Hierarchical Dictionary Learning and Sparse Coding for Static Signature Verification

Elias N. Zois
University of West Attica
Agiou Spiridonos Str., 12243 Egaleo, Greece
ezois@teiath.gr

Marianna Papagiannopoulou, Dimitrios Tsourounis and George Economou
University of Patras
http://www.upcv.upatras.gr

Abstract

An assortment of review papers as well as newly quoted literature indicates that usually, the most important link in the chain of designing signature verification systems (SV's) is the feature extraction one. These methods are divided in two main categories. The first one, includes handcrafted features which are methods smanually engineered by scientists to be optimal for certain type of information extraction-summarization from signature images. Examples of this kind include global-local and/or grid-texture oriented features. The second feature category addresses signature modeling and verification with the use of dedicated features, usually learned directly from raw signature image data. Typical representatives include Deep Learning (DL) as well as Bag of Visual Words (BoW) or Histogram of Templates (HOT). Recently, sparse representation (SR) methods (dictionary learning and coding) have been introduced for signature modeling and verification with promising results. In this paper, we propose an extension of the SR framework by introducing the idea of embedding the atoms of a dictionary in a directed tree. This is demonstrated with an l_0 tree-structured sparse regularization norm. The efficiency of the proposed method is shown by conducting experiments with two popular datasets namely the CEDAR and MCYT-75.

1. Introduction

The handwritten signature is a specific outcome of handwriting and hence a part of the behavioral biometric traits family. Signatures are considered to be the product of a personal motoric pattern formed from a combination of letters and/or a sophisticated flourish [1, 2]. The signature trace, depicted usually onto a sheet of paper or an electronic device is considered to be the joint outcome of a person's specific motoric procedure and his/hers taught scripting customs. In addition, the signature production process conveys information related also to his/her writing system and psychophysical state [3]. A substantial amount of research efforts towards the modeling and verification of signatures [4-14] provide evidence that the handwritten signature is a member of a popular behavioral club for affirming the identity or consent of a person in cases like forensics and/or administration. Thus, it is considered to be a dynamic and open research topic.

The offline or static SV addresses motionless images as a result of a scanning procedure. In this case, computer vision and pattern recognition (CVPR) systems complete the task of authenticating an individual, by means of his/hers signature, with potential applications to a non-invasive, friendly and secure interface for security oriented e-society applications [15]. Static SV's are reported to perform inferior when compared to systems which exploit dynamic information (i.e. as a function of time) [16, 17]. However, their use may sometimes be compulsory specifically in forensic cases of interest [18, 19]. This makes the offline signature verification problem a challenging and hard one [20].

The problem that a SV system typically addresses is to discriminate genuine samples against the following types of forgery [21]: a) Random: defined as genuine signatures of a writer different from the authentic author, b) Simple/Casual/Zero Effort: defined as signature samples that are formed by an imitator who owns the name of the genuine writer or samples that are formed by someone who knows the original signatures but his efforts are without practicing, c) Simulated/Skilled: samples that are formed by an experienced imitator or a calligrapher after practicing unrestricted number of times and d) Disguised: samples that actually are not a forgery, but instead they are the outcome of an effort of a genuine author to imitate his/her own signature in such a way that he/she can deny it at a later time.

The most crucial step in the design of a SV system relies on the feature extraction algorithm that assigns any signature image into a multidimensional vector space. Since signatures are carriers of intrinsic ambiguity, expressed with the term variability or with the inverse term stability, the feature extraction stage ideally must retain all the essential intrapersonal information, which is vital for the subsequent verification stage. Review papers as well as recent research efforts [20] state that the offline feature extraction methods may be distributed into two major categories: a) handcrafted features which are outcomes of methods created for other image related applications; examples of this branch include methods with global-local and/or grid-texture oriented features [22-24] and b) signature verification dedicated features, learned directly from images, with representatives Deep Learning (DL) [9,25,26] as well as Bag of Visual Words (BoW) [27, 28] or Histogram of Templates (HOT) [29].

Methods which learn features directly from image pixels, serving as initial data, have also appeared in the literature with noteworthy results. In general, these methods exploit any spatial associations of pixels that exist into the static signature images. Early attempts include the use of Restricted Boltzmann Machines (RBMs) in [30] and Convolutional Neural Networks (CNNs) in [31]. Lately, in [25] Soleimani et al. has proposed the use of Deep Neural Networks for Multitask Metric Learning by employing a distance metric between pairs of signatures in which LBP's were used as an intermediate feature vector. Also quite recently, Hafemman et al. in a series of publications, proposed methods for learning features from images. Specifically, the authors in [32] introduced their formulation for learning features from the genuine signatures of a development dataset, and then utilized them in order to test another set of users. In [33], the authors analyzed the learned feature space and optimized their CNN architecture, obtaining state-of-the-art results on the GPDS dataset. Finally in [20], the previously described formulations were extended with supplementary experiments on two other datasets (MCYT and CEDAR), providing a richer explanation of the method and the experimental protocol, as well as a novel formulation that leverages knowledge of skilled forgeries for feature learning.

Quite recently also, another method for static SV has been presented with the use of parsimonial coding techniques. Specifically, sparse representation methods by means of the KSVD and OMP were tested successfully to local patches of signature images followed by average pooling [22]. This was justified by the fact that signatures being a particular class of image signals exhibit a degenerate structure and lie on a low dimensional subspace. On the other, sparse representation is well suited for handling this kind of problem by approximating this subspace with the sparsity principle and an overcomplete set of basis signals. The concept of representing pixel intensities as linear combinations of few dictionary elements (or atoms) is a popular technique in a number of image processing, as well as machine learning applications [34-36].

In this paper, we propose a novel extension of the classic SR conceptual framework to the structured SR. This is done by introducing the idea of embedding the atoms of a dictionary in a directed tree in both dictionary learning and SR stages. Justification of this key-concept idea comes from the intuition that the structure of any problem regarding spatial arrangement of image pixels encourages the search for relationships between dictionary elements [37]. Structured sparsity has been a main research area for quite a long time [37-44]. For the purpose of this work, we address a special type of structured sparsity, which we will define hereafter as hierarchical sparse coding (HSC). In HSC the dictionary atoms are embedded in a directed and rooted tree T which is fixed and known beforehand while the sparsity patterns expressed by the representation matrix $\mathbf{A}$ are subject to the constraint of being members of a connected and rooted sub-tree of T [38, 39, 45].

The conducted experiments with the use of HSC for SV employ a l_0^{tree} tree-structured sparse regularization norm which has been found to be useful in a number of cases. For comparison purposes, the pooling operation on the representation matrix $\mathbf{A}$ was chosen to be similar to the one presented in [22, 23]. To the best of the author's knowledge, this is a new and novel work that exploits a type of structured sparsity (SS), namely the hierarchical sparsity dedicated for offline signature verification. Codebooks have been also proposed for offline SV by utilizing first order HOG's and then coding each feature to the nearest word in the codebook with K-means [46]. This is clearly not our case since we apply hierarchical dictionary learning from signature patches instead of K-means feature cluster algorithms for creating the codebook. We justify the use of this method as it is said that "*whenever using k-means to get a dictionary, if you replace it with sparse coding it'll often work better*" [47].

The remaining of this paper is organized as follows: Section 2 quotes the necessary elements of hierarchical sparse dictionary coding and representation. Section 3 summarizes the method and presents the feature extraction while section 4 describes the conducted experiments and the corresponding experimental results. Finally, section 6 draws the conclusions.

2. Elements of hierarchical sparse coding

2.1. Terminology

Following the notation proposed of Jenatton et al. [37] vectors are represented with bold lower case letters ($\mathbf{x}$) while matrices with upper case ones ($\mathbf{X}$). The l_q-norm for $q \geq 1$ of a vector $\mathbf{x} \in R^m$ as: $\|\mathbf{x}\|_q \triangleq \left(\sum_{i=1}^{m} |\mathbf{x}_i|^q\right)^{1/q}$ where $\mathbf{x}_i$ designates the i-th component of $\mathbf{x}$ and

$\|\mathbf{x}\|_\infty \triangleq \max_{i=1,\dots,m}(|\mathbf{x}_i|) = \lim_{q\to\infty}(\|\mathbf{x}\|_q)$. Let us also denote the l_0-pseudo-norm, or with the slightly abusive term l_0-norm, to be the number of non-zero elements in a vector:

$$\ell_0(\mathbf{x}) = \|\mathbf{x}\|_0 \triangleq \#\{\text{j s.t. } \mathbf{x}[j] \neq 0, j = 1:m\} \quad (1)$$

Finally, the Frobenius norm of a matrix $\mathbf{X} \in R^{m\times n}$ is defined as: $\|\mathbf{X}\|_q \triangleq \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|\mathbf{X}_{i,j}^2|\right)^{1/2}$, with $\mathbf{X}_{i,j}$ to denote the (i,j) element of the matrix $\mathbf{X}$.

2.2. Elements of sparse representation

In general, SR encodes (or represents) a set of *N*-columnwise signals (or patches, or samples) $\mathbf{x}^i \in R^m$, denoted with a patch matrix $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^{2,}...\mathbf{x}^N] \in \mathrm{R}^{m\times N}$ as a linear combination of a set of basis vectors (or atoms) symbolized with the columns $\mathbf{d}^j \in R^m$ of an overcomplete dictionary or lexicon $\mathbf{D} = [\mathbf{d}^1, \mathbf{d}^2, ..., \mathbf{d}^K] \in \mathrm{R}^{m\times K}$, with $K > n$, that is the number of atoms is greater than the number of signal dimensionality. Specific, for one signal $\mathbf{x} \in R^m$ the formulation of SR is usually expressed with the following unconstrained optimization problems (2) or (3):

$$\min_{\boldsymbol{\alpha}\in R^K}\left(\frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \psi(\boldsymbol{\alpha})\right) \quad (2)$$

$$\min_{\boldsymbol{\alpha}\in R^K}\left(\frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\boldsymbol{\Omega}(\boldsymbol{\alpha})\right), \text{ with } \lambda \geq 0 \quad (3)$$

In eq. (2) $\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2$ represents the square loss measure for one patch and subsequent signal reconstruction error, $\boldsymbol{\alpha} \in R^K$ is the solution or representation vector for the patch $\mathbf{x} \in R^m$. The decomposition vector $\boldsymbol{\alpha}$ obeys the sparsity property, $|\boldsymbol{\alpha}|_0 \leq s$ which states that only a few ($s < K$) coefficients of $\boldsymbol{\alpha}$ should be non-zero valued. This is expressed with the $\psi(\bullet)$ norm or the regularizer, a term which imposes on the representation solution $\boldsymbol{\alpha}$ and dominates significantly the optimization problem.

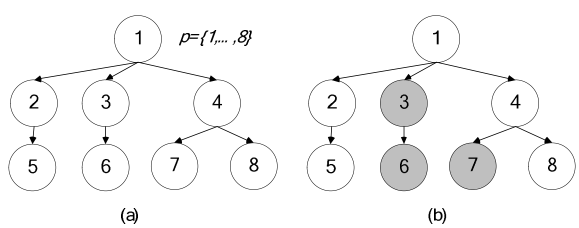

Figure 1: a) Map of an example with a solution vector $\boldsymbol{\alpha} \in R^8$ embedded within a tree T. b) Solution with non-zero valued (non-shadowed) nodes in which the ancestor property apply: $\boldsymbol{\alpha}_8 \neq 0 \Rightarrow \boldsymbol{\alpha}_{(ancestors(8))} \neq 0$.

Equation (3) simply replaces the generic term $\psi(\bullet)$ with the sparsity inducing norm $\boldsymbol{\Omega}$ multiplied by an appropriate parameter λ (or Lagrange multiplier). In the proposed work the sparsity-inducing norm $\boldsymbol{\Omega}$ embeds a fixed and a-priori known hierarchical tree structure between the atoms of $\mathbf{D}$ according to the material exposed in the sections below.

2.3. Penalization with hierarchy

We extend the classic SR problems expressed with eq. (3) by emphasizing on specific sets of the nonzero $\boldsymbol{\alpha}$ coefficients. Given a tree T, whose *p* nodes are denoted as $j = \{1, ..., p\}$, we are concentrating on visual patterns that the non-zero $\boldsymbol{\alpha}$ coefficients are forming by enabling the hypothesis that they are forced to be part of a *connected and rooted subtree* of a tree structure T. This concept can be clarified with the following example depicted also in figure 1. Suppose that a solution vector $\boldsymbol{\alpha} \in R^8$ exists. Let us define the *ancestors(j)* of a node j to be the subset of indices corresponding to the ancestors of the node j. Then, the solution vector $\boldsymbol{\alpha} \in R^8$ is subject to the following condition:

$$\mathbf{a}_j \neq 0, \Rightarrow [\mathbf{a}_k \neq 0] \text{for all } k \text{ in } ancestors(j) \quad (4)$$

Intuitively, we code any signal $\mathbf{x} \in R^m$ by means of a dictionary $\mathbf{D} \in \mathrm{R}^{m\times 8}$ with the representation vector $\boldsymbol{\alpha} \in R^8$ by imposing a rule that the $\mathbf{d}^j \in R^m$ atom contributes in the reconstruction of the $\mathbf{x}$ signal, only if its *ancestors(j)* are also contributing. For a given s-level of sparsity a penalized version of (3) takes the form (5) which has been considered by Donoho [45]:

$$\min_{\boldsymbol{\alpha}\in R^K}\left(\frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda|\boldsymbol{\alpha}|_0\right), \text{ s.t: eq. } (4) \quad (5)$$

Complementary to the non-convex nature of the constraint (4) and subsequent eq. (5), Jenatton et al. [37] provides a contrapositive description of (4) by providing definition for the *descendants(j)* of a *j*-node as follows:

$$\mathbf{a}_j = 0, \ \Rightarrow [\mathbf{a} = 0] \text{for all } k \text{ in } descendants(j) \quad (6)$$

The intuition behind this is that if a $\mathbf{d}^j$ dictionary atom does not contribute in the reconstruction of the $\mathbf{x}$ signal then, neither its descendants in the tree T should do. To summarize, the following rules are equivalent: *(i)* if a node participates, then the same holds for all its ancestors, *(ii)* if a node does not participate, then neither its descendants do. To continue and for simplicity purposes let us denote *T*

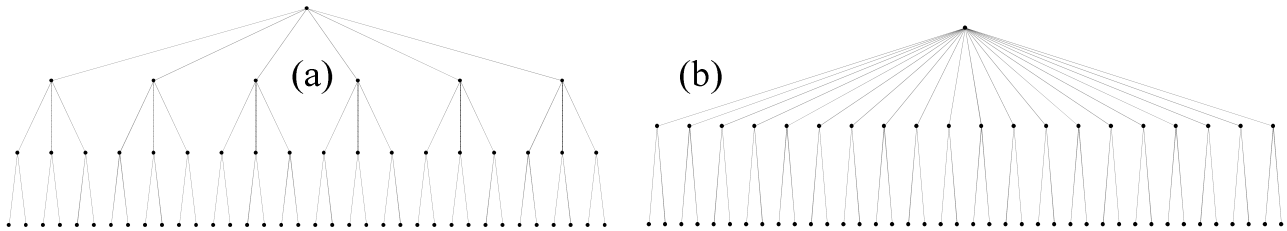

Figure 2: An example of a balanced tree of a) depth 4, with branching factors $\{p_1, p_2, p_3\} = \{6, 3, 2\}$ and b) depth 3 with branching factors $\{p_1, p_2\} = \{20, 2\}$. Both cases have a similar number of 61 corresponding nodes-atoms.

to be a set with elements the *descendants(j)* of each node: $T \triangleq \{descendants(j)\}$, with $j = \{1,\dots, p\}$. Each element g of T is referred as a *group*. In this context, an intuitive penalization was provided by Baraniuk et al. [38] which penalized the number of groups g in G, that participate in the representation of $\mathbf{x}$, by using an $\Omega(\boldsymbol{\alpha}) \triangleq l_0^{tree}(\boldsymbol{\alpha})$ norm which records at least one nonzero coefficient of $\boldsymbol{\alpha}$ as:

$$\Omega(\boldsymbol{\alpha}) \triangleq l_0^{tree}(\boldsymbol{\alpha}) = \sum_{g\in T} \delta^g(\mathbf{a}), \text{ with}$$
$$\delta^g = \begin{cases} 1 & \text{if there exists } j \in g \text{ such that } \boldsymbol{\alpha}_j \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

As a result, eq. (3) along with the constraint of eq. (7) results to the following nonconvex optimization problem:

$$\min_{\boldsymbol{\alpha}\in R^K}(\frac{1}{2}\left\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\right\|_2^2 + \lambda\sum_{g\in T}\delta^g),\ \lambda \geq 0 \tag{8}$$

The above simplified presentation is in the context of a hierarchical norm which contains a single tree equipped with one element for each node. However the above formulation can be expanded to the case of trees comprised with an arbitrary number of atoms at each node. Although in this work we will not examine situations of this kind, for completeness we provide the definition of tree-structured groups according to the following [37]:

Definition: A set of groups $G \triangleq \{g\}_{g\in G}$ is said to be tree structured in $\{1,\dots, p\}$ if $\bigcup_{g\in G}\{g\} = \{1,...,p\}$ and if for all $(g,h)\in G, (g\cap h \neq \varnothing) \Rightarrow (g \subseteq h \ or\ h \subseteq g)$. For such a set of groups, there exists a (non-unique) total order relation $\prec$ such that: $g \prec h \Rightarrow \{g \subseteq h \ or\ g\cap h = \varnothing\}$.

In this work we employed two different balanced trees of depths 4 and 3 with branching factors $\{p_1, p_2, p_3\} = \{6, 3, 2\}$ or $\{p_1, p_2\} = \{20, 2\}$ resulting to the same total number of dictionary nodes-atoms equal to 61. Figure 2 presents the structure of both trees. The reason for selecting a depth not exceeding four is that in this case, the complexity $O(\log(p))$ of the optimization algorithm is almost linear thus providing reasonable execution times.

The optimization problem expressed by eq. (8) for a given dictionary $\mathbf{D}$ and for the l_0^{tree} norm is algorithmically tackled in the more general context of approximation from dyadic partitions; a method initially introduced by Donoho [45]. The algorithm solves the following problem:

$$\min_{\boldsymbol{\alpha}\in R^K}(\frac{1}{2}\left\|\mathbf{u}-\mathbf{a}\right\|_2^2 + \lambda\sum_{g\in T}\delta^g(\mathbf{a})) \tag{9}$$

where $\mathbf{u}\in R^K$ is a fixed signal, and the other parameters retain their previous notions. This problem is considered as a proximal operator for the nonconvex regularization $\lambda \cdot l_0^{tree}(\boldsymbol{\alpha})$. As Jenatton et al., [37] has shown this can be solved efficiently, and provides approximate solutions for the nonconvex problem presented in eq. (5). The algorithm performs a chain of thresholding operations on the variable $\mathbf{a}$ by means of the Iterative Shrinkage Thresholding Algorithm (ISTA) [48]. In this work, we implemented this method as a function within the SPAMS toolbox [49] (*mexFISTAtree*, with ISTA option).

2.4. Dictionary Learning

Dictionary learning is one of the key factors in SR. An effective dictionary can lead to excellent reconstruction results [50]. The framework of dictionary learning manages a joint optimization problem with respect to the dictionary $\mathbf{D}$, which is now learned, and the coefficient matrix $\mathbf{A}\in R^{K\times N}$, with $\mathbf{A} = \{\boldsymbol{\alpha}^i \in R^K\} = \{\boldsymbol{\alpha}^i[j], j = 1:K\}$. In the proposed HSC framework dictionary learning is expressed with the use of the following mathematical notations for a $\mathbf{X}\in \mathrm{R}^{m\times N}$ patch matrix:

$$\begin{aligned}\min_{\mathbf{D}\in C,\mathbf{A}\in R^{K\times M}}(\frac{1}{2}\left\|\mathbf{X}-\mathbf{D}\mathbf{A}\right\|_F^2 + \lambda\Omega(\mathbf{A})) = \\ \min_{\mathbf{D}\in C,\mathbf{A}\in R^{K\times M}}\frac{1}{N}\sum_{i=1}^{N}(\frac{1}{2}\left\|\mathbf{x}^i-\mathbf{D}\mathbf{a}^i\right\|_2^2 + \lambda\Omega(\mathbf{a}^i))\end{aligned} \tag{10}$$

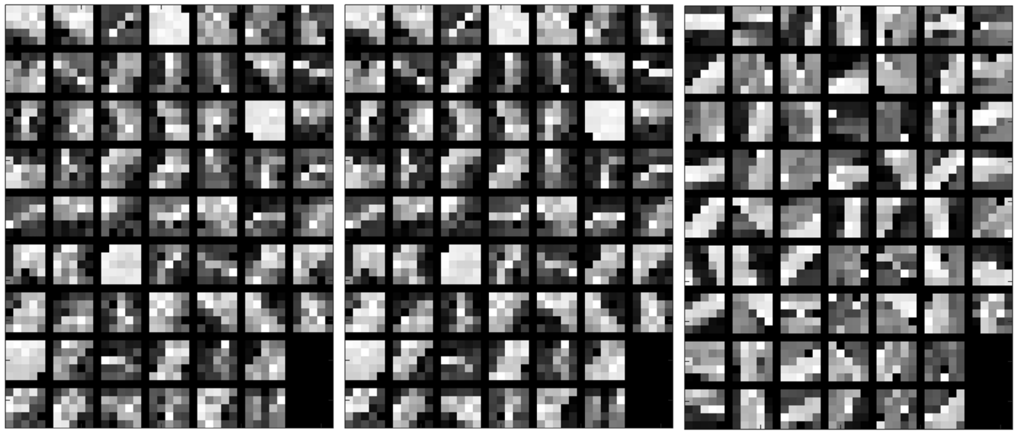

Figure 3: Sixty one learned atoms of a dictionary for three levels of sparsity a) low, b) medium, c) high.

where C is a convex set, the convex set of matrices satisfying the following constraint: $C \triangleq \mathbf{D} \in R^{n \times K}$, s.t: $\forall_{j=1:M}$, $(\mathbf{d}^j)^T \mathbf{d}^j \leq 1$. As seen, the $\mathbf{d}^j$ atoms usually require their $\ell_2(\mathbf{d}^j) = \|\mathbf{d}^j\|_2$ norm be less than or equal to one in order to avoid arbitrarily small values of the $\mathbf{A}$ coefficients. The replacement of the l_0^{tree} norm of eq. (7) into eq. (10) embeds the dictionary to the tree structure and attempts to exploit possible relations between them. Again the implementation was made possible with the use of the SPAMS toolbox (*mexStructTrainDL*, with ISTA option).

3. System design

In the proposed approach, for each writer a population N_G^{REF}, of some genuine reference signature samples $R_{G_i}^{writer}$, $i = 1 : N_G^{REF}$, is enrolled in order to create the reference signature dataset. Following typical image pre-processing steps, which include thresholding and thinning the solution of the dictionary learning algorithm described by eqs. (7), (10) provides a dictionary $\mathbf{D}$ which embeds a balanced tree with branching factors of $\{p_1, p_2, p_3\} = \{6, 3, 2\}$ or $\{p_1, p_2\} = \{20, 2\}$. The dictionary learning algorithm is successively activated by each one of the writer's genuine reference signatures $R_{G_i}^{writer}$ in order to initialize and further update the writer's individual characteristic dictionary $\mathbf{D}_{l_o}^{tree}$ based on the entire images. In the dictionary learning phase the patch matrix $\mathbf{X} = \{\mathbf{x}^i\}$, $i = 1 : N$ with N being the number of signature pixels is formed by a) densely placing a 5×5 window centered on the pixels of the preprocessed signature, and b) reshaping them into a columnwise format thus resulting to $\mathbf{X} \in R^{25 \times N}$. To familiarize with the regularizer parameter λ we conducted experiments with several values of it. Figure 3 presents the dictionary atoms for three cases of having low ($\lambda = 5 \times 10^{-4}$), medium ($\lambda = 10^{-3}$) and high sparsity ($\lambda = 10^{-2}$). It is observed that as the sparsity level gets higher a) the number of dimensions that come into play in order to create an atom decrease, thus making the atoms more "distinct"; i.e. there is an entropy reduction of quantization levels and b) the reconstruction error increases inversely. In the conducted experiments we have selected the medium sparsity level. Figure 4, also presents a detail of the hierarchical tree with the embedded atoms.

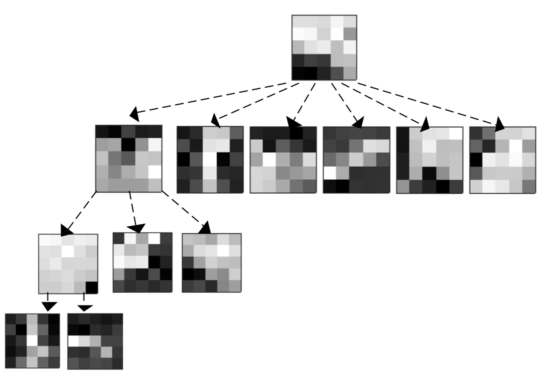

Figure 4: Detail of the figure 2 hierarchical tree structure with some embedded dictionary atoms.

For any other signature, its corresponding patches, the claimed writer's hierarchical dictionary and eq. (8) provides a structured code by means of the coefficients matrix $\mathbf{A}$. Two final signature descriptors f^{F_1}, f^{F_2} are formed as the outcome of two predefined types of pooling [51] of the hierarchical matrix $\mathbf{A}$:

$$f^{F_1} = \{f^{F_1}[j]\} = \{\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\alpha}^i[j]\}, j = 1 : K \quad (12)$$

TABLE 1. VERIFICATION ERROR RATES (%) FOR THE CEDAR SIGNATURE DATASET WITH l_0^{tree} -NORM FOR A TREE STRUCTURE OF DEPTH 4 AND BRANCH LEVELS EQUAL TO [6 3 2]. NUMBER OF REFERENCE TRAINING SAMPLES EQUALS TO FIVE.

Pooling method	Segments: (1+2×2), Dim=305				Segments: (1+3×3), Dim=610			
	PFAR(S)	PFRR	EERusertheesh	PFAR(R)	PFAR(S)	PFRR	EERusertheesh	PFAR(R)
Average	6.44	7.19	**2.56**	0.40	6.67	7.42	**2.72**	0.33
Max	14.8	18.2	**13.6**	7.85	15.1	18.1	**13.7**	7.96

$$f^{F_2} = \{f^{F_2}[j]\} = \max(|\boldsymbol{\alpha}^i[j]|),\ i=1:\mathrm{N},\ j=1:K \qquad (13)$$

Further expansion of the feature dimensionality is achieved by employing the same pooling strategy to specific patches that are parts of image segments designated by a specially designed equimass spatial pyramid of 2×2 or 3×3 segments, which apply to images of varying size [52], resulting to total dimensionality of 305 or 610.

4. Experiments

4.1. Datasets

Two popular signature datasets were used in order to demonstrate the proposed system architecture. The first one was created at CEDAR, Buffalo University [53]. 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 [54]. 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. Due to the diverse acquisition settings for the two aforesaid signature databases and given that the patch size exploits information within a 5×5 grid, the thinning levels for the CEDAR, MCYT and datasets have been set to one, and two correspondingly in order that primitives strokes fit within it. Both datasets have a noteworthy property which is that their samples have been acquired with the use of only one bounding box.

4.2. Experimental scenarios

The experimental setup is quite straightforward as we implement a writer dependent SVs, that is a specific model is being built for every participant. The number of genuine reference samples for each writer N_G^{REF} in this work has been set to five, imitating cases in which a limited number of samples is available. Then, given the two tree topologies the $\mathbf{D}_{l_0}^{tree}$ hierarchical dictionary is evaluated each time and is hereafter regarded to symbolize 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 or 74 samples (depending on the CEDAR or MCYT75 dataset), by selecting one random sample from 10 out of the remaining writers. Each one of the N_G^{REF} and N_{RF} samples is analyzed with the use of the SR algorithm expressed with eq, (8) and the claimed $\mathbf{D}_{l_0}^{tree}$ dictionary matrix. Next, the corresponding sparse coefficients $\mathbf{A}$ are evaluated along with their final features f^{F_1}, f^{F_2} : in order to provide the positive and negative class $\omega^{\oplus} \in R^{N_G^{REF} \times \dim}$ and $\omega^{-} \in R^{2\times N_G^{REF} \times \dim}$. Summarizing, a corresponding training feature set $[\omega^{\oplus}, \omega^{-}] \in R^{3\times N_G^{REF} \times \dim}$ is used as an input to a binary, radial basis SVM classifier. A holdout cross-validation procedure returns the optimal operation values of the margin and the scale parameters with respect to the maximum value of the **A**rea **U**nder **C**urve. Furthermore, the cross-validation procedure extracts for each writer its output scores conditioned on the positive only $\omega^{\oplus}$ class samples $CVS^{\oplus}$. Concluding, the testing stage makes use of

TABLE 2. VERIFICATION ERROR RATES (%) FOR THE CEDAR SIGNATURE DATASET WITH l_0^{tree} -NORM FOR A TREE STRUCTURE OF DEPTH 3 AND BRANCH LEVELS EQUAL TO [20 2]. NUMBER OF REFERENCE TRAINING SAMPLES EQUALS TO FIVE.

Pooling method	Segments: (1+2×2), Dim=305				Segments: (1+3×3), Dim=610			
	PFAR(S)	PFRR	EERusertheesh	PFAR(R)	PFAR(S)	PFRR	EERusertheesh	PFAR(R)
Average	6.06	6.95	**2.30**	0.35	6.32	6.41	**2.42**	0.24
Max	12.9	17.1	**10.2**	6.67	13.1	17.2	**10.4**	6.98

TABLE 3. VERIFICATION ERROR RATES (%) FOR THE MCYT75 SIGNATURE DATASET WITH l_0^{tree} -NORM FOR A TREE STRUCTURE OF DEPTH 4 AND BRANCH LEVELS EQUAL TO [6 3 2]. NUMBER OF REFERENCE TRAINING SAMPLES EQUALS TO FIVE.

Pooling method	Segments: (1+2×2), Dim=305				Segments: (1+3×3), Dim=610			
	$P_{FAR(S)}$	P_{FRR}	$EER_{usertheesh}$	$P_{FAR(R)}$	$P_{FAR(S)}$	P_{FRR}	$EER_{usertheesh}$	$P_{FAR(R)}$
Average	10.3	9.80	**4.29**	0.40	10.4	7.38	**3.70**	0.18
Max	26.3	25.7	**21.3**	8.02	20.2	19.8	**15.1**	11.0

the remaining genuine signatures, its associated skilled forgeries (S) and a number of 44 or 64 random forgeries (R) by taking one random sample from the remaining writers which do not participate to the formation of the training set. Results are reported by means of the following receiver operating characteristic (ROC) probabilities: the $P_{FAR(\mathrm{S})}$ and P_{FRR} error rates which are computed as a function of a rolling threshold- value- whose extremes exists between the minimum and maximum values of the $CVS^{\oplus}$ cross validation procedure. Two different verification approaches are reported. In the first, a hard threshold is utilized to separate the genuine sample from skilled forgeries. The selection of this hard threshold depends only on the genuine reference samples as they are the only ones available during training. Therefore its determination depends only on the values that are obtained during the cross validation stage. In a typical scenario, this hard threshold is set to 50% of the genuine scores. Additionally, we report also the equal error rate per user defined as $EER_{\text{user-threshold}}$: to be the point in which the two errors coincide $P_{FAR(\mathrm{S})} = P_{FRR}$. The experiments were repeated ten times and their average values are reported. For completeness, at this specific threshold point, the random forgery-(R) $P_{FAR(\mathrm{R})}$ error rate is evaluated by employing the genuine samples of the remaining writers of the testing set.

4.3. Results

Tables 1-4 present the verification results derived by the realization of the aforementioned experimental protocols. Apparently, the optimal error rates for the CEDAR dataset are related with the use of a 2×2 spatial pyramid while for the MCYT-75 this occurs when a 3×3 spatial pyramid is employed combined with the average pooling operation. This is probably due to the fact that CEDAR signatures have been scanned with a resolution of 300 dpi, while the MCYT-75 ones with a resolution of 600 dpi, leading to higher pixel density inside the segments. Another observation is that there is a consistent drop, although small, in the verification error for the l_0^{tree} associated with a tree of depth 3 compared to the ones derived with a tree of depth 4. It is the intuition of the authors however that this effect relates to the selection of the average and max pooling techniques which does not convey the tree structure to the feature extraction method.

Finally, Table 5 presents a comparative summary between the proposed method and a number of state of the art approaches for offline signature verification. Table 5 provides evidence that the proposed method achieves a low verification error when a few genuine samples are available. This is considered to be at least comparable to other state-of-the-art methods for static signature verification. Comparing to classical SR methods as presented in [22] the method seems to slightly outperform in all cases.

5. Conclusions

In this work hierarchical dictionary learning and sparse coding which is a particular instance of structured sparsity has been applied, as a novel extension of the classic SR conceptual framework, for modeling and consequent verifying offline signatures. This concept introduces for the first time the idea of embedding the atoms of a dictionary in a rooted and directed tree in both dictionary learning and SR stages of an offline signature verifier. In the conducted experiments we employed a regularizer upon two different tree structures of depths 3 and 4 and different branch factors. The verification results expressed with the user specific EER at CEDAR and MCYT signature datasets indicate that this method seems to outperform the classic SR approach and constitutes a

TABLE 4. VERIFICATION ERROR RATES (%) FOR THE MCYT75 SIGNATURE DATASET WITH l_0^{tree} -NORM FOR A TREE STRUCTURE OF DEPTH 3 AND BRANCH LEVELS EQUAL TO [20 2]. NUMBER OF REFERENCE TRAINING SAMPLES EQUALS TO FIVE.

Pooling method	Segments: (1+2×2), Dim=305				Segments: (1+3×3), Dim=610			
	$P_{FAR(S)}$	P_{FRR}	$EER_{usertheesh}$	$P_{FAR(R)}$	$P_{FAR(S)}$	P_{FRR}	$EER_{usertheesh}$	$P_{FAR(R)}$
Average	9.78	9.67.	**4.01**	0.32	9.67	9.41	**3.52**	0.11
Max	24.4	26.1	**21.2**	8.38	26.3	25.5	**21.8**	10.2

TABLE 5. COMPARISON WITH OTHER STATE OF THE ART OFFLINE SIGNATURE VERIFICATION METHODS FOR CEDAR AND MCYT DATASETS.

1st author / Ref #	Dataset	Method	# Samples	AER/EER
Kumar [55]	CEDAR	Signature Morphology	24 [29] or 1 [20]	11.6 / 11.8
Kumar [56]		Surroundness	24 [29] or 1 [20]	8.33
Chen [57]		Gradient+concavity	16	7.90
Chen [57]		Zernike moments	16	16.4
Kalera [53]		G. S & C	16	21.9
Zois [24]		Partially Ordered Sets	5	4.12
Guerbai [58]		Curvelet Transform	12	5.60
Serdouk [59]		Gradient LBP + LRF	16	3.52
Hafeman [20]		SigNet-F	12	4.63
Bharathi [60]		Chain code	12	7.84
Okawa [28]		B.O.W with KAZE	16	1.60
Okawa [27]		V.L.A.D with KAZE	16	1.00
Ganapathi [61]		Gradient Direction	14	6.01
Shekar [62]		Morphological Pattern Spectrum	16	9.58
Hamadene [63]		Directional Co-occurrence	5	2.11
Zois [23]		Archetypes	5	2.07
Dutta [46]		Compact Correlated Features	N/A	0.00
Zois [22]		KSVD/OMP	5	2.78
Proposed		l_0^{tree} **: branch factors [20, 2]**	**5**	**2.30**
Vargas [64]	MCYT75	L.B.P	10	7.08
Zois [24]		Partially Ordered Sets	5	6.02
Gilperez [65]		Contours	10	7.08
Alonso-Fernandez [66]		Slant and Envelope	5	22.4
Fierrez-Aguilar [67]		Global and Local Slant	10	9.28
Wen [68]		Invariant Ring Peripheral	5	15.0
Yin Ooi [69]		Discrete Radon Transform	10	9.87
Soleimani [25]		HOG-Deep Multitask Metric	10	9.86
Hafeman [20]		SigNet	10	2.87
Serdouk [29]		H.O.T	10	18.15
Zois [22]		KSVD/OMP	5	3.67
Proposed		l_0^{tree} **: branch factors [20, 2]**	**5**	**3.52**

potential candidate for offline signature verification. We plan to expand this work as follows: First, in addition to the l_0^{tree} norm, we plan to extend our study by applying a number of other tree structured norms such as: the l_2^{tree} and the l_∞^{tree} as well as the multi-task tree-structured sum of l_∞ norms. Second, we plan to pursue a line of research which will employ optimization tools for structured norms with general overlapping groups or particular instances of other group-oriented norms such as the *sparse group Lasso*. Third, we also plan to study the effect of positivity of the coefficients or the dictionary atoms. Fourth, we plan to explore the effect to the verification performance of selecting various tree depth levels with several branch factors; although it has been already pointed out that a large depth level may result into pointless execution times. Another important option will be to explore specific pooling which will also exploit the structure of the problem instead of using unstructured pooling styles like the averages. Finally we intend to incorporate several other signature datasets with parameters like different bounding boxes in order to simulate actual life conditions.

References

[1] L. C. Alewijnse, "Analysis of Signature Complexity, Master Thesis," University of Amsterdam, 2008.

[2] H-L. Teulings, Handwriting Movement Control, pp. 561-614, London: Academic Press, 1996.
[3] G. Pirlo, M. Diaz, M. A. Ferrer, D. Impedovo, F. Occhionero and U. Zurlo, "Early Diagnosis of Neurodegenerative Diseases by Handwritten Signature Analysis," New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops. pp. 290-297, 2015.
[4] R. Plamondon, and G. Lorette, “Automatic signature verification and writer identification -- the state of the art,” Pattern Recognition, 22(2):107-131, 1989.
[5] F. Leclerc, and R. Plamondon, “Automatic Signature Verification: The State Of The Art: 1989–1993,” International Journal of Pattern Recognition and Artificial Intelligence, 8(3):643-660, 1994.
[6] R. Plamondon, and S.N. Srihari, “Online and off-line handwriting recognition: a comprehensive survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):63-84, 2000.
[7] L. Batista, D. Rivard, R. Sabourin et al., "State Of The Art In Off-Line Signature Verification " Pattern Recognition Technologies and Applications: Recent Advances B. Verma and M. Blumenstein, eds., pp. 39-62, 2008.
[8] 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.
[9] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Offline Handwritten Signature Verification - Literature Review.,” in 7th International Conference on Image Processing Theory, Tools and Applications --IPTA 2017, pp. 1-8, 2017.
[10] L. G Hafemann, R. Sabourin, and L. S. Oliveira, “Offline handwritten signature verification-literature review,” arXiv preprint arXiv:1507.07909, 2015.
[11] A. S. Shah, M. Khan, and A. Shah, “An appraisal of off-line signature verification techniques,” International Journal of Modern Education and Computer Science, 7(4):67-75, 2015.
[12] D. Impedovo, G. Pirlo, and R. Plamondon, “Handwritten Signature Verification: New Advancements and Open Issues,” in 2012 International Conference on Frontiers in Handwriting Recognition, pp. 367-372, 2012.
[13] R. Plamondon, G. Pirlo, and D. Impedovo, "Online Signature Verification", Handbook of Document Image Processing and Recognition, D. Doermann and K. Tombre, eds., pp. 917-947, London: Springer London, 2014.
[14] S. Pal, M. Blumenstein, and U. Pal, “Off-line signature verification systems: a survey”, in Proceedings of the International Conference, Workshop on Emerging Trends in Technology, Mumbai, Maharashtra, India, pp. 652-657, 2011.
[15] G. Pirlo, D. Impedovo, and M. Fairhurst, "Front matter," Advances in Digital Handwritten Signature Processing, pp. i-xiii: World Scientific, 2014.
[16] J. Wen, B. Fang, L. Zhang et al., “Off-line Signature Verification Based on Multi-scale Local Structural Pattern,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 31, no. 6, pp. 1-16, 2017.
[17] J. Galbally, M. Diaz-Cabrera, M. A. Ferrer et al., “On-line signature recognition through the combination of real dynamic data and synthetically generated static data,” Pattern Recognition,. 48(9):2921-2934, 9, 2015.
[18] M. I. Malik, M. Liwicki, and A. Dengel, "Local features for off-line forensic signature verification " Advances in Digital Handwritten Signature Processing, pp. 95-109: World Scientific, 2014.
[19] M. I. Malik, M. Liwicki, A. Dengel et al., “Automatic Signature Stability Analysis and Verification Using Local Features,” in Proceedings 14th International Conference on Frontiers in Handwriting Recognition, pp. 621-626, 2014.
[20] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Learning features for offline handwritten signature verification using deep convolutional neural networks,” Pattern Recognition, 70(10):163-176, 2017.
[21] M. I. Malik, and M. Liwicki, “From Terminology to Evaluation: Performance Assessment of Automatic Signature Verification Systems,” in Proceedings International Conference on Frontiers in Handwriting Recognition, pp. 613-618.
[22] E. N. Zois, I. Theodorakopoulos, D. Tsourounis et al., "Parsimonious Coding and Verification of Offline Handwritten Signatures." pp. 636-645.
[23] E. N. Zois, I. Theodorakopoulos, and G. Economou, “Offline Handwritten Signature Modeling and Verification Based on Archetypal Analysis,” in Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 5515-5524, 2017,
[24] E. N. Zois, L. Alewijnse, and G. Economou, “Offline signature verification and quality characterization using poset-oriented grid features,” Pattern Recognition, vol. 54, pp. 162-177, 2016.
[25] A. Soleimani, B. N. Araabi, and K. Fouladi, “Deep Multitask Metric Learning for Offline Signature Verification,” Pattern Recognition Letters, 80:84-90, 2016.
[26] S. Dey, A. Dutta, J. I. Toledo et al., “SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification,” arXiv preprint arXiv:1707.02131, 2017.
[27] M. Okawa, “Vector of locally aggregated descriptors with KAZE features for offline signature verification,” in Proceedings of IEEE 5th Global Conference on Consumer Electronics, pp. 1-5, 2016.
[28] 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, pp. 184-190, 2016,
[29] Y. Serdouk, H. Nemmour, and Y. Chibani, “Handwritten signature verification using the quad-tree histogram of templates and a Support Vector-based artificial immune classification,” Image and Vision Computing, 66: 26-35, 2017.
[30] B. Ribeiro, I. Gonçalves, S. Santos et al., “Deep Learning Networks for Off-Line Handwritten Signature Recognition,” in Proceedings of the 16th Iberoamerican Congress conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, CIARP-11, Berlin, Heidelberg, pp. 523-532, 2011.
[31] Khalajzadeh Hurieh, M. Mansouri, and M. Teshnehlab, “Persian Signature Verification using Convolutional Neural

Networks," International Journal of Engineering Research and Technology (IJERT), 1(2):7-12, 2012.
[32] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-independent feature learning for Offline Signature Verification using Deep Convolutional Neural Networks", Proceedings of International Joint Conference on Neural Networks (IJCNN), pp. 2576-2583, 2016.
[33] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Analyzing features learned for Offline Signature Verification using Deep CNNs," in Proceedings of 23rd International Conference on Pattern Recognition (ICPR), pp. 2989-2994, 2016.
[34] J. Wright, Y. Ma, J. Mairal et al., "Sparse Representation for Computer Vision and Pattern Recognition," Proceedings of the IEEE, 98(6):1031-1044, 2010.
[35] F. Bach, R. Jenatton, J. Mairal et al., "Optimization with Sparsity-Inducing Penalties," Foundations and Trends in Machine Learning, 4(1):1-106, 2012.
[36] H. Cheng, Z. Liu, L. Yang et al., "Sparse representation and learning in visual recognition: Theory and applications," Signal Processing, 93(6):1408-1425, 2013.
[37] R. Jenatton, J. Mairal, G. Obozinski et al., "Proximal methods for hierarchical sparse coding," Journal of Machine Learning Research, 12:2297-2334, 2011.
[38] R. G. Baraniuk, V. Cevher, M. F. Duarte et al., "Model-Based Compressive Sensing," IEEE Transactions on Information Theory, 56(4):1982-2001, 2010.
[39] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection", The Annals of Statistics, 37(6A):3468-3497, 2009.
[40] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," Journal of Machine Learning Research, 12:3371-3412, 2011.
[41] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada, pp. 433-440, 2009
[42] Z. S. Jie Gui, Shuiwang Ji, Dacheng Tao, Tieniu Tan "Feature Selection Based on Structured Sparsity: A Comprehensive Study," IEEE Transactions on Neural Networks and Learning Systems, 28(7):1490-1507, 2017.
[43] C. A. Micchelli, J. M. Morales, and M. Pontil, "Regularizers for structured sparsity", Advances in Computational Mathematics, 38(3):455-489, 2013.
[44] J. Morales, C. A. Micchelli, and M. Pontil, "A family of penalty functions for structured sparsity", Advances in Neural Information Processing Systems, pp. 1612-1623, 2010.
[45] D. L. Donoho, "CART and best-ortho-basis: a connection," Ann. Statist., 25(5):1870-1911, 1997.
[46] A. Dutta, U. Pal, and J. Lladós, "Compact correlated features for writer independent signature verification", Proceedings of 23rd International Conference on Pattern Recognition (ICPR), pp. 3422-3427, 2016.
[47] Ng Andrew, and Yu Kai. "ECCV-2010 Tutorial: Feature Learning for Image Classification, http:\\ufldl.stanford.edu/eccv10-tutorial/
[48] A. Beck, and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," SIAM Journal on Imaging Sciences, 2(1):183-202, 2009.
[49] 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.
[50] Z. Zhang, Y. Xu, J. Yang et al., "A Survey of Sparse Representation: Algorithms and Applications", IEEE Access, 3:490-530, 2015.
[51] J. Feng, B. Ni, Q. Tian et al., "Geometric lp-norm feature pooling for image classification," in Proceedings of CVPR 2011, pp. 2609-2704, 2011.
[52] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree", in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, pp. 464—472, 2016.
[53] 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.
[54] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon et al., "MCYT baseline corpus: a bimodal biometric database," IEE Proceedings Vision, Image and Signal Processing, 150(6):395-401, 2003.
[55] R. Kumar, L. Kundu, B. Chanda et al., "A writer-independent off-line signature verification system based on signature morphology," in Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia, pp. 261-265, 2010.
[56] R. Kumar, J. D. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," Pattern Recognition Letters, 33(3):301-308, 2012.
[57] S. Chen, and S. Srihari, "A New Off-line Signature Verification Method based on Graph", in Proceedings of 18th International Conference on Pattern Recognition, pp. 869-872, 2006.
[58] 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, 48(1):103-113, 2015.
[59] 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.
[60] R. K. Bharathi, and B. H. Shekar, "Off-line signature verification based on chain code histogram and Support Vector Machine", in Proceedings of International Conference on in Advances in Computing, Communications and Informatics (ICACCI), pp. 2063-2068, 2013.
[61] G. Ganapathi, and R. Nadarajan, "A Fuzzy Hybrid Framework for Offline Signature Verification". in Proceedings of 5th International Conference on Pattern Recognition and Machine Intelligence: PReMI 2013, P. Maji, A. Ghosh, M. N. Murty et al., eds., pp. 121-127, 2013.
[62] B. H. Shekar, R. K. Bharathi, and B. Pilar, "Local Morphological Pattern Spectrum Based Approach for Off-line Signature Verification," Proceedings of 5th

International Conference on Pattern Recognition and Machine Intelligence:, PReMI 2013, P. Maji, A. Ghosh, M. N. Murty et al., eds., pp. 335-342, 2013.
[63] 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.
[64] J. F. Vargas, M. A. Ferrer, C. M. Travieso et al., “Off-line signature verification based on grey level information using texture features,” Pattern Recognition, 44(2):375-385, 2011.
[65] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez-Aguilar and J. Ortega-Garcia, “Off-line Signature Verification Using Contour Features”, in Proceedings of 11th International Conference on Frontiers in Handwriting Recognition, 2008.
[66] F. Alonso-Fernandez, M. C. Fairhurst, J. Fierrez-Aguilar and J. Ortega-Garcia, “Automatic Measures for Predicting Performance in Off-Line Signature”, in IEEE International Conference on Image Processing, pp. 369-372, 2007.
[67] 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., pp. 295-306: Springer Berlin Heidelberg, 2004.
[68] J. Wen, B. Fang, Y. Y. Tang et al., “Model-based signature verification with rotation invariant features,” Pattern Recognition, 42(7):1458-1466, 2009.
[69] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang et al., “Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network,” Applied Soft Computing, vol. 40(3):274-282, 2016.