Egyptian Computer Science Journal, ECS, Vol. 37 No. 3, May 2013 ISSN-1110-2586

Simulation and Evaluation of Signature Recognition Techniques

Dina Darwish
International Academy for Engineering and Media Science – Egypt
dina.g.darwish@gmail.com

Abstract

The digital signature verification has become a widely important demand. Due to the increase of the use of online and offline digital signature in bank transactions and user authentication and other activities, methods of verification of these signatures have been diversified. The methods of verification include both online (or dynamic) and off-line (static) signature verification algorithms. This paper, investigates simulation and evaluation of signature recognition techniques. Several features are developed for signature recognition. A simulation technique is proposed for evaluation of signature recognition algorithms. Three signature recognition techniques are considered for evaluation based on the proposed simulation technique namely: radial basis network, nearest neighbor classifier, and k-nearest neighbor search using exhaustive search and the results of evaluation are discussed and analyzed.

Keywords: *neural networks, nearest neighbor classifier, signature verification, features extraction, signature recognition, simulation***.**

1. Introduction

A problem of personal verification and identification is an actively growing area of research. The methods are numerous, and are based on different personal characteristics. Voice, lip movements, hand geometry, face, iris, retina, and fingerprint are the most commonly used authentication methods. All of these psychological and behavioral characteristics are called biometrics. The biometrics is most commonly defined as measurable psychological or behavioral characteristic of the individual that can be used in personal identification and verification. [1],[2] The driving force of the progress in this field is, above all, the growing role of the Internet and electronic transfers in modern society. Therefore, considerable number of applications is concentrated in the area of electronic commerce and electronic banking systems. [3],[4]

The biometrics have a significant advantage over traditional authentication techniques (namely passwords, PIN numbers, smartcards etc.) due to the fact that biometric characteristics of the individual are not easily transferable, are unique of every person, and cannot be lost, stolen or broken. Another advantage of the use of signature recognition as an authentication method is that most of the modern portable computers and personal digital assistants (PDAs) use handwritten inputs, thus there is no need in invention of principally new devices for biometric information collection.

At the same time there are very few signature recognition solutions that can provide sufficiently high recognition rates at a reasonable level of efficiency. However, this area of research is vastly growing and has a promising future. Signature verification systems can generally be divided into two vast areas: static methods (or sometimes called off-line) that assume no time-relayed information, and dynamic (sometimes called on-line) with time-related information available in the form of p-dimensional function of time, where p represents the number of features of the signature. (see e.g. [5]-[8])

The task would be to extract some characteristics from the recorded information of the signing process and further compare them with the characteristics of the reference signature. The question that arises in this case is which kinds of characteristics should be recorded and extracted in order to identify the person in question in the most efficient and accurate way.

Some researchers as [1],[2],[9] suggested using the Vertical and Horizontal Projection Profiles (VPP and HPP) in the spatial domain and Discrete Cosine Transform in the transform domain to verify offline signatures. Other researchers [3,10] suggested using pen position, time, velocity, and pressure parameters for signature recognition.

A novel combination of the Modified Direction Feature (MDF) and additional distinguishing features such as the centroid, surface area, length and skew are used for classification. A Resilient Back propagation (RBP) neural network and a Radial Basis Function (RBF) network were compared in terms of verification accuracy. [6]

. In [8], an offline signature recognition and verification scheme is proposed based on extraction of several features including one hybrid set from the input signature and compare them with the already trained forms. Feature points are classified using statistical parameters like mean and variance. In [9] authors propose visualization technique based approach for digital signature authentication.

This paper, investigates simulation and evaluation of signature recognition techniques. Several features are developed for signature recognition. A simulation technique is proposed for evaluation of signature recognition algorithms. and the results of evaluation are discussed and analyzed.

In section 2, the features extraction to be used for signature recognition are introduced. In section 3 a proposed simulation technique is developed for evaluation of signature recognition techniques. were. In sections 4, evaluation of three signature recognition techniques are investigated . The simulation results are discussed and analyzed in section 5 , finally in section 6 conclusions are presented.

2. Features Extraction for Signature Recognition

The features needed to be extracted to identify signature are:

1 - The new curve of the signature after rotating the original curve of the signature points around the center x and y coordinates of the original signature curve based on making the original signature curve rotate around its center x and y coordinates, to make the new signature curve that will be used in pattern recognition.

2 - The number of pixels in the signature based on calculating the total number of pixels of the signature.

3 - The occupancy Ratio of the signature to the whole image which is described as :

Occupancy ratio = total number of pixels of the signature/total number of pixels of the signature image * 100

4 - The minimum Eigen value of the signature curve

Where the eigen values of a matrix A are obtained from the solution of the characteristic equation :

$$\det(A - \lambda I) = 0 \quad \text{.......................} \quad (1)$$

where det is the determinant of the matrix $(A - \lambda I)$ and I is the $n \times n$ identity matrix, λ is the eigen value

5 - The maximum height of the signature is based on the following:

Maximum height of the signature = maximum x coordinate of signature – minimum x coordinate of signature

6 - The maximum width of the signature is based on the following

Maximum width of the signature = maximum y coordinate of signature – minimum y coordinate of the signature

7 - The Euclidean distance between every two consecutive points in the signature curve

8 - The angle between every two consecutive points in the signature curve

9 - The height to width ratio of the signature

3. Proposed Simulation Technique

The proposed simulation technique is composed of 500 signatures pictures with the same size. After transforming the fixed-sizes signatures images to black and white colors only, we extract the black pixels from the signatures to catch the signature curve. Then, we calculate the number of pixels in the signature curve. Also, we have to calculate the occupancy ratio of the signature to the whole signature image, which is the amount occupied by the signature curve to the image in terms of number of pixels. Then, we calculate the minimum eigen value of the signature, the maximum height of the signature, the maximum width of the signature, the Euclidean distance between every two consecutive points in the signature curve, the angle between every two consecutive points in the signature curve and the height to width ratio of the signature. For the proposed simulation using 500 signatures for 100 persons, each person has 5 signatures. We used 60% of the signatures for training, and the other 40% were used for testing. Figure (1) shows the proposed simulation technique.

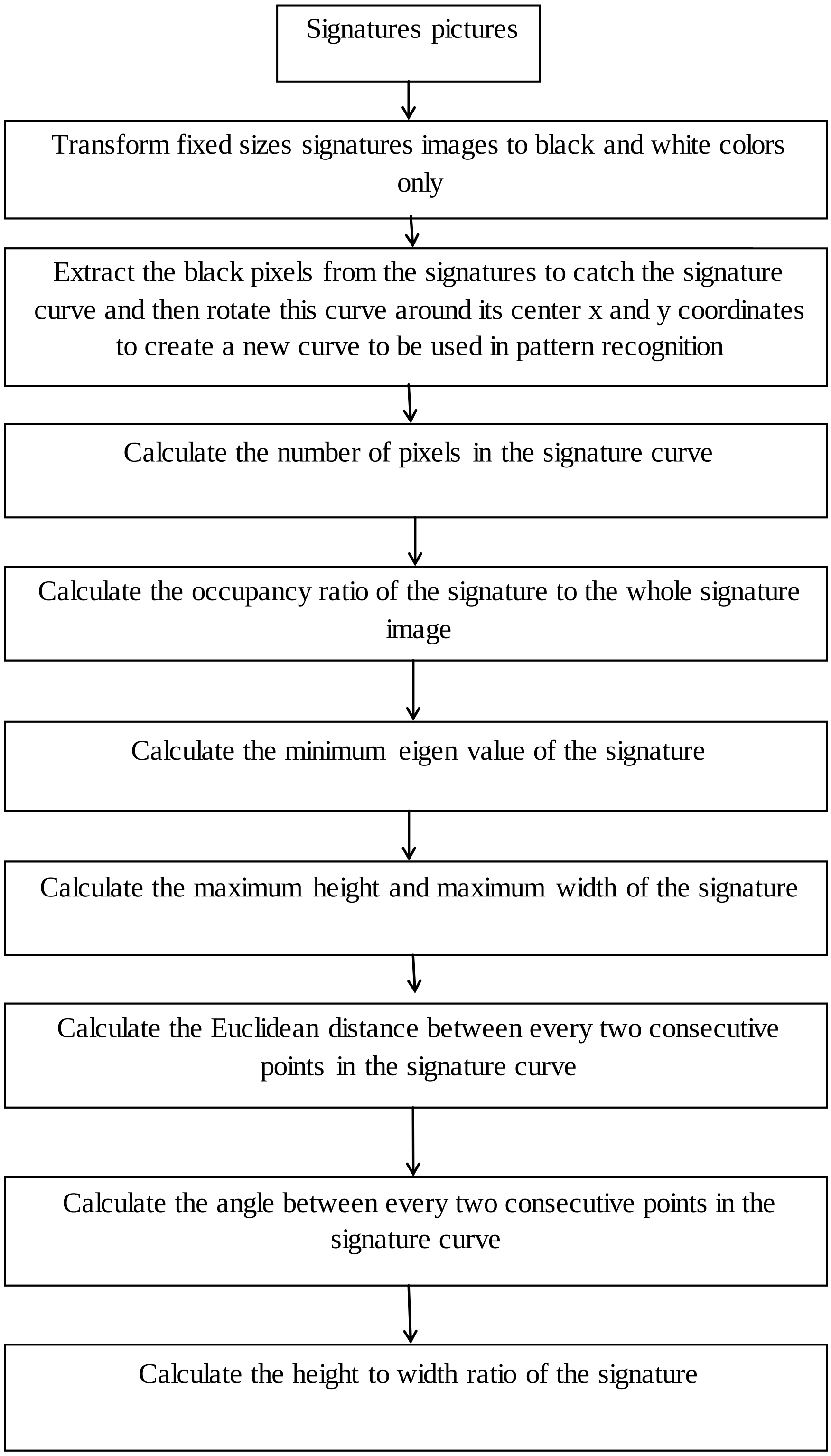

Fig.1 The proposed simulation technique

4. Evaluation of Signature Recognition Techniques

Three techniques for signature recognition are used for evaluation based on the proposed simulation technique described in section (3) , the techniques are :

1) Radial Basis Functions Networks
2) Nearest neighbor classifier
3) k- nearest neighbor search using exhaustive search

4.1 Radial Bases Functions Networks (RBF networks)

A radial basis function network [11,12] is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. They are used in function approximation, time series prediction, and control. Figure (2) shows the architecture of RBF network

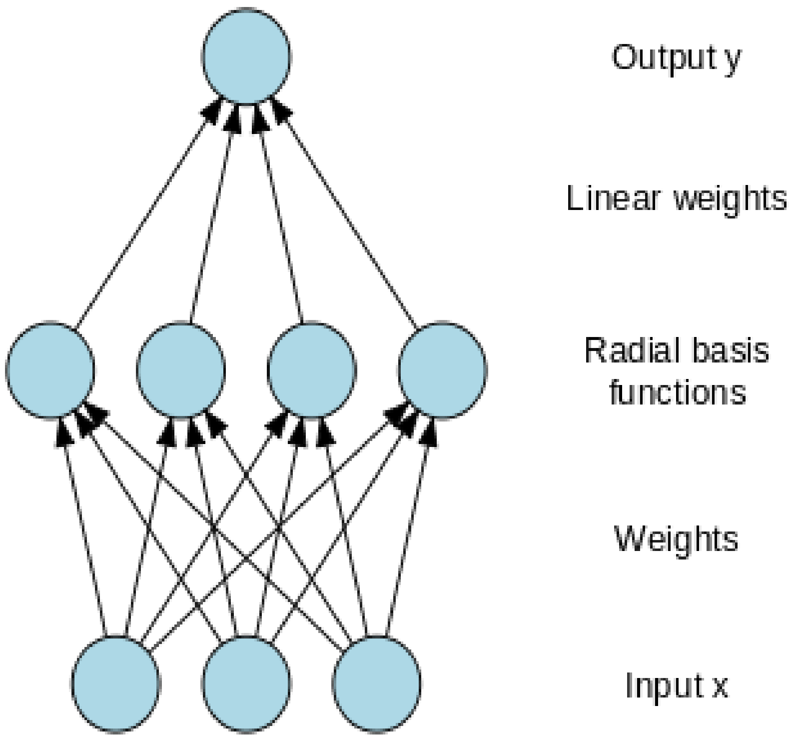

Fig. 2 Architecture of a radial basis function network

An input vector **x** is used as input to all radial basis functions, each with different parameters. The output of the network is a linear combination of the outputs from radial basis functions.

Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer.

The output, $\varphi : \mathbb{R}^n \to \mathbb{R}$, of the network is

$$\varphi(\mathbf{x}) = \sum_{i=1}^{N} a_i \rho(||\mathbf{x} - \mathbf{c}_i||) \qquad \text{……………………….(2)}$$

where N is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i, and a_i are the weights of the linear output neuron. In the basic form all inputs are connected to each hidden neuron. The norm is typically taken to be the Euclidean distance where

$$\rho\big(\left\|\mathbf{x} - \mathbf{c}_i\right\|\big) = \exp\left[-\beta \left\|\mathbf{x} - \mathbf{c}_i\right\|^2\right]. \qquad \text{…………………..(3)}$$

RBF networks are universal approximators on a compact subset of $\mathbb{R}^n$. This means that a RBF network with enough hidden neurons can approximate any continuous function with arbitrary precision.

The weights a_i, $\mathbf{c}_i$, and β are determined in a manner that optimizes the fit between φ and the data.

In a RBF network there are three types of parameters that need to be chosen to adapt the network for a particular task: the center vectors $\mathbf{c}_i$, the output weights w_i, and the RBF width parameters β_i. In the sequential training of the weights are updated at each time step as data streams in.

4.2 Nearest Neighbor Classifier

The nearest neighbor classifier [13] is based on categorizing query points based on their distance to points in a training dataset, it can be a simple yet effective way of classifying new points. We used various metrics to determine the distance as follows :

Given an mx-by-n data matrix X, which is treated as mx (1-by-n) row vectors $x_1, x_2, ..., x_{mx}$, and my-by-n data matrix Y, which is treated as my (1-by-n) row vectors $y_1, y_2, ..., y_{my}$, the various distances between the vector x_s and y_t are defined as follows:

- Euclidean distance

$$d_{st}^2 = (x_s - y_t)(x_s - y_t)' \qquad \text{……………………………(4)}$$

The Euclidean distance is a special case of the Minkowski metric, where p = 2.

- Standardized Euclidean distance

$$d_{st}^2 = (x_s - y_t)V^{-1}(x_s - y_t)' \qquad \text{……………………….. (5)}$$

where V is the *n*-by-*n* diagonal matrix whose *j*th diagonal element is $S(j)^2$, where S is the vector containing the inverse weights.

- Mahalanobis distance

$$d_{st}^2 = (x_s - y_t)C^{-1}(x_s - y_t)' \qquad \text{.................................(6)}$$

where C is the covariance matrix.

- Correlation distance

$$d_{st} = 1 - \frac{(x_s - \bar{x}_s)(y_t - \bar{y}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(y_t - \bar{y}_t)(y_t - \bar{y}_t)'}} \qquad \text{...................... (7)}$$

where

$$\bar{x}_s = \frac{1}{n}\sum_j x_{sj}$$

and

$$\bar{y}_t = \frac{1}{n}\sum_j y_{tj}$$

- Spearman distance

$$d_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'}\sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \qquad \text{...........................(8)}$$

where

- r_{sj} is the rank of x_{sj} taken over $x_{1j}, x_{2j}, \ldots x_{mx,j}$
- r_{tj} is the rank of y_{tj} taken over $y_{1j}, y_{2j}, \ldots y_{my,j}$
- r_s and r_t are the coordinate-wise rank vectors of x_s and y_t, i.e., $r_s = (r_{s1}, r_{s2}, \ldots r_{sn})$ and $r_t = (r_{t1}, r_{t2}, \ldots r_{tn})$.

$$\bar{r}_s = \frac{1}{n}\sum_j r_{sj} = \frac{(n+1)}{2}.$$

$$\bar{r}_t = \frac{1}{n}\sum_j r_{tj} = \frac{(n+1)}{2}.$$

4.3 k-Nearest Neighbor Search using Exhaustive Search

In pattern recognition, the *k*-nearest neighbor algorithm (*k*-NN) [14] is a method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. (A common weighting scheme is to give each neighbor a weight of $1/d$, where d is the distance to the neighbor. This scheme is a generalization of linear interpolation.)

The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data.

Nearest neighbor rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity.

When the input data meets any of the following criteria, k-Nearest Neighbor (kNN) search uses the exhaustive search method by default to find the k-nearest neighbors:

- The number of columns of X is more than 10.
- X is sparse.
- The distance measure is either:
    - 'Euclidean'
    - 'Mahalanobis'
    - 'correlation'
    - 'spearman'
    - A custom distance function

*k*NN search also uses the exhaustive search method if your search object is an exhaustive searcher object. The exhaustive search method finds the distance from each query point to every point in X, ranks them in ascending order, and returns the *k* points with the smallest distances.

5. Simulation Results

We used 500 patterns for simulation, representing 500 signatures for 100 persons, each person having 5 signatures. We used 60% of the patterns for training and the rest 40% for testing. We tested all the 500 patterns, which are all vectors of the same size. Each vector representing a signature. The applying the proposed simulation technique developed in section (3) ,the results we obtained as shown in table (1).

Table (1) Simulation Results

Signature Recognition Techniques	Percentage of correctly classified signatures
Radial basis networks	60.8%
Nearest neighbor classifier	77.4%
k-nearest neighbor search using exhaustive search	73.6%

From the table 1, we noticed that the percentage of correctly classified signatures using the radial basis networks is 60.8%, which represents the lowest percentage of correctly classified patterns among the three simulation environments. Using the nearest neighbor classifier, 77.4% of the signatures were correctly classified, which represents the highest percentage of correctly classified patterns among the three simulation environments. Using the *k*-nearest neighbor search using exhaustive search, the percentage of the correctly classified patterns was 73.6%, which is closest to the percentage of the nearest neighbor classifier.

6. Conclusions

A simulation technique is proposed for evaluating signature recognition techniques. Three techniques are considered for evaluation namely: radial basis network, nearest neighbor classifier, and k-nearest neighbor search using exhaustive search. From the simulations, as expected, the nearest neighbor classifier showed the best performance with signature recognition ratio of 77.4%, then, the k-nearest neighbor search using exhaustive search follows it with a signature recognition ratio of 73.6%, then, comes the radial basis functions networks with a signature recognition ratio of 60.8%, which is the least recognition ratio among the three signature recognition techniques.

References

[1] S. Kumar, K. Braja, R. Chihotaray and S. Pattanaik ;, " Off-line Signature Verification Based on Fusion of Grid and Global Features using Neural Networkks " , International Journal of Engineering Science and Technology , Vol.2 No. 12 ,2010 pp. 7035-7044

[2] A. Talwai, B. Sasikanth and De. Sengupta ," Offline Signature Verification", a report submitted on completion of Project Work, , Indian Institute ofTechnology April 2009.

[3] M.Mailahi and L. Hans, " Biometric Signature Verification using Pen Position, Time Velocity and Pressure Parameters ", *Journal of Technology*, vol.48(A) , Jun 2008: pp. 35 - 54

[4] C.Oz ,F. Erealand Z. Demir , " Signature Recognition and Verification with ANN "Proceeding of the third international conference on Electrical and Electroincs Engineering EMO – Odasi , 2003

[5] S. Armand, Mi. Blumenstein and V. Muthukkumarasamy, " Off-line Signature Verification using the Enhanced Modified Direction Feature and Neural-based Classification ", International joint conference on Neural Networks , IEEE 2006

[6] S. Patil and S. Dewangan " Neural Network-based Offline Handwritten Signature Verification System using Hu's Moment Invariant Analysis ", International Journal of Engineering and Advanced Technology (IJEAT), Volume-1, Issue-1, October 2011.

[7] A.Bansal , B.Gupta , G Khandelwal, and S. Chakraverty , " Offline Signature Verification Using Critical Region Matching " ,International Journal of Signal Processing, Image Processing and Pattern, Vol. 2, No.1, March, 2009.

[8] M. Bhuyan, K. Sarma and H. Das , " Signature Recognition and Verification using Hybrid Features and Clustered Artificial Neural Network" , International Journal of Electrical and Computer Engineering Vol. 5, No 2 , June 2010.

[9] A Visualization Based Approach for Digital Signature Authentication, Songhua Xu, Wenxia Yang, Francis C.M. Lau, IEEE-VGTC Symposium on Visualization H.-C. Hege, I. Hotz, and T. Munzner (Guest Editors), *Vol. 28 No.3*, 2009

[10] S. Armand , M. Blumenstein and V. Muthukkumarasamy ," Off-line Signature Verification based on the Modified Direction Feature " , Proceedings of 18th international conference on pattern recognition , IEEE computer society - 2006

[11] http://en.wikipedia.org/wiki/Radial_basis_function

[12] http://en.wikipedia.org/wiki/Radial_basis_function_network

[13] http://www.mathworks.com/help/stats/classification-using-nearest- neighbors.html

[14] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm