A Hybrid ANN-Based Technique for Signature Verification

ASHRAF A. ZAHER* and ABDULNASSER ABU-REZQ
Physics Department – Science College
Kuwait University
P. O. Box 5969 – Safat 13060
Kuwait
*ashraf.zaher@ku.edu.kw  http://physics.kuniv.edu.kw

Abstract: - This paper introduces a real-time system for verifying handwritten signatures that relies on a hybrid methodology, for which consistency checking is performed prior to enrolling signatures for further processing. Only the best six signatures are retained out of 10 signatures for each signer, during the enrollment phase, based on the deviation in both the total signing time and the binary pattern of the pen movement. The proposed system consists of three consecutive phases, where the first one is an online approach that is quite similar to the enrollment phase, and acts as an initial bottleneck for the verification process so that simple forgeries are quickly filtered out. The second phase uses a combination of neural networks and linear predictive coding to construct a majority voting committee in a pattern recognition context to decide on the authenticity of the signatures that passed the first phase of the verification process. The third phase is an offline technique that processes the real-time data features after converting them into stationary image frames. A digitizing tablet was used to collect eight features during the implementation of the proposed system resulting in a 2.9% for the false acceptance rate and 8.8% for the false rejection rate.

Key-Words: - Signature Verification, Pattern Recognition, Hybrid Systems, Neural Networks.

1 Introduction

Automatic signature verification has been the subject of considerable research in the recent years. This was perhaps motivated by the lack of security for password-based identification techniques that can sometimes be lost, stolen, cracked, or even forgotten. Handwritten signatures are unique, subconscious, and extremely individualistic, representing a permanent part of a person's physiology. Over the course of time, people tend to master handwritten signatures that reflect their unique individual styles, size and dexterity of their fingers, the way they hold the writing instrument, their postures, etc. This makes signature verification an ideal tool for applications requiring moderately high level of security.

Signature forgeries are usually classified into three main categories: random, simple and skilled. Random, or zero-effort, forgery simply corresponds to an invalid signature that the forger tries to improve without any knowledge of the original person's name or style of writing. Simple forgeries that represent the majority of forgery cases involve using the original name of the true identity, but without knowing the style in which it should be produced. Skilled forgeries, which are the most difficult to detect, even by experts, look almost as exact as their original counterparts as the forger is aware of both the name and style of the original signature and tries to mimic it as best as he can.

Approaches to signature verification fall into two categories according to the method of acquisition of the data: online and offline. Reviews of the literature on signature verification can be found in [1] and [2], while a comprehensive survey of online and offline handwriting recognition can be found in [3]. Online systems record the time profiles of the motion of the writing instrument while the signature is produced, and includes position, velocity, acceleration, and possibly pressure and tilt angle. On the other hand, offline systems use a static image of the signature. The offline case is more difficult because information about pen movement is not available.

There exist two different tasks of signature verification and signature recognition. The former is easier, requiring only to decide whether a given signature does belong to the claimed signer or not. On the other hand, signature recognition systems need to decide to which writer(s) a given signature belongs. Sometimes the two systems overlap via sharing the same methodology, and given a small database, they can even be used for the same purpose.

Two types of errors are generally used to evaluate the effectiveness and robustness of a given signature verification; type I error, which is a measure of the percentage of genuine signatures that are being falsely rejected by the system, and type II error, which is a measure of the percentage of the forged signatures that are being falsely accepted by
the system. These two errors are abbreviated as FRR and FAR respectively. The point at which both errors have the same value is called equal error rate, EER, and is usually used for evaluating the performance of the signature verification system.

2 Literature Review

Research done in the field of signature verification can be categorized based on whether the approach is shape-based or time-based or a combination of both. When considering the signature features, there is always the choice between using local and global features. In addition, based on the methodology, there are stochastic, mathematical, or computational. The targeted forgery method is yet another key factor when classifying signature verification systems. In the following, the literature of signature verification systems based on whether they are offline, online, or hybrids will be surveyed.

2.1 Offline Systems

Offline signature verification systems use a digitized static image of the signature that is usually acquired using a scanner. These systems are of particular interest when only hardcopies of signatures are available, e.g. when authenticating massive amounts of documents. Offline signature verification systems are more difficult to design because of the absence of temporal data, as only spatial data about signatures are available. Generally, two different classes of features can be extracted from static images, global and local. Global features are used to describe the entire signatures and they include the use of discrete Wavelet transform [4], Hough transform [5], horizontal and vertical projections [6] and smoothness features [7]. On the other hand, local features are extracted at the stroke and substroke levels, and include unballistic motion and tremor information [8], stroke elements [6], local shape descriptors [9], and pressure and slant features [10]. Recently, geometric parameters using fixed-point arithmetic were also reported in the literature [11].

Most of the previous work on offline signature verification has dealt primarily with random forgeries, where the signatures may not be the same name, and simple forgeries, where stylistic differences are prevalent. Many researchers therefore found it sufficient to consider only global features of the signature [7,8]. In addition, many pattern recognition methodologies have been reported in the literature in conjunction with offline signature verification systems. The template matching technique was used in [4,6,8]. Using Hidden Markov Models (HMMs) was also investigated in [12,13]. The pixels densities of the segmented signature image were used as local features in [18] where the cross-validation principle was utilized to detect random forgeries. Each questionable signature was tested, at different resolution levels, and a majority-vote rule was used to make the decision. The same methodology was extended in [19] to include three features from the segmented image, namely the pixel density, the pixel distribution, and the axial slant.

2.2 Online Systems

Unlike offline signature verification, the features used for online verification are always those with dynamic properties. A variety of techniques for online signature verification systems have been reported that depend on the signature capture instrument, temporal information such as velocity, acceleration, pen pressure, pen tilt, and spatial information such as x and y coordinates. These different techniques can be categorized according to the model used for verification. Since online signals contain a significant amount of data, data acquisition was the focus of this type of verification. Earlier work of online verification made use of global profiles of features such as speed and pressure [14-17]. In addition, local comparison was also made possible via using model-guided segmentation techniques. HMM-based methods along with other techniques such as elastic matching or time warping have also been widely used, including half regional correlation and tree matching. In general, the choice of a particular method is based primarily on processing time and on the features and parameters needed for each technique.

2.3 Hybrid and Other Systems

The success of other real-time methods in similar fields such as image processing, speech processing, and signal processing in general motivated trying to extend the application of such methods to signature verification. A combination of linear prediction coding (LPC) and neural networks were used in [18] for signature verification of Chinese characters, where each signature consists of several symbols. After normalizing the signatures with respect to the size and number of sample points, they were subdivided into frames. The LPC-cepstrum of both the x and y coordinates were computed, and later compared against a stored template using a three-layer perceptron network that is trained using the backpropagation method. The comparison was done for each symbol separately, and the compiled output of all the networks was compared to a threshold. The best AER was 4% for a database of 27 signers.

A scanner is usually used to digitize the static
image of the signature when applying offline techniques, while a digitizing tablet is used to capture the dynamic characteristics. Other forms of hardware have been tried in order to implement signature verification systems, such as gloves in [19], where individual movements of the fingers were recorded. When using web cameras, sound signals were also incorporated into the research activities, and this was justified by two reasons; firstly, web cameras have the capabilities of capturing sound and secondly, because sound as a digital signal possesses almost the same global features as images and signatures. This suggests applying the same techniques to sound signals as a mean for verifying peoples’ identities. Similar work on integrating sound with text was reported recently in the literature [20].

3 The Proposed System

Three major components were used in this paper, which include a Wacom intuos 3 (A6 Wide) digitizing tablet, a fast calculation-optimized Fujitsu-Siemens Celsius R540 workstation, and a complete version of MATLAB to implement the software. Two introductory steps were performed. Firstly, using MATLAB data acquisition toolbox to interface the required hardware to the computer in order to collect the required data in a digital format. Secondly, constructing the required database of the multimedia signals, digitizing them and storing them in a special format that is suitable for digital processing.

A detailed flow chart of the proposed system is given in Fig. (1), illustrating the steps involved in both the enrollment and verification phases. As indicated, the two phases are almost similar, and they share the same preprocessing and feature extraction steps. The enrollment phase terminates with storing the template in an external database, while the verification phase ends with a decision whether the questionable signature is genuine or forged.

![Diagram of enrollment and verification steps](image_url)

**Fig. (1): Enrollment and verification steps.**

### 3.1 Structure of the Proposed System

The proposed signature verification system involves three consecutive steps as follows:

**Step (1):** This is an online stage, during which the profile of the pen status was examined. Since the pen status is a binary signal, this step was rather simple, as it was only required to match the bit pattern of the questionable signature against the stored template. In addition, this step was slightly affected by the preprocessing phase, as it is both translation and rotation independent; however, it was required to undergo a time-normalization preprocessing step to ensure robust matching. The spline toolbox of MATLAB was used to resample the questionable signature such that the number of its samples is identical to that of the stored template.

**Step (2):** This step is also an online stage, during which the LPC Coefficients as well as the LPC-cepstrum of the magnitude of the velocity (speed) signal were examined. Although this stage could have been extended to include the magnitude of the acceleration signal as well, it was decided to only consider the speed to minimize the effect of the preprocessing phase. In addition, simulation results proved the effectiveness of this particular choice. This step represents a typical pattern recognition process as the neural network is used to decide on to whom the questionable signature belongs to. Using majority voting, the decision of this step can be used to support or weaken the decision of the first step.

**Step (3):** After passing both steps (1) and (2), the questionable signature undergoes an offline verification stage, during which the image of the signature is examined. Each image is split into a number of sub images via considering the acquired whole image as a video stream that could be split into frames with a predefined time interval.

### 3.2 Structure of the Proposed System

The x and y coordinates of the signature, as functions of time, depend on both the baseline and orientation of the signing frame. Two different signatures are related by:

\[
\begin{bmatrix}
  x_2 \\
  y_2 \\
  1
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & -\sin \theta & \Delta x \\
  \sin \theta & \cos \theta & \Delta y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  y_1 \\
  1
\end{bmatrix}
\]

where \(x_i\) and \(y_i\) and the position coordinates of signature \#i, \(\Delta x\) and \(\Delta y\) are the shift in coordinates between the starting point of both signatures, and \(\theta\) is the rotation angle that is measured in the counterclockwise direction. Thus, the velocity components are given by:

\[
\begin{align*}
  v_{x2} &= \dot{x}_2 = \dot{x}_1 \cos \theta - \dot{y}_1 \sin \theta = v_{x1} \cos \theta - v_{y1} \sin \theta \\
  v_{y2} &= \dot{y}_2 = \dot{x}_1 \sin \theta + \dot{y}_1 \cos \theta = v_{x1} \sin \theta + v_{y1} \cos \theta
\end{align*}
\]
where it is seen that only the effect of orientation is retained, while the effect of the initial starting point is removed. The magnitude of the velocity is:

\[ v_z = \sqrt{v_x^2 + v_y^2} = v_t \]

which can be considered a robust dynamic feature for the signature as it is independent of variations in both the initial position of the signature as well as the orientation. The same argument applies to the magnitude of the acceleration; however, the velocity is less sensitive to noise and to any artificial errors introduced to the signature data during the pre-processing phase. For \( n \) genuine signatures, from every signer, the average velocity can be calculated as:

\[ \bar{v} = \frac{1}{n} \sum_{i=1}^{n} v_i \]

which was stored as a template for comparison purposes during the verification phase. The translation effect, resulting from starting the signing process at different points was compensated for, as both \( \Delta x \) and \( \Delta y \) were calculated and then subtracted from the coordinates of the. The first and second moments of the signature coordinates, having \( N \) samples, are given by Eq.s (5) and (6) respectively:

\[ \mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{and} \quad \mu_y = \frac{1}{N} \sum_{i=1}^{N} y_i \]

\[ \sigma_x^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_x)^2 \quad \text{and} \quad \sigma_y^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mu_y)^2 \]

The orientation of the axis of least inertia was then calculated as the orientation of the least Eigen vector of the matrix in Eq. (7)

\[ \Theta = \begin{bmatrix} \sigma_x^2 & \mu_x \mu_y \\ \mu_x \mu_y & \sigma_y^2 \end{bmatrix} \]

(7)

Once this angle was obtained, all the points in the signature curve were rotated with this angle. This only changes the position coordinates of the signature and has no effect on the magnitude of both the velocity and acceleration. Other pre-processing phases were also needed to refine the data collected during the signing process. This includes extraneous signal removal, resampling, and normalization.

Figure (2) shows the complete motion profile of a sample signature, via augmenting the spatial coordinates of the signing pen while moving from one segment to another. Both the \( x \) and \( y \) coordinates are shown to be continuous functions of time, while the pen status is shown to be a binary signal that can serve two goals:

1. Firstly, it was used as a consistency check measure in the data acquisition phase, and consequently the samples that did not have the same binary pattern for a particular individual were rejected. This resulted in robustifying the data collected in the signature enrollment phase, and therefore improving the overall performance of the system.

2. Secondly, it was used as a feature in the verification phase for which the matching process is simple and straightforward. During the first online verification step, the questionable signature went through the same pre-processing steps and then its binary pattern was compared against the genuine signature, stored in the database.

Figure (3) shows the normalized preprocessed velocity and acceleration signals for the signature, shown in Fig. (2), where it was demonstrated that the magnitude of the velocity signal is both translation and rotation independent, while being less sensitive to noise and pre-processing artifacts compared to the acceleration signal.

**3.3 Consistency Check & Feature Extraction**

A database of 37 participants, each providing 10 signatures was used in this paper. The signatures have different sizes, time intervals, styles, and some of them are in Arabic (from right to left), and others are in English (from left to right). The 10 signatures from each participant were checked for consistency using the following two features:

1. the total signing time,
2. the binary pattern of the pen status.

The slowest and fastest signatures for each participant were removed from the database, while retain-
ing only the remaining eight. Then the mean and variance of the total signing time were calculated and compared to a certain threshold. In addition, the time-normalized signatures were used to extract the binary pattern of the pen status (up or down) and matched against each other. The results of these two steps were combined such that if they both satisfy the consistency criteria (mean and variance threshold as well as the binary pattern), the signatures were enrolled, otherwise, the participant had to repeat the enrollment process again for a maximum of two more trials, and otherwise his signature was rejected. Based of the aforementioned discussion, signatures 1-17 were passed, signatures 18-20 were rejected, and signatures 21-37 had to repeat the enrollment phase more than once. Thus, the signatures database reduced to only 34 consistent signatures. To establish a consistency measure (CM) for the samples collected from each participant, the following equation was proposed that calculates a normalized value of the data variance:

\[
CM = \frac{\sigma}{\mu}
\]

where \(\sigma\) is the variance and \(\mu\) is the mean value. For a questionable signature to pass this phase, it must meet the following three conditions:

1. The total signing time agrees with the default value. The agreement was achieved via assigning a threshold value for the allowable deviation between the measured signing time and the one corresponds to the template stored in the database. Threshold values in the range from 2% to 10% were tried, and it was found that the value of 5% was optimum. It should be emphasized that this step requires no preprocessing.

2. The consistency measure is again rechecked. This was done by augmenting the questionable (test) signature to the ones stored in the database (the default best 6 signatures), and then recalculating the new consistency measure of the new vector constituting the 7 signatures. If the change between the new and old CMs falls below a certain threshold, the test was passed, otherwise the signature was considered no to be genuine.

3. The binary pattern of the time-normalized test signature was matched against the one stored in the template. The resampling process using the "spline" function was a bottleneck in this step, as both the template and test signatures must have the same number of "bits" in order for the comparison to be meaningful.

This three-step verification phase is an online method as the temporal information about the signature data are needed. The three conditions were combined using an AND function; thus, a test signature must meet all the conditions in order for it to pass to the next verification phase.

The second phase of the verification process is also an online phase for which the magnitude of the velocity signal (speed) is used. To demonstrate the importance of the time and normalization preprocessing steps, Fig. (4) shows the improvement in matching two speed profiles before and after normalization in (a) and (b) respectively for participant #2.

![Fig. (4): Matching speed profiles.](image_url)

### 3.4 NNs for Matching Speed Signals

Speed signals were encoded using LPC, such that they were represented by vectors of equal lengths, even when the original signatures have different time intervals. This is an important fact, as the template signatures will have different number of samples after the time normalization process. In addition, the size normalization process will only scale the speed signal up or down, without affecting the shape of the time-profile. Thus representing the speed signal using LPC is an effective way to retain the temporal features of the signature regardless of the translation, rotation, or resampling effects.

Choosing the order of the LPC was done using trial and error, as different orders were applied to all signatures and the error between the original and reconstructed signals were estimated in order to arrive at the best one. The sum of the squared errors was used as a cost function and after a few runs it was decided to choose an order of 20 as it resulted in the best fit for the constructed signals, while accommodating different time intervals for the signatures falling in the range from 1 to 6 seconds approximately. After only keeping the best 6 signatures for every participant, the LPC coefficients were calculated for each one, and then their average was stored as a template for future comparison with test signatures.

Now that a database was created for storing the LPC coefficients of the speed signals for individual signatures, the second phase of the verification can be reduced to a pattern recognition problem for which neural networks were employed. This can be achieved by building a neural network with three...
layers, input, hidden, and output, and using back-
propagation, or any other training method such that
the network can recognize to whom a questionable
signature belongs to. It should be emphasized that
the recognition problem was somewhat different
from verification, as in the former, the answer can
belong to any of the enrolled users, while in the lat-
er, the answer is simply yes or no. Thus, recognition
is harder than verification, and consequently if the
questionable signature passes this phase, there
should be a high confidence in the overall result of
the verification system.

The 6 signatures stored in the database for each
participant were divided into two sets, 4 for training
and 2 for testing. Thus, the composite training vec-
ror has dimension 34×4×20 corresponding to the
number of participants, number of signatures, and
order of LPC. In addition, the target vector has di-
ension 34×1 corresponding to the number of par-
ticipants. Five different neural networks were built
and a majority voting technique was adopted such
that a result was accepted only if three out of theive agree on one decision. The MATLAB toolbox
of neural networks was used to build the five net-
works using radial basis (newrb), probabilistic radial
basis (newpnn), linear vector quantization (newlvq),
and feed-forward backpropagation network (newff).

Two different structures were used to implement
the linear vector quantization neural network via
changing the number of hidden neurons. Several
runs were done to arrive at the best parameters for
each network regarding speed of convergence,
learning algorithm, and number of epochs. For each
run, 4 different samples were used for the training,
while the remaining two for testing, resulting in a
conclusive answer for 91% of the time. Changing
the number of training sets to 5 and using only one
test signatures improved the results slightly to 92%.

3.5 Final Offline Phase
The previous two phases of the proposed signature
recognition system were considered online, as tem-
poral information about the signing process was in-
cluded. Although only the pen movement and the
speed were considered, the outlined methodology
could have been applied to other dynamic signals
such as the pressure and tilt angle. Depending on
the signing instrument, the pressure signal could be
converted into gray levels that are reflected in the
static image of the signature and consequently using
image processing techniques to isolate segments of
low and high pressure. This may be done using both
spatial and/or frequency techniques.

In the third and last phase of the proposed signa-
ture verification system, the following two steps
were performed:
1. The static image of the motion of the pen while
not touching the tablet (moving between signa-
ture segments) was investigated and compared
against a stored template. The Euclidean dis-
tances between the image of the questionable
signature and its corresponding template were
calculated, and compared with a certain thre-
shold. The particular choice of the pen-up
movements instead of the pen-down was to faci-
litate this offline phase, as they have fewer sam-
iples than the signature image itself.
2. Each enrolled signature (pen-down movements)
is treated as a video stream, which can be seg-
mented into a number of frames. Each frame
represents a static image of part of the signature.
Thus, the set of the static images encapsulate the
temporal information of the signature.

This phase is used to enforce the decision made
about the authenticity of the questionable signature
from the previous two phases, i.e. when a question-
able signature passes the first phase, it was allowed
to undergo the second phase, and if a conclusive
decision was obtained from the neural networks
committee, the signature was declared authentic,
otherwise it was passed to the third phase for further
investigation. This adds flexibility and versatility to
the proposed signature verification system.

3.6 Results
Integrating the three verification phases into one
software package was done and a simple MATLAB
program with an easy to use graphical user interface
was implemented to collect the data from different
participant where the acceptance/rejection decision
was made based on the consistency measure. The
main features used in the verification were:
1. total signing time (online),
2. consistency measure (online),
3. binary pattern of the pen movements (online),
4. LPC of the magnitude of the speed signal (online),
5. static image of the pen-up signature (offline),
6. time-based frames of the pen-down signature (of-
line/online).

while the auxiliary features were:
1. pressure signal:
   i. Time-based profile (online),
   ii. Gray level of the signature samples (offline),
2. LPC-cepstrum of the speed signal (online).

In this proposed system, signatures that failed to
pass the first phase were declared not authentic and
those that pass it were allowed to proceed to the
second phase. If they pass the second phase they are
declared authentic, otherwise, they will be passed to
the third and final phase for further checking. This
structure is flexible and several variations were at-
tempted to arrive at the best performance, but finally it was agreed to adopt the aforementioned structure due to its superiority. The best results were 2.9% and 8.8% for the FAR and FRR respectively when using the first phase followed by either the second or third phases.

4 Summary and Conclusion

In this paper a robust combination of both offline and online features of signatures were combined to implement a signature verification system. The proposed system was a hybrid approach that consists of three consecutive phases for which the first phase was mandatory, while the second phase was only needed if the second (main) phase resulted in an inconclusive result. The first phase examined three online features that included the total signing time, the consistency between the questionable signature and a stored database consisting of 6 signatures, and the binary pattern of the pen movements. This phase had the advantage of being online; thus capable of rejecting simple as well as skilled forgeries. Another advantage was that it does not undergo the normalization process, i.e. it was directly applied to the raw data. The second phase was mainly a pattern recognition problem in which a committee of five neural networks were used to examine the speed profile of a questionable signature. Linear predictive coding was used to convert the variable-length speed profile into a vector of 20 coefficients. Because recognition is harder than verification, the output of this phase was considered sufficient if a conclusive voting result was obtained; otherwise, a third offline phase was put in action.

Acknowledgment

The results of this paper were the outcome of project SP03/06 conducted at Kuwait University, Science College, Physics department.

References


