Combining distances through an auto-encoder network to verify signatures

Milena R. P. Souza
Center of Informatics
Federal University of Pernambuco
Recife, PE, 50732-970, Brazil
mrps@cin.ufpe.br

Leandro R. Almeida
Center of Informatics
Federal University of Pernambuco
Recife, PE, 50732-970, Brazil
lra@cin.ufpe.br

George D. C. Cavalcanti
Center of Informatics
Federal University of Pernambuco
Recife, PE, 50732-970, Brazil
gdcc@cin.ufpe.br

Abstract

In this paper we present a system for off-line signature verification. The paper's contributions are: i) Five distances were calculated and evaluated over the signature database, they are: furthest, nearest, template, central and ncentral. Also, a normalization procedure is established to turn each distance scale invariant; ii) These distances are combined using the following rules: product, mean, maximum and minimum; iii) The calculated distances can be used as a feature vector to represent a given signature. So, the feature vectors found and their combination were finally used as input vector for an auto-encoder neural network. All the experimental study is done using one-class classification, which demands only the genuine signature to generalize. The proposed approaches achieved very good rates for the signature verification task.

1. Introduction

Signature verification is a very important commercial application of handwritten recognition. Every bank has a recognition system which operates automatically or manually. Signature verification consists of distinguishing genuine signature from forged ones. Forged signatures can be split in, at least, two classes: skilled and random. A skilled forgery is signed by someone who has had contact with the genuine signature. The random forgery is signed without any information about the signature. Signature verification is traditionally separated into two categories according to the input data used: on-line and off-line. On-line (dynamic) signature verification uses information captured from an electronic device (e.g. PDA, laptop). This information could be the pen pressure, speed, number and order of the strokes, etc. Off-line (static) signature verification takes images as inputs. Generally, techniques that use on-line signature verification present better results. On the other hand, off-line techniques preserve the natural process of writing and do not need any special equipment.

Many techniques have been proposed to deal with the signature verification problem. Neural networks have been applied for off-line and on-line signature verification. The multilayer perceptron network (MLP) trained with backpropagation is the most common model used in signature verification. Various different approaches of MLP had been used in the literature, but in this paper, we particularly investigate the artificial neural network based on the auto-associative model. In [2] a model for banknote verification is done based on auto-associators, in [4] is proposed the use of multilayered auto-associators for problems of speech verification and in [9] Cascade-Correlation (Cascor) under the auto-associative approach. Here we will use an auto-associative neural network with different distance combinations. The advantage of the neural networks related with many other techniques is their ability to learn and realize the class separability. The class of neural network used here has a special advantage for the signature verification problem: in the training of an auto-associative neural network it is only needed genuine patterns. This is an important aspect because acquiring actual forgery signatures from each class of the system is a difficult task. Thus, the forged ones, in the proposed system, are used only in the test phase. This paper is organized as follows. In Section 2 the system architecture is described. This includes: three different feature extrac-

1522-4899/08 $25.00 © 2008 IEEE
DOI 10.1109/SBRN.2008.10

tion techniques: Shadow code [10], Leung [1] and Lee [8]; how we use theses distances to calculate five different distances [11], how this distances are combined, and finally how use this distances as inputs of an auto-encoded neural network [6]. Soon after, Section 2.3 shows our experiments and the results obtained using auto-encoded network. In the last section the conclusion and future work are presented.

2. System Architecture

The system architecture could be split in three stages. First a feature extraction was done using three different approaches, then five distance measures were calculated for these features, and finally the distances obtained were used as input to an auto-encoder neural network. These stages will be detailed in the next subsections.

2.1. Feature Extraction

To represent the signature we have experimented three approaches: Shadow code [10], Leung [1] and Lee [8]. In the Shadow code strategy the signature is divided in blocks of 32 by 32 pixels, from each block is extracted six features: two horizontal, two vertical and two diagonal. The feature values range from 0 to 32 (horizontal and vertical bars) and from 0 to $32\sqrt{2}$ (diagonal bars), the value of the feature is calculated according to the number of black pixels projected on the bars as shown in Figure 1. The original strategy has a pre-processing stage that resizes the signature to 512x128 pixels, but in our work there is not any pre-processing, the original signature is used to extract the features.

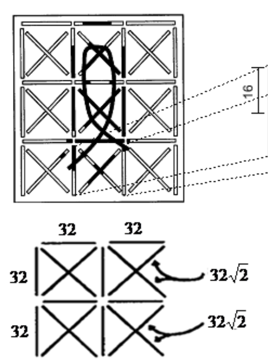

Figure 1. Shadow code feature extraction.

The second feature group evaluated was the peripheral and differential peripheral described in [1]. The signature image is divided into 3 horizontal and vertical strips as shown in Figure 2(a) and (b). In Figure 2(a) each strip is the area between the edge of the virtual frame (bounding box) and the first white-to-black pixel transition is calculated (peripheral feature). For the horizontal strips, the pixels are traversed row by row. In figure 2(b) each strip is the distance between the first black-to-white transition and the second white-to-black transition is obtained, within each strip, the distances so obtained are summed up (differential peripheral feature). This processes gives a 48-dimensional feature vector (24 from peripheral plus 24 for differential peripheral).

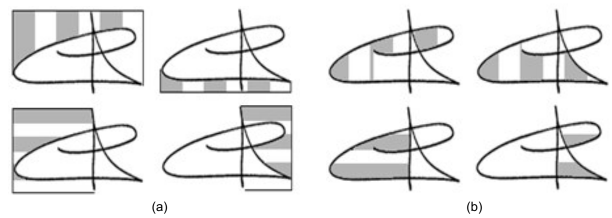

Figure 2. (a) Extraction of peripheral features and (b) Extraction of differential peripheral features.

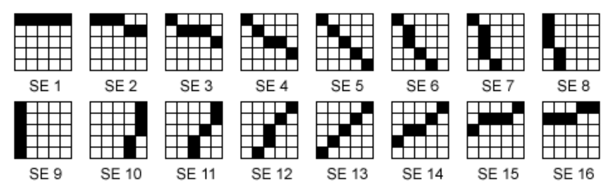

Figure 3. Structure Elements.

In [8] the feature set regards the stroke orientation in a signature image. Before extracting such information, we implement a set of 16 five-by-five structuring elements (SEs) as shown in Figure 3. The goal is to identify the degree of inclination of signature strokes. The signature is divided in blocks as the Shadow code strategy, but in blocks of 5 by 5 pixels, for each block the 16 SEs are compared with the signature and counted the frequency of matches of the black pixels. This process gives 16-dimensional feature vector that are the frequencies of the SEs.

2.2. Distances

Five different distances are calculated for each signature, using Euclidian distance:

$$d_e(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

d_{min} the minimum distance between a signature and the elements of the reference set. It represents the distance to the nearest neighbor;

d_{max} the maximum distance between a signature and the elements of the reference set. It represents the distance to the farthest neighbor in the reference set;

$d_{central}$ the distance between a signature and the mass center of the reference set;

$d_{template}$ The distance between a signature and the template. The template is the reference signature with the minimum average distance to all other signatures in the reference set;

$d_{ncentral}$ an average between the distance to each element in the reference set and the $d_{central}$.

These distances are illustrated in Figure 4, where X_i represents each element of the reference set and Q is a query signature whose distances will be calculated. Aiming to turn these distances scale independent, they are normalized by the corresponding averages of the reference set (RS). These distances form a five-dimensional feature vector (F_Q):

$$F_Q = \begin{bmatrix} d_{min}(x, RS) \backslash \bar{d}_{min}(RS) \\ d_{max}(x, RS) \backslash \bar{d}_{max}(RS) \\ d_{central}(x, RS) \backslash \bar{d}_{central}(RS) \\ d_{template}(x, RS) \backslash \bar{d}_{template}(RS) \\ d_{ncentral}(x, RS) \backslash \bar{d}_{ncentral}(RS) \end{bmatrix} \tag{2}$$

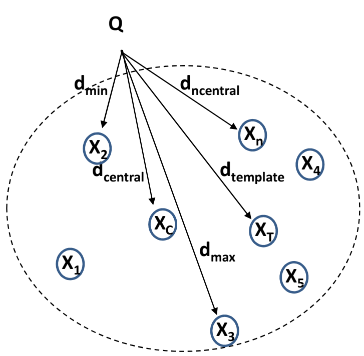

Figure 4. The used distances.

As exposed by Kholmatov and Yanikoglu [6], the normalization procedure eliminates the need to calculate user-dependent thresholds.

The normalization factors are calculated over a reference set RS. The computed average values over RS are: $\bar{d}_{min}(RS)$ distances of reference signatures to their nearest neighbor; $\bar{d}_{max}(RS)$ distances of reference signatures to their farthest neighbor; $\bar{d}_{central}(RS)$ distance of the reference signatures to their mass center; $\bar{d}_{template}(RS)$ distances of reference signatures to the template signature; $\bar{d}_{ncentral}(RS)$ distances of the reference signatures using $\bar{d}_{ncentral}(x, RS \setminus x)$, where x is an element of RS and $\setminus$ is the difference set operation.

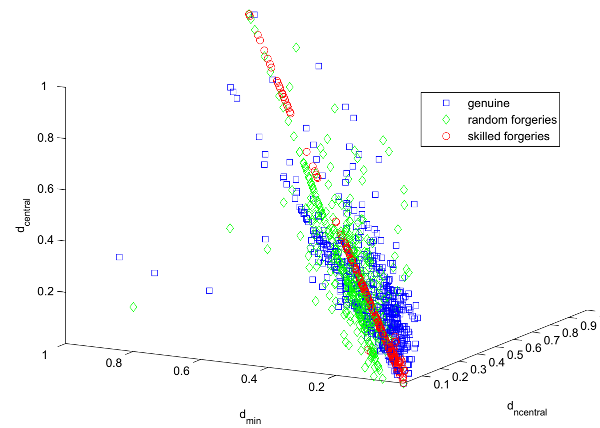

Figure 5. A three-dimensional plot of signatures distances for the shadow code feature extraction.

A plot of the data feature space is shown in Figure 5, it is formed by the distances: $d_{central}$, $d_{ncentral}$ and d_{min}.

2.3. Auto-encoder Neural Network

Gori et al. [5] investigated the ability of multi-layer perceptrons (MLP) in the creation of bounded domains in the pattern space and, in particular, they related this analysis to applications of pattern verification. They proved that, regardless of the function used in the processing units, architectures with less units in the first hidden layer than inputs cannot yield closed separation surfaces. When using more hidden units than inputs, they have also proven that an MLP can either create open or closed surfaces. Moreover, no choice of the sigmoidal function in the neurons can transform open separation surfaces into closed separation surfaces, and deciding whether or not they are open is NP-hard. There are alternative approaches to pattern verification using neural networks which do not suffer from the problems pointed out above. For instance, MLPs used as auto-associators (auto-encoders) [7], where the weights are adjusted so as to copy the inputs to the outputs, can profitably be used for designing pattern verification systems. For each pattern, the verification criterion is based on the

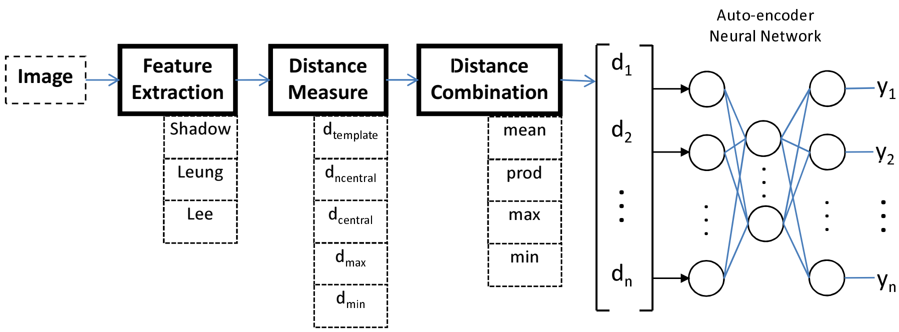

Figure 6. System architecture.

input/output Euclidean distance, that is, given a threshold t, pattern x is accepted if and only if $|f(x) - x| \leq t$. The basic idea is that only the patterns of the class used for training the auto-associator are likely to be reproduced with enough approximation at the output. It has been shown that in this case the separation surfaces are always closed (e.g., [4]).

3. Experimental Study

The complete system architecture is shown in Figure 6. The aim of the first phase is to extract a feature vector from a signature image. After that, five different distances are calculated based on the features obtained before. Each calculated distance results in a vector of n positions (d_1, d_2, ... d_n) that will be combined using four combinations strategies. Finally we use these combined distances, one by one, as input for an auto-encoder neural network.

3.1. Database

The proposed method was evaluated on a database that contains 1057 authentic signatures and 675 forgeries, the latest are divided in 343 skilled forgeries and 332 random forgeries. The authentic signatures were collected from 53 volunteers, each person contributing with, approximately, 20 signatures. The average of skilled forgeries and random forgeries is 7 for each user [3].

The training set was formed by 5 of the 20 genuine signatures (total of 265 signatures - 53 users). All the other signatures (genuine and forgeries) formed the test set (total of 1528).

3.2. Results

From the 20 authentic signatures for each volunteer, 5 were taken for training, and the remaining authentic signatures with the forgeries signatures were used to test the system. For each reference signature taken using the feature extraction methods described (Shadow code, Leung and Lee), five different distances were calculated. The first experiment done is based on the architecture scheme exposed in Figure 7. There the distance combination and the auto-encoder neural network are not used, only the distances are evaluated separately.

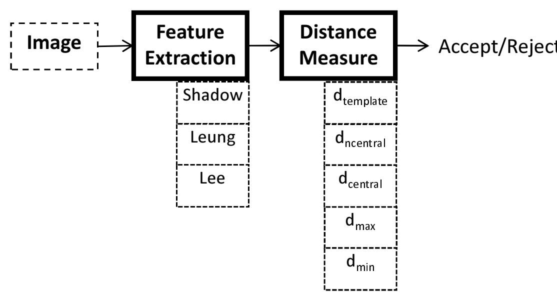

Figure 7. First experiment scheme.

The results obtained for the five distances calculated can be seen in Table 1. All the results are shown based on the Area Under the ROC Curve (AUC) value. The *Random* label in the table means that the AUC were calculated using only genuine and random forgeries. While the *Skilled* one the AUC value used genuine and skilled forgeries. In the *Global* label all kinds of signatures were used: genuine and random and skilled forgeries. As we can see in Table 1, the Leung feature extraction achieved the best results for random signatures, but was worse than the Shadow code for skilled ones. Considering the global AUC, Shadow code got the best result with an AUC of 0.8862 ($d_{central}$). In our second experiment, Figure 8, these five distances were used, separately, as input to an auto-encoder network. The results, in terms of the AUC, can be seen in Table 2. Using the neural network the global AUC increased from 0.8862 (first experiment using shadow code and $d_{central}$) to 0.9043 (second experiment using shadow code and d_{max}) over the best result. It can be verified that the application of the neural network in general caused a decrease on the AUC for

Distance	Feature	Random	Skilled	Global
$d_{central}$	Shadow Code	*0.9547*	*0.8236*	*0.8862*
	Leung	0.9662	0.7142	0.8386
	Lee	0.8623	0.6300	0.7447
d_{max}	Shadow Code	0.9177	0.7915	0.8538
	Leung	0.9487	0.6871	0.8162
	Lee	0.8529	0.6215	0.7357
d_{min}	Shadow Code	0.9044	0.6737	0.7875
	Leung	0.9298	0.5465	0.7357
	Lee	0.7792	0.6004	0.6824
$d_{template}$	Shadow Code	0.9161	0.7598	0.8369
	Leung	0.9515	0.6809	0.8144
	Lee	0.8680	0.6367	0.7509
$d_{ncentral}$	Shadow Code	0.9500	0.8209	0.8846
	Leung	0.9671	0.7149	0.8394
	Lee	0.8693	0.6356	0.7510

Table 1. Area Under ROC Curve for each distances.

the random forgeries and an increase on the AUC for the skilled ones. For most of the cases the global results for the application of the neural network was positive.

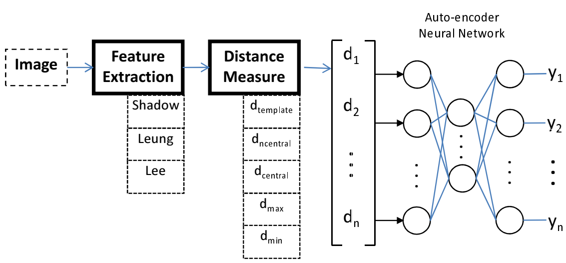

Figure 8. Second experiment scheme.

The third experiment used the complete architecture showed before in Figure 6. The four combination strategies (product, mean, minimum and maximum) of the five distances are calculated. To perform the combination, we pairwise the five distance vectors and calculate the product, mean, maximum and minimum of each index. Forming four new vectors with the same size of the five distance vectors calculated before. Each one of these four distance combination vectors is separately used as input to the auto-encoded neural network. Table 3 shows the AUC for these combinations. For the best result the global AUC was the min combination with 0.9234 (using the Shadow code and the minimum combination). This result is better than previous ones, but we can notice that in general the combination presents worse results.

For each best achieved result ($d_{central}$ for the first experiment, d_{max} for the second and min for the third, all of them using shadow code), it was collected the true pos-

Distance	Feature	Random	Skilled	Global
$d_{central}$	Shadow Code	0.9013	0.7988	0.8375
	Leung	0.9628	0.7728	0.8467
	Lee	0.8762	0.7252	0.7874
d_{max}	Shadow Code	*0.9646*	*0.8424*	*0.9043*
	Leung	0.9585	0.7527	0.7912
	Lee	0.8421	0.7233	0.7793
d_{min}	Shadow Code	0.8742	0.7286	0.8196
	Leung	0.9503	0.7465	0.8258
	Lee	0.8545	0.6179	0.7109
$d_{template}$	Shadow Code	0.9063	0.7593	0.8589
	Leung	0.9242	0.7347	0.8149
	Lee	0.8585	0.6925	0.7247
$d_{ncentral}$	Shadow Code	0.9398	0.8027	0.8898
	Leung	0.9642	0.7552	0.8366
	Lee	0.8728	0.7385	0.7527

Table 2. Area Under ROC Curve for each distances applied to an auto-encoder network.

Distance	Feature	Random	Skilled	Global
product	Shadow Code	0.9547	0.8079	0.8885
	Leung	0.9607	0.7728	0.8344
	Lee	0.8812	0.7173	0.7902
mean	Shadow Code	0.9446	0.7903	0.8293
	Leung	0.9628	0.7728	0.8467
	Lee	0.7305	0.3518	0.5582
max	Shadow Code	0.7222	0.6869	0.6968
	Leung	0.7103	0.5991	0.6530
	Lee	0.4702	0.2748	0.3559
min	Shadow Code	*0.9738*	*0.8717*	*0.9234*
	Leung	0.9566	0.6821	0.7922
	Lee	0.5780	0.2748	0.2913

Table 3. Area Under ROC Curve for the combination of the distances applied to the auto-encoder network.

itive rate for 5 and 10% of false positive. The rates obtained are exposed in Table 4. For all the experiments the Shadow code feature extraction presented the best results. Figure 9 illustrates the ROC curves for $d_{ncentral}$ and the $mean$ combination of distances applied to an auto-encoder network, both using Shadow code. We can clearly notice the benefits of using the distance combination and the auto-encoder network, especially for random and skilled forgeries. As expected, the results are much better for random forgeries, with approximately, 7% of error rate for false negative against 10% of false positive. When the results over the skilled forgery signatures are analyzed, it is possible to note that they represent a much more difficult task; only half of the tested images were corrected classified (FP=10%).

tion techniques: Shadow code [10], Leung [1] and Lee [8]; how we use theses distances to calculate five different distances [11], how this distances are combined, and finally how use this distances as inputs of an auto-encoded neural network [6]. Soon after, Section 2.3 shows our experiments and the results obtained using auto-encoded network. In the last section the conclusion and future work are presented.

2. System Architecture

The system architecture could be split in three stages. First a feature extraction was done using three different approaches, then five distance measures were calculated for these features, and finally the distances obtained were used as input to an auto-encoder neural network. These stages will be detailed in the next subsections.

2.1. Feature Extraction

To represent the signature we have experimented three approaches: Shadow code [10], Leung [1] and Lee [8]. In the Shadow code strategy the signature is divided in blocks of 32 by 32 pixels, from each block is extracted six features: two horizontal, two vertical and two diagonal. The feature values range from 0 to 32 (horizontal and vertical bars) and from 0 to $32\sqrt{2}$ (diagonal bars), the value of the feature is calculated according to the number of black pixels projected on the bars as shown in Figure 1. The original strategy has a pre-processing stage that resizes the signature to 512x128 pixels, but in our work there is not any pre-processing, the original signature is used to extract the features.

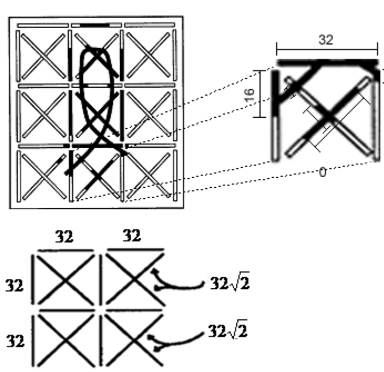

Figure 1. Shadow code feature extraction.

The second feature group evaluated was the peripheral and differential peripheral described in [1]. The signature image is divided into 3 horizontal and vertical strips as shown in Figure 2(a) and (b). In Figure 2(a) each strip is the area between the edge of the virtual frame (bounding box) and the first white-to-black pixel transition is calculated (peripheral feature). For the horizontal strips, the pixels are traversed row by row. In figure 2(b) each strip is the distance between the first black-to-white transition and the second white-to-black transition is obtained, within each strip, the distances so obtained are summed up (differential peripheral feature). This processes gives a 48-dimensional feature vector (24 from peripheral plus 24 for differential peripheral).

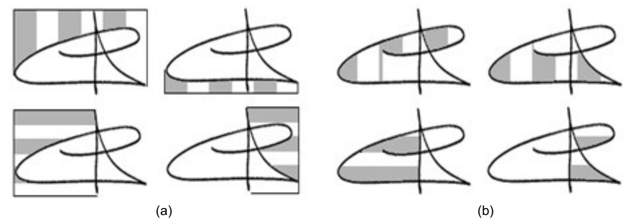

Figure 2. (a) Extraction of peripheral features and (b) Extraction of differential peripheral features.

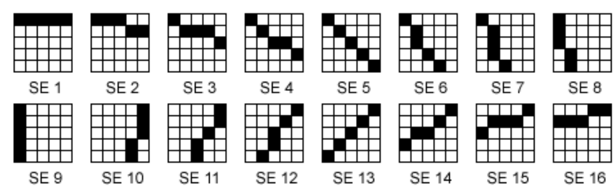

Figure 3. Structure Elements.

In [8] the feature set regards the stroke orientation in a signature image. Before extracting such information, we implement a set of 16 five-by-five structuring elements (SEs) as shown in Figure 3. The goal is to identify the degree of inclination of signature strokes. The signature is divided in blocks as the Shadow code strategy, but in blocks of 5 by 5 pixels, for each block the 16 SEs are compared with the signature and counted the frequency of matches of the black pixels. This process gives 16-dimensional feature vector that are the frequencies of the SEs.

2.2. Distances

Five different distances are calculated for each signature, using Euclidian distance:

$$d_e(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

d_{min} the minimum distance between a signature and the elements of the reference set. It represents the distance to the nearest neighbor;

d_{max} the maximum distance between a signature and the elements of the reference set. It represents the distance to the farthest neighbor in the reference set;

$d_{central}$ the distance between a signature and the mass center of the reference set;

$d_{template}$ The distance between a signature and the template. The template is the reference signature with the minimum average distance to all other signatures in the reference set;

$d_{ncentral}$ an average between the distance to each element in the reference set and the $d_{central}$.

These distances are illustrated in Figure 4, where X_i represents each element of the reference set and Q is a query signature whose distances will be calculated. Aiming to turn these distances scale independent, they are normalized by the corresponding averages of the reference set (RS). These distances form a five-dimensional feature vector (F_Q):

$$F_Q = \begin{bmatrix} d_{min}(x, RS)\backslash\bar{d}_{min}(RS) \\ d_{max}(x, RS)\backslash\bar{d}_{max}(RS) \\ d_{central}(x, RS)\backslash\bar{d}_{central}(RS) \\ d_{template}(x, RS)\backslash\bar{d}_{template}(RS) \\ d_{ncentral}(x, RS)\backslash\bar{d}_{ncentral}(RS) \end{bmatrix} \quad (2)$$

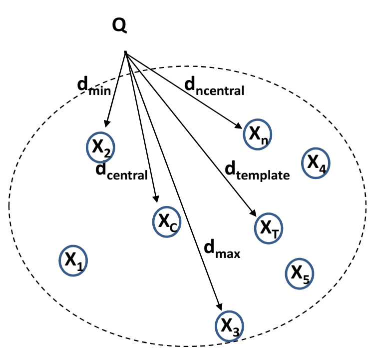

Figure 4. The used distances.

As exposed by Kholmatov and Yanikoglu [6], the normalization procedure eliminates the need to calculate user-dependent thresholds.

The normalization factors are calculated over a reference set RS. The computed average values over RS are: $\bar{d}_{min}(RS)$ distances of reference signatures to their nearest neighbor; $\bar{d}_{max}(RS)$ distances of reference signatures to their farthest neighbor; $\bar{d}_{central}(RS)$ distance of the reference signatures to their mass center; $\bar{d}_{template}(RS)$ distances of reference signatures to the template signature; $\bar{d}_{ncentral}(RS)$ distances of the reference signatures using $\bar{d}_{ncentral}(x, RS \setminus x)$, where x is an element of RS and $\setminus$ is the difference set operation.

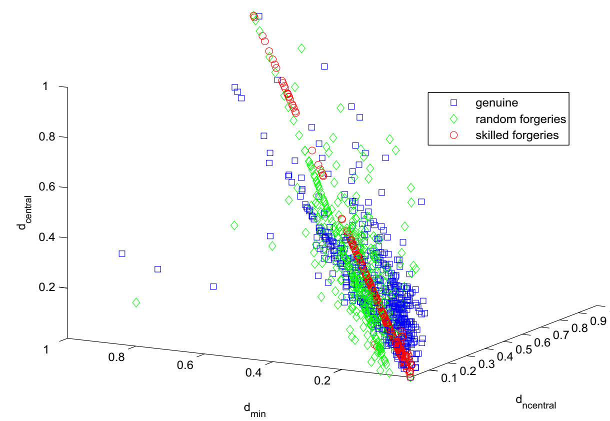

Figure 5. A three-dimensional plot of signatures distances for the shadow code feature extraction.

A plot of the data feature space is shown in Figure 5, it is formed by the distances: $d_{central}$, $d_{ncentral}$ and d_{min}.

2.3. Auto-encoder Neural Network

Gori et al. [5] investigated the ability of multi-layer perceptrons (MLP) in the creation of bounded domains in the pattern space and, in particular, they related this analysis to applications of pattern verification. They proved that, regardless of the function used in the processing units, architectures with less units in the first hidden layer than inputs cannot yield closed separation surfaces. When using more hidden units than inputs, they have also proven that an MLP can either create open or closed surfaces. Moreover, no choice of the sigmoidal function in the neurons can transform open separation surfaces into closed separation surfaces, and deciding whether or not they are open is NP-hard. There are alternative approaches to pattern verification using neural networks which do not suffer from the problems pointed out above. For instance, MLPs used as auto-associators (auto-encoders) [7], where the weights are adjusted so as to copy the inputs to the outputs, can profitably be used for designing pattern verification systems. For each pattern, the verification criterion is based on the

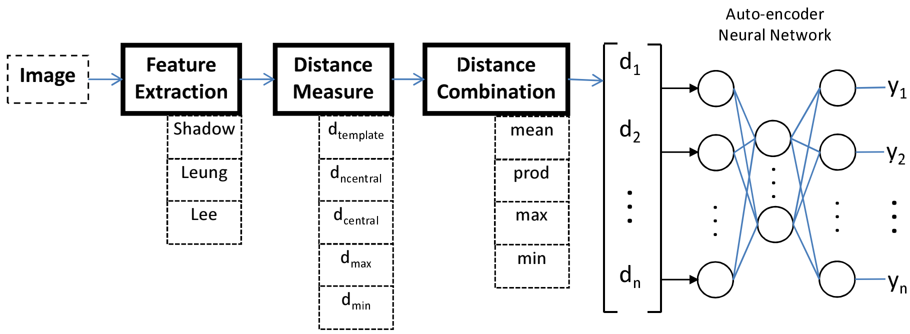

Figure 6. System architecture.

input/output Euclidean distance, that is, given a threshold t, pattern x is accepted if and only if $|f(x) - x| \leq t$. The basic idea is that only the patterns of the class used for training the auto-associator are likely to be reproduced with enough approximation at the output. It has been shown that in this case the separation surfaces are always closed (e.g., [4]).

3. Experimental Study

The complete system architecture is shown in Figure 6. The aim of the first phase is to extract a feature vector from a signature image. After that, five different distances are calculated based on the features obtained before. Each calculated distance results in a vector of n positions (d_1, d_2, ... d_n) that will be combined using four combinations strategies. Finally we use these combined distances, one by one, as input for an auto-encoder neural network.

3.1. Database

The proposed method was evaluated on a database that contains 1057 authentic signatures and 675 forgeries, the latest are divided in 343 skilled forgeries and 332 random forgeries. The authentic signatures were collected from 53 volunteers, each person contributing with, approximately, 20 signatures. The average of skilled forgeries and random forgeries is 7 for each user [3].

The training set was formed by 5 of the 20 genuine signatures (total of 265 signatures - 53 users). All the other signatures (genuine and forgeries) formed the test set (total of 1528).

3.2. Results

From the 20 authentic signatures for each volunteer, 5 were taken for training, and the remaining authentic signatures with the forgeries signatures were used to test the system. For each reference signature taken using the feature extraction methods described (Shadow code, Leung and Lee), five different distances were calculated. The first experiment done is based on the architecture scheme exposed in Figure 7. There the distance combination and the auto-encoder neural network are not used, only the distances are evaluated separately.

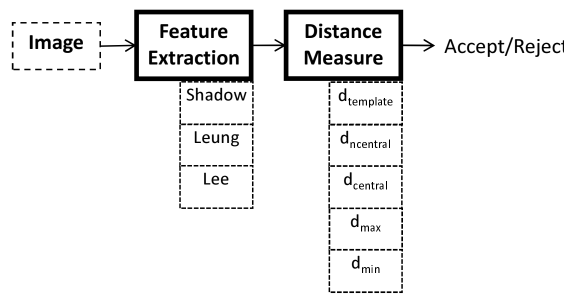

Figure 7. First experiment scheme.

The results obtained for the five distances calculated can be seen in Table 1. All the results are shown based on the Area Under the ROC Curve (AUC) value. The *Random* label in the table means that the AUC were calculated using only genuine and random forgeries. While the *Skilled* one the AUC value used genuine and skilled forgeries. In the *Global* label all kinds of signatures were used: genuine and random and skilled forgeries. As we can see in Table 1, the Leung feature extraction achieved the best results for random signatures, but was worse than the Shadow code for skilled ones. Considering the global AUC, Shadow code got the best result with an AUC of 0.8862 ($d_{central}$). In our second experiment, Figure 8, these five distances were used, separately, as input to an auto-encoder network. The results, in terms of the AUC, can be seen in Table 2. Using the neural network the global AUC increased from 0.8862 (first experiment using shadow code and $d_{central}$) to 0.9043 (second experiment using shadow code and d_{max}) over the best result. It can be verified that the application of the neural network in general caused a decrease on the AUC for

Distance	Feature	Random	Skilled	Global
$d_{central}$	Shadow Code	*0.9547*	*0.8236*	*0.8862*
	Leung	0.9662	0.7142	0.8386
	Lee	0.8623	0.6300	0.7447
d_{max}	Shadow Code	0.9177	0.7915	0.8538
	Leung	0.9487	0.6871	0.8162
	Lee	0.8529	0.6215	0.7357
d_{min}	Shadow Code	0.9044	0.6737	0.7875
	Leung	0.9298	0.5465	0.7357
	Lee	0.7792	0.6004	0.6824
$d_{template}$	Shadow Code	0.9161	0.7598	0.8369
	Leung	0.9515	0.6809	0.8144
	Lee	0.8680	0.6367	0.7509
$d_{ncentral}$	Shadow Code	0.9500	0.8209	0.8846
	Leung	0.9671	0.7149	0.8394
	Lee	0.8693	0.6356	0.7510

Table 1. Area Under ROC Curve for each distances.

Distance	Feature	Random	Skilled	Global
$d_{central}$	Shadow Code	0.9013	0.7988	0.8375
	Leung	0.9628	0.7728	0.8467
	Lee	0.8762	0.7252	0.7874
d_{max}	Shadow Code	*0.9646*	*0.8424*	*0.9043*
	Leung	0.9585	0.7527	0.7912
	Lee	0.8421	0.7233	0.7793
d_{min}	Shadow Code	0.8742	0.7286	0.8196
	Leung	0.9503	0.7465	0.8258
	Lee	0.8545	0.6179	0.7109
$d_{template}$	Shadow Code	0.9063	0.7593	0.8589
	Leung	0.9242	0.7347	0.8149
	Lee	0.8585	0.6925	0.7247
$d_{ncentral}$	Shadow Code	0.9398	0.8027	0.8898
	Leung	0.9642	0.7552	0.8366
	Lee	0.8728	0.7385	0.7527

Table 2. Area Under ROC Curve for each distances applied to an auto-encoder network.

the random forgeries and an increase on the AUC for the skilled ones. For most of the cases the global results for the application of the neural network was positive.

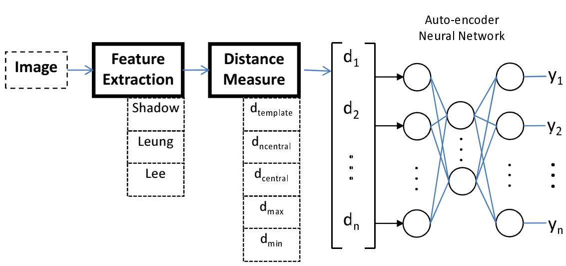

Figure 8. Second experiment scheme.

The third experiment used the complete architecture showed before in Figure 6. The four combination strategies (product, mean, minimum and maximum) of the five distances are calculated. To perform the combination, we pairwise the five distance vectors and calculate the product, mean, maximum and minimum of each index. Forming four new vectors with the same size of the five distance vectors calculated before. Each one of these four distance combination vectors is separately used as input to the auto-encoded neural network. Table 3 shows the AUC for these combinations. For the best result the global AUC was the min combination with 0.9234 (using the Shadow code and the minimum combination). This result is better than previous ones, but we can notice that in general the combination presents worse results.

For each best achieved result ($d_{central}$ for the first experiment, d_{max} for the second and min for the third, all of them using shadow code), it was collected the true positive rate for 5 and 10% of false positive. The rates obtained are exposed in Table 4. For all the experiments the Shadow code feature extraction presented the best results. Figure 9 illustrates the ROC curves for $d_{ncentral}$ and the $mean$ combination of distances applied to an auto-encoder network, both using Shadow code. We can clearly notice the benefits of using the distance combination and the auto-encoder network, especially for random and skilled forgeries. As expected, the results are much better for random forgeries, with approximately, 7% of error rate for false negative against 10% of false positive. When the results over the skilled forgery signatures are analyzed, it is possible to note that they represent a much more difficult task; only half of the tested images were corrected classified (FP=10%).

Distance	Feature	Random	Skilled	Global
product	Shadow Code	0.9547	0.8079	0.8885
	Leung	0.9607	0.7728	0.8344
	Lee	0.8812	0.7173	0.7902
mean	Shadow Code	0.9446	0.7903	0.8293
	Leung	0.9628	0.7728	0.8467
	Lee	0.7305	0.3518	0.5582
max	Shadow Code	0.7222	0.6869	0.6968
	Leung	0.7103	0.5991	0.6530
	Lee	0.4702	0.2748	0.3559
min	Shadow Code	*0.9738*	*0.8717*	*0.9234*
	Leung	0.9566	0.6821	0.7922
	Lee	0.5780	0.2748	0.2913

Table 3. Area Under ROC Curve for the combination of the distances applied to the auto-encoder network.

		$d_{central}$	d_{max} + AANN	min + AANN
global	FP=5%	47%	47%	49%
	FP=10%	64%	67%	69%
skilled	FP=5%	26%	34%	36%
	FP=10%	50%	51%	52%
random	FP=5%	82%	84%	87%
	FP=10%	89%	91%	93%

Table 4. True Positives for global, skilled and random forgeries when the False Positive Rates are 5 and 10% for the best result obtained using Shadow code.

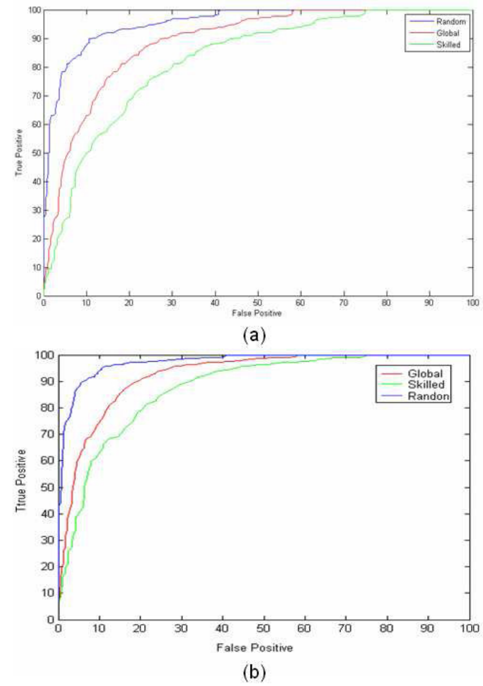

Figure 9. ROC curve for (a) $d_{central}$ and (b) min combination of distances applied to an auto-encoder network, both using Shadow code.

4. Final Remarks

An off-line signature system using different combinations of the feature vectors and submitted to an auto-encoder neural network was presented. To evaluate the system three experiments were done. The first experiment uses just the five distances calculated with the feature vectors. The second experiment shows how the results of the first experiment were improved by using an auto-encoder network. Finally, the third experiment illustrates the complete system architecture, when the distances calculated are combined in four different ways and each combination was used in the auto-encoder network. The main contribution of our system is to associate the feature extractions techniques of various normalized distances measures and the combination of them with an auto-encoder network. The idea was to treat the problem using only the genuine patterns to train the system, this approach of one-class classification has advantages, once only patterns of one class are required. We have also studied the effect of combining various distances, which produced better results than the distances alone.

References

[1] B. Fang, C. Leung, Y. Tang, P. Kwok, K. Tse, and Y. Wong. Offline signature verification with generated training samples. *IEEE Proceedings-Vision, Image and Signal Processing*, 149(2):85–90, 2002.

[2] A. Frosini, M. Gori, and P. Priami. A neural network-based model for paper currency recognition and verification. *IEEE Transactions on Neural Networks*, 7(6):1482–1490, 1996.

[3] H. M. Gomes. Investigacão de técnicas automáticas para reconhecimento off-line de assinaturas. Master's thesis, Federal University of Pernambuco, 1995.

[4] M. Gori, L. Lastrucci, and G. Soda. Autoassociator-based models for speaker verification. *Pattern Recognition Letters*, 17(3):241–250, 1996.

[5] M. Gori and F. Scarselli. Are Multilayer Perceptrons Adequate for Pattern Recognition and Verification? *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(11):1121–1132, 1998.

[6] A. Kholmatov and B. Yanikogl. Identity authentication using improved online signature verification method. *Pattern Recognition Letters*, 26(15):2400–2408, 2005.

[7] M. A. Kramer. Autoassociative Neural Networks. *Computers Chem Engng*, 16(4):313–328, 1992.

[8] L. L. Lee, M. G. Lizarraga, N. R. Gomes, and A. L. Koerich. A Prototype for Brazilian Bankcheck Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(4):549–570, 1997.

[9] J. N. G. Ribeiro and G. C. Vasconcelos. Off-line signature verification using an autoassociator cascade-correlation architecture. *International Joint Conference on Neural Networks*, 4:2882–2886, 1999.

[10] R. Sabourin, M. Cheriet, and G. Genest. An extended-shadow-code based approach for off-line signature verification. In *International Conference on Document Analysis and Recognition*, pages 1–5, 1993.

[11] M. Umeda. Recognition of Multi-font Printed Chinese Characters. In *International Conference on Pattern Recognition*, volume 2, pages 793–796, 1982.