Online Signature Verification using Deep Representation: A new Descriptor

Article · January 2016

CITATIONS
0

READS
482

4 authors:

Mohammad Hajizadeh Saffar
Iran University of Science and Technology
8 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Mohsen Fayyaz
University of Bonn
13 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Mohammad Sabokrou
Institute for Research in Fundamental Sciences (IPM)
26 PUBLICATIONS **120** CITATIONS
SEE PROFILE

Mahmood Fathy
Iran University of Science and Technology
336 PUBLICATIONS **2,633** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Action Recognition View project

Project Optimal RSU Placement in Urban Scenarios View project

Noname manuscript No.
(will be inserted by the editor)

Online Signature Verification using Deep Representation: A new Descriptor

Received: date / Accepted: date

Abstract This paper presents an accurate method for verifying online signatures. The main difficulty of signature verification come from: (1) Lacking the enough training samples (2) The methods must be spatial change invariant. To deal with these difficulties and modeling the signatures efficiently, we propose a method that a one-class classifier per each user is built on discriminative features. First, we pre-train a sparse auto-encoder using a large number of unlabeled signatures, then we applied the discriminative features which are learned by auto-encoder to represent the training and testing signatures as a *self-thought learning* method (i.e. we have introduced a *signature descriptor*). Finally, user's signatures are modeled and classified using a one-class classifier. The proposed method is independent on signature datasets thanks to self-taught learning. The features have been learned from 17,500 signatures (ATVS dataset) and verification process of the proposed system is evaluated on SVC2004 and SUSIG datasets, which contain genuine and skilled forgery signatures. The experimental results indicate significant error reduction and accuracy enhancement in comparison with state-of-the-art methods.

Keywords Online Signature Verification · Deep Representation · Self-thought Learning · Auto-encoder

1 Introduction

Authentication, has been known as an intrinsic part of social life. Recent years have seen a growing interest toward personal identity authentication. Increasing security requirements have placed biometrics at the centre of so much attention. Biometric technology has become an important field in verifying people and has been used in people identification and authentication. The term biometric refers to individual recognition based on a person's distinguishing characteristics [1]. In biometric systems, attributes do not have the weaknesses of token-based approaches that can be

Address(es) of author(s) should be given

lost or stolen or knowledge-based approaches that can be forgotten. Therefore, biometric authentication systems have been used in a wide range of applications, such as; banking consumer verification, access control systems, etc.

Generally, the people recognition systems based on biometrics have two main categories [2]

1. *Physiological biometrics* are based on recognizing some physical part of the human body, such as; fingerprint, retina, hand scan, etc.
2. *Behavioural biometrics* are based on measuring some characteristics and behaviours of the human, such as; handwritten signature, voice, etc.

Person recognition has been applied by several biometric modalities, such as; fingerprint [3], iris [4], face [5], vein [6,7] and signature [8,9]. Handwritten signature recognition is one of the most common techniques to recognize the identity of a person. However, when dealing with signatures, most of the proposed systems focus on verification rather than identification because of daily usage of signature verification systems [10].

Recognition refers to two different tasks: identification and verification. Identification specifies which user provides a given biometric parameter among a set of known users. Therefore, the input used for identification only contains genuine data. However, verification determines if the given biometric parameter is provided by a specific known user or is a forgery. In this situation, forgery consist of three types:

1. *Random forgery*: Produced with no knowledge about the signature shape or signer's name.
2. *Simple forgery*: Produced by knowing only the name of the signer.
3. *Skilled forgery*: Produced by looking at the original signature sample.

There are two types of signature verification: Offline (static) and Online (dynamic) verification. In the offline setting, the shape of the signature has been got by capturing or scanning them from papers and the system must extract features from the picture of the signature. Therefore, in offline verification systems, input data contains x, y coordinates of signatures. However, in the online setting, the system uses devices for capturing additional information while the user is signing [11]. Online signatures have extra information for extraction such as; time, pressure, pen up and down, azimuth, etc.

Generally, two types of features can be extracted from a signature [1]:

1. *Function-Features*: The signature is characterized in terms of a time function whose values constitute the feature set, such as; position, velocity, pressure, etc.
2. *Parameter-Features*: The signature is characterized as a vector of elements each representing a value of a feature. Parameters are generally classified into two main categories: local and global. Local features are so-called because of their relation to each point of the signature, such as; height or width ratio of the stroke, stroke orientation, pixel density, etc. Global features are so-called because of their relation to the whole of the signature and signing process, such as; total time, average pressure, average speed, etc.

Using the features explained, the Verification approaches which are used in previous works can be described in three categories [1]:

1. *Template Matching*: A questioned sample has been matched against templates of signatures, such as; Euclidean distance and Dynamic Time Wrapping (DTW) [10, 11, 12].
2. *Statistical*: In this approaches, distance-based classifiers can be considered, such as; Neural Networks [13] and Hidden Markov Models (HMM) [10, 14].
3. *Structural*: This approach is related to structural representations of signatures and compared through graph or tree matching techniques [15].

Recently, *deep learning* provides state-of-the-art results for various biometric systems such as; iris [16], face and fingerprint [8] and finger-vein [7]. In this paper, a signature verification system based on deep learning has been proposed. A sparse linear auto-encoder has been implemented to learn the signature pattern of each user by learning features based on an *unsupervised self-taught* method. This feature learning is done on a large number of unlabelled signatures which are provided in ATVS dataset. As the number of labelled signature samples are limited, so learning the features on labelled ones is not feasible and the self-taught is a good choice for dealing with this restriction. Furthermore, one-class classifier has been used for classifying test signatures. The results of this paper confirm that the learned features are more discriminative rather than state-of-the-art methods where hand-crafted features have been used. As the best of our knowledge, we are the first in introducing a descriptor for verifying the signatures using deep learning. The main contributions of this paper are three folds:

1. We introduce an efficient descriptor for online signatures which can be applied in different datasets. Our experiment confirms our claim that we have achieved state-of-the-art results in two classic benchmarks.
2. We consider an online signature as an image with two channels (one channels is related to time information and another is related to pressure.)
3. We propose an one-class classifier to reject (i.e. detect) the forgery signature as an outlier.

This paper is organized as follows: Section 2, presents previous works have been done in the field of signature verification. Section 3 introduces the adopted methodology for system architecture while section 4 presents the proposed system with details. Experimental results and their comparisons have been described in section 5. Finally, section 6 presents the conclusion for this paper and suggestions for future work.

2 Related Work

Most recent approaches in the field of online signature verification have been described in [1, 2, 17]. The process of signature verification is usually divided into three phases:

2.1 Pre-processing

The signature dataset must take some pre-processes since there is no guarantee that different signatures of one user will always be the same. Several processes have been proposed for this phase, which generally consist of smoothing, rotation and normalization.

Cubic splines can be employed for smoothing purposes to solve the jaggedness in the signatures. Signatures can become rotation-invariant by rotating each one based on orthogonal regression (Eq. 1) [11].

$$\theta = \tan^{-1} \frac{s_y^2 - s_x^2 + \sqrt{(s_y^2 - s_x^2)^2 + 4cov_{x,y}^2}}{2cov_{x,y}} \tag{1}$$

Where s_x and s_y are variance and $cov_{x,y}$ is covariance of the horizontal and vertical components. The signatures of one person must have the same size for better performance. The horizontal and vertical components of the signatures can be normalized to make a standard size (Eq. 2) [12].

$$x_n = \frac{x - \min(x)}{\max(x) - \min(x)} \times 100$$

$$y_n = \frac{y - \min(y)}{\max(y) - \min(y)} \times 100 \tag{2}$$

Where x and y are original and x_n and y_n denote the normalized coordinates.

2.2 Feature Extraction

Feature selection and feature extraction play an important role in verification systems. Many studies have been performed in the field of feature selection to choose the best set of features for extraction. List of common features have been described in Table 1 [11].

Table 1 List of common features

#	Description
1	Coordinate $x(t)$
2	Coordinate $y(t)$
3	Pressure $p(t)$
4	Time stamp
5	Absolute position $r(t) = \sqrt{x^2(t) + y^2(t)}$
6	Velocity in x, $v_x(t)$
7	Velocity in y, $v_y(t)$
8	Absolute velocity $v(t) = \sqrt{v_x^2(t) + v_y^2(t)}$
9	Velocity of $r(t)$, $v_r(t)$
10	Acceleration in x, $a_x(t)$
11	Acceleration in y, $a_y(t)$
12	Absolute acceleration $a(t) = \sqrt{a_x^2(t) + a_y^2(t)}$

Furthermore, some non-common features have been described in other papers [12,15,18,19,20,21,22]. Recently, unlike the traditional biometric authentication systems for face, iris and fingerprint instead of hand-crafted features, some more discriminative features which are provided using deep learning are exploited [8,23].

2.3 Classification

After the feature extraction phase, the system must create a model from reference signatures. For classification phase, each signature must be compared against reference signatures and the difference between features of test signature and reference signatures would be calculated. By calculating the distances between test and reference signatures, the system can decide to accept or reject the test signature.

There are different options for distance calculation such as $d_{min/max}$ which is minimum/maximum distance between a signature and the patterns of the reference set, and $d_{central}$ which is the distance between a signature and the centre of mass of the reference set [24]. One of the important parameter in verification system is the threshold value for accepting or rejecting a signature. Consequently, choosing the best threshold is a crucial step. There are two types of threshold: global and local. In global threshold, the system will choose one threshold value for all users. On the other hand, for local threshold, the system must choose one threshold per user so that, this approach could lead to a better result [24].

As mentioned, the signature recognition problem is an abstract concept, which comprises signature identification and signature verification. In daily usage of authenticating systems such as banking systems, handwritten signature of users have been used to verify the identity of official documents. In these sets of problem, the main goal is verifying whether a signature belongs to one identified person or not. In contrast with multi-class classifiers, the aim for one-class classifiers is distinguishing one type of class (target) from other classes (outlier). Thus, For classifying a signature as genuine or forgery, one-class classifiers have been commonly used [24] to divide the set into two categories: target and outlier (Fig. 1). As it shown, detecting the random forgery ones is very easier rather than skilled forgery ones.

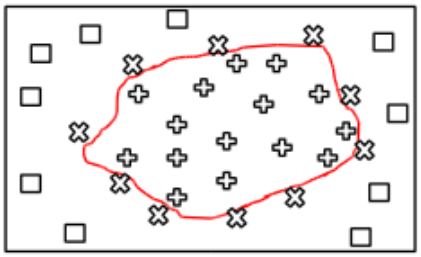

Fig. 1 Example of signature model for each user, $\times$, $+$ and $-$ indicate the skilled forgery, genuine and random forgery signatures. The red is boundary of genuine samples with others

Jain and Gangrade [13] proposed a system by using angle, energy and chain code features to differentiate the signatures. In this approach, a Neural Network has been applied for classification.

Faundez-Zanuy [10] studied four pattern recognition algorithms for online signature recognition: Vector Quantization (VQ), Nearest Neighbour, Dynamic Time Warping (DTW) and Hidden Markov Model (HMM). The author proposed two methods based on VQ and Nearest Neighbour.

Rashidi, et al. [11] evaluated 19 dynamic features viewpoint classification errors and discrimination capability between genuine and forgery signatures. They used a modified distance of DTW for improving performance of verification phase.

Ansari, et al. [12] presented an online signature verification system based on fuzzy modelling. The point of geometric extrema has been chosen for signature segmentation and a minimum distance alignment between samples has been made by DTW techniques. Dynamic features have been converted to a fuzzy model and a user-dependent threshold used for classification.

Barkoula, et al. [15] studied the signatures Turning Angle Sequence (TAS), the Turning Angle Scale Space (TASS) representations, and their application to online signature verification. In the matching stage, the authors have employed a variation of the longest common sub-sequence matching technique.

Yahyatabar, et al. [18] proposed a method based on efficient features defined in persian signatures. A combination of shape based and dynamic extracted features has been applied and a SVM has been used for classification phase.

Alhaddad, et al. [19] explored a new technique by combining back-propagation Neural Network (BPNN) and the probabilistic model. BPNN has been used for local features classification, while probabilistic model has been used to classify global features.

Mohammadi and Faez [20] proposed a method based on the correspondence between important points in the direction of wrap for the time signal provided to maximize the distinction between the genuine and forged signatures.

Napa and Memon [21] Presented a simple and effective method for signature verification in which an online signature is represented with a discriminative feature vector derived from attributes of several histograms that can be computed in linear time. For testing phase, the authors proposed a method on finger drawn signatures on touch devices by collecting a dataset from an uncontrolled environment and over multiple sessions.

Souza, et al. [24] proposed an off-line signature verification system, which uses a combination of five distance measurements, such as; furthest, nearest, template and central using four operations: product, mean, maximum, and minimum as a feature vector.

Fallah, et al. [25] presented a new signature verification system based on Mellin transform. The features have been extracted by Mel Frequency Cepstral Coefficient (MFCC). Neural Network with multi-layer perception architecture and linear classifier in conjunction with Principal Component Analysis (PCA) have used for classification.

Iranmanesh, et al. [26] proposed a verification system by using multi-layer perceptron (MLP) on a subset of PCA features. This approach used a feature selection

method on the information that has been discarded by PCA, which significantly reduced the error rate.

Cpaka, et al. [27] explored a new method by using area partitioning of high and low speed of the signature and high and low pen's pressure. The template for each partition has been generated and by calculating the distance between signatures and template in each partition, a fuzzy classification has been implemented to classify the signatures.

Lopez-Garcia, et al. [28] presented a signature verification system implemented on an embedded system. In this approach, a template for each user has been generated and a DTW algorithm has been used for distance calculation. Finally, the features extracted and passed through a Gaussian Mixture Model (GMM) to calculate the similarity between the test signature and the generated template.

Gruber, et al. [29] proposed a technique based on Longest Common Sub-sequences (LCSS) detection. Authors have used a LCSS kernel of SVM for classifying the similarity of signature time series.

3 Methodology

Deep learning (*Feature Learning* or *Representation Learning*) is a new era of machine learning which aims to learn the high-level features from raw data to achieve a better performance in classification tasks. Deep learning is part of a field of machine learning methods based on learning representation of data [30].

Feature learning tries to learn discriminative features autonomously which is considered as one of its advantages. The other advantage of feature learning process is its capability to be completely unsupervised. One of the goals of deep learning is hierarchical feature extraction. For achieving that goal, feature learning tries to learn a new representation of the input data which is the observed data and continue learning new representations of previously learned features at each level, which are able to reconstruct the original data.

One of the scopes of machine learning, which plays a key role in deep learning, is *self-taught learning*. The main promise of self-taught learning is using unlabeled data in supervised classification tasks [31]. The key point of such algorithms is that unlabeled data are not supposed to follow the same class labels. Indeed, unlabeled data are exploited to teach the system recognizing patterns or relations for the supervised learning task. In summary, self-taught learning learns a concise, higher-level feature representation of the raw data using unlabeled data. Having a concise high level feature representation provides an easier classification task by having features that are more significant [31].

In following of this section, we explain the auto-encoder architecture in 3.1 which is used for learning and extracting sparse and discriminative features of signatures. Furthermore, in 3.2, we explain how convolution and pooling techniques are exploited to the extracted features to become spatial-changing invariant.

3.1 Auto-encoder

Auto-encoder is one of the unsupervised feature learning tools. There is one kind of auto-encoder algorithms, which is based on multi-layer perceptron neural networks. In contrary to traditional neural networks, MLP based auto-encoders are unsupervised learning algorithms which try learning weights of each layer to set the output values to be equal to the inputs for the neural network The structure of auto-encoder is shown in Fig. 2.

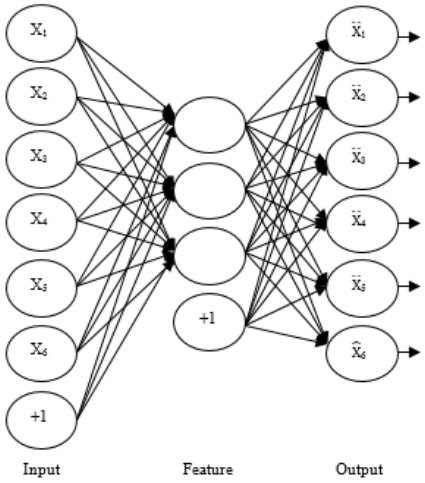

Fig. 2 Architecture of an auto-encoder for learning kernel of convolutional layer

Suppose $x \in \mathbb{R}$ is the set of input features. To learn features from input features, the basic auto-encoder with regularization term to prevent over-fitting, attempts reconstructing input features by minimizing following cost function (Eq. 3):

$$J(w,b) = argmin_{w,b} \frac{1}{m} \sum_{i} \| h_{w,b}(x^{(i)}) - x^{(i)} \|^2 + \lambda \sum_{l} \sum_{i,j} (w_{i,j}^{l})^2 \tag{3}$$

Where $w \in \mathbb{R}$ is weight matrix mapping nodes of each layer to next layer nodes, and $b \in \mathbb{R}$ is a bias vector.

The cost function of auto-encoder mentioned in (Eq. 3) only focuses on the differences between input and output data of auto-encoder. This brings a network with the ability of representing raw data with learned feature without any guarantee of having sparse represented features, which plays a key role in classification task. In order to learn features that are more effective and having a sparser dataset of represented features, the sparsity constraint can impose on the auto-encoder network. The objective

function is as follows (Eq. 4):

$$\begin{aligned} J_{sparse}(w,b) &= J(w,b) + \beta \sum_i KL(\rho \parallel \hat{\rho}_j) \\ KL(\rho \parallel \hat{\rho}_j) &= \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \\ \hat{\rho}_j &= \frac{1}{m} \sum_i [a_j^2(x^{(i)})] \end{aligned} \tag{4}$$

Where $KL(\rho \parallel \hat{\rho}_j)$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$, which is the average activation of hidden unit j. The notation summary of Eq. 4 is described in Table 2.

Table 2 Auto-encoder cost function notation summary

Symbol	Description
x	Input features for a training example
y	Output/Target values. y is a vector. In the case of an auto-encoder, $y = x$
$x^{(i)}$	The i-th training example
w	The parameter associated with the connection between units of layers
b	The bias term associated with the connection between two layers
ρ	Sparsity parameter, which specifies the desired level of sparsity
$\hat{\rho}_j$	The average activation of hidden unit i (in the sparse auto-encoder)
β	Weight of the sparsity penalty term (in the sparse auto-encoder objective)
λ	Weight decay parameter

A *sparse auto-encoder* model can effectively realize feature extraction and dimension reduction of the input data, which play a vital role in classification tasks [23].

3.2 Convolution and Pooling

Raw input data are usually stationary. In other words, randomly selected parts of the data have the same statistics. This characteristic shows that not all the features are useful. It is obvious that having more features results in increasing the computational complexity especially in a classification task. In order to avoid high complexity, redundant data have been neglected by picking up random patches of raw data and convolving them. After obtaining convolved features, pooling method can be exploited in order to obtain pooled convolved features (Fig. 3).

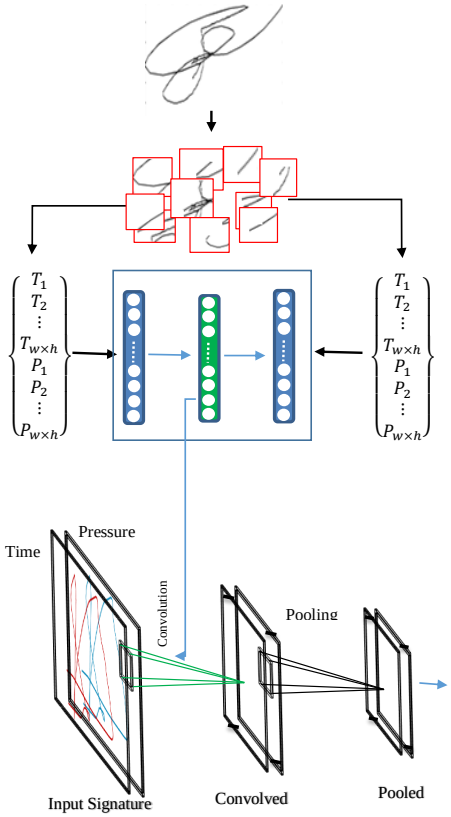

Fig. 3 *Top*: Learning features from ATCV dataset. *Bottom*: Convolving and plooling the learned feature for representing the signatures

4 Proposed System

One of the important problems in signature verification is choosing features due to diverse difficulties in signature verification, such as; differences between same user signatures, different circumstances of signing, various shapes of signatures, etc. Among these, exploiting an unsupervised feature learning method results in system compatibility improvement with various types of signatures and automatic feature selecting from signatures. To achieve more discriminative features, each signature has been considered as an image with two channels where the intensity of these channels are the pressure and time of each position of signature. First, an efficient descriptor has been learned for describing the signatures. This procedure is done based on self-thought learning using ATVS dataset. Then, for each user the training samples have been described using the learned features and a reference model has been fitted on the training user's signatures. In test phase and for verifying a signature, it has been checked with all models. If it has not be fitted to any of defence models of users, the

signature is labelled as forgery, otherwise it is labelled as genuine for a specific user. Fig. 4 shows an overview of proposed approach for verifying the user's signatures.

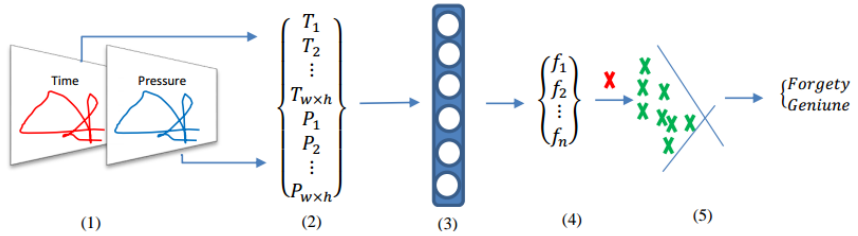

Fig. 4 Work-flow of signature verification. Left to right: (1) A test signature which is represented as an image with two channels: pressure and time. (2) Reshaping the input sample to vector. (3) The vector is represented by an auto-encoder where is pre-trained on ATVS dataset. (4) The input signature is described by n features. It is the output of auto-encoder (5) Classify the sample using one-class classifier which is learned by training samples of each user. (6) Final decision

The proposed signature verification system comprises two main steps:

Step 1:Learning a signature descriptor
First step consists of creating signature descriptor based on self-tough learning. All signature samples in ATVS dataset have divided to 17500×64 Patches with the size of 10×10 and given to a sparse auto-encoder. After the auto-encoder is completely learned, it can be used as an efficient feature extractor from the signatures patches.

Step 2:Creating references models for users
In the second step, all signatures have divided into small patches and they have described by the learned descriptor. The descriptions of small patches have pooled and the mean of them (i. e. mean of the features which are extracted from each sample) have considered as the descriptions of the signatures. This procedure has done by considering the learned descriptor as a filter and convolve it on the signature and operating a mean pooling on the result if convolution. Based on the explained procedure, all training signatures for each user have been described and a reference signature model for each user has been created. These models are considered as a set of one-class classifiers.

As described, In the first step, features are learned by the auto-encoder. In this step, an unlabelled dataset, which is discretized from train and test datasets, is used based on self-taught method. In the next step, a reference model of the system is built using classified represented data from user's reference signatures. These two steps are parts of the system training phase [21]. Finally, in verification phase, which is system-working section, new unknown signatures are compared against the system reference model (classified data) to be verified. There are three principal parts among described steps, which are pre-processing, feature learning using auto-encoder, and classification. These parts are explained as follows:

4.1 Pre-processing

As mentioned, in the pre-processing phase, normalizing size of the signature is the first step. This aim can be achieved by scaling the signature size. At the next step, the mean of the data must become equal to zero for data normalization.

Signatures data in databases are based on time, pressure, pen up/down, etc. in x, y positions. To make representation become similar to reality, points of signatures have been continued. This object achieved by using time of the points to observe the sequence of data and pen up/down to check if the pen has gone up, the point must be separated from the next one. Finally, signatures have been represented base on two layer: pressure and time (Fig. 5).

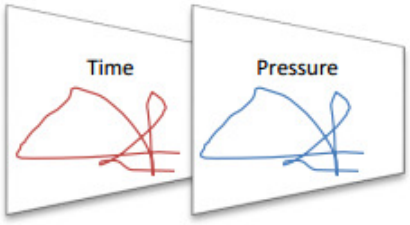

Fig. 5 Illustration of input signature

Principal Component Analysis (PCA) is an algorithm that reduces dimensions of signature data and can be used to significantly speed up unsupervised feature learning algorithm. Since the system is trained based on signature images, adjacent pixel values are highly correlated. Whitening can make the input less redundant, the features become less correlated with each other and all become the same variance. Therefore, these two algorithms have been used to reduce the dimension.

4.2 Feature Learning using Auto-encoder

For learning features from signatures, a linear auto-encoder with sparsity have been used. The signature has been set for input and output and auto-encoder has been checked to maps input to output. This auto-encoder has been designed based on gradient descent.

Unsupervised learning algorithms have high computational cost. In order to increase performance of learning phase, raw data (large patch of a signature) has been divided into small patches and have been used in feature learning phase as input. Then, learned features have been convolved with large patch. After obtaining features using convolution, mean pooling method has been exploited in order to obtain pooled convolved features. These pooled features have been used for classification.

4.3 Model creation and classification

The significant issues of classification in this type of problems are differences between same user's signatures, diverse circumstances of signing, low amount of signature samples, and forgery signatures. For resolving such issues, selecting an appropriate classifier is very important.

The one-class classifier in the proposed system has a target class, which is class of the user whose signature is being compared with input signature, and the outlier class is other user's sample signatures. As a result, the classifier must create a model of target class for each user.

5 Experimental Results

In the evaluation process of proposed approach, test signatures have been comprised by comparing their features against reference signatures. In this section, short description of benchmarks and evaluation parameters have been described. In addition, two main steps of the experiments have been explained.

5.1 Benchmarks

For evaluation of the proposed approach, three public datasets have been used which are *SVC2004* [32], *SUSIG* [33] and *ATVS* [34,35]. The structure of the mentioned datasets have been explained as follows:

5.1.1 SVC2004

This is the first international signature verification competition[1]. The aim of holding SVC2004 competition was allowing researchers to evaluate the performance of their signature verification methods based on benchmark datasets. In this regard benchmarking rules that resulted in creating a benchmark dataset named SVC2004.

SVC2004 main database has 100 sets of signature data. SVC2004 public database, which has been released before the competition, consists of 40 signature sets. Each set includes 20 genuine signatures of one signature contributor and 20 skilled forgeries of at least four other contributors (Fig. 6).

[1] Available at http://www.cse.ust.hk/svc2004/download.html

Fig. 6 Examples of Genuine (first row) and Forgery (second row) signatures in SVC2004 database

In data collection process of the signature sets, contributors were asked not to use their real signatures for privacy reasons. On the other hand, made-up signatures are shortcoming of this database, which will result in having higher variance and higher error rates. For decreasing effect of the mentioned problem, contributors were reminded that, not only should their signatures have spatial consistency in signature shape but also should have temporal consistency of dynamic features as well. Contributors were asked to contribute 20 genuine signatures in two sessions during two weeks. At least four other contributors forged the skilled forgeries for each contributor's signature.

In SVS2004 database, each signature includes a sequence of points, which contains x, y coordinates, time and pen up/down, azimuth, altitude and pressure.

5.1.2 SUSIG

Sabanci University Signature database (SUSIG)[2] is a database of online signatures, which aim is overcoming some of the shortcomings of its contemporary databases.

The SUSIG database consist of two sub-corpora, which are visual and blind. In both sub-corpora, contributors used their real signatures for creating genuine signatures sets, which is one of this database advantages in contrary to SVC2004 database (Fig. 7).

[2] Available at http://biometrics.sabanciuniv.edu/susig.html

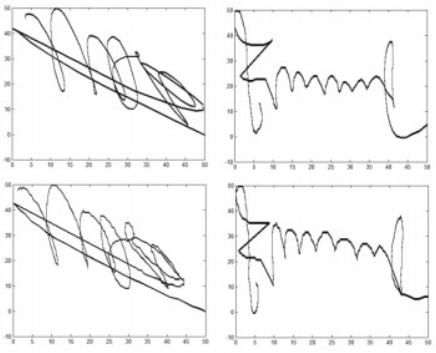

Fig. 7 Examples of Genuine (first row) and Forgery (second row) signatures in SUSIG database

In blind sub-corpus data collection process, collection has been done on a tablet without visual feedback. It consists of signatures of 100 contributors. First group of 30 contributors provided eight genuine signatures, while the other 70 contributors provided 10 genuine signatures each. For providing forgery signatures, forgers were shown the genuine signature's drawing replay several times. After training well, forgers supplied ten forgeries for each set of contributor's genuine signatures. Additionally, there is a separate ten-person validation set with ten genuine and ten forged signatures per person.

In visual sub-corpus data collection process, collection has been done on a tablet with a LCD, which provided visual feedback to the contributors while they were signing signatures. Visual sub-corpus data were collected in two separate sessions. Each contributor has provided 20 samples of his/her signature. In this database, in the visual sub-corpus, there are two types of forgery signatures: skilled and highly skilled forgeries. For providing skilled forgeries, contributors were shown the genuine signature's drawing like the blind sub-corpus. Each forger was asked to provide five forgeries of the signature. For providing highly skilled forgeries, the replay of the reference signature shown on both a monitor in front of forgers and the LCD screen of the tablet which provided forgers the ability of tracing the reference signature signing process. Like the normal skilled ones, forgers have forged five highly skilled forgeries for each set of genuine signatures. In summary, 20 genuine signatures, five skilled and five highly skilled forgeries were collected for each person in the sub-corpus. Additionally, there is a separate 10 person-validation set with 10 genuine and 10 forged signatures per person acquired in a single session for tuning system parameters.

5.1.3 ATVS

All two mentioned databases (SVC2004 and SUSIG) are human made database. Although they have advantages, such as; having real signature of a human, and having real world situations for sampling, they have restrictions, including limited amount of data, privacy issues, subdued to legal aspects. Synthetic signature databases are solu-

tion of this problem. They are not restricted to limitations mentioned above. However, they miss the advantage of having real world situations, and real human signature. In spite of suffering from such problems, synthetic databases have had good approaches to simulation of real signatures, which involves the effect of real situation of sampling. ATVS database[3] is one of the synthetic databases (Fig. 8).

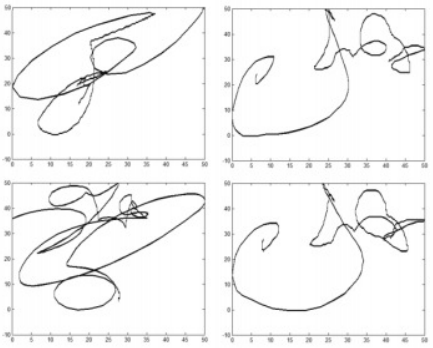

Fig. 8 Examples of Genuine (first row) and Forgery (second row) signatures in ATVS database

The artificial samples produced by ATVS follow the pattern of the western signatures, which are left-to-right concatenated handwritten signature [34]. ATVS signature generation contains two steps: First, a master signature corresponding to a synthetic individual is produced using a generative model based on information obtained. This information has been acquired by analysing real signatures using a spectral analysis approach and the kinematic theory of rapid human movements. Second, various samples of the same synthetic subject are created using the master signature.

ATVS has two parts, named "direct modification of the time functions" and "modification of the sigma-log-normal parameters (LN-Parameters)".

In direct modification of the time functions, time sequence of the reference signature have been modified according to a model simulating the distortions introduced by a given channel.

In Modification of the Sigma-Log-normal Parameters, researchers have decomposed the velocity function v, derived from the coordinate functions x and y, into simple strokes. Each stroke is used with different velocity functions to set the Sigma-Log-normal parameters.

In summary, ATVS especially the ATVS-SSig have two types of data. In Modification time functions, the time functions of the master signature is changed to generate the duplicated samples, as described previously [27]. In modification LN-Parameters, duplicated samples are generated modifying the log-normal parameters of the master. Both methods use 25 signatures from 350 users. Of the 25 signatures, the first five follow an intra-session variability and the next 20 follow an inter-session variability.

[3] Available at http://atvs.ii.uam.es/databases.jsp

5.2 Evaluation Parameters

Different parameters have been used in verification systems. In the following, a short description of most commonly used parameters have been summarized.

1. **Receiver Operating Characteristic (ROC) Curve**:A one-class classifier can be evaluated based on small fraction false negative (false reject rate) and false positive (false accept rate). ROC curve shows how the fraction false positive varies for varying fraction false negative. Traditionally the fraction true positive is plotted versus the fraction false positive. The smaller these fractions are, the more this one-class classifier is to be preferred.
2. **Equal Error rate (EER)**:If a line connects the points (1, 0) and (0, 1) in the ROC curve of a classifier, EER can be defined such that false positive and false negative fractions are equal. This parameter is a simple way to compare system accuracies. The smaller the EER rate is, the more accurate the system is.
3. **Area Under the ROC Curve (AUC)**:AUC is one way to summarize an ROC curve in a single number. This integrates the fraction true positive over varying thresholds (or equivalently, varying fraction false positive). Higher values indicate a better separation between target and outlier objects.

5.3 Feature Learning

In feature learning phase, a methodology has been set to learn features based on signatures except of test and train sets. Therefore, all of the signatures in ATVS database have been used for feature learning using auto-encoder. The size of hidden layer and iteration value of auto encoder have been selected based on an experiment on auto-encoder with hidden size of 500, 1000, 1500, 2000, 2500 and 3000 nodes in which the iteration value was set from 100 to 700. Finally, based on experimental results that described in the next subsection, the auto-encoder comprises one hidden layer with 2000 nodes and the limited BroydenFletcherGoldfarbShanno algorithm (L-BFGS) method with 700 iteration for minimization function has been chosen.

5.4 Classification and Verification

In this phase, SVC2004 and SUSIG databases have been used for a K-Fold Cross-Validation process that has been implemented to categorize train and test signature groups. Several experiments have been done to achieve the best values for system parameters. EER and AUC results for SVC2004 and SUSIG databases have been shown in Table 3 and Table 4, respectively.

Table 3 EER Experiment results with different hidden size for SVC2004 and SUSIG

Iteration	100		200		300		400		500		600		700	
Hidden size	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG
500	1.7	5.02	1.65	4.94	1.60	4.77	1.60	4.74	1.55	4.57	1.45	4.07	1.03	3.23
1000	1.25	4.90	1.14	4.57	1.14	4.24	1.08	3.72	1.06	3.72	1.03	3.51	1.03	2.70
1500	1.25	3.06	1.20	2.91	1.20	2.87	1.15	2.87	1.15	2.56	1.10	2.53	1.05	2.40
2000	**1.00**	**2.00**	**0.93**	**1.98**	0.92	**1.75**	0.90	**1.51**	0.88	**1.26**	0.85	**1.02**	0.83	**0.77**
2500	1.05	2.56	1.00	2.52	**0.90**	2.39	**0.89**	2.36	**0.80**	2.32	**0.77**	2.20	**0.73**	2.15
3000	1.03	2.67	1.01	2.57	0.98	2.52	0.96	2.41	0.88	2.34	0.88	2.16	0.78	2.05

Table 4 AUC Experiment results with different hidden size for SVC2004 and SUSIG

Iteration	100		200		300		400		500		600		700	
Hidden size	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG
500	99.1	98.0	99.2	98.0	99.2	98.0	99.2	98.1	99.3	98.2	99.3	98.2	99.3	98.8
1000	99.3	98.1	99.3	98.3	99.4	98.4	99.4	98.6	99.4	98.6	99.4	98.7	99.4	99.0
1500	99.3	98.9	99.3	98.9	99.4	98.9	99.4	98.9	99.4	99.0	99.5	99.0	99.5	99.1
2000	**99.5**	**99.2**	**99.5**	**99.2**	**99.5**	**99.2**	**99.5**	**99.3**	**99.5**	**99.3**	**99.6**	**99.4**	**99.6**	**99.5**
2500	99.4	98.9	**99.5**	99.0	**99.5**	99.1	**99.5**	99.1	**99.5**	99.1	**99.6**	99.1	**99.6**	99.2
3000	**99.5**	99.0	**99.5**	99.0	**99.5**	99.0	**99.5**	99.1	**99.5**	99.1	99.5	99.1	**99.6**	99.2

The results shown a decrement in EER and an increment in AUC rate while facing iteration value increment. Due to change mitigation in more than 700 iterations, the iteration value has been set to 700. Although for hidden size parameter, the rate of EER enhancement and AUC rates decreased for hidden sizes larger than 2000 while computational costs increased and had been prone to over fitting and curse of dimensionality. Finally, the size of 2000 has been selected because of its computational efficiency and appropriate accuracy.

For better evaluation of the system, the performance of the classification and verification have been tested with different percentages of each user's genuine signatures used as training data to create each user's genuine signature model. The EER and AUC of this experiment have been illustrated in Figure 9-left and Figure 9-Right, respectively. The graphs in figure 9-left show the EER improvement during the increase of the size of genuine samples signatures used for training. It can be seen that EER decrement rate from about the percent 10 up to percent 20 is high due to the lack of genuine signatures sample for creating the model. But after this point, the smoother EER decrement rate can be seen due to the correlation of the genuine signatures samples. In other word, the classifier has seen enough samples to create a unique model for each user. This behaviour is somehow obvious in the figure Figure 9-Right which shows the AUC of the system tested with different percentages of user's genuine signatures. It can be concluded that after the percentage of 30, the increment rate of AUC graph decreases and becomes smoother.

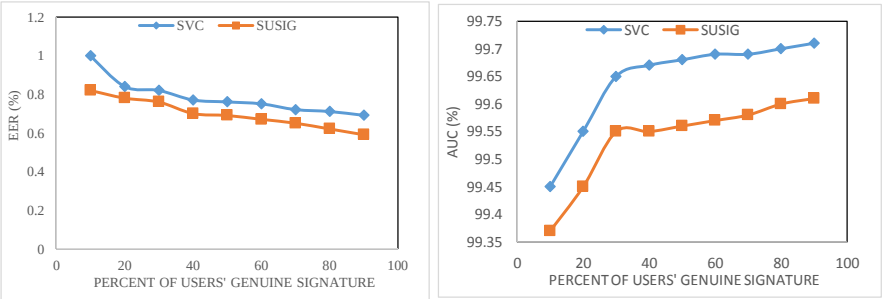

Fig. 9 *Left:*Average EER vs. percent of user's genuine signatures used as training data. *Right:*Average AUC vs. percent of user's genuine signatures used as training data

As a comparison between the proposed system and other approaches, verification protocols must be similar. Based on random and skilled forgery verification protocol [32,33], 25 percent of each user's genuine signatures have been used for training to create the user model. The remaining 75 percent of user's genuine signatures, all of the skilled forgery signatures of his/her and all of the genuine signatures of other users have been used for testing based on a local threshold for each user. For evaluating the proposed method, multiple classifiers have been tested based on author's previous work [9]. These classifiers are available in Matlab open source Data Description toolbox[4] (dd_tools). This toolbox has the ability of obtaining optimal coefficients for classifiers. Finally, based on achieved experimental results, Gaussian classifier has been used. The results of proposed method in comparison with state-of-the-art methods for two standard benchmarks (SVC2004 and SUSIG) are shown in Table 5 and Table 6, respectively.

Table 5 Different online signature verification methods for SVC2004

Method	EER (%)
Gruber, et al. [29]	6.84
Mohammadi and Faez [20]	6.33
Barkoula, et al. [15]	5.33
Yahyatabar, et al. [18]	4.58
Yeung, et al. [32]	2.89
Ansari, et al. [12]	1.65
Fayyaz, et al. [9]	2.15
Proposed Method	**0.83**

[4] Available at http://www.prtools.org

Table 6 Different online signature verification methods for SUSIG

Method	EER (%)
Khalil, et al. [36]	3.06
Napa and Memon [21]	2.91
Kholmatov and Yanikoglu [33]	2.10
Ibrahim, et al. [37]	1.59
Ansari, et al. [12]	1.23
Proposed Method	**0.77**

These tables indicate that proposed method have the best performance in comparison with competing algorithms. This method's EER on SVC2004 dataset is 0.83 percent, where the next best method is 1.65 percent reported for the method Ansari, et al. [12]. This verification system is 0.82 percent superior to the otherwise best result. On SUSIG benchmark, implemented method's EER is equal to 0.77 percent as it is 0.46 percent superior to the next best method.

Table 5 and table 6 illustrate that in contrast to all reported methods, the results on two datasets are very close (0.06 percent difference in EER). This similarity is indicated that proposed method is dataset invariant.

6 Conclusions And Future Work

In this paper, a new approach has been introduced based on self-thought learning to verify the signatures. As it can be inferred from experimental results and inherited properties of self-thought learning, the proposed system is independent from specific benchmarks, which means that it is signature shape invariant.

The features, which are used to verify the signatures, have been learned from ATVS dataset by using a sparse auto-encoder with one hidden layer. By applying convolution and pooling methods, system has achieved pooled convolved features to verify the signatures. In addition, one-class classifier has been applied as it models the signatures of each user.

To compare with similar works, two standard benchmarks have been used which are named as SVC2004 and SUSIG datasets. Our results have shown superiority on both datasets. The features have been used in this paper can be used in other benchmarks, as this is the main component of the method proposed in this paper.

his method has proved its ability to learn the best set of features in problems that need to define hand-crafted features. Therefore, it can be used in a wide range of machine learning problems. As a future work, this method can be tested on offline signatures. In addition, the impact of deep convolutional networks can be tested on both online and offline signature datasets.

References

1. D. Impedovo and G. Pirlo, "Automatic Signature Verification: The State of the Art", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, pp. 609-635, (2008)

2. D. Impedovo, G. Pirlo and R. Plamondon, "Handwritten Signature Verification: New Advancements and Open Issues", 2012 International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 367-372, (2012)
3. H. Hasan and S. Abdul-Kareem, "Fingerprint image enhancement and recognition algorithms: a survey", Neural Computing and Applications, vol. 23, pp. 1605-1610, (2013)
4. K. Bowyer, K. Hollingsworth, and P. Flynn, "A Survey of Iris Biometrics Research: 20082010", Handbook of Iris Recognition, pp. 15-54, Springer, London, (2013)
5. J. M. Pandya, D. Rathod and J. J. Jadav, "A survey of face recognition approach", International Journal of Engineering Research and Applications (IJERA), vol. 3, pp. 632-635, (2013)
6. K.-S. Wu, J.-C. Lee, T.-M. Lo, K.-C. Chang and C.-P. Chang, "A secure palm vein recognition system", Journal of Systems and Software, vol. 86, pp. 2870-2876, (2013)
7. M. Fayyaz, M. Hajizadeh, M. Sabokrou, M. Hoseini and M. Fathy, "A Novel Approach For Finger Vein Verification Based on Self-Taught Learning", 9th Iranian Conference on Machine Vision and Image Processing (MVIP), Tehran, Iran, (2015)
8. D. Menotti, G. Chiachia, A. Pinto, W. Schwartz, H. Pedrini and A. Falcao, "Deep Representations for Iris, Face, and Fingerprint Spoofing Detection", IEEE Transactions on Information Forensics and Security, vol. 10, (2015)
9. M. Fayyaz, M. Hajizadeh, M. Sabokrou, M. Hoseini and M. Fathy, "Online Signature Verification Based on Feature Representation", International Symposium on Artificial Intelligence and Signal Processing, Mashhad, Iran, (2015)
10. M. Faundez-Zanuy, "On-line signature recognition based on VQ-DTW", Pattern Recognition, vol. 40, pp. 981-992, (2007)
11. S. Rashidi, A. Fallah and F. Towhidkhah, "Authentication based on signature verification using position, velocity, acceleration and Jerk signals", 9th International ISC Conference on Information Security and Cryptology (ISCISC), pp. 26-31, (2012)
12. A. Q. Ansari, M. Hanmandlu, J. Kour and A. K. Singh, "Online signature verification using segment-level fuzzy modelling", IET Biometrics, vol. 3, pp. 113-127, (2014)
13. P. Jain and J. Gangrade, "Online Signature Verification Using Energy, Angle and Directional Gradient Feature with Neural Network", International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, pp. 211-216, (2014)
14. J. Fierrez, J. Ortega-Garcia, D. Ramos and J. Gonzalez-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modeling", Pattern Recognition Letters, vol. 28, pp. 2325-2334, (2007)
15. K. Barkoula, G. Economou and S. Fotopoulos, "Online signature verification based on signatures turning angle representation using longest common subsequence matching", International Journal on Document Analysis and Recognition (IJDAR), vol. 16, pp. 261-272, (2013)
16. N. Liu, M. Zhang, H. Li, Z. Sun and T. Tan, "DeepIris: Learning Pairwise Filter Bank for Heterogeneous Iris Verification", Pattern Recognition Letters, 2015
17. Z. Zhang, K. Wang and Y. Wang, "A Survey of On-line Signature Verification", Biometric Recognition. vol. 7098, pp. 141-149, (2011)
18. M. E. Yahyatabar, Y. Baleghi and M. R. Karami, "Online signature verification: A Persian-language specific approach", 21st Iranian Conference on Electrical Engineering (ICEE), pp. 1-6, (2013)
19. M. J. Alhaddad, D. Mohamad and A. M. Ahsan, "Online Signature Verification Using Probablistic Modeling and Neural Network", Spring Congress on Engineering and Technology (S-CET), pp. 1-5, (2012)
20. M. H. Mohammadi and K. Faez, "Matching between Important Points using Dynamic Time Warping for Online Signature Verification", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Bioinformatics (JBIO), (2012)
21. S.-B. Napa and N. Memon, "Online Signature Verification on Mobile Devices", IEEE Transactions on Information Forensics and Security, vol. 9, pp. 933-947, (2014)
22. A. Reza, H. Lim and M. Alam, "An Efficient Online Signature Verification Scheme Using Dynamic Programming of String Matching", Convergence and Hybrid Information Technology, pp. 590-597, (2011)
23. R. Wang, C. Han, Y. Wu and T. Guo, "Fingerprint Classification Based on Depth Neural Network", The Computing Research Repository (CoRR), (2014)
24. M. R. P. Souza, G. D. C. Cavalcanti and R. Tsang Ing, "Off-line Signature Verification: An Approach Based on Combining Distances and One-class Classifiers", 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 7-11, (2011)

25. A. Fallah, M. Jamaati and A. Soleamani, "A new online signature verification system based on combining Mellin transform, MFCC and neural network", Digital Signal Processing, vol. 21, pp. 404-416, (2011)
26. V. Iranmanesh, S. M. S. Ahmad, W. A. W. Adnan, S. Yussof, O. A. Arigbabu and F. L. Malallah, "Online Handwritten Signature Verification Using Neural Network Classifier Based on Principal Component Analysis", The Scientific World Journal, (2014)
27. K. Cpaka, M. Zalasiski and L. Rutkowski, "New method for the on-line signature verification based on horizontal partitioning", Pattern Recognition, vol. 47, pp. 26522661, (2014)
28. M. Lopez-Garcia, R. Ramos-Lara, O. Miguel-Hurtado and E. Canto-Navarro, "Embedded System for Biometric Online Signature Verification", IEEE Transactions on Industrial Informatics, vol. 10, pp. 491-501, (2014)
29. C. Gruber, T. Gruber, S. Krinninger and B. Sick, "Online Signature Verification With Support Vector Machines Based on LCSS Kernel Functions", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 40, pp. 1088-1100, (2010)
30. H. Song and S.-Y. Lee, "Hierarchical Representation Using NMF", Neural Information Processing. vol. 8226, pp. 466-473, (2013)
31. R. Raina, A. Battle, H. Lee, B. Packer and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data", 24th international conference on Machine learning, Corvalis, Oregon, USA, (2007)
32. D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi and T. Matsumoto, "SVC2004: First International Signature Verification Competition", Biometric Authentication. vol. 3072, pp. 16-22, (2004)
33. A. Kholmatov and B. Yanikoglu, "SUSIG: an on-line signature database, associated protocols and benchmark results", Pattern Analysis and Applications, vol. 12, pp. 227-236, (2009)
34. J. Galbally, j. Plamondon, J. Fierrez and J. Ortega-Garcia, "Synthetic on-line signature generation. Part I: Methodology and algorithms", Pattern Recognition, vol. 45, pp. 2610-2621, (2012)
35. J. Galbally, J. Fierrez, J. Ortega-Garcia and j. Plamondon, "Synthetic on-line signature generation. Part II: Experimental validation", Pattern Recognition, vol. 45, pp. 2622-2632, (2012)
36. M. I. Khalil, M. Moustafa and H. M. Abbas, "Enhanced DTW based on-line signature verification", 16th IEEE International Conference on Image Processing (ICIP), pp. 2713-2716, (2009)
37. M. T. Ibrahim, M. Kyan and G. Ling, "On-line signature verification using global features", Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 682-685, (2009)