compared with that proposed by Baltzakis and Papamarkos[1]. Fig. 10 records error rates of individual feature set in combination with neural network.

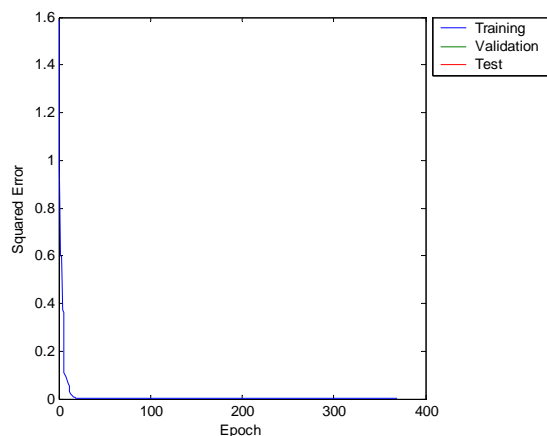


Fig. 10 Squared Error

VII. CONCLUSION

Transforming the input before training yields much lower error, but is more sensitive. Most importantly, we have presented system can vary in security depending on the situation.

Uses for such a system range from securing a credit card transaction at the point of sale to user authentication on tablet PCs. We hope that this system will help future research in creating variable security HSV systems as well as systems which can select feature sets which are optimal for a specific user.

ACKNOWLEDGMENT

My deepest thanks are to Prof. Dr. Senol Bektas, and to Prof. Dr. Fakhraddin Mamedov. I would like to express my gratitude to my colleagues in the department. Also I would like to express my gratitude to my family.

REFERENCES

- [1] Sabourin R, Drouhard J P, 1992, Offline signature verification using directional PDF and neural networks, Proceedings 11th international conference on pattern recognition, 2:321-325.
- [2] Qi Y Y, Hunt B R, 1994, Signature verification using global and grid features, Pattern Recognition, 22(12): 1621-1629.
- [3] Bajaj R, Chaudhury S, 1997, Signature verification using multiple neural classifiers, Pattern Recognition, 30(1):1-7.
- [4] Sansone C, Vento M, 2000, Signature verification: increasing performance by a multistage system, Pattern Analysis & Application, 3:169- 181.
- [5] Baltzakis H, Papamarkos N, 2001, A new signature verification technique based on a two stage neural network classifier, *Engineering*.
- [6] Eric W Brown, feneric@ccs.neu.edu *Applying Neural Networks to Character Recognition*.
- [7] Bazzi, Issam, Richard Schwartz and John Makhoul (1999) An omnifont open-vocabulary OCR system for English and Arabic. Pattern Analysis and Machine Intelligence 21 495-504.
- [8] D. A. Mighell, "Backpropagation and its application to handwritten signature verification", Advances in Neural Information Processing Systems I, pp. 340-347, 1989.
- [9] Holger Schwenk, and Yoshua Bengio, Adaptive Boosting of Neural Networks for Character Recognition.



Jamal Fathi Abu Hasna was born in Tulkarm on November 19, 1964. He came to the North Cyprus in 1997, and graduated with a Bachelor of Science degree in Electrical & Electronics Engineering from Near East University in 1998-99. He continued to the graduate program at Near East University and completed the Master of Science in 2000-2001 and continued the Doctor of Philosophy degrees in 2001 and now in

process.

His current interests include neural computing, adaptive signal processing, cellular communications, control systems, and wavelets.