International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 6, June-2014, pp: (39-46), Impact Factor: 1.252, Available online at: www.erpublications.com.

Offline Signature Verification and Recognition System

Rosy Vig[1], Dr. Mahesh Kumar[2]
[1] Dept. of CSE, Mata Raj Kaur Institute of Engg & Technology, Rewari, Haryana, India
[2] Maharishi Dayanand University, Rohtak, Haryana, India

Abstract: Authentication of a person is the major concern in this era for security purposes. A number of biometric techniques have been proposed for personal identification in the past. Among the vision based ones are face recognition, fingerprint recognition, iris scanning and retina scanning. The primary advantage that signature verification systems have over other type's technologies is that signatures are already accepted as the common method of identity verification. As signatures continue to play an important role in financial, commercial and legal transactions, truly secured authentication becomes more and more crucial. A signature by an authorized person is considered to be the "seal of approval" and remains the most preferred means of authentication.

The method presented in this paper consists of image prepossessing, geometric feature extraction, neural network training and verification. In this paper signatures from database are extracted before feature extraction. After the signature is scanned through device and entered in to the system, it then be converted to gray image representing black and white in two dimensional image. A verification stage includes applying the extracted features of test signature to a trained neural network which will classify it as genuine or forged. The signature verification system is designed using MATLAB. These extracted features are then applied as input to a trained neural network which will classify it as a genuine or forged signature.

Keywords: offline signature, gray scale image, back propagation algorithm, Pattern Recognition Tool, Neural Network.

Introduction

Since a long time signature has been distinguishing feature for person identification. Even today signatures are being used for a number of transaction especially financials. Hence methods of automatic signature verification must be developed to achieve authenticity. Signature verification is a necessary task in society, as signatures have a well-established and accepted place as a formal means of personal verification. Due to this, they are used in government, legal and commercial transactions and they are the most accepted method of identity verification. An inevitable side-effect of signatures is that they can be exploited for the purpose of feigning a document's authenticity. In signature verification, there are two overall methods that are employed for determining whether a signature is a genuine or a forgery. Alexander T. Ihler, John W. Fisher III, and Alan S. Willsky present modeling dynamical processes in 2002[1]. They use this methodology to model handwriting stroke data, specifically signatures, as a dynamical system and show that it is possible to learn a model capturing their dynamics for use either in synthesizing realistic signatures and in discriminating between signatures and forgeries even though no forgeries have been used in constructing the model.

The system should neither be too sensitive nor too coarse. It should have an acceptable trade-off between a low False Acceptance Rate (FAR) and a low False Rejection Rate (FRR). The designed system should also find an optimal storage and comparison solution for the extracted feature points [2]. There are widely two types of approaches used: on-line and off-line signature verification. On-line signatures are carrying dynamic characteristics which are specific to each individual and sufficiently stable as well as repetitive. Off-line data is 2-D image of signature. Alan McCabe et al. proposed a method for verifying handwritten signatures by using NN architecture. Various static (e.g., height, slant, etc.) and dynamic (e.g., velocity, pen tip pressure, etc.) signature features are extracted and used to train the NN Several Network topologies are tested and their accuracy is compared [3].

Steps to be followed in our approach:

- Image pre-processing.
- Feature extraction.
- Neural network training.
- Verification

1. Image Pre-Processing

For image processing, the signatures are scanned into gray image. The images can be of .jpg, .tiff, .bmp etc. the reason for this step is to make image ready for the extraction. After the image has been pre-processed, the quality of image gets improved and further helps in feature extraction stage.
The process followed in pre-processing of image is as follows.

Figure 1 Pre- processing steps

1.1 Converting color image to gray image:

A gray scale image is converted to binary image through following MATLAB command:
I=rgb2gray(A) where A is image. It converts true color image A into gray scale intensity image I by eliminating the hue and saturation information but retaining the luminence.

Figure2.(a) shows Input Signature image; (b) Output Gray image

1.2 Converting gray image to binary image:

The gray scale signature after conversion will be converted to binary with following command:
BW= im2bw(I, level) converts the gray scale image I to a binary image. The output image BW replaces all pixels in the input image with luminence greater than level with the value 1(white) and replaces all other pixels with the value 0(black). The level should be in range of [0,1]. In a binary image, each pixel assumes one of only two discrete values: 1 or 0. A binary image is stored as a logical array.

1.3 Noise removal from images:

Images are prone to variety of noise. Noise is result of some errors in image acquisition process. Images are often corrupted due to positive and negative impulses coming from decoding errors or noisy channels. To remove noise introduced in the image, median filtering method is mainly used. Median filtering is 2-D filter and specific case of order statistics filtering also known as rank filtering.(L=medfilt2(J)) It is used for smoothing and restoring images corrupted by noise. We can also use an inbuilt function bw area open in matlab to remove the noise from background. It is non-linear process used in reducing impulsive or salt-and-pepper type noise.

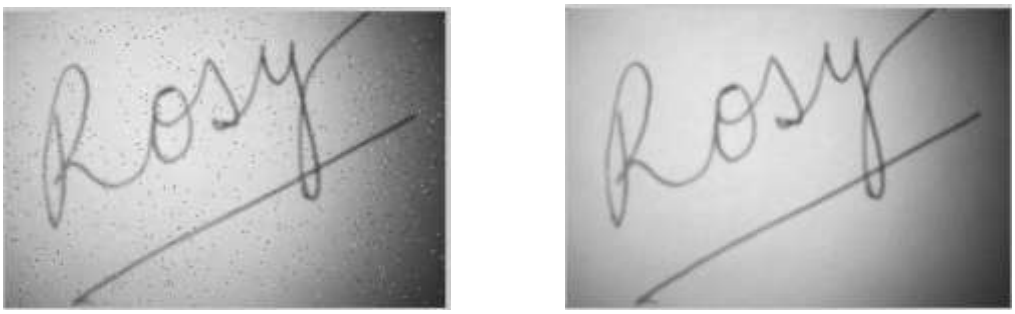

Figure 3. (a) Noise in image (b) Noise removed imag

1.4 Image Resizing:

The signatures obtained from signatories are in different sizes, resizing is performed so as to bring them in standardize size. To resize an image, imresize function is being used as follows:-
B = imresize (A, scale)
B = imresize (A, scale) returns image B that is scale times the size of A. The input image A can be a grayscale, RGB, or binary image. If scale is between 0 and 1.0, B is smaller than A. If scale is greater than 1.0, B is larger than A.

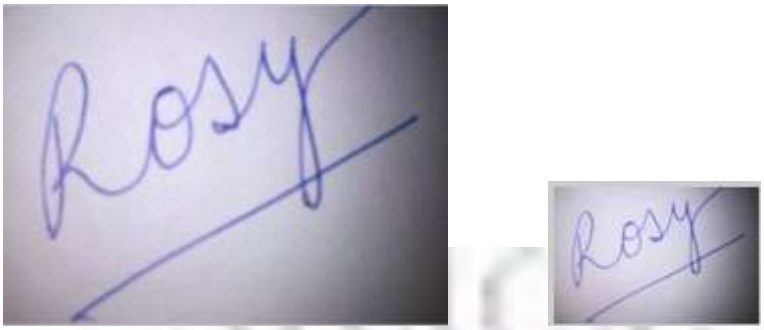

Figure 4 (a) Original Image **(b) Image after Resizing**

Use the im2bw function to convert the grayscale image into a binary image by using thresholding. The function graythresh automatically computes an appropriate threshold to use to convert the grayscale image to binary [4].

```
level = graythresh(I3);
bw = im2bw(I3,level);
bw = bwareaopen(bw, 50);
figure, imshow(bw);
```

1.5 Image Thinning:

Image thinning is also performed to make it invariant to change of stylus (pen or marker) used by user for signature purpose. By doing so each signature's thickness is changed to one pixel length. For image thinning purpose MATLAB inbuilt function is used which uses the algorithm proposed by (Lam et al., 1992)and [5] as described below:

1. Divide the image into two distinct subfields in a checkerboard pattern.
2. In the first sub-iteration, delete pixel p from the first subfield if and only if the conditions G1, G2, and G3 are all satisfied.
3. In the second sub-iteration, delete pixel p from the second subfield if and only if the conditions G1, G2, and G3 are all satisfied.

Condition G1:

$$X_H(p) = 1 \quad \text{(i)}$$

Where

$$X_H(p) = \sum_{i=1}^{4} b_i \quad \text{(ii)}$$

$b_i = \{1, \text{if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1)$
$0, \text{otherwise}$

X1,X2,……,X8 are the values of the eight neighbors of p , starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \leq \min\{n_1(p), n_2(p)\} \leq 3 \quad \text{(iii)}$$

Where

$n1(p) = \sum_{k=1}^{4} (x_{2k-1}) \vee x_{2k}$

$n2(p) = \sum_{k=1}^{4} (x_{2k-1}) \vee x_{2k+1}$

Condition G3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0 \qquad \text{(iv)}$$

Condition G3':

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0 \qquad \text{(v)}$$

The two sub-iterations together make up one iteration of the thinning algorithm. When the user specifies an infinite number of iterations (n=Inf), the iterations are repeated until the image stops changing.

Figure5. (a)Image after thinning (b) Another Image after thinning

1.6 Bounding the signature:

The solution is based on the identification of edges of the signature in the signature box. It scan the rectangular area from its four(i.e. top, bottom, left, right) sides. Thus now each image consists only of the signature part. And after that the actual signature area has been extracted. This reduces the area of signature to be used for further processing and saves time. During the process, the aspect ratio between width and height of a signature is kept intact[5]. The process made use of the following equations.

$$x_i = \frac{x'_i - x_{min}}{x_{max} - x_{min}} * M$$

$$y_i = \frac{y'_i - y_{min}}{y_{max} - y_{min}} * M \qquad \text{(vi)}$$

In above equations:

xi,,yi : pixel coordinates for the normalized signature,
x'i,,y'i : pixel coordinates for the original signature,
M : one of the dimensions (width or height) for the normalized signature

Figure 6 Signature image with bounding box

1.7 Image Rotation

It is well known that a person's situation differs in each signature from time to time or at the same time. There are many reasons behind this, such as psychology, health, or others. This leads to changes in inclination angles of the same person's signatures. Hence, the rotation algorithm must be used to unify signature orientation in a horizontal manner to overcome this problem. Figure shows the slant angel removed image.

Figure 7(a) Image before rotation **(b) Rotated Image**

2. Feature Extraction

The basic data structure in MATLAB is the array, an ordered set of real or complex elements [6]. This object is naturally suited to the representation of images, real-valued ordered sets of color or intensity data. MATLAB stores most images as two-dimensional arrays (i.e., matrices), in which each element of the matrix corresponds to a single pixel in the displayed image**.**

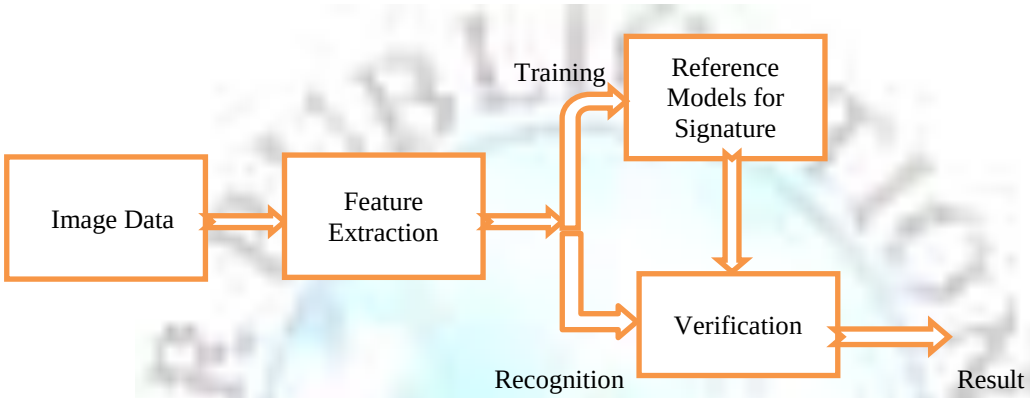

Figure 8: steps in feature extraction

Feature extraction performed in three steps:

(i) After image preprocessing we divide the image into 4 equal parts from center of the image and computes the COM (gx, gy) of each obtained sub-image by using following equation:

$$g_x = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad \text{(vii)}$$

$$g_y = \frac{1}{N} \sum_{i=1}^{N} y_i$$

Where N is the number of foreground pixels in the image (xi, yi).

(ii) Now divide each sub-image obtained from step (i) into another four sub-images from their respective computed COM.

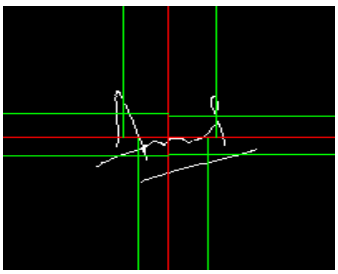

Figure 9. Sub-Images of an Image Space after step – (i) and (ii)

(iii) Now out of sixteen sub-images that are obtained after step (ii) four sub-images are chosen which are closer to the center of the original image space and their respective COM are computed. The set of four COM which are obtained in step (i) are represented as COM1 and set of four COM which are obtained in step (ii) are represented as COM2 as shown below in figure 10.

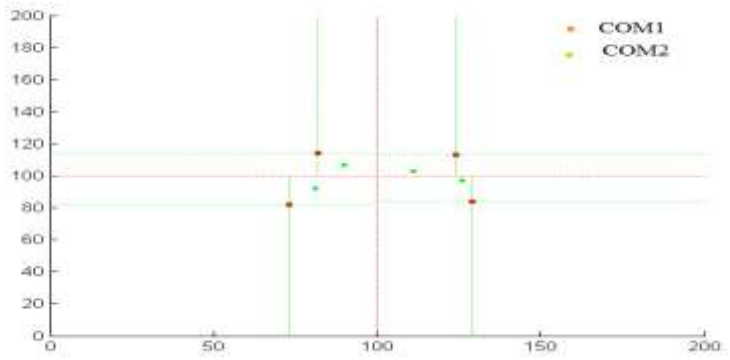

Figure 10: Selected Feature Points of above Image Space

Finally, we come up with total of eight critical feature points four from COM1 and four from COM2 for an input signature to our proposed signature verification system and are used for verification purpose.

3. Verification

For the purpose of signature verification we use the k-NN classifier.

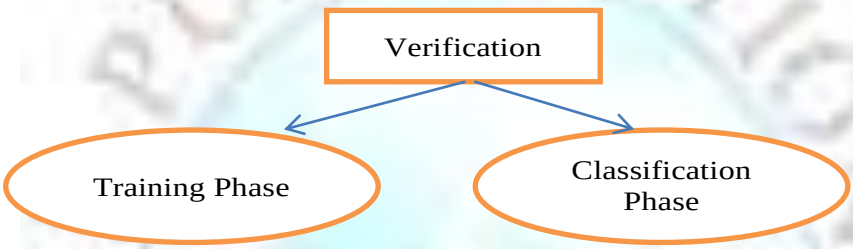

Figure 11 shows two steps of verification of signature.

Classification using Nearest Neighbors is a part of neural network training [7]. It is kind of supervised learning. The k-NN search technique and k-NN-based algorithms are widely used as benchmark learning rules — the relative simplicity of the k-NN search technique makes it easy to compare the results from other classification techniques to k-NN results. Different k-NN classifiers used different distance metrics to find the neighbour of query input out of which two most common used distance metrics are mentioned below:

$$d_2(a_1, a_2) = \sqrt{\sum_{j=1}^{k}(a_{1,j} - a_{2,j})^2} \quad \text{Euclidean or } L_2 \text{ distance} \quad \text{(viii)}$$

$$d_1(a_1, a_2) = \sqrt{\sum_{j=1}^{k}|a_{1,j} - a_{2,j}|} \quad \text{Manhatten or } L_1 \text{ distance} \quad \text{(ix)}$$

Where d (a1, a2) is the distance between any pair a1 = (a1, 1…a1, k) and a2 = (a2, 1…..a2, k).
The first thing you must do to apply k-NN is to find a measure of the distance between attributes in the data and then calculate it.[7] The mean-squared error (MSE) is a commonly used error function which tries to minimize the average error between the network's output and the target value.

- **Training Phase**

We choose four genuine signatures and four forged signatures for training purpose and two classes are created one for set of genuine trained signatures and designated as 'Genuine' while other class belongs to set of forged trained signatures and designated as 'Forged'.

- **Classification Phase**

Once the system is trained with given signatures for a user now it is ready to classified the coming test signature which is classified as genuine or forged based on the number of feature points of test signature neighbor with or close to either 'Genuine' or 'Forged' class if more feature points belong to 'Genuine' class then it is classified as genuine signature otherwise as forged signature.

4. RESULTS AND DISCUSSION

In this, data sets are being used for signature verification and result sets are being described based on data sets. We have constructed 3 data sets to evaluate the system performance, where 15 people contributed. Each person supplied 10 genuine signatures signed at different period of time.

Data set used to evaluate the performance of the system.

Table 1: Data Set For Evaluation

Data Set	No. of Signatures	Type
G1	90	Genuine
F1	45	Skilled
F2	45	Simple

The analysis of a signature scheme involves the evaluation of 2 parameters: FAR & FRR.

FAR – False Acceptance Ratio.
The false acceptance ratio is given by the number of fake signatures accepted by the system with respect to the total number of comparisons made.

FRR – False Rejection Ratio – The false rejection ratio is the total number of genuine signatures rejected by the system with respect to the total number of comparisons made.

Both FAR and FRR depend on the threshold variance parameter taken to decide the genuineness of an image [8][9]. If we choose a high threshold variance then the FRR is reduced, but at the same time the FAR also increases. If we choose a low threshold variance then the FAR is reduced, but at the same time the FRR also increases. ERR – Error rejection Rate – If the FAR of a system is same as the FRR then the system is said to be in an optimal state. In this condition, the FAR and FRR are also known as ERR.

Performance results of proposed system:

Table 2: Result Of Data Set

Data Set	FRR	FAR
F1	--	2.4%
F2	--	3.3%
G1	14%	--

G1- Genuine Signature

F1- Skilled Forgeries

F2- Simple Forgeries

Once the features are extracted, two neural networks, resilient back propagation and radial basis function, are used to classify the signatures.

Conclusion

The paper presents a new approach for offline signature verification system that uses neural network with MATLAB. The method uses feature extraction from pre-processed images of signature. The image is first converted to graysacle and then converted into binary image. The main feature extracted is COM of an image space to increase system efficiency. We have centralized the focus on center of mass of an image and divide our image into further sub images. The accuracy of any signature verification system highly depends upon three factors:

(i) Image preprocessing
(ii) Feature extraction and
(iii) Classifier used for classification

References

[1]. A.T. Ihler, J.W. Fisher, and A.S. Willsky,"Nonparametric estimators for online signature authentication", International Conference on Acoustics, Speech, and Signal Processing, May 07-11, 2001, Salt Lake City, UT , USA, Vol. 6,
[2]. Banshidhar Majhi, Y Santhosh Reddy, D Prasanna Babu, "Novel Features for Off-line SignatureVerification" International Journal of Computers, Communications & Control, Vol. I, No.2006
[3]. Alan McCabe, Jarrod Trevathan and Wayne Read, Neural Network-based Handwritten Signature Verification, Journal of computers,vol. 3, no. 8, August 2008.
[4]. Gonzalez, Woods, Eddins "Digital Image Processing Using Matlab" Pearson Education.
[5]. http://www.mathworks.com/help/tooolbox/pixelvaluesand statics/regionprops
[6]. A. Piyush Shanker, A.N. Rajagopalan, "Off-line signature verification using DTW", Pattern Recognition Letters 160 A. Karouni et al. / Procedia Computer Science 3 (2011) 155–161Ali Karouni / Procedia computer Science 00 (2010) 000–00028 (2007)
[7]. http://www.mathworks.in/help/stats/classification-using-nearest-neighbors.html
[8]. Rapid Off-line Signature Verification Based on Signature Envelope and Adaptive Density Partitioning by Vahid Malekian Department of Biomedical Engineering Amirkabir University of Technology Tehran, Iran.
[9]. Bradley Schafer, Serestina Viriry "An Offline Signature Verification system" IEEE International conference on signals and image processing application, 2009.
[10]. Banshidhar Majhi, Y Santhosh Reddy, D Prasanna Babu, "Novel Features for Off-line SignatureVerification" International Journal of Computers, Communications & Control, Vol. I, No.2006.