Parsimonious Coding and Verification of Offline Handwritten Signatures

Elias N. Zois
Athens Technological & Educational Inst.
Agiou Spiridonos Str., 12243 Egaleo, Greece
ezois@teiath.gr

Ilias Theodorakopoulos, Dimitrios Tsourounis
and George Economou
University of Patras
Rio, 26504, Greece
http://www.upcv.upatras.gr

Abstract

A common practice for addressing the problem of verifying the presence, or the consent of a person in many transactions is to utilize the handwritten signature. Among others, the offline or static signature is a valuable tool in forensic related studies. Thus, the importance of verifying static handwritten signatures still poses a challenging task. Throughout the literature, gray-level images, composed of handwritten signature traces are subjected to numerous processing stages; their outcome is the mapping of any input signature image in a so-called corresponding feature space. Pattern recognition techniques utilize this feature space, usually as a binary verification problem. In this work, sparse dictionary learning and coding are for the first time employed as a means to provide a feature space for offline signature verification, which intuitively adapts to a small set of randomly selected genuine reference samples, thus making it attractable for forensic cases. In this context, the K-SVD dictionary learning algorithm is employed in order to create a writer oriented lexicon. For any signature sample, sparse representation with the use of the writer's lexicon and the Orthogonal Matching Pursuit algorithm generates a weight matrix; features are then extracted by applying simple average pooling to the generated sparse codes. The performance of the proposed scheme is demonstrated using the popular CEDAR, MCYT75 and GPDS300 signature datasets, delivering state of the art results.

1. Introduction

In spite of the technological advancements of our digital era which are applicable to numerous smart city applications, the most familiar manner for validating the identity of a person in a wide range of daily transactions, utilizes the handwritten signature [1]. This behavioral trait is considered to be the outcome of the joint interaction between a person's specific motoric procedure and his/hers taught scripting customs. Typically any acquisition methods of the handwritten signature can be broadly divided into two major categories [2]: In the online case, the characteristics are captured in real time using specialized acquisition devices, while in the offline or static case the formed signature trace is being depicted with a digital image, as a result of a scanning procedure. Offline signature verification systems (SV's) which rely solely on motionless images have been reported to perform poorer when compared to dynamic systems; however their use may be inevitably required in several occasions such as forensic applications [3]. In their effort to provide efficient SV',s researchers have considered a plethora of feature extraction methods, i.e. ways that transform any input image into a multidimensional vector. Given a number of reference signature samples, it is anticipated that a reliable feature space will retain a number of significant characteristics of the signing process by means of forming compact clusters. A considerable number of feature extraction methods rely on the evaluation of global and/or local signature descriptors [2]. Throughout the years, particularly during the last decade, a diversity of methods which employ feature extraction for offline SV has been presented with notable results for verification tasks; examples are Granulometric Size Distributions [4], Extended Shadow Code (ESC) and Directional Probabilistic Density Functions (DPDF) [5, 6], variations of Local Binary Patterns (LBP) [7, 8, 9] Histogram of Oriented Gradients (HOG) [10, 11], surroundness [12], curvelet transform [13], Directional Code Co-occurrence Matrix (DCCM) [14], Speeded Up Robust Features (SURF) [15, 16], partially ordered sets [17], Bag-Of-Visual Words [18], Slant [19], Discrete Radon Transform (DRT) [11, 20], Modified Direction Feature (MDF) [21], Optical Flow [22], Graphometrics [23], Scale Invariant Feature Transform (SIFT) [24], Morphology [25]. Another distinction between SV's is whether they are writer dependent (WD) or independent (WI) with the majority of SV's being WD's i.e. for each writer a personal and dedicated model is created [26]; recently mixed implementations have been also presented [6, 27] with notable results.

The contribution of this work is to propose a new and novel parsimonious modeling of the static handwritten signatures. Parsimony, a biologically inspired notion has

been exploited lately in a plethora of pattern recognition, machine learning, and computer vision applications. As an example, sparse representation (SR) provides a compact signal representation, and offers better compression performance compared to methods based on orthonormal transforms. Sparse representation has been the object of scientific interest for quite a long time [28, 29, 30]. Its aim is to provide a parsimonious representation of a signal by means of a linear combination of only a few atoms, which are members of an overcomplete set or dictionary. Sparse representation has been found extremely useful in scientific disciplines which among others include face recognition, image super-resolution and denoising [29].

In this work, the problem of writer-dependent signature verification is addressed [27], in which one model is being built for each writer, based on dictionary learning algorithms. This methodology utilizes a feature vector, constructed by average pooling of the corresponding sparse-representation coefficients, delivering state-of-the-art performance in terms of verification error. In addition, the proposed method employs only genuine samples for creating the dictionaries. To the best of author's knowledge, this is the first work in the literature that exploits sparse representations for offline signature verification. In [31] the authors present a method for writer identification based on sparse representation of handwritten structural primitives, called graphemes or fraglets, while in [32] the authors propose a novel online signature verification technique based on discrete cosine transform (DCT) and sparse representation. This is clearly not our case since a) The proposed method exploits static-offline signature samples and generates the characteristic feature space of a person with the use of a limited number of only genuine samples via the K-SVD algorithm [29], b) the dictionary elements are learned and not hand-crafted like i.e. DCT, and c) we perform signature verification and not writer identification. In the proposed method there is no need for any kind of predetermined feature extraction analysis which corresponds to any fixed image analysis models; the signal patches used are simply one dimensional rearranged sets of overlapping local 5 5 neighborhoods of the signature pixels. Then, given a specific writer dictionary all questionable signature samples, authentic or simulated, are represented with the use of this specific dictionary by means of the Orthogonal Matching Pursuit algorithm (OMP). An important issue of the proposed method is that it actually requires a limited set of genuine signatures for dictionary learning; consequently the extracted features are adapted to those samples only, something that can be useful in cases of forensic interest. Dictionary learning takes place over the entire signature image plane.

The training procedure employs only genuine and random forgery samples since it has been stated that the use of simulated forgeries for training SV's is intractable [5, 27]. The computation of different writer's dictionaries is performed only for the genuine class of each writer, since usually the properties of the simulated forgery class are not available during the training stage. Evaluation of the testing stage is carried out with the use of the of the equal error rate (EER) between the false acceptance rate of simulated forgeries (FAR(S)), and false rejection rate of the genuine forgeries (FRR), with the use of the receiver operating characteristic (ROC) curves. Experiments with three well-known datasets namely CEDAR, MCYT and GPDS300 have been carried out in order to provide the corresponding verification metrics.

This work is organized as follows: Section 2 provides an overview of the proposed approach. Section 3 gives a brief description of sparse representation and K-SVD. Section 4 describes the feature extraction method while section 5 discusses the verification protocol and the experimental results. Finally, section 6 draws the conclusions.

2. System architecture

Figure 1, presents the overall system architecture. In the proposed WD approach, during the enrollment stage and for one specific writer, a population N_{Gen} of some genuine reference signature samples R_{Gi}^{writer}, $i=1:N_{Gen}$ is enrolled along with a number of random forgeries R_{RF}^{writer}, $i=1:N_{RF}$ in order to create the reference signature dataset $RSD^{writer}=\{R_{Gi}^{writer},R_{RF}^{writer}\}$ of size $(N_{Gen}+N_{RF})$. After typical preprocessing steps which include thresholding and thinning, the K-SVD dictionary learning algorithm is sequentially activated by each one of the

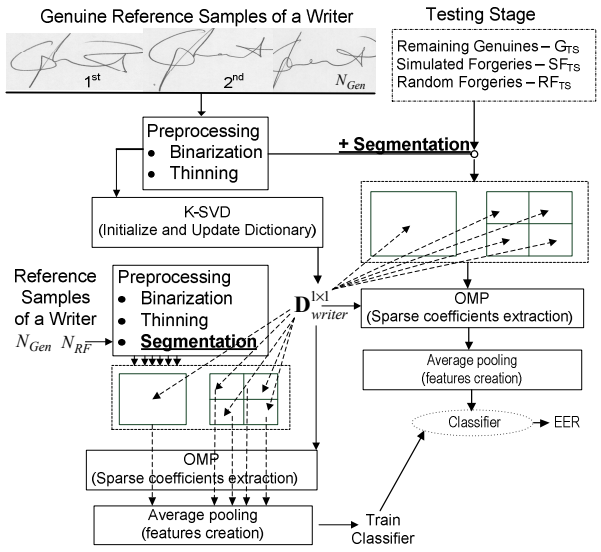

Figure 1: Overview of the proposed system.

writer's genuine reference signatures R_{Gi}^{writer} in order to initially create and further update the writer's individual characteristic dictionary $\mathbf{D}_{writer}^{l\times l}$ based on the entire images. For the first genuine signature sample that belongs to the reference set the initial dictionary $\mathbf{D}_{writer}^{l\times l,1^{st}\,sample}$ is being created with the use of the K-SVD algorithm. Next and for each one of the remaining genuine reference samples the initial dictionary is being updated in order to provide the final $\mathbf{D}_{writer}^{l\times l}$ dictionary according to figure's 2 content.

Next to the enrollment stage is the learning stage. For each input signature, the preprocessing procedure is applied with the addition of an equimass segmentation step [33, 34], which creates four equimass segments denoted hereafter as I_g^{2x2}, $g=1:4$. The OMP algorithm computes the sparse coefficients given the $\mathbf{D}_{writer}^{l\times l}$ dictionary for the entire signature image along with its four equimass I_g^{2x2} segments. Figure 2 depicts an algorithmic illustration of the dictionary learning procedure for the $\mathbf{D}_{writer}^{l\times l}$ case. The result is the creation of five representation matrices $\mathbf{A}$ which contain the sparse coefficients derived by the $\mathbf{D}_{writer}^{l\times l}$ dictionary. The learning stage of the classifier, incorporates the genuine and its random forgery reference subsets in order to provide the best training parameters according to a cross-validation procedure which maximizes the Area under Curve (AUC).

During the testing stage, any questioned signature sample is first subjected to the preprocessing and sparse coding sequence (similar to the learning phase) and the corresponding feature is being presented to the trained classifier which computes a score value. Posterior analysis provides the EER performance metrics.

3. Sparse representation principles

Sparse representation or approximation is the scientific discipline which represents a signal with a linear combination of a set of dictionary elements or atoms in such a way that only few of the resulting coefficients are non-zero. In this context, dictionaries are usually overcomplete in order to promote sparsity while the actual atoms can be derived either from the discretization of functions like over-complete DCT [31] or directly learned from the data, as in the presented scheme. In the latter case, the problem of simultaneously learning an appropriate dictionary and computing the sparse representation of a set of M - training signals, stored in the matrix $\mathbf{X}\in R^{n\times M}$, $\mathbf{X}=\{(\mathbf{x}_j)^T\}, \mathbf{x}_j\in R^n$ as n-dimensional column vectors, can be defined as follows:

$$\min_{\mathbf{A},\mathbf{D}}\left\{\left\|\mathbf{X}-\mathbf{D}\mathbf{A}\right\|_F^2\right\} \quad \text{s.t.} \quad \left\|\mathbf{a}_i\right\|_0 \le \rho \tag{1}$$

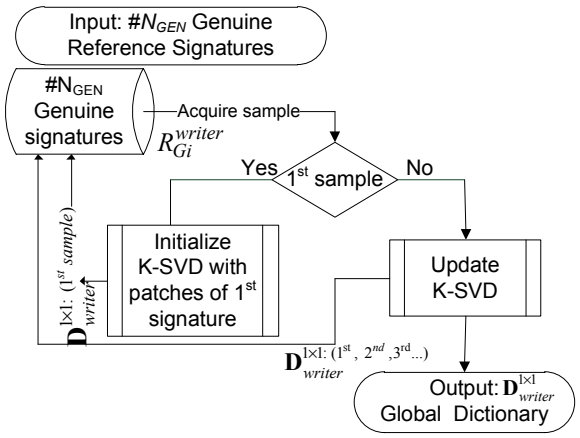

Figure 2: Sequential $\mathbf{D}_{writer}^{l\times l}$ learning algorithm for N_{GEN} genuine reference samples.

where $\mathbf{A}\in R^{K\times M}$, $\mathbf{A}=\left\{(\mathbf{a}_i)^T\right\}, i=1...M$, $\mathbf{a}_i\in R^K$ is the matrix which holds the corresponding representation coefficients, K is the number of atoms in the dictionary $\mathbf{D}\in R^{n\times K}$, and ρ is the largest number of non-zero coefficients allowed in each signal's representation defined hereafter as sparsity level. The solution to the above optimization problem is usually approximated by following a two-step iterative procedure, consisting of the sparse coding and dictionary update phases. During the coding-approximation phase, the sparse representation of the training samples is estimated given a (temporary) dictionary. In the second phase, the dictionary atoms are updated so as to better represent the samples, given the coefficients derived from the previous step. This procedure is repeated until some termination criteria are met. The dictionary obtained at the final iteration can be utilized for the sparse representation of new samples, using the same coding technique as in training. Although the sparse coding problem is a known NP-hard combinatorial problem it is usually approximated using either greedy algorithms [35] or convex relaxations [36]. In this work we follow the approach of Orthogonal Matching Pursuit (OMP), known to be effective in a variety of applications and computationally lightweight. For the dictionary learning we utilize the K-SVD algorithm [37].

3.1. OMP representation

Given a dictionary $\mathbf{D}=\left[d_1,d_2,...,d_K\right]$, $d_K\in R^n$ and any sample $\mathbf{x}$, OMP follows a greedy approach by sequentially selecting the atoms with the highest correlation to the respective sample's residual. At a step $s: 0<s\le\rho$ the selected atom is given by:

$$k_s=\arg\max_k\left|d_k\mathbf{r}_{s-1}\right| \tag{2}$$

where $\mathbf{r}_{s-1}$ is the current residual. Once an atom is selected, the signal is projected onto the span of currently selected atoms as: $\hat{\mathbf{a}}_s = \left(\mathbf{D}_{C_s}\right)^{+}\mathbf{x}$, where $C_s = C_{s-1} \cup k_s$ is the set of indices pointing at the currently selected dictionary atoms, and $\mathbf{D}_{C_s}$ is the subset of dictionary indexed by C_s. The new residual is now given by $\mathbf{r}_s = \mathbf{x} - \mathbf{D}_{C_s}\hat{\mathbf{a}}_s$. The process is repeated until ρ atoms are selected or the residual magnitude becomes zero. The initial conditions are $C_0 = \varnothing$ and $\mathbf{r}_0 = \mathbf{x}$, and the final sample's sparse representation is formed by a vector $\mathbf{a}^T \in R^K$ whose only non-zero entries are given by $\hat{\mathbf{a}}_{final}$ located at the respective positions indicated by C_{final}. In this work we utilize the batch-OMP implementation [38], which makes use of Cholesky decomposition in order to reduce the computational cost of repeated re-projections.

3.2. K-SVD

Given a set of training samples $\mathbf{X}$ and the corresponding set of sparse coefficients $\mathbf{A}^{(t)}$ computed using OMP for the dictionary $\mathbf{D}^{(t)} = \left[d_1^{(t)}, d_2^{(t)}, ..., d_K^{(t)}\right]$ at iteration t, the goal is to produce a new dictionary $\mathbf{D}^{(t+1)}$ such that:

$$\min_{\mathbf{D}^{(t+1)}}\left\{\left\|\mathbf{X} - \mathbf{D}^{(t+1)}\mathbf{A}^{(t)}\right\|_F^2\right\} \quad \text{s.t.} \quad \left\|\mathbf{a}_i^{(t)}\right\|_0 \le \rho, \; i = 1, ..., M \qquad (3)$$

Now, let's define a group of indices ω_k, pointing to training samples that use the atom $d_k^{(t)}$ as follows:

$$\omega_k = \left\{i \,|\, 1 \le i \le M, \mathbf{a}_T^k(i) \ne 0\right\} \qquad (4)$$

where $\mathbf{a}_T^k$ denotes the *k*-th row of the matrix $\mathbf{A} \in R^{K \times N}$, which holds the coefficients that correspond to the *k*-th atom. With the use of ω_k, we define a matrix $\Omega_k \in R^{K \times |\omega_k|}$ with ones on the $(\omega_k(i), i)$ entries and zeros elsewhere. An overall-representation-error matrix can now be defined as:

$$\mathbf{E}_k = \mathbf{X} - \sum_{j \ne k} d_j^{(t)}\mathbf{a}_T^j \qquad (5)$$

expressing the current error in the representation of training samples, due to all atoms except the *k*-th. In other words, $\mathbf{E}_k$ contains the information that remains unexplained by the current representation. Thus, the *k*-th atom can be optimized so as to better represent this information. The multiplication $\mathbf{a}_\omega^k = \mathbf{a}_T^k\Omega_k$ creates a vector using only the non-zero coefficients of $\mathbf{a}_T^k$. In addition, $\mathbf{E}_k^\omega = \mathbf{E}_k\Omega_k$ shrinks the matrix $\mathbf{E}_k$ by discarding the columns corresponding to training samples that are not currently using the *k*-th atom. The K-SVD algorithm updates the *k*-th atom by applying Singular Value Decomposition (SVD) on the matrix $\mathbf{E}_k^\omega$ by finding the closest rank-1 matrix so that $\mathbf{E}_k^\omega = \mathbf{U}\Delta\mathbf{V}^T$. The updated value $d_k^{(t+1)}$ of the kth atom is defined as the first column of $\mathbf{U}$, and the new coefficient vector $\mathbf{a}_\omega^k$ as the first column of $\mathbf{V}$ multiplied by $\Delta(1,1)$. This process is repeated for each atom, producing an updated dictionary. It is important to note that during the update, each atom's support stays the same, thus retaining the overall sparsity conditions derived from sparse coding. The new dictionary is consequently used by OMP in order to produce the updated sparse coefficients $\mathbf{A}^{(t+1)}$, and the procedure is repeated until the overall representation error becomes smaller than a threshold, or a maximum number of iterations t_{max} is reached. The initial dictionary $\mathbf{D}^{(0)}$ is usually filled with random values from a normal distribution with zero mean and unit variance.

4. Feature extraction

Following the description provided in the previous section and prior to: a) the dictionary learning stage and b) any OMP representation, the matrix $\mathbf{X} \in R^{n \times M}$ of any input signature image $\mathbf{I} \in \{\mathbf{I}^{1\times1}, \mathbf{I}_g^{2\times2}\}$ must be properly defined and organized. For any signature pixel in $\mathbf{I} \in \{\mathbf{I}^{1\times1}, \mathbf{I}_g^{2\times2}\}$ with (i,j)- coordinates belonging to either the entire signature trace $\mathbf{I}^{1\times1}$ (e.g. for the case of dictionary learning) or an equimass *g*-segment $\mathbf{I}_g^{2x2}$, $g = 1:4$ an elementary rectangular window defined hereafter as the patch $p_N(i,j)$ of size $N_p \times N_p$ is imposed in order to locate and store the local neighborhood pixel intensities. Next, each patch $p_{N_p}(i,j)$ is vectored and stored with a one-dimensional column vector: $(\mathbf{x}^{(i,j)} \in R^n) \triangleq {}^{1D}p_{N_p}^{i,j}(k)$, with $k = 1:n$ and $n = N_p \times N_p$. The union of the image patches $(\{\mathbf{x}^{(i,j)}\} \in R^n)$ on the entire image $\mathbf{I}^{1\times1}$ or any $\mathbf{I}_g^{2x2}$ segment finally defines the matrix $\mathbf{X} \in R^{n \times M}$ as:

$$\mathbf{X} = \{\cup_{(i,j)\in \mathbf{I}^{1x1}}\mathbf{x}^{(i,j)}\} \text{ or } \mathbf{X} = \{\cup_{(i,j)\in \mathbf{I}_g^{2x2}}\mathbf{x}^{(i,j)}\}, \; g = 1:4 \qquad (6)$$

where the number $M = |(i,j)| \in \{\mathbf{I}^{1\times1} \; or \; \mathbf{I}_g^{2\times2}\}$ corresponds

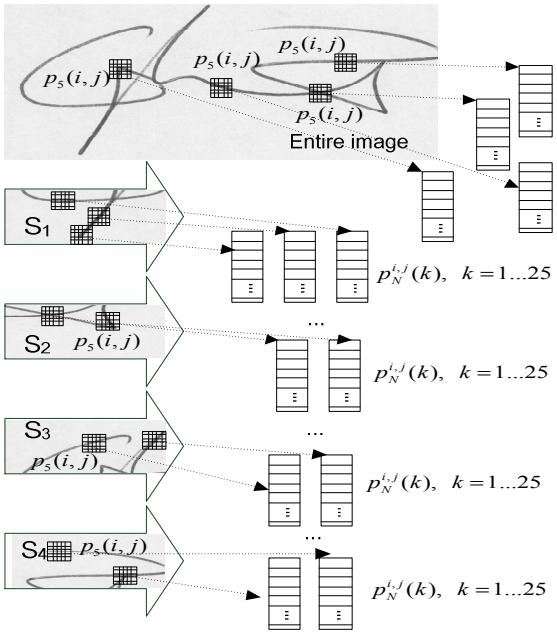

Figure 3: Patch formation and creation of the representative set X from the $\mathbf{I}^{1\times1}$ or the $\mathbf{I}_g^{2\times2}$ signature traces. The patch size N_p equals five.

to the number of the $\mathbf{I}^{1\times1}$ or $\mathbf{I}_g^{2x2}$ signature pixels. The value of the patch size N_p has been to five for this work, i.e. $n=25$, since previous literature research [17] indicates that patterns detected within this window size provide state of the art results. It has been observed also that the increase of the patch size N_p to values way beyond five (e.g. eleven and above) usually results to conditions in which the number of atoms is greater than the number of training patches-signals something that causes the fail of the K-SVD algorithm. The K-SVD also fails for the cases of applying the equimass segmentation of non-complex types of signatures into a number of segments greater than four (e.g. a segmentation grid of 8×8; instead of a 2×2 which is the case for this work). A reduction in the number of atoms may resolve this, issue but at the expense of risking a poorer verification performance. Furthermore, the feature dimensionality increases unnecessary. Figure 3 depicts an example of patch formation for some pixels of the $\mathbf{I}^{1\times1}$ and the four ($\mathbf{I}_g^{2\times2}$) image segments. Throughout the conducted experiments, the working parameters of the K-SVD algorithm were set as follows: number of maximum iterations t_{max}=50, number of atoms K=60 in order to ensure the over-completeness (as a rule of thumb we use a number of atoms greater than twice the patch size dimension) and $\rho=3$ for the sparsity level in order to provide an overall 5% total sparsity. The systematic study of how all the parameters involved in the sparse representation stages affect the verification performance, although interesting is beyond the scope of this work.

Following, for any other input signature after preprocessing and segmentation, the feature extraction stage utilizes the OMP in order to create the $\mathbf{A}_{1\times1}\in R^{K\times|\mathbf{I}^{1\times1}|}$ along with the local $\mathbf{A}_g\in R^{K\times|\mathbf{I}_g^{2\times2}|}$ sparse coefficient matrix. Then, inspired by [39], the feature vector f is simply formed by average pooling the sparse coefficients corresponding to each atom and concatenating the resulting values to a vectors. Let us denote the feature

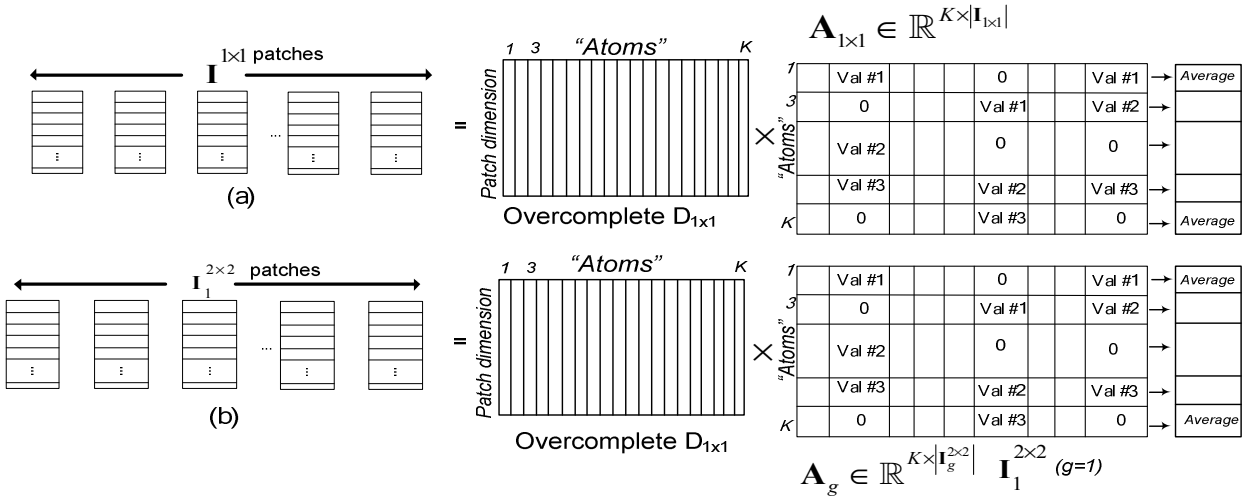

Figure 4: Feature extraction example. Features are the averages of each decomposition row. a) Rearranged 1-D $\mathbf{I}^{1\times1}$ patches and corresponding $\mathbf{A}_{1\times1}$ sparse coefficients. b) Rearranged 1-D $\mathbf{I}_1^{2\times2}$ patches and corresponding $\mathbf{A}_g$, (g=1) sparse coefficients. K=60. Feature dimensionality equals to 300.

vector $f=\left\{f_{\mathbf{I}^{1\times1}},\{f_{\mathbf{I}_g^{2\times2}}\}\right\}$ where:

$$f_{\mathbf{I}^{1\times1}}=\{f_{h,\mathbf{I}^{1\times1}}\},h=1...K,:\ f_{h,\mathbf{I}^{1\times1}}=\frac{1}{\left|\mathbf{I}^{1\times1}\right|}\sum_{c=1}^{\left|\mathbf{I}^{1\times1}\right|}\mathbf{A}_{1\times1}(h,c)$$
$$f_{\mathbf{I}_g^{2\times2}}=\{f_{h,\mathrm{g}}\},h=1...K,:\ f_{h,g}=\frac{1}{\left|\mathbf{I}_g^{2\times2}\right|}\sum_{c=1}^{|\mathbf{Sg}|}\mathbf{A}_g(h,c) \quad (7)$$

Given that the number of atoms K equals to 60, the overall feature dimensionality is easily perceived to be 300 $((\mathbf{I}^{1\times1}+4\mathbf{I}_g^{2\times2})\times 60)$. Figure 4 provides the graphical depiction of the feature extraction method.

5. Verification protocol

5.1. The datasets

Three widely used signature datasets were used in order to test the proposed system architecture. The first one was created at CEDAR, Buffalo University [40]. For each one of the 55 enrolled writers, a total of forty-eight signature specimens (24 genuine and 24 simulated) confined in a 50 mm by 50 mm square box were provided and digitized at 300 dpi. The simulated signatures found in the CEDAR database are composed from a mixture of random, simple and skilled forgeries. The second signature database used was the off-line version of the MCYT signature database [41]. A whole of thirty (15 genuine and 15 simulated) signature samples were recorded for each one of the 75 enrolled writers at a resolution of 600 dpi. The last signature database used was the GPDS300 signature database [7]. It contains 24 genuine signatures and 30 simulated forgeries of 300 individuals stored in an 8-bit, grey level format. A remarkable property of this dataset stems from the fact that the acquisition of signature specimens was carried out with the aid of two different box sizes: the first box had dimensions of 5×1.8 cm wide, while the second box had dimensions of 4.5×2.5 cm. As a result, the files of the GPDS dataset include images having two different aspect ratios; this phenomenon conveys a structural distortion highlighted during the feature extraction procedure. Due to the diverse acquisition settings for the three aforesaid signature databases and given that the patch size exploits information within a 5×5 grid, while the GPDS dataset images are present within two different aspect ratios, the trimming levels for the CEDAR, MCYT and GPDS datasets have been set to one, two and four correspondingly.

5.2. Methodology

For each one of the writers-users of a specific dataset, the building of a classifier begins by enrolling randomly N_{Gen} genuine reference signature samples for the dictionary learning procedure. The number of N_{Gen} in this work has been set to five (N_{Gen}=5) in order to address cases with a limited number of samples. Then, the $\mathbf{D}_{writer}^{1\times1}$ dictionary is evaluated sequentially by employing all the genuine reference samples and is considered as the handwriting model of the specific writer.

In the training stage, the genuine enrollment set is complemented by the negative class representatives in order to form the training set. The negative class is composed from N_{RF}=10 out of 54/74/299 samples (according to the selected dataset), by picking up one random sample from 10 out of the remaining writers. Each one of the N_{Gen} and N_{RF} samples is analyzed with the use of the OMP algorithm and the claimed $\mathbf{D}_{writer}^{1\times1}$ dictionary matrix. Next, the corresponding sparse coefficients $\mathbf{A}$, are evaluated along with their final features: $f\in R^{(Ngen+10)\times300}$ in order to provide the positive and negative class $\omega^{\oplus}\in R^{Ngen\times300}$ and $\omega^{-}\in R^{10\times(300)}$ Thus, a corresponding training feature population $[\omega^{\oplus},\omega^{-}]$ is used as an input to a binary, radial basis SVM classifier. A holdout cross-validation procedure returns the optimal operation values of the $\{C^{opt},\ \gamma^{opt}\}$ parameters with respect to a maximum value of the Area Under Curve. In addition, the cross-validation procedure (*CV*) returns each writer's classifier output scores (*CVS*) conditioned on the positive $\omega^{\oplus}$ class samples $CVS^{\oplus}$.

Finally, the testing stage utilizes the remaining genuine signatures, the skilled forgeries (S) and a number of 44/64/289 random forgeries (R) by taking one random sample from the other writers which were not part of the training set. In order to obtain results that are comparable to those reported in the literature, we employ for evaluation the receiver operating characteristic (ROC) parameters. The FAR(S) and 1-FRR error rates are computed as a function of the sliding threshold whose extremes are located between the minimum and maximum values of the $CVS^{\oplus}$ cross validation procedure of each user. The EER: FAR(S) = FRR is then computed. The experiments were repeated ten times and their average values are reported. As in [27], two forms of EER were considered. The first was the $EER_{user\text{-}threshold}$: using user-specific decision thresholds; and the second was the $EER_{global\_threshold}$ using a global decision threshold. Furthermore, the calculation of the FAR and FRR rates with the utilization of a predetermined threshold is accomplished with the a-priori knowledge of the cross validation procedure scores $CVS^{\oplus}$. Specifically, this hard threshold value is the one which corresponds to the 50% of the average of the genuine $CVS^{\oplus}$ scores for each writer. For completeness, at this specific threshold point, the

TABLE 1: VERIFICATION ERROR RATES (%) FOR CEDAR, MCYT75 AND GPDS300. N_{GEN}=5.

Dataset	FRR	FAR_{random}	$FAR_{skilled}$	$EER_{global\_threshold}$	$EER_{user\_threshold}$
CEDAR	7.32	0.34	6.83	7.58	2.78
MCYT-75	7.32	0.35	10.4	8.43	3.67
GPDS300	6.91	0.33	6.22	7.21	2.70

random forgery-(R) FAR(R) error rate is evaluated by employing the genuine samples of the remaining writers of the testing set.

5.3. Results

Table 1 presents the associated verification results according to the aforementioned protocol. We noticed that the performance of the proposed system was quite good in both terms of equal error rates as well as hard decision error rates. However, it is clear that there is a difference between the error rates that are reported between the $EER_{user\text{-}thresholds}$ and the $EER_{global\_threshold}$. The $EER_{user\text{-}thresholds}$ results are superior to the ones provided from the $EER_{global\_threshold}$. In other words the best decision threshold is not the same for each writer. This can be partly explained, by the absence of simulated samples during the training phase which conveys undoubtedly a bias in the selection of the optimal SVM parameters and the distribution of the genuine $CVS^{\oplus}$ scores.

The feature extraction method relies on the average pooling of the sparse coefficient matrices $\mathbf{A}_{1\times l}$ and/or $\mathbf{A}_g$ as equation (7) describes. Another popular type of pooling in sparse feature extraction is the max pooling [30]. In this work the max pooling has been also tested both independently and in cooperation with the average pooling. However we report that the verification rates obtained in both cases through the max pooling operation were not as high as those of the average pooling.

The magnitude of the sparse coefficients and their relation to the verification efficiency has also been addressed in this work. Traversing of any sparse representation matrix $\mathbf{A} \in R^{K\times(\#\text{pixels})}$ across a fixed line

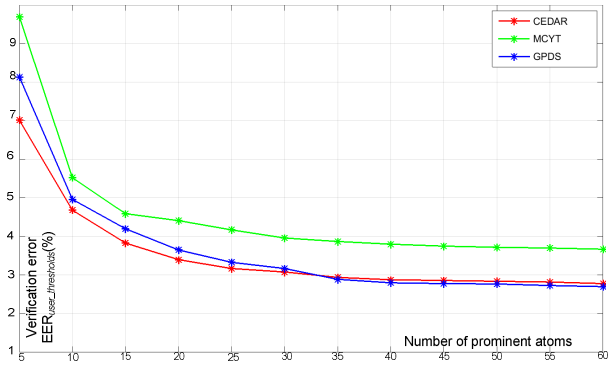

Figure 5: $EER_{user\_threshold}$ for the three datasets as a function of the number of most prominent atoms.

number $\mathbf{A}_L^T \; L \in [1...k]$, provides a way for assessing the contribution of the L-atom to the reconstruction of the neighborhood of pixels for the entire signature image. In order to study the impact of the magnitude of each atom into the verification process and identify potentially important feature components the following protocol was followed: for each batch of sixty average pooled features a sorting was carried out according to their absolute value. Then the verification protocol was adapted to use only the s-most significant features with s having values of five up to sixty with an increasing step of five with the corresponding feature dimensionality to lie in the interval from 25 up to 300. The produced results showed clearly that although there is a monotonically decrease of the verification error with respect to the number of significant features, the use of the entire sixty features per pooling provides the best verification results. Figure 5 provides a graphical depiction of the $EER_{user\text{-}thresholds}$ verification error as a function of the s-most significant features.

An appraisal of the verification efficiency of various signature verification systems is always a clouded task given the diverse parameters used in the training and testing stages like a) the number of training samples, b) the use of genuine or simulated forgeries during the training stage, something which affects seriously any selection of the decision threshold, c) the use of random or/and simulated forgeries during the testing stage and d) the selection of WD or WI strategy [17, 26]. One possible solution for addressing this diversity could emerge from the use of datasets from the signature verification contests initiated by E. v. d. Heuvel [42] and continued for many years. This is especially important, because at some contests very few genuine signatures were provided. Table 2 outlines and compares the best results of our proposed method ($EER_{user\_thresholds}$) with a number of WD and WI related methods for the three utilized datasets on an EER basis. According to these results, for all the exploited signature datasets the proposed method provides verification results which surpass those reported in the literature at the time of the submission of this work.

6. Conclusions

Parsimony, a biologically inspired notion has been exploited lately in a plethora of pattern recognition, machine learning, and computer vision applications. Besides representation, sparse algorithms and techniques

TABLE 2: ERROR RATES (%) OF THE PROPOSED METHOD COMPARED TO OTHER SYSTEMS WITH THE CEDAR/MCYT75/GPDS DATASETS

First Author	Dataset	Method	# Genuine Sigs. for training	EER
Kumar R. [25]	CEDAR	Signature Morphology	24	11.6
Kumar R. [12]	CEDAR	Surroundness	24	8.33
Kalera [40]	CEDAR	Gradient, Structural and Concavity	16	21.9
Zois [17]	CEDAR	Partially Ordered Sets	5	4.12
Proposed	**CEDAR**	**K-SVD dictionary learning – OMP**	**5**	**2.78**
Vargas [7]	MCYT75	Local binary patterns (LBP)	5	11.3
Zois [17]	MCYT75	Partially Ordered Sets	5	6.02
Alonso-Fernandez [43]	MCYT75	Slant and envelope	5	22.4
Fierrez-Aguilar [44]	MCYT75	Global and local slant	5	11.0
Wen [45]	MCYT75	Invariant ring peripheral	5	15.0
Yin Ooi [46]	MCYT75	Discrete Radon transform	5	13.9
Soleimani [11]	MCYT75	Deep multitask metric	**5**	13.44
Proposed	**MCYT75**	**K-SVD dictionary learning – OMP**	**5**	**3.67**
Hu [48]	GPDS 150	LBP & HOG & GLCM	10	7.66
Yilmaz [47]	GPDS 160	Local binary patterns	12	9.64
Yilmaz [47]	GPDS 160	LBP & HOG	12	9.67
Pirlo[34]	GPDS 300	Cosine similarity	12	7.20
Pirlo [22]	GPDS 300	Optical flow	6	4.60
Parodi [23]	GPDS 300	Circular Grid	13	4.21
Kumar [12]	GPDS 300	Surroundness	24	13.8
Zois [17]	GPDS 300	Partially Ordered Sets	5	5.48
Hafemann [27]	GPDS 300	Deep Convolutional N. N.	5	4.53
Hafemann [27]	GPDS 300	Deep Convolutional N. N.	14	3.47
Proposed:	**GPDS 300**	**K-SVD dictionary learning – OMP**	**5**	**2.70**

find an ever increasing number of applications related to image classification and recognition. Signatures being a particular class of image signals exhibit a degenerate structure and lie on a low dimensional subspace. Sparse representation is well suited for handling this problem, approximating the subspace utilizing the sparsity principle and an overcomplete set of basis signals. In this work, a SR-based method is tested successfully to the signature verification problem.

Initially a dictionary is extracted from the signature training samples utilizing square patches, densely sampled, along the signature trace. Dictionary atoms are adapted to the particular characteristics of the extracted image patches, and represent important directions of this space. Next, in the coding stage every patch of the query signature is approximated utilizing a few dictionary atoms with the corresponding weight coefficients. The sparse matrix of the weight coefficients contains all the required information for signature reconstruction while at the same time this information can be used for classification purposes. In the feature extraction stage of the proposed scheme, the representation coefficients are used to produce a global signature descriptor, by simply averaging the coefficients corresponding to every atom, over all patches. Despite the averaging operation that destroys and disregards the spatial information, the produced feature vector proves to be quite discriminative, delivering state-of-the-art verification performance. This is attributed to the compact and efficient signal description attained by the sparse representation.

Regarding future research, we intend to explore at least the possibility to use sparse representation and dictionary learning for designing writer independent dictionaries along with the use of local average pooling. In addition, we plan to test this algorithm to several new datasets in order to provide a more generic infrastructure regarding signature verification. The code implementing the proposed scheme can be provided upon request.

References

[1] G. Pirlo, D. Impedovo, and M. Fairhurst, Front matter, Advances in Digital Handwritten Signature Processing: i-xiii: World Scientific, 2014.

[2] 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, 38(5):609-635, 2008.

[3] M. I. Malik, M. Liwicki, and A. Dengel, Local Features for Forensic Signature Verification, New Trends in Image Analysis and Processing – ICIAP 2013 International Workshops, Naples, Italy, Proceedings, A. Petrosino, L. Maddalena and P. Pala, eds.: 103-111, Springer Berlin Heidelberg, 2013.

[4] R. Sabourin, G. Genest, and F. J. Preteux, Off-line signature verification by local granulometric size distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(9):976-988, 1997.

[5] D. Rivard, E. Granger, and R. Sabourin, Multi-feature extraction and selection in writer-independent off-line signature verification, International Journal on Document Analysis and Recognition,16(1): 83-103, 2013.

[6] G. S. Eskander, R. Sabourin, and E. Granger, Hybrid writer-independent writer-dependent offline signature verification system, IET Biometrics,2(4):169-181, 2013.

[7] J. F. Vargas, M. A. Ferrer, C. M. Travieso and J.B. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognition, 44(2): 375-385, 2011.

[8] Y. Serdouk, H. Nemmour, and Y. Chibani, New off-line Handwritten Signature Verification method based on Artificial Immune Recognition System, Expert Systems with Applications, 51:186-194, 2016.

[9] M. A. Ferrer, J. F. Vargas, A. Morales and A. Ordonez, Robustness of Offline Signature Verification Based on Gray Level Features, IEEE Transactions on Information Forensics and Security, 7(3): 966-977, 2012.

[10] M. B. Yılmaz, and B. Yanıkoğlu, Score level fusion of classifiers in off-line signature verification, Information Fusion, 32(B): 109-119, 2016.

[11] A. Soleimani, B. N. Araabi, and K. Fouladi, Deep Multitask Metric Learning for Offline Signature Verification, Pattern Recognition Letters, 80: 84-90, 2016.

[12] R. Kumar, J. D. Sharma, and B. Chanda, Writer-independent off-line signature verification using surroundedness feature, Pattern Recognition Letters, vol. 33, no. 3, pp. 301-308, 2012.

[13] Y. Guerbai, Y. Chibani, and B. Hadjadji, The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters, Pattern Recognition, vol. 48, no. 1, pp. 103-113, 2015.

[14] A. Hamadene, and Y. Chibani, One-Class Writer Independent Offline Signature Verification Using Feature Dissimilarity Thresholding, IEEE Transactions on Information Forensics and Security, 11(6): 1226-1238, 2016.

[15] M. I. Malik, M. Liwicki, A. Dengel, S. Uchida, and V. Frinken, Automatic Signature Stability Analysis and Verification Using Local Features, Proceedings of 14th International Conference on Frontiers in Handwriting Recognition: 621-626, 2014.

[16] S. Pal, S. Chanda, U. Pal, K. Franke, M. Blumenstein, Off-line signature verification using G-SURF, 12th International Conference on Intelligent Systems Design and Applications: 586-591, 2012.

[17] E. N. Zois, L. Alewijnse, and G. Economou, Offline signature verification and quality characterization using poset-oriented grid features, Pattern Recognition, vol. 54: 162-177, 2016.

[18] M. Okawa, Offline Signature Verification Based on Bag-Of-Visual Words Model Using KAZE Features and Weighting Schemes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops: 184-190, 2016.

[19] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez and J. Ortega-Garcia, An Off-line Signature Verification System Based on Fusion of Local and Global Information , Biometric Authentication, Lecture Notes in Computer Science D. Maltoni and A. K. Jain, eds.: 295-306: Springer Berlin Heidelberg, 2004.

[20] J. Coetzer, B. M. Herbst, and J. A. d. Preez, Offline Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model, EURASIP Journal on Advances in Signal Processing, 2004(4): 1-13, 2004.

[21] V. Nguyen, Y. Kawazoe, T. Wakabayashi, U. Pal and M. Blumenstein, Performance Analysis of the Gradient Feature and the Modified Direction Feature for Off-line Signature Verification, International Conference on Frontiers in Handwriting Recognition: 303-307, 2010.

[22] G. Pirlo, and D. Impedovo, Verification of Static Signatures by Optical Flow Analysis, IEEE Transactions on Human-Machine Systems, 43(5): 499-505, 2013.

[23] M. Parodi, J. C. Gomez, and A. Belaid, A Circular Grid-Based Rotation Invariant Feature Extraction Approach for Off-line Signature Verification, Proceedings 11th International Conference on Document Analysis and Recognition,1289-1293, 2011.

[24] J. Ruiz-del-Solar, C. Devia, P. Loncomilla and F. Concha, Offline Signature Verification Using Local Interest Points and Descriptors, Progress in Pattern Recognition, Image Analysis and Applications, Lecture Notes in Computer Science J. Ruiz-Shulcloper and W. Kropatsch, eds.: 22-29: Springer Berlin Heidelberg, 2008.

[25] R. Kumar, L. Kundu, B. Chanda and J. D. Sharma, A writer-independent off-line signature verification system based on signature morphology, Proceedings of the 1st International Conference on Intelligent Interactive Technologies and Multimedia: 261-265, 2010.

[26] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, Offline handwritten signature verification-literature review, arXiv preprint arXiv:1507.07909, 2015.

[27] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, Analyzing features learned for Offline Signature Verification using Deep CNNs, arXiv preprint arXiv:1607.04573, 2016.

[28] B. A. Olshausen, and D. J. Field, Emergence of Simple-Cell Receptive-Field Properties by Learning a Sparse Code for Natural Images, Nature, 381(6583): 607-609, 1996.

[29] R. Rubinstein, A. M. Bruckstein, and M. Elad, Dictionaries for Sparse Representation Modeling, Proceedings of the IEEE, 98(6): 1045-1057, 2010.

[30] Z. Zhang, Y. Xu, J. Yang, X. Li and D. Zhang, A Survey of Sparse Representation: Algorithms and Applications, IEEE Access, 3: 490-530, 2015.

[31] R. Kumar, B. Chanda, and J. D. Sharma, A novel sparse model based forensic writer identification, Pattern Recognition Letters,35: 105-112, 2014.

[32] Y. Liu, Z. Yang, and L. Yang, Online Signature Verification Based on DCT and Sparse Representation, IEEE Transactions on Cybernetics, 45(11): 2498-2511, 2015.
[33] J. T. Favata, and G. Srikantan, A multiple feature resolution approach to handprinted digit and character recognition, International journal of imaging systems and technology, 7(4): 304-311, 1996.
[34] G. Pirlo, and D. Impedovo, Cosine similarity for analysis and verification of static signatures, IET Biometrics,2(4): 151-158, 2013.
[35] S. G. Mallat, and Z. Zhifeng, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing, 41(12): 3397-3415, 1993.
[36] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic Decomposition by Basis Pursuit, SIAM Review, 43(1): 129-159, 2001.
[37] M. Aharon, M. Elad, and A. Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE Transactions on Signal Processing, 54(11): 4311-4322, 2006.
[38] R. Rubinstein, M. Zibulevsky, and M. Elad, Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit, CS Technion, 40(8): 1-15, 2008.
[39] J. Mairal, F. Bach, and J. Ponce, Sparse Modeling for Image and Vision Processing, Found. Trends. Comput. Graph. Vis., 8(2-3): 85-283, 2014.
[40] M. K. Kalera, S. Srihari, and A. Xu, Offline signature verification and identification using distance statistics, International Journal of Pattern Recognition and Artificial Intelligence, 18(7): 1339-1360, 2004.
[41] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.J. Igarza, C. Vivaracho, D. Escudero, and Q.I. Moro, MCYT baseline corpus: a bimodal biometric database, IEE Proceedings Vision, Image and Signal Processing, 150(6): 395-401, 2003.
[42] C. E. van den Heuvel, K. Franke, and L. Vuurpijl, The ICDAR 2009 signature verification competition, Proceedings of International Conference on Document Analysis and Recogntion: 1403-1407, 2009.
[43] Alonso-Fernandez, M.C. Fairhurst, J. Fierrez and J. Ortega-Garcia, Automatic Measures for Predicting Performance in Off-Line Signature, IEEE International Conference on Image Processing: 369-372, 2007.
[44] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez and J. Ortega-Garcia, An Off-line Signature Verification System Based on Fusion of Local and Global Information , Biometric Authentication, Lecture Notes in Computer Science, D. Maltoni and A. K. Jain, eds.: 295-306: Springer Berlin Heidelberg, 2004.
[45] J. Wen, B. Fang, Y. Y. Tang and T. Zhang, Model-based signature verification with rotation invariant features, Pattern Recognition, 42(7): 1458-1466, 2009.
[46] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang and B. Y. Hiew, Image-based handwritten signature verification using hybrid methods of discrete Radon transform principal component analysis and probabilistic neural network, Applied Soft Computing, 40(3): 274-282, 2016.
[47] M. B. Yilmaz, Offline signature verification with user-based and global classifiers of local features, Ph.D dissertation, Sabancı University, 2015.
[48] J. Hu, and Y. Chen, Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudo-dynamic Features, Proceedings of 12th International Conference on Document Analysis and Recognition: 1345-1349, 2013.