Signal Processing: An International Journal (SPIJ)
Singaporean Journal of Scientific Research(SJSR)
Vol6.No.3 2014Pp.163-171
Available at:www.iaaet.org/sjsr
Paper Received:17-02-2014
Paper Accepted:10-03-2014
Paper Reviewed by:1Prof.Dr.Ponnambalam2. Chai Cheng Yue
Editor:Dr. BinodKumar

OFFLINE SIGNATURE RECOGNITION & VERIFICATION USING SVM

S.JAYASUDHA
Asst. Professor
Dept. of ECE, MRK Institute of Technology
Nattarmangalam, Tamilnadu
jaya17sudha@gmail.com

ABSTRACT

The signature of a person is an important biometric attribute of a human being which can be used to authenticate human identity. However human signatures can be handled as an image and recognized using computer vision and neural network techniques. With modern computers, there is need to develop fast algorithms for signature recognition. There are various approaches to signature recognition with a lot of scope of research. In this paper, off-line signature recognition & verification using neural network is proposed, where the signature is captured and presented to the user in an image format. Signatures are verified based on parameters extracted from the signature using various image processing techniques. The Off-line Signature Recognition and Verification is implemented using C# (C-sharp). This work has been tested and found suitable for its purpose. **General Terms** Security

Keywords Signature, Biometric, Neural Networks, Off-line Signature Recognition and Verification

1. INTRODUCTION

In our society, traditional and accepted means for a person to identify and authenticate himself either to another human being or to a computer system is based on one or more of these three (3) general principles:

- What the person knows
- What he possesses or
- What he is

The written signature is regarded as the primary means of identifying the signer of a written document based on the implicit assumption that a person's normal signature changes slowly and is very difficult to erase, alter or forge without detection. The handwritten signature is one of the ways to authorize transactions and authenticate the human identity compared with other electronic identification methods such as fingerprints scanning and retinal vascular pattern screening. It is easier for people to migrate from using the popular pen-and-paper signature to one where the handwritten signature is captured and verified electronically. The signature of a person is an important biometric attribute of a human being and is used for authorization purpose. Various approaches are possible for signature recognition with a lot of scope of research. Here, we deal with an off-line signature recognition technique. Signatures are composed of special characters and flourishes and therefore most of the time they can be unreadable. Also intrapersonal variations and interpersonal differences make it necessary to analyze them as complete images and not as letters and words put together [6]. Signature recognition is the process of verifying the writer's identity by checking the signature against samples kept in the database. The result of this process is usually

between 0 and 1 which represents a fit ratio (1 for match and 0 for mismatch). Signature recognition is used most often to describe the ability of a computer to translate human writing into text. This may take place in one of two ways either by scanning of written text (off-line method) or by writing directly on to a peripheral input device. The first of these recognition techniques, known as Optical Character Recognition (OCR) is the most successful in the main stream. Most scanning suites offer some form of OCR, allowing user to scan handwritten documents and have them translated into basic text documents. OCR is also used by some archivist as a method of converting massive quantities of handwritten historical documents into searchable, easily-accessible digital forms.

2. OVERVIEW OF SIGNATURE RECOGNITION

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 movement, hand geometry, face, odor, gait, iris, retina and fingerprint are the most commonly used authentication methods. All these psychological and behavioral characteristics are called biometrics. 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 [9]. The biometrics have a significant advantage over traditional authentication techniques (namely passwords, PIN numbers, smart cards 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.

The choice of one of the biometric solutions depends on several factors which include [3]:

- User acceptance o Level of security required
- Accuracy
- Cost and implementation time

The method of signature verification reviewed in this paper benefits the advantage of being highly accepted by potential customers. The use of the signature has a long history which goes back to the appearance of writing itself [9]. Utilization of the signature as an authentication method has already become a tradition in the western civilization and is respected among the others. The signature is an accepted proof of identity of the person in a transaction taken on his or her behalf.

Thus the users are more likely to approve this kind of computerized authentication method [10]. Signature verification systems differ in both their feature selection and their decision methodologies. More than 40 different feature types have been used for signature verification [8]. Features can be classified into two major types: local and global [4]. Global features are features related to the signature as a whole, for instance the average signing speed, the signature bounding box and Fourier descriptors of the signatures trajectory. Local features correspond to a specific sample point along the trajectory of the signature. Examples of local features include distance and curvature change between successive points on the signature trajectory [4]. Most commonly used online signatures acquisition devices are pressure sensitive tablets capable of measuring forces exerted at the pen-tip, in addition to the coordinate of the pen. The pressure information at each point along the signature trajectory is another example of commonly used local feature. Some of these features are compared in order to find the more robust ones for signature verification purposes. Other systems have used genetic algorithms to find the most useful features. Due to the high sampling rate of the tablet, some consecutive sample points may mark the same trajectory point especially when the pen movement is slow. Most verification systems resample the input so as to obtain a trajectory consisting of equidistant points. This is often done in order to remove redundant points to speed up the comparisons and to obtain a shape-based representation, removing the time dependencies, separately keep track of the local velocity values and use them in aligning two signatures. Signature recognition and verification involves two separate but strongly related tasks: one of them is identification of the signature owner, and the other is the decision about whether the signature is genuine or forged. Also, depending on the need, signature recognition and verification problem is put into two major classes: (i) On-line signature recognition and verification systems (SRVS) and (ii) Off-line SRVS. On-line SRVS requires some special peripheral units for measuring hand speed and pressure on the human hand when it creates the signature. On the other hand, almost all Off-line SRVS systems rely on image processing and feature extraction techniques [1].

2.1 Types of Signature Verification

Based on the definitions of signature, it can lead to two different approaches of signature verification.

2.1.1 Off-Line or Static Signature Verification Technique

This approach is based on static characteristics of the signature which are invariant [6]. In this sense signature verification, becomes a typical pattern recognition task knowing that variations in signature pattern are inevitable; the task of signature authentication can be narrowed to drawing the threshold of the range of genuine variation. In the offline signature verification techniques, images of the signatures written on a paper are obtained using a scanner or a camera.

2.1.2 On-line or Dynamic Signature Verification Technique

This is the second type of signature verification technique. This approach is based on dynamic characteristics of the process of signing. This verification uses signatures that are captured by pressure sensitive tablets that extract dynamic properties of a signature in addition to its shape. Dynamic features include the number of order of the strokes, the overall speed of the signature and the pen pressure at each point that make the signature more unique and more difficult to forge. Application areas of Online Signature Verification include protection of small personal devices (e.g. PDA, laptop), authorization of computer users for accessing sensitive data or programs and authentication of individuals for access to physical devices or buildings [4].

2.2 Nature of Human Signature

It is supposed that the features of the process of signing originate from the intrinsic properties of human neuromuscular system which produces the aforementioned rapid movements. Knowing that this system is constituted by a very large number of neurons and muscle, fibers is possible to declare based on the central limit theorem that a rapid and habitual movement velocity profile tends toward a delta-log normal equation [10]. This statement explains stability of the characteristics of the signature. Thus, the signature can be treated as an output of a system obscured in a certain time interval necessary to make the signature. This system models the person making the signature [7].

2.3 Types of Forgeries

The main task of any signature verification system is to detect whether the signature is genuine or counterfeit. Forgery is a crime that aims at deceiving people. Since actual forgeries are difficult to obtain, the instrument and the results of the verification depend on the type of the forgery [9]. Basically there are three types that have been defined:

Random forgery: this can normally be represented by a signature sample that belongs to a different writer i.e. the forger has no information whatsoever about the signature style and the name of the person.

Simple forgery: this is a signature with the same shape or the genuine writer's name. **Skilled forgery**: this is signed by a person who has had access to a genuine signature for practice [4].

3. OVERVIEW OF NEURAL NETWORK

In this work there is a challenge of creating a system with the ability to recognize hand written signature and verify its authenticity. 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.

How the Human Brain Works

The workings of the neuron of the human brain remain incomplete till date. But from what we understand so far about the human brain a system with similar properties can be developed. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes [5].

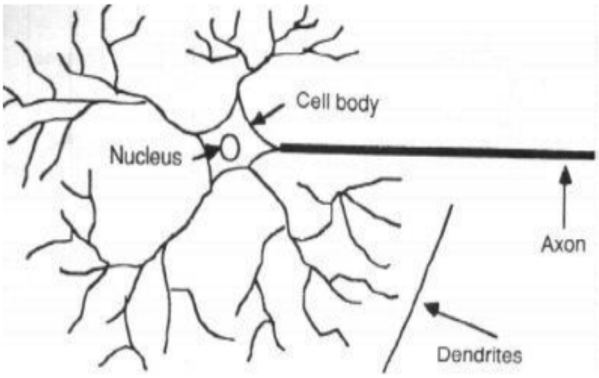

Figure 1: Brain Neuron

3.1 Artificial Neuron

An artificial neuron is designed to simulate the brain neuron under limiting conditions such as a smaller processing power than the human brain, can also be optimized to have similar properties which are

1. Training – Which utilizes several inputs to create an output
2. Utilization – Which uses an already created output to solve a problem

For each of these stages, a general rule governs the operation. This rule can be called the "firing rule" [5]. The rule accounts for the high flexibility of a neural network. It simply defines how a neuron should optimize for an input, irrespective of whether it has been previously trained for that input or not. A firing rule based on the Hamming distance principle is stated below.

Take a collection of training patterns for a node, some of which cause it to fire (the 1-taught set of patterns) and others which prevent it from doing so (the 0-taught set). Then the patterns not in the collection cause the node to fire if, on comparison, they have more input elements in common with the 'nearest' pattern in the 1-taught set than with the 'nearest' pattern in the 0-taught set.

If there is a tie, then the pattern remains in the undefined state. For example, a 3-input neuron is trained to output 1 when the input (X1, X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001. Then, before applying the firing rule, the truth table is;

Table 1 (NEURAL NETWORKS by Christos Stergiou and DimitriosSiganos)

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0/1	0/1	0/1	1	0/1	1

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000 which belongs in the 0- taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 are equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1). By applying the firing in every column the following truth table is obtained;

Table 2 (NEURAL NETWORKS by Christos Stergiou and DimitriosSiganos)

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0	0/1	0/1	1	1	1

The difference between the two truth tables is called the *generalization of the neuron.* Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training [5].

3.2 Back Propagation Artificial Neural Network

There are several algorithms that can be used to create an artificial neural network, but the Back propagation was chosen because it is probably the easiest to implement, while preserving efficiency of the network. Backward Propagation Artificial Neural Network (ANN) use more than one input layers (usually 3). Each of these layers must be either of the following:

- Input Layer – This layer holds the input for the network Output Layer – This layer holds the output data, usually an identifier for the input.
- Hidden Layer – This layer comes between the input layer and the output layer. They serve as a propagation point for sending data from the previous layer to the next layer.

A typical Back Propagation ANN is as depicted below. The black nodes (on the extreme left) are the initial inputs. Training such a network involves two phases. In the first phase, the inputs are propagated forward to compute the outputs for each output node. Then, each of these outputs are 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 is adder

output signal and = is output signal of nonlinear element. Signal *y* is also output signal of neuron.

To teach the neural network, we need data set. The training data set consists of input signals 1 2 assigned with corresponding target (desired output). The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set. Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals values for each neuron in each network layer.

3.3 Propagation of Signals through the hidden layer

Symbols represent weights of connections between output of neuron and input of neuron in the next layer. In the next algorithm step, the output signal of the network is compared with the desired output value (the target), which is found in training data set. The difference is called error signal . of output layer neuron. It is impossible to compute error signal for internal neurons directly, because output values of these neurons are unknown. For many years the effective method for training multilayer networks has been unknown. Only in the middle eighties the back propagation algorithm has been worked out.

The idea is to propagate error signal (computed in single teaching step) back to all neurons, which output signals were input for discussed neuron. The weights' coefficient used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed (signals are propagated from output to inputs one after the other). This technique is used for all network layers. If propagated errors came from few neurons they are added. The illustration is below:

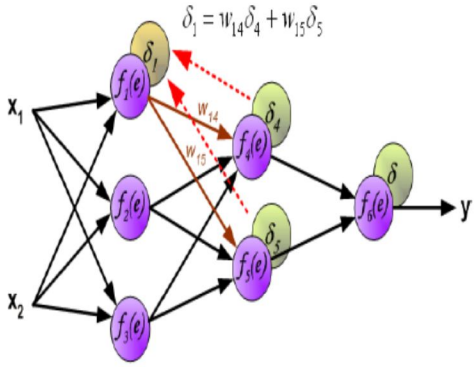

Figure 2: A 3-layer neural network using back propagation with 2 inputs & 1 output

When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In the formulas below represents derivative of neuron activation function (which weights are modified). Coefficient affects network teaching speed. There are a few techniques to select this parameter. The first method is to start teaching process with large value of the parameter. While weights coefficients are being established the parameter is being decreased gradually. The second, more complicated, method starts teaching with small parameter value. During the teaching process the parameter is being increased when the teaching is advanced and then decreased again in the final stage. Starting teaching process with low parameter value enables to determine weights coefficients signs. [2]

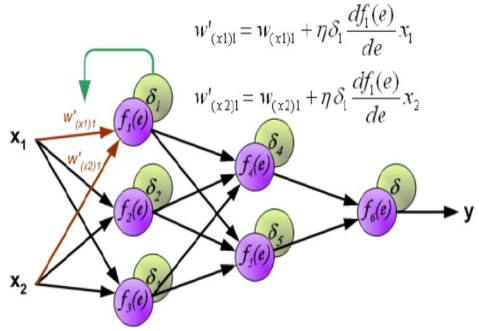

Figure 3: A 3-layer neural network using back propagation with modified input node

4. PROPOSED DESIGN

The system we introduce is in two major parts:

i. Training signatures
ii. Recognition and verification of given signature

The block diagram of the system is given in figure 4.

I. Preprocessing: The preprocessing step is applied in both the training and testing phases. Signatures are scanned in gray. In this phase signatures are made standard and ready for feature extraction. The preprocessing stage includes 3 steps: (a) background elimination, (b) Width Normalization and (c) thinning.

a) Background Elimination: Many image processing applications require differentiation of objects from the image background. Thresholding is the most trivial and easily applicable method for this purpose. It is widely used in image segmentation. We used threshold technique for differentiating the signature pixels from the background pixels. Clearly in this application, we are interested in dark objects on a light background and therefore a threshold value called the brightness threshold is appropriately chosen and applied to image pixels. After the thresholding, the pixels of the

signature would be 1 and the other pixels which belong to the background would be 0.

b) Width Normalization: Irregularities in the image scanning and capturing process may cause signature dimensions to vary. Furthermore, height and width of signatures vary from person to person and sometimes even the same person may use different size signatures. First there is the need to eliminate the size differences and obtain a standard signature size for all signatures. During the normalization process, the aspect ratio between the width and height of a signature is kept intact and after the process, all the signatures will have the same dimension.

c) Thinning: The goal of thinning is to eliminate the thickness differences of pen by making the image one pixel thick.

II. Feature Extraction: Extracted features in this phase are the inputs of the training phase. The features in this system are global features, mask features and grid features.

a) Global Features: provides information about specific cases of the signature shape like signature area, signature height-to-width ratio, maximum horizontal histogram and maximum vertical histogram, horizontal and vertical centre of the signature, local maxima numbers of the signatures and edge point of the signature.

b) Mask Features: provides information about directions of the lines of the signature because the angles of signature have inter-personal differences.

c) Grid Features: Provide over-all signature appearance information. An input into Signature Recognition greatly affects the accuracy level of training and the overall performance of the application. As a result, Signature Recognition accepts predominantly 2 types of input:

- User Bio-data
- 8 Signature images

USER Bio-Data- Since Signature Recognition is designed to be an application to be used in a banking environment, it accepts the name and account number of the bank user, through its interface, which will be used for identifying his/account during training and verification.

8 Signature images- For a higher level of accuracy, Signature Recognition accept 8 inputs for training. This is based on the mathematical mean theory, i.e. when we have 8 different signatures, we can create a marginal image which will be the basis of signature verification. Each of these images is however captured using an image scanner. Also during the training process, the application requests for the account number (ID) of the bank user, which is entered through the keyboard in the interface of the application.

4.1 System Pseudo code

When the application launches, it waits for the user to determine whether he wishes to train or verify a set of signatures. At the training stage, based on the back propagation neural network algorithm, the user gives eight (8) different images as input, of which the real input to the network are the individual pixels of the images. When input is confirmed and accepted, it passes through the back propagation neural network algorithm to generate an output which contains the network data of the trained images. The back propagation artificial neural network simply calculates the gradient of error of the network regarding the networks modifiable weights. In this paper we implement a multi-layer neural network, specifically a three-layer neural network, consisting of:

- input layer – which may have as many input nodes as possible,
- middle layer
- Output layer – which is usually just a single binary file containing the results of the training. Display "Splash screen" while program stability is checked.

Display "Home Screen" giving the users options of what to do.

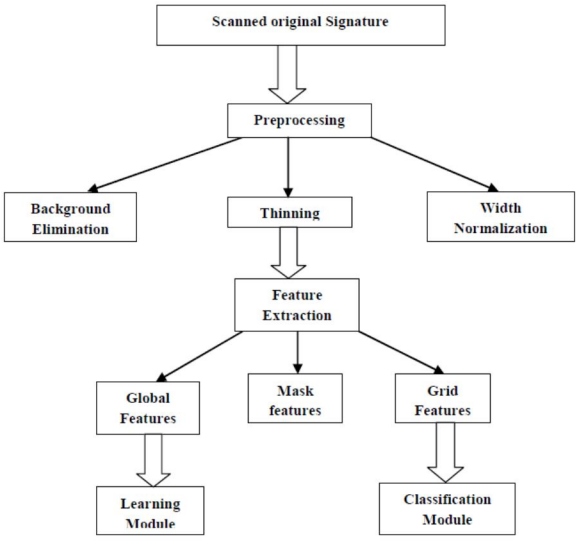

Figure 4: Block Diagram of Proposed system

The system pseudo code is as given below:

Signature training process

Accept ID from standard input
Accept Name from standard input
Accept a sequence of 8 images from the scanner Start training process on confirmation

DO Training process

Create Pre-input array of pattern from input images. Create input layer from pre-input. Begin forward propagation Create inputs and errors for the input layer. Create errors and outputs for the hidden using inputs from the input layer. Create target outputs and errors for the output layer using inputs from the hidden layer. Begin Back propagation Correct errors for the input layer using results from the forward propagation of the input layer. Correct errors for the hidden layer using the results from the forward propagation of the hidden layer. Correct errors for the output layer using the results from the forward propagation of the output layer.

Until all image pixels have been exhausted

Save network training data to filesystem. Accept Account holder ID from standard input. Accept Signature to be verified from scanner. Start verification on confirmation

Verification Process

Begin only back propagation training using the single input image. Compare the output of new input with existing network data. Return verification result in percentage.

Application Settings

Change maximum error level. Make BackUp. Accept backup location. Save application data to backup location.

View Help

Load help file from file systém. Display help file using integrated viewer.

About

Display short application description.

5. RESULTS

When verification begins, the application updates the user of the current state of events. For instance, at the first stage, settings are initialized, indicated by "Initializing settings..." and "initializing settings...Done", when completed. At the second stage, the training set for the inputs is generated, indicated by the output "Generating training set..." and "Generating training set...Done", when completed. At third stage, when training on the images begins, the program notifies with "Began training process..." and when done, the final notification states "Completed training process successfully." After the entire process of training, a file is generated and stored in the files system. This file contains the network details of the training process in binary.

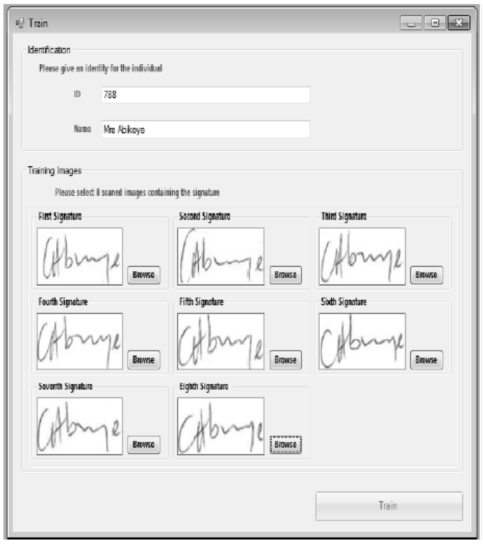

Figure 5: Eight (8) different scanned signatures with each possessing about 85% dominance

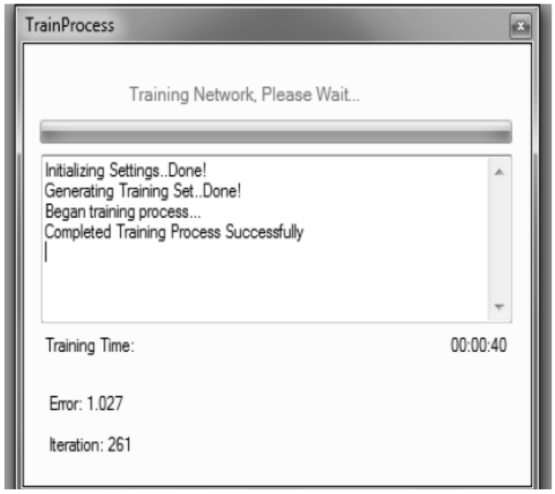

Figure 6: Training of the 8 scanned signatures completed

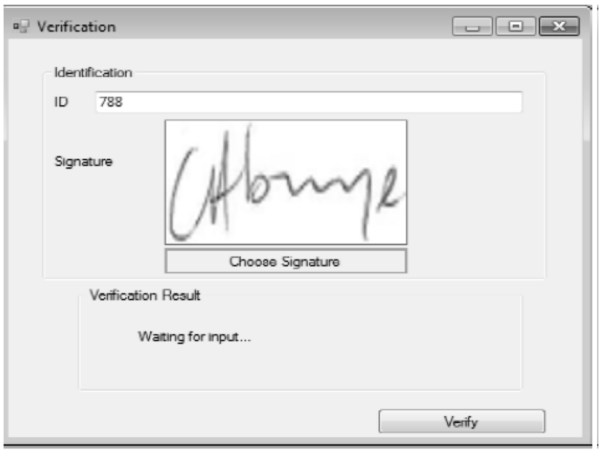

Figure 7a (i): Verification process of the signed scanned signature against the trained neural network

Figure7a(ii): Signature accepted

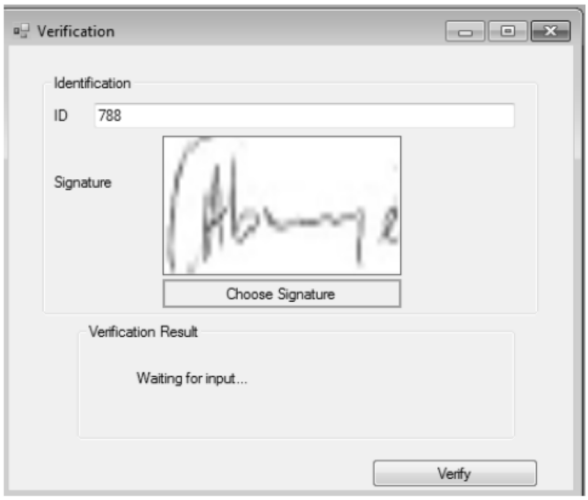

Figure7b(i): Verification process of the signed scanned signature

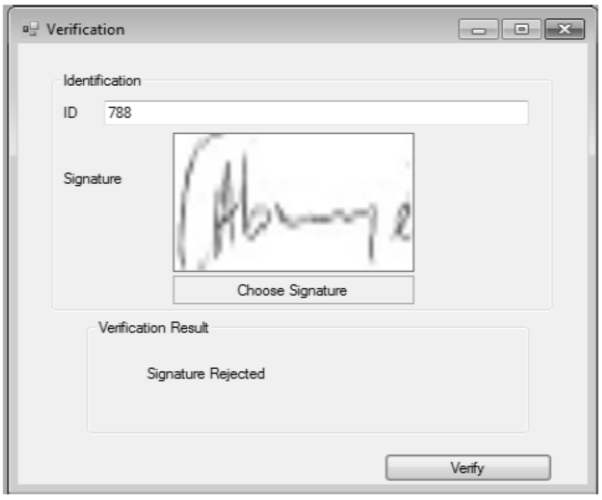

Figure 7b (ii): Signature rejected

6. DISCUSSION AND CONCLUSION

In this study, we presented Off-Line Signature Recognition and Verification System which is based on image processing, moment invariants, some global properties and neural networks. Both systems used a three-step process; in the first step, the signature is separated from its image background. Second step performs normalization and digitization of the original signature. Moment invariants and some global properties which are used as input features for the neural network (NN) are obtained in the third step. Our recognition system exhibited 100% success rate by identifying correctly all the signatures that it was trained for. However, it exhibited poor performance when it was presented with signatures that it was not trained for earlier. We did not consider this a "high risk" case because recognition step is always followed by verification step and these kinds of false positives can be easily caught by the verification system. Generally the failure to recognize/verify a signature was due to poor image quality and high similarity between 2 signatures. Recognition and verification ability of the system can be increased by using additional features in the input data set. This study aims to reduce to a minimum the cases of forgery in business transactions.

REFERENCES

[1]. OZ, C. Ercal, F. and Demir, Z. Signature Recognition and Verification with ANN.

[2]. Golda, A. 2005. Principles of Training multi-layer neural network using back propagation.

[3]. Jain, A., Bolle, R., and Pankarti. 1999. Biometrics: Personal Identification in Networked Society. The Springer International series in Engineering & Computer Science. vol. 479.

[4]. Aykanat C. et. al ,(Eds). 2004. Proceedings of the 19th International Symposium on Computer and

Information Sciences, ISCIS 2004. Springer-Verlag Berlin Heidelberg New York. pp. 373-380.

[5]. Stergiou, C. and Siganos, D. 2003. Neural Networks Retrieve April 1, 2011 from www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs #/report. html

[6]. Ozgunduz, E., Karsligil, E., and Senturk, T. 2005.Off-line Signature Verification and Recognition by Support Vector Machine. Paper presented at the European Signal processing Conference.

[7]. Pacut, A. and Czaja, A. 2001. Recognition of Human Signatures. Neural Network, in proceedings of the International Conference on Neural Network, IJCNN'01, vol.2, pp 1560-1564.

[8]. Jain, A., Griess, F., and Connel1, S. "Online Signature Recognition", Pattern Recognition, vol.35,2002, pp 2963- 2972.

[9]. Kalenova, D. 2005. Personal Authentication using Signature Recognition.

[10]. Plamondon, R.1995. The Handwritten Signature as a Biometric Identifier: Psychophysical Model & System Design. IEE Conference Publications, Issue CP408, 23-27

[11]. Sonsone and Vento. "Signature Verification: Increasing Performance by Multi-Stage System", Pattern Analysis & Application, vol.3, no. 2, 2000, pp.169-181

[12]. Velez, J.F., Sanchez, A. and Moreno, A.B. 2003. Robust Off-Line Signature Verification using Compression Networks and Position Cuttings.