Off-line Signature Verification Using Curve Fitting Algorithm with Neural Networks

Vaibhav Shah, Umang Sanghavi, Udit Shah
Dwarkadas J. Sanghvi College of Engineering, Mumbai.

***Abstract*— As signature is widely used as a means of personal verification, it is necessary for an automatic verification system. Offline and Online are two methods of verification based on the application. Online systems use dynamic information of a signature captured at the time the signature is made. Offline systems work on the scanned image of a signature. Processing Off-line is complex due to the absence of stable dynamic characteristics and also due to highly stylish and unconventional writing styles. A simple and a reliable system has to be designed which should detect various types of forgeries. Hence this paper proposes architecture for off-line signature verification. Our approach makes use of runtime signature instead of scanned images for recognition. This Offline verification of signatures uses a set of shape based geometric features and more importantly focuses on the distance based parameters such as the continuity of the signature and matching of the curves of the signatures generated by the critical points of the respective signature by analyzing the polynomial equation. Curve fitting and the analyzing of polynomial equations is one of the least explored topics till date but yet very efficient and hence we have implement this novel technique.**

***Index Terms*—aspect ratio, base angle inclination, normalized area of the signature, centre of gravity, edge points, cross points, looplength ,continuity breaks, matching of curves, analysis of polynomial equations, distance.**

I. INTRODUCTION

In online verification, the online information of pen tip (position, speed, and pressure) is obtained by using a hand pad together with an instructed pen [4] or a video camera. Therefore the input is a sequence of features. In the offline case, scanner or other device captures the two dimensional signature images. Online signature verification has been shown to achieve much higher verification rate than offline verification [3]. Online verification achieves equal error rates (EERs) ranging from 2% to 5% [4], while the EERs of offline verification are still as high as 10%-30% [3, 5]. This difference is mainly due to the availability of run-time information in online system. Although online verification outperforms the offline one, its use of special gadgets for noting down the pen-tip trajectory increases its hardware cost and brings limitations on its applications. In some situations such as check transaction and document verification, offline signature is a must. This paper therefore focuses on offline signature verification and our prime objective is to discriminate between genuine signatures and skilled forgeries. It should have an acceptable trade-off between a low False Acceptance Rate (FAR) and a low False Rejection Rate (FRR). The FRR measures the percent of valid inputs which are incorrectly rejected. The FAR measures the percent of invalid inputs which are incorrectly accepted. There are three kinds of forgeries–Skilled, Random and Casual. Examples of different forgeries are shown in Fig. 1.

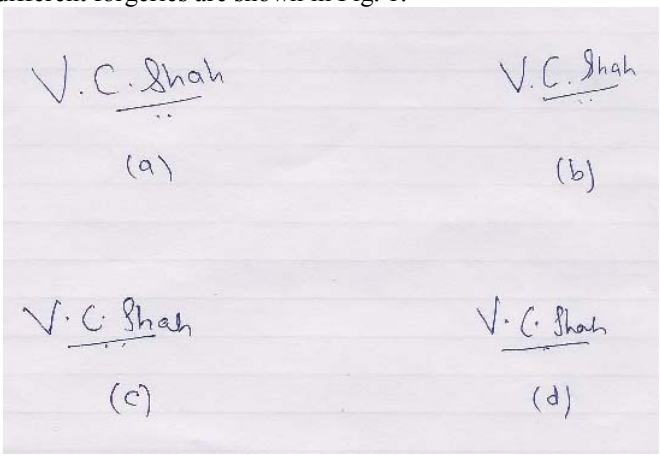


Fig.1.
Example of (a) genuine signature,(b)skilled forgery
(c) Casual forgery and (d) random forgery.

II. ALGORITHM

Our proposed solution is as follows. Initially a set of signatures are obtained from the subject and fed to the system. These signatures are pre-processed. Then the pre-processed images are used to extract relevant geometric parameters that can distinguish signatures of different persons. Then the curves of signatures are produced by using the critical points and their equations are analysed. The extracted features are fed into the neural network and trained accordingly. In the next step the signature taken during runtime to be verified is given to the system and stored in the database. It is pre-processed to be suitable for extracting features. It is fed to the system and various features are extracted from it. The corresponding curve of the signature is also calculated. Then these values are also fed into the neural network which produces an output indicating whether the signature is a

``

genuine one or forged. Depending on whether the input signature satisfies the threshold the system either accepts or rejects the signature. A flow chart illustrating the various steps that have been used is shown in Fig. 2.

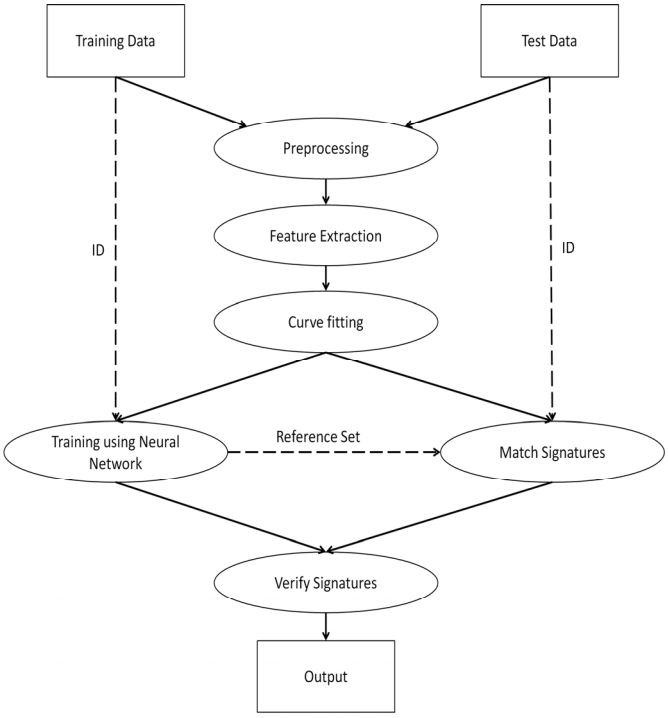


Fig. 2. Flowchart

Different from other offline signature verification algorithms, our solution is based on the recovered trajectories which do not have dynamic writing information such as speed, pressure, and orientation. To compensate the loss of the dynamic information, we have introduced the various geometric features and use the curve fitting method and then analyse their respective polynomial equations. We have developed a verification criterion which compares the two signatures based on the respective parameters. Experimental results show that our method achieves EERs which are comparable to the online methods and better than the other offline methods.

III. PRE-PROCESSING

The run-time signature image may contain spurious noise and has to be removed to avoid errors in the further processing steps. Niblack algorithm is used to remove the noise from the image. It is a simple and efficient method for adaptive thresholding. The algorithm reads the image, converts the image into greyscale and plots the histogram of the resultant image. The niblack filter size has to be selected appropriately. The local means and variables are calculated based on the order of the filter and the filtered image is then displayed. The result obtained after applying Niblack algorithm on Fig.3 is shown in Fig 4.

Fig.3.Original image

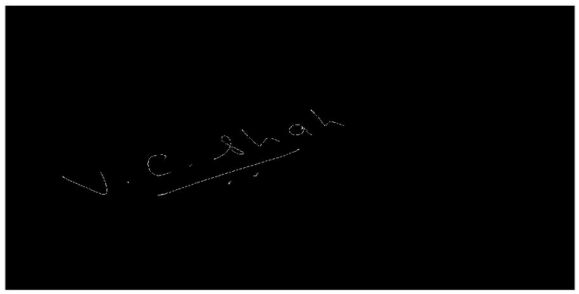


Fig.4. Image after applying Niblack Algorithm

Once Niblack algorithm is applied, further feature extraction is done on the cropped image. In order to crop the image, we got the co-ordinates of the bounding box by scanning image in four steps: scanning to get the topmost white pixel point on the right, right bottom white pixel point, left topmost and bottommost white pixels. The co-ordinates of the four points thus obtained are then used to crop the image in MATLAB. Cropping of signature removes the excess areas of black region which contains no information. The resultant image after cropping will be such that a bounding box will be formed around only white pixel elements. The cropped image is shown in Fig.6.

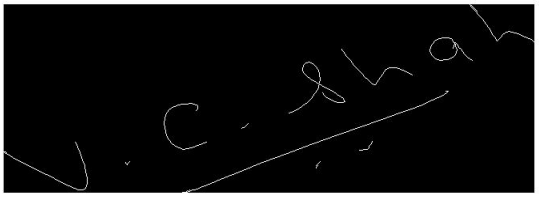


Fig.5.Cropped Image

IV. FEATURE EXTRACTION

Using a set of features, we can uniquely characterize a candidate's signature. These features are geometrical features based on the shape and dimensions of a signature image. The various shape features that can be used are:

1) Baseline Slant Angle

Baseline is the imaginary line about which the signature is assumed to rest. The angle of inclination of this line to the horizontal is called the Slant Angle Θ. To determine the slant angle the ratio of the maximum horizontal projection to the width of the projection is maximized over a range of values of angle of rotation θ.

``

$$P_H(i) = \sum_{j=0}^{N-1} I_T(i,j)$$

$$\rho(\theta) = \frac{H(\theta)}{W(\theta)} \quad \theta_1 < \theta < \theta_2$$

$$H(\theta) = Max \;\; P_H(i)$$

$$W(\theta) = number \;\; of \;\; non\ zero\ elements\ in\ P_H(i)$$

Θ is the value of θ at which $\rho(\theta)$ attains maximum. The ratio $\rho(\theta)$ is smaller at every angle other than the baseline slant angle. The threshold image I_T is rotated by this angle to obtain the slant normalized signature image I_R.

2) Aspect Ratio

The aspect ratio (A) is the ratio of width to height of the signature. The bounding box coordinates of the signature are determined and the width (D_x) and height (D_y) are computed using these coordinates.

$$A = \frac{D_x}{D_y}$$

3) Normalized area of the signature

Normalized area (NA) is the ratio of the area occupied by signature pixels to the area of the bounding box.

$$NA = \frac{\Delta}{D_x D_y}$$

$$where\ \Delta\ is\ the\ area\ of\ \ signature\ pixels.$$

4) Center of Gravity

The Center of Gravity is the 2-tuple (X,Y) given by,

$$X = \sum_{j=0}^{N-1} P_V(j) \otimes \frac{j}{\Delta}$$

$$Y = \sum_{i=0}^{M-1} \; P_H(i) \otimes \frac{i}{\Delta}$$

$$where\ P_V\ and\ P_H\ are\ the\ vertical\ and\ horizontal\ projections\ respectively.$$

Using these formulae the center of gravity for the two halves i.e. for left and right half is calculated.

5) Number of Edge Points

The edge point is a point that has only one 8- neighbour. In order to extract the edge points in a given signature, a 3×3 structuring element should have only one pixel equal to 1 and others equal to 0.

6) Number of cross points

Cross point is a signature point that has at least three 8-neighbours i.e. atleast three neighbouring pixels should be 1 and remaining equal to 0.

7) *Number* of closed loops (CL)

The number of closed loops can be defined as CL=1+ [(EL-EP) ÷2] with EP denoting the number of edge points and EL the number of extra departures, defined as

EL = ∑ (Number of 8-neighbors – 2)

All cross points

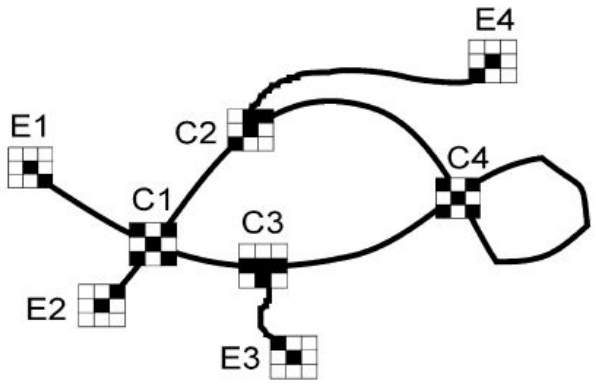


Fig.6.
Examples of Cross points (C1, C2, C3, C4) and edge points (E1, E2, E3, E4) and CL=2 [6]

8) Looplength

This parameter gives the length of all the closed loops encountered in the signature i.e summation of perimeter of all the closed loops.

9) Number of continuity breaks

Breaks in the signature are the number of points the user lifts the pen while signing. Fluidity is an important factor and it is unique for each user and hence calculating the number of breaks is a critical parameter.

V. Curve Fitting

Curve fitting is the process of constructing a curve, or mathematical function that has the best fit to a series of data points, possibly subject to constraints. It is derived from the trajectory of the signature. In this paper, we have used the Polynomial type of fit to generate the curve which is extremely peculiar of each user as each one has a style of its own. Then we have used the corr2 function to correlate the curves of the training and the test data which verifies the signature depending on the threshold. The below figure depicts the curve of the above cropped signature after using Polynomial type of Curve Fit.

``

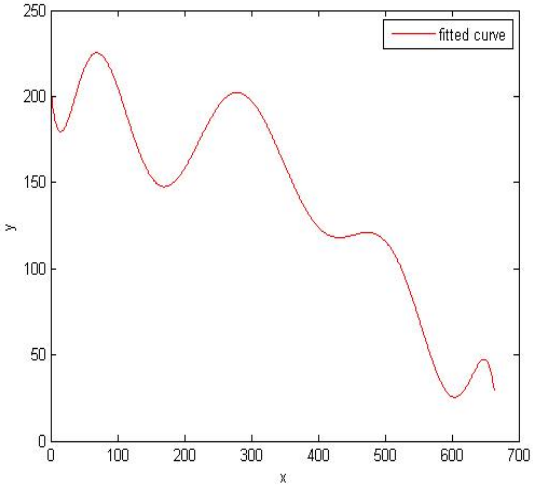


Fig.7. Curve after Fitting

Corr2 computes the correlation coefficient using

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

where $\bar{A}$ = mean2(A), and $\bar{B}$ = mean2(B).

In addition to the Corr2 function, we also generate the 9th degree polynomial equation of the developed curve. As we now have all the coefficients of the respective equation, we use its peculiarity which is summation of all the coefficients of the equation which is fed to the neural network. If the signature is genuine then the summation of the coefficients of its curve will be same or a multiple of the data given for training.

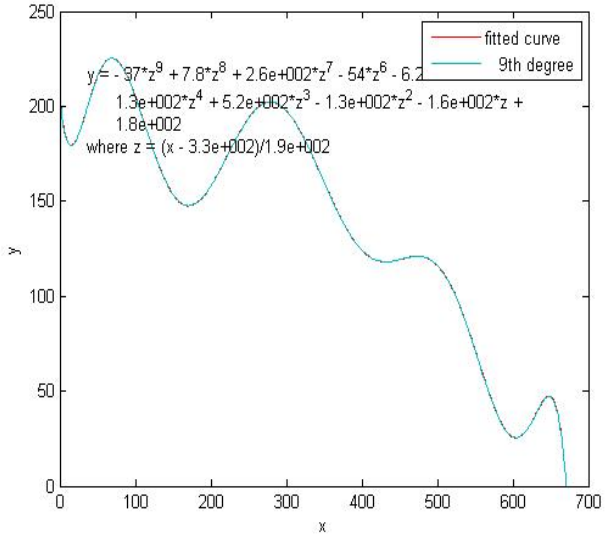


Fig.8 Curve with its 9th degree polynomial equation

VI .Training using Neural networks

We use the feed forward back propagation neural network to verify the authenticity of the signatures. The various parameters discussed above are given as inputs to the neural network. Based on the inputs, the neural network is trained and according to the target values specified, the corresponding outputs and error values are obtained for the particular parameter under test. This procedure is repeated for all parameters under consideration. The outputs are exported to the testing code for the two signatures. If any of the parameters has error greater than the threshold specified by the programmer, the signatures will not be verified. The neural network has to be trained for satisfactory working. Greater the number of samples trained; more satisfactory will be the output.

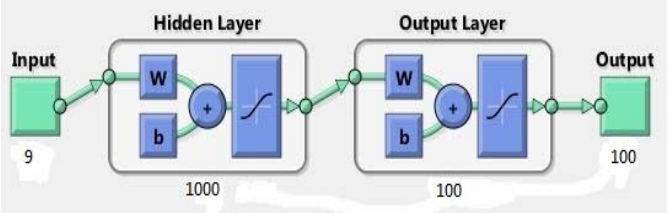


Fig.9. Feed Forward Back Propagation Network

VII. EXPERIMENTAL RESULTS

The results of the various features as extracted from the above signature are shown in Table 1.

Table 1: Values of the various features Extracted for the two samples.

Mean Values of Features	Sample Signature
Baseline Slant Angle	10.5967
Aspect Ratio	2.8455
Normalized Area	0.0078
Centre of Gravity	(260.743,101.405)
Number of Edge Points	25
Number of Cross Points	2
Number of Closed Loops	2
Looplength	160
Number of Continuity Breaks	3

After extracting the features, then we generated the curve using Polynomial type of fit. Then the curve is used to find the correlation coefficient(r) between the test and the training data. The correlation coefficient(r) value lies between 0 and 1 and depending on its value; the authenticity of the signature is verified.

After curve fitting, for extreme accuracy and precision, we have trained the network using supervised learning with the help of neural network. We have chosen Feed forward Back Propagation Network as it is fault tolerant and its learning rate α and β are high. The neural network is trained for 100 users and each user provides 10 signature samples as there is always a slight variation in the two signatures signed by the same

``

user. Each user is given a unique identification number and based on the output of neural network, the ID is verified.

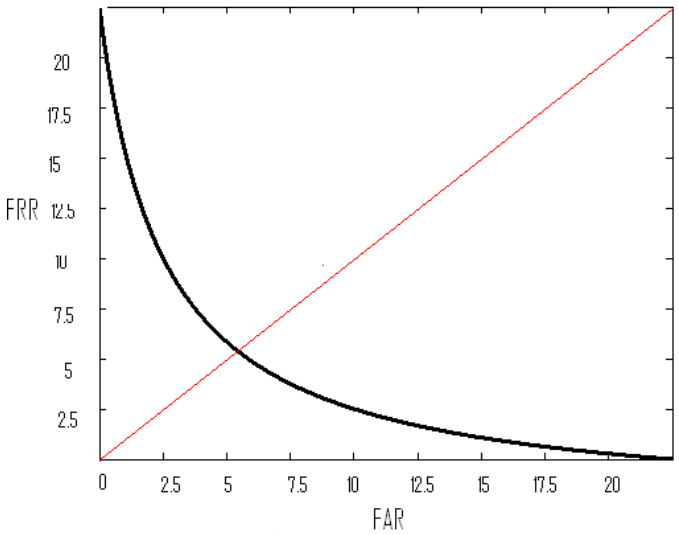
The false rejection rate (FRR) and the false acceptance rate (FAR) are used as quality performance measures. When the decision threshold is altered so as to decrease the FRR, the FAR will invariably increase, and vice versa. Out of the 76 genuine signatures that were fed in, 4 were rejected as forgeries. This yielded a False Rejection Rate (FRR) of 5.26%. Also out of 50 skilled forgeries fed into the system, 5 signatures were accepted. This gave us a False Acceptance Rate (FAR) of 10%.Other results which were obtained are shown in Table 2 below.

Table 2: FAR and FRR for original signature, casual forgery, skilled forgery.

Nature of signature	Samples	False acceptance rate	False rejection rate
Original	75	2%	5.26%
Casual forgery	75	3%	5%
Skilled forgery	50	11%	4.5%

VIII. CONCLUSION AND FUTURE WORK

We implemented our code using 75 samples of genuine signatures and received FAR=2% and FRR=5.26%.Thus we get the following graph:



Graph 1: FRR versus FAR

This graph shows EER to be about 3%-6% which is very much comparable to online algorithms. The algorithm uses curve matching technique and various geometric features to characterize signatures that effectively serve to distinguish signatures of different persons. The system is robust and can detect random, simple and semi-skilled forgeries but the performance deteriorates in case of skilled forgeries. Using a higher dimensional feature space and also incorporating dynamic information gathered during the time of signature can also improve the performance. The concepts of Wavelet transforms hold a lot of promise in building systems with high accuracy. The above algorithm is used to verify signatures online. Automatic on-line signature verification is an intriguing intellectual challenge with many practical applications.

``

REFERENCES

[1] Ashish Dhawan and Aditi R. Ganesan. Handwritten Signature Verification. *MS Thesis*, The University of Wisconsin Madison.

[2] Yu Qiao, Jianzhuang Li and Xiaoou Tan. Offline Signature Verification Using Online Handwriting Registration. *Technical Report*, The Chinese University of Hong Kong.

[3] M. Kalera, S. Srihari, and A. Xu. Offline signature verification and identification using distance statistics, 2004.

[4] V. Nalwa. Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2):215–239, 1997.

[5] R. Sabourin, M. Cheriet, and G. Genest. An extended shadow-code based approach for off-line signature verification. *Proc. ICDAR*, pages 1–5, 1993.

[6] H. Baltzakis, N. Papamarkos. A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence* 14 (2001) 95±103.

[7] Abhay Bansal, Bharat Gupta, Gaurav Khandelwal, and Shampa Chakraverty. Offline Signature Verification Using Critical Region Matching. *International Journal of Signal Processing, Image Processing and Pattern* Vol. 2, No.1, March, 2009