A novel sparse model based forensic writer identification

Rajesh Kumar[a,*], Bhabatosh Chanda[b], J D Sharma[c]

[a] *Forensic Science Laboratory, Govt. of NCT of Delhi, India*
[b] *ECS Unit, Indian Statistical Institute, Kolkata, India*
[c] *Dr. HSG University, Sagar, M.P., India*

Abstract

The paper presents a novel method for writer identification based on sparse representation of handwritten structural primitives, called *graphemes* or *fraglets*. The proposed method is different from the existing grapheme based methods as the earlier methods use vector quantization based coding (clustering method) to get a document descriptor, while the proposed method uses *sparse coding* for the same. Literature shows that the sparse coding outperforms vector quantization in many real life applications including face recognition. Sparse coding can achieve comparatively much lower reconstruction error. Secondly, the sparsity allows representation to be specialized and can capture a writer specific features more accurately. Graphemes (fraglets) extracted from a document are represented in terms of Fourier and wavelet descriptors because the fraglet contour may be well described by its global as well as local characteristics. Wavelet descriptors also give a multi-resolution

*Corresponding Author. Fax:- +91 33 2575 2914
Email addresses: rajeshkumar512@gmail.com (Rajesh Kumar), chanda@isical.ac.in (Bhabatosh Chanda), jdsharma29@gmail.com (J D Sharma)

representation of the shape. Results have shown that even with a smaller codebook (than the earlier reported systems), the proposed method achieves better performance.
Keywords: Sparse coding, grapheme, Fourier descriptors, wavelet descriptors, writer identification.

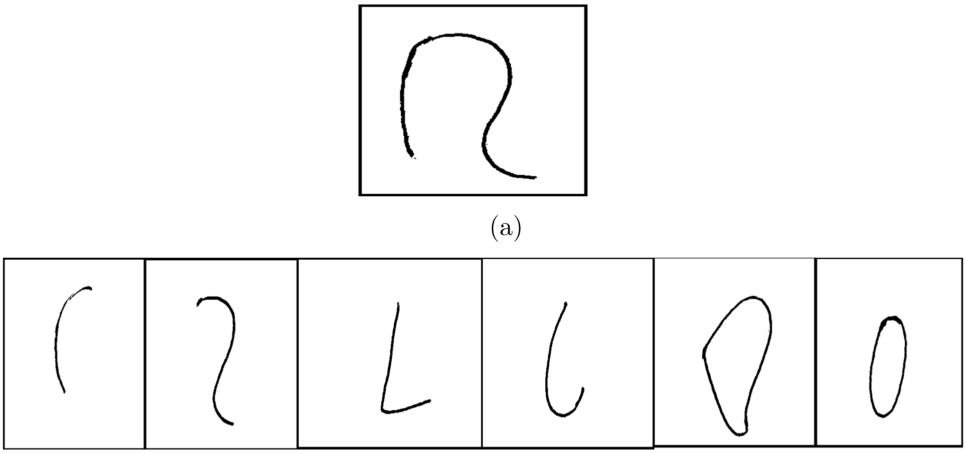
1. Introduction

Authentication of a person based on his (her) handwriting is one of the oldest biometric hallmarks. Even in the modern digital era, handwriting based authentication is frequently used for legal and general official purposes. Research in the area of automated writer identification is more than three decades old and there are enormous literature available in this domain. Besides the extensive reserach in this domain, the large intra-writer variations (also called *natural variation* in forensic literature) has made this problem still open in pattern recognition framework.

One of the earliest researchers who has contributed towards systematic examination of questioned documents including writer recognition, though manual, is Osborne (Osborne, 1929). A comprehensive review of the work done before 1989 is given in (Plamondon and G.Lorrete, 1989). Recent findings of the work done in this domain, may be found in (Bulacu and Schomaker, 2007; Plamondon and Srihari, 2000).

Automatic writer identification is using one of the two strategies: (i) text dependent and (ii) text independent (Bulacu and Schomaker, 2007). The text dependent approaches are based on the semantic content of segmented characters/words/lines. And the respective characters/words/lines are sought for the identification, details may be found in (Zois and Anastassopoulos, 2000; Franke and Koppen, 2001; Tomai et al., 2004; Srihari et al., 2002; Zhang and Srihari, 2003). Text independent approaches, on the other hand, do not depend on the semantic content. Text independent methods may also be attempted in one of the two ways: (i) whole document is considered as a text block and structural and textural features are extracted (Said et al., 1998; Marti et al., 2001; Hertel and Bunke, 2003) from the entire document/paragraph/line, or (ii) the grapheme based *codebooks* (Schomaker and Bulacu, 2004; Bensefia et al., 2005b,a; Bulacu and Schomaker, 2007) are prepared. In case of grapheme based method, content of the document is segmented into graphemes (or small fragments of text) and a codebook is prepared using some clustering techniques. Each of the fragments in the test document is represented in terms of exactly one codeword (or fraglet) of the codebook (collection of fragments obtained from clustering). The histogram of fragments is used as a descriptor of the document and used for the identification purpose.

The main drawback of clustering based method is that each fragment of the document is represented by one and only one fraglet of the codebook. There may be situations when it is not possible to represent a fragment with exactly one fraglet of the codebook. The fact may be clear from the illustration shown in Figure 1. One can see from the figure that the test fragment (a) can not be precisely represented by any of the fraglets shown in the codebook (b). This kind of problem may be avoided, at least for the given illustration, by taking a combination of some fraglets present in the codebook. For example, the test fragment (a) of the figure may closely be represented as some combination of the first and the second fraglets of the codebook (b).



(a)

(b)

Figure 1: Illustrates coding of fraglets. (a) Query fraglet, and (b) codebook of fraglets.

Sparse coding is based on similar concepts. The contributions of the present work are two-fold; first, to propose a text independent writer identification system, which is based on sparse coding. The approach is based on the assumption that an individual writer generates a particular kind of structural primitives (graphemes or fraglets) and the writing habits of that particular writer can be represented as a combination of such structural primitives. Representation of graphemes as a combination of Fourier and wavelet coefficients may be considered as another significant contribution of the paper. By doing so the fragment signatures have got both global as well as local characteristics. Another advantage of such representation is that it can easily be made translation, rotation and scale invariant.

The paper is organized as follows: Section 2 and 3 give a brief description of Fourier and wavelet descriptors and sparse coding respectively. The proposed writer identification system is discussed in Section 4. In Section 5, we discuss experimental results along with the description of the database used. Section 6 concludes the paper mentioning the scope of future work.

2. Fourier and wavelet descriptors

Shape is one of the best forms of visual information to describe an object. Various shape descriptors are reported in the literature, which are broadly classified as contour based features and region based features (Zhang and Lu, 2001). In case of handwritten samples, which are mostly in binary form, contour based features are found effective to represent the shape. Among contour based features, spectral descriptors like Fourier coefficients are popular ones. One of the advantages of the Fourier descriptors is that the first few low frequency coefficients of the Fourier transform capture the overall shape while the higher frequency terms capture its finer details. Besides, Fourier descriptors are easy to normalize (rotation, scale and translation) and preserve the overall shape information. In spite of so many advantages, Fourier descriptors fail to give multi-resolution representation. Wavelet descriptors, on the other hand, provide a multi-resolution representation and give a coarse-to-fine details of the shape. Unlike Fourier descriptors, wavelet descriptors achieve localization of shape feature in both spatial and frequency domains. We provide, next, a brief description of both Fourier and wavelet descriptors.

2.1. Fourier descriptors

A digital boundary, consisting of R points in an order (clockwise or anticlockwise), can be written as a sequence of complex numbers as $s(r) = x(r) + jy(r)$ for $r = 0, 1, \cdots, R-1$. Here, $(x(r), y(r))$, the coordinates of a contour point, is assumed to lie on a complex plane where x-axis is considered as real axis and y-axis as imaginary axis. The main advantage of this representation is that it converts a 2-D problem to a 1-D problem. The discrete Fourier transform (DFT) of $s(r)$ is

$$S(v) = \frac{1}{R} \sum_{r=0}^{R-1} s(r) e^{j2\pi vr/R}; \qquad v = 0, 1, \cdots, R-1 \tag{1}$$

The complex coefficients $S(v)$s are called the Fourier descriptors of the contour. Details can be found in (Gonzalez and Woods, 2008). The main advantage of the Fourier descriptors is that we can make the descriptors rotation, translation and scale invariant using some simple operations. For example, by taking $|S(v)|$s, one can make the Fourier descriptors rotation invariant. Similarly, the descriptors can be made translation and scale invariant making first coefficient zero and one (by normalization) respectively.

2.2. Wavelet descriptors

Wavelets are the functions that are generated from a single function ψ, known as *mother wavelet*, by dyadic down sampling and translations (Antonini et al., 1992; Chang and Kuo, 1993). Based on the mother wavelet, the k-th generation daughter Haar wavelet may be defined as

$$\psi_{k,n}(r) = 2^{-k/2}\psi(2^{kr} - n) \tag{2}$$

where k and n are integers denoting scale and position respectively. Due to orthonormal property, the wavelet coefficients of a signal $f(r)$ can be easily computed via

$$F_{k,n}(r) = \sum_{r=0}^{R-1} f(r)\psi_{k,n}(r) \tag{3}$$

Here $f(r)$ represents the signature of grapheme, which is nothing but a sequence of distance of the contour points from its centroid $(\bar{x}, \bar{y})$, defined as $f(r) = \sqrt{(x(r) - \bar{x})^2 + (y(r) - \bar{y})^2}$ for $r = 0, 1, 2, \cdots, R - 1$.

3. Sparse coding

Representation of a signal in terms of a few items of a learned dictionary has shown state-of-the-art performance in many application areas of signal processing, image processing, computer vision and pattern recognition (Wright et al., 2010; Mairal et al., 2010). Like a vector quantization based algorithm, sparse code based algorithms can also be broken into two parts: (i) to learn a set of basis vectors or codewords, often called dictionary or codebook, and (ii) given a dictionary of codewords, to encode an input vector in terms of those codewords or basis vectors. Although both steps are connected, it is not necessary to use the same encoding algorithm for both parts (Coates and Ng, 2011).

The formulation of sparse coding relies on the basic assumption that a vector $\mathbf{x}$ of m elements can be represented as a combination of (not necessarily orthogonal) a set of m dimensional basis vectors or codewords $\mathbf{D}_{m\times K} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \cdots, \mathbf{d}_K]$ from a dictionary (Olshausen and Field, 1996):

$$\mathbf{x} = \sum_{i=1}^{K} a_i \mathbf{d}_i = \mathbf{D}\mathbf{a} \tag{4}$$

where $\mathbf{d}_i \in \mathbb{R}^m$ and $\mathbf{a} \in \mathbb{R}^K$ is the sparse code vector, where $\mathbf{a} = (a_1, a_2, \cdots, a_K)^T$.

Choosing an appropriate dictionary $\mathbf{D}$ is an important task in learning phase of the sparse coding based method. Here, a_is are the coefficients and are desired mostly to be zero for a sparse representation. Now, a sparse model may be formulated as an optimization problem by constructing the following cost functions to be minimized (Olshausen and Field, 1996):

$$E = [Reconstruction\ error] + \lambda[sparseness\ of\ weight\ vector\ \mathbf{a}] \tag{5}$$

where λ is a positive constant that regulates the importance of the first term with respect to the second. The first term tries to minimize the reconstruction error, and the second term introduces the sparsity and tries to make only a few a_is to be non-zero.

The reconstruction error may be defined as a least square error between the actual vector and the reconstructed vector as

$$[Reconstruction\ error] = \sum_{h=1}^{m} [x_h - \sum_{i=1}^{K} a_i d_{ih}]^2 = ||\mathbf{x} - \mathbf{Da}||^2 \tag{6}$$

such that the p-th norm of the weight vector be sufficiently small, i.e., $||a_i||_p < T_{th}$. The threshold T_{th} being a small integer number imposes sparsity (2nd term of equn. (5)) in the representation. Detailed treatment of sparse coding and its applications may be found in (Wright et al., 2010; Mairal et al., 2010; Coates and Ng, 2011). Suppose the collection of training data is represented as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$. During the learning stage, the dictionary $\mathbf{D}$ and the corresponding weight vectors $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N)$ for N training data are both unknown and are both estimated alternatively until the system stabilizes. That means, given a set of training data $\mathbf{X}$ and size K of the dictionary, the training phase gives out a dictionary $\mathbf{D}$ such that each of the data vector $\mathbf{x}_i$ can be represented using $\mathbf{D}$ maintaining sparsity. In the analysis phase, on the other hand, given the dictionary $\mathbf{D}$, a test vector $\mathbf{x}$ is sparse coded following the equn. (4).

4. Proposed writer identification system

As shown in the Figure 2, the proposed method is based on representing the graphemes in terms of Fourier and wavelet descriptors. Then a dictionary or codebook is learnt from these descriptors of graphemes extracted from a

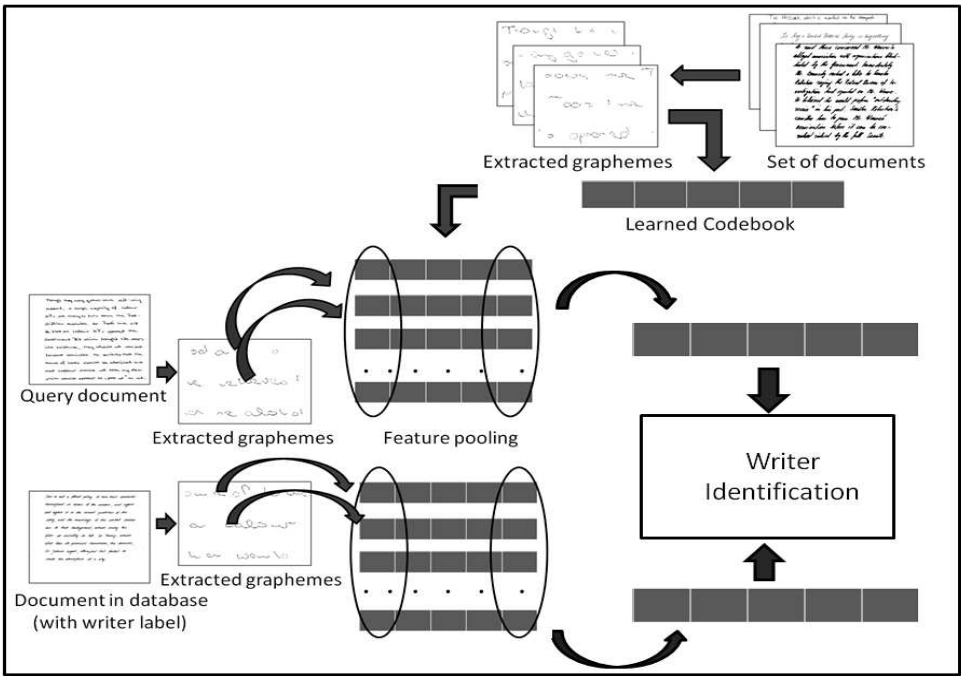


Figure 2: Schematic diagram of the proposed method. Major steps are grapheme extraction, sparse representation of graphemes based on codebook followed by pooling and finally classification.

set of document written by various writers. When a test document pertaining to a particular writer is examined for its authenticity, the various connected components present in the document is extracted. Each connected component is further segmented into graphemes. Each grapheme is represented by a vector formed by concatenating Fourier and wavelet descriptors. Since the dictionary of vectors representing graphemes is already learnt, the grapheme in hand is represented as a combination of vectors present in the dictionary. To represent the whole document, a pooling over all the sparse coded vectors is performed. Finally, the test document descriptor (which here is nothing but the writer descriptor) is matched with the document descriptors (with writer label) present in the database using simple nearest neighbor classifier. Each step is explained elaborately in the following subsections.

4.1. Extraction of graphemes

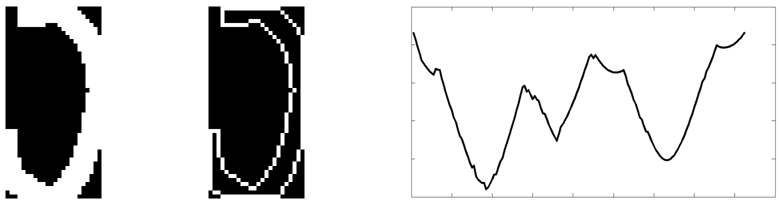
The documents, which are used for writer identification, are mostly in gray-scale. The document is first converted into binary image by Otsu method (Otsu, 1979) followed by connected component labeling. Then contour of each component is traced. To segment the connected component into graphemes, minima of lower and upper contours are sought. The contour of connected component is then segmented at the minima of lower contour which is proximal to a minima in upper contour (Schomaker and Bulacu, 2004).

All the graphemes extracted from the documents may not possess significant information for writer identification. We select only those graphemes, which have number of contour points greater than a threshold. This threshold is taken to be (mean - standard error) computed over all the graphemes present in all the documents. Standard error is the standard deviation of means of all the samples extracted from a population. This criterion filters out the graphemes of contour length less than the true mean of whole population of graphemes.

4.2. Representation of graphemes using Fourier and wavelet features

Once the graphemes of desired length are extracted from a document, we represent the contour of each grapheme in terms of Fourier and wavelet descriptors. To represent a grapheme in terms of Fourier descriptors, we take first 32 Fourier coefficients of coordinate of contour points. The number of Fourier coefficients is decided based on some preliminary experiments on the dataset. Finally, the Fourier coefficients are made rotation, translation, scaling and starting point invariant, the same way as mentioned in Section 2. To do so, the first and the second Fourier descriptors has to be zero and one respectively. Thus only 30 Fourier coefficients are actually contributing for the representation.

For the present problem, a grapheme is also represented using Haar wavelet descriptors. First, the centroid of the contour points is determined. Then a signature curve is generated by computing distance of each contour point from this centroid. The signature curve is normalized and is treated as a one-dimensional periodic function. A graphical representation of a signature curve for a grapheme is shown in Figure 3. Wavelet features are then extracted from this signature curve. For the present problem, we decompose the signature function up to fifth level using Haar wavelet transform and then the energy of each of the frequency bands are calculated. Thus a wavelet descriptors of 6 elements is obtained. On a preliminary experimentation on the dataset, it was found that the energy of these six frequency bands provide a satisfactory representation of the shape of a grapheme and makes the descriptors suitable for multi-resolution analysis.



(a) (b) (c)

Figure 3: Illustrates extraction of contour signature from graphemes. (a) Grapheme extracted from a document, (b) contour extracted from the grapheme, and (c) the signature curve obtained from the contour.

As discussed in Section 2, Fourier and wavelet gives two different kinds of representation of a grapheme. While Fourier descriptors give some kind of global nature of grapheme shape; wavelet descriptors provide localized multi-resolution information with coarse-to-fine details. A combination of these two descriptors may represent a grapheme more accurately. For appropriate representation of the shape of a grapheme, we concatenate these two descriptors as:

$$< \; Fourier\ descriptor, \;\; \mu \; \times \; wavelet\ descriptor \; >$$

where μ controls their relative importance. Thus each grapheme is represented by a feature vector of 36 elements. We call this vector as grapheme vector.

4.3. Learning a codebook of graphemes

A number of graphemes (N) are extracted from all the training documents written by writers. Each grapheme is represented as a column vector (say, $\mathbf{x}_i \in \mathbb{R}^m$) of m elements. The whole data set $\mathbf{X} \in \mathbb{R}^{m \times N}$ considered as $\mathbf{X} =$

$[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ is used to learn a dictionary $\mathbf{D} \in \mathbb{R}^{m \times K}$. Each column of the dictionary represents a codeword or a basis vector, $\mathbf{d}_k \in \mathbb{R}^m$. Each grapheme vector $\mathbf{x}_i \in \mathbf{X}$ is supposed to be reconstructed using a linear combination of vectors present in the dictionary. Let $\mathbf{a}_i \in \mathbb{R}^K$ be the reconstruction coefficients vector for the grapheme vector $\mathbf{x}_i$ corresponding to the dictionary $\mathbf{D}$. The coefficient matrix for the whole dataset used to learn the dictionary, may be written as $\mathbf{A}_{K \times N} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \cdots, \mathbf{a}_N]$

The goal is to design (learn) a dictionary $\mathbf{D}$ such that $\mathbf{X} \backsimeq \mathbf{DA}$, where the value of $||\mathbf{a}_i||_p$ should be sufficiently small, the small value of $||\mathbf{a}_i||_p$ as well as sparsity of $\mathbf{a}_i$ is usually determined by a threshold T_{th}. This requirement may be formulated as an optimization problem as

$$(\mathbf{A}^{\star}, \mathbf{D}^{\star}) = \arg\min_{\mathbf{A},\mathbf{D}}(||\mathbf{X} - \mathbf{DA}||^2 + \lambda||\mathbf{A}||_p) \tag{7}$$

The value of p may be 0 or 1 or 2, depending on either L_0 or L_1 or L_2 norms respectively is utilized to solve the problem. In the dictionary learning phase, as stated in Section 3, both $\mathbf{D}$ and $\mathbf{A}$ are unknown. So, they are estimated alternatively keeping the other one fixed and iteration continues until the system stabilizes. In the present work, we use SPAMS to learn the dictionary.

SPAMS (SPArse Modelling Software) is an open-source optimization toolbox which provides a very fast and efficient platform to learn the dictionary following equn. (7). For learning the dictionary, the matrix factorization algorithm due to (Mairal et al., 2010) which has been included in SPAMS, is used. Also a model selection method, called 'Least Angle Regression Algorithm (LARS)' (Efron et al., 2004) which is utilized for efficient sparse decomposition, has also been included in SPAMS. Details of these algorithms may be found in the respective papers while the details of SPAMS may be found at (SPAMS, 2011).

4.4. Document (writer) descriptor

Each grapheme vector of a test document, written by a specific writer, can be represented in terms of a combination of basis vectors or codewords present in the estimated dictionary of size K. Given the dictionary $\mathbf{D}^{\star}$, each grapheme vector is now sparse coded as

$$a = \arg\min_{\alpha}(||\mathbf{x} - \mathbf{D}^{\star}\alpha||^2 + \lambda||\alpha||_p) \tag{8}$$

Since the document contains many graphemes, we need to combine these grapheme vectors by means of some statistics of the codewords to represent the document. Two kinds of statistics are utilized for the task in the literature (Yang et al., 2009).

A document having M number of graphemes, may be written as $\mathbf{Y}_{K\times M}$. Thus $\mathbf{Y}_{K\times M}$ consists of M number of sparse code vectors $\mathbf{a}_i$ that are arranged in it as columns of K elements. Finally, the document is represented by a vector $\mathbf{z}$ of K elements formed by combining the sparse code of the document either by average pooling or by max pooling.

The $k-$th element of vector $\mathbf{z}$ due to average pooling over sparse codes may be expressed as

$$z_k = \sum_{j=1}^{M} |y_{kj}| \tag{9}$$

whereas the one due to max pooling may be defined as

$$z_k = \max\{|y_{k1}|, |y_{k2}|, |y_{k3}|, \cdots, |y_{kM}|\} \tag{10}$$

Thus the vector $\mathbf{z}$ represents a kind of distribution of different kinds of graphemes present in a document which is supposed to have a strong relation to the writing style of a writer.

4.5. Identification of a writer

Finally, for the identification of a writer, a nearest neighbor classifier is utilized in the space of coefficient vectors $\mathbf{z}$. The results are reported both in terms of nearest neighbor (Top 1) and also a hit in a list of 10 nearest matches (Top 10). Results are explained in the following section.

5. Experimental results and discussion

This section presents the results of the proposed method for writer identification. We have tested the algorithm on IAM dataset (Marti and Bunke, 2002). The proposed system is analyzed and compared with some existing systems which used similar framework (i.e., codebook or dictionary of graphemes) and reported results on the same database.

5.1. Description of dataset

The IAM off-line handwriting database is one of the frequently used handwriting database for evaluation purpose. The database contains forms of handwritten English text. It includes as many as 657 writers. The number of forms (transcript) written by a writer varies from 1 to 59. Out of 657 writers, 356 writers have written only a single page. Other 301 writers have written two pages or more. Here, for the experimentation, we include at most two documents written by each of the 650 writers. Note that among these 650 writers, 350 writers have only one document while other 300 writers have 2 documents. To utilize all the 650 writers, in experimentation, a fraction of document (say, $P\%$) of each of the writer is taken randomly for dictionary learning as well as for training and the remaining data ($(100 - P)\%$) for testing. A fraction of document is also used to optimize the parameters like regularization constant (λ), the weight parameter μ used to concatenate Fourier and wavelet descriptors and the size of the dictionary, K.

5.2. Results and discussion

Besides having a number of advantages, one of the drawback of the text-independent writer recognition is that there should be an adequate amount of writing such that the characteristics of a writer should be captured. In our experiment, we have divided each document into 10 parts. Out of these 10 parts, say, P parts are used for dictionary learning as well as for training and $(10 - P)$ parts are used for testing. Performance of the system versus P is shown in Figure 4 , keeping other parameters fixed ($\mu = 0.15$ and $\lambda = 0.15$). Since a document written by a writer in IAM database contains text ranging from 2 to 11 lines (or 80 to 600 graphemes), a small part of a document may not contain an adequate amount of writing to represent a writer. Figure 4 reveals the dependency of the system on the fraction of writing used for dictionary learning. It shows poor performance (due to improper dictionary learning) for low value of P.

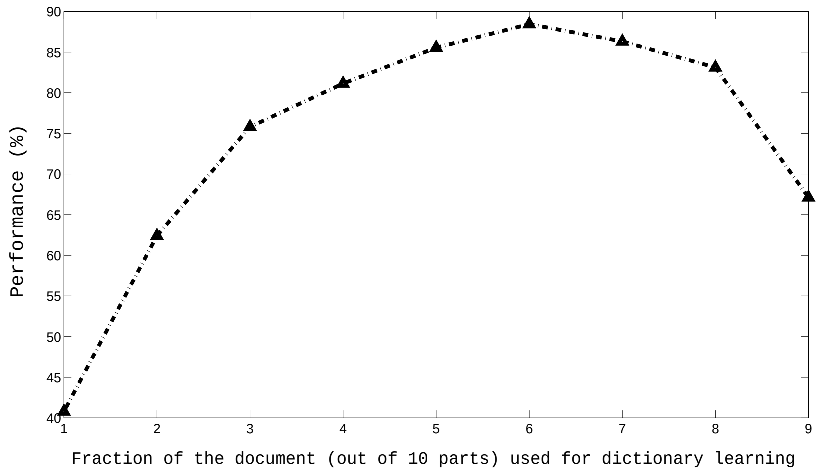


Figure 4: Variation in performance with the fraction of document (P out of 10 parts) used for dictionary learning

As shown in the figure, as the amount (i.e., P) of writings (or graphemes) increases the performance of the system also increases. On the other hand, the performance of the system drops if value of P is too high. The reason for decrease in performance of the system for higher value of P is the decrease in amount (i.e., $10 - P$ parts) of writings available for testing, which is unable to represent the characteristics of test writer properly. So, as expected, the best result is achieved where training and test part of the data are balanced, i.e., keeping the sixty percent of the data for dictionary learning and forty percent of data for testing. For rest of the experiments, we follow the same partition of the data.

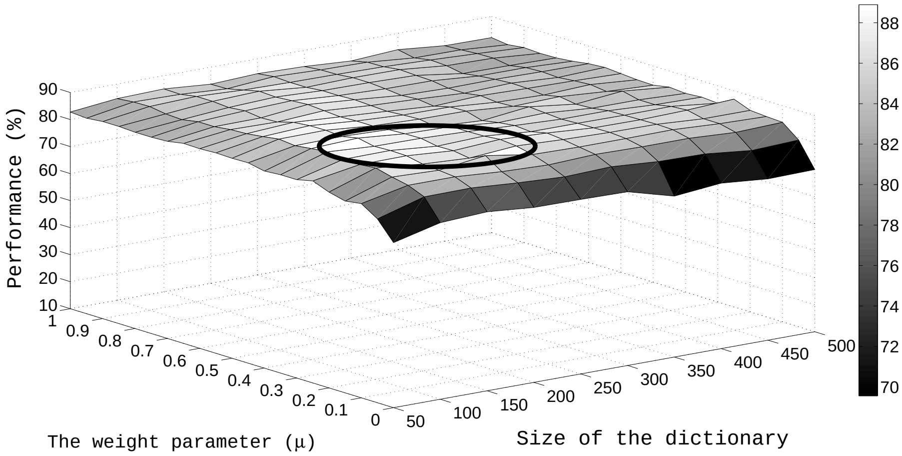
A broader picture of the performance on varying P has been shown in Table 1. In this context, it may be noted that performance of such recognition system for Top L candidates is quite appealing for many forensic applications as it reduces the search space for human experts. In addition to the top most candidate, Table 1 also reveals the performance of the proposed system for $L = 5$ and 10. From the table, it is clear that although there is a huge variation in the performance for Top 1 candidate when P is too high, the performance for Top 5 and Top 10 candidates are comparatively better and more stable even if the size of data used for learning the dictionary is high or the size to test the data is low.

Size of the dictionary (K) is one of the important parameters for any sparse model. In the present case, the weight parameter μ, used in concatenating Fourier and wavelet descriptors, is equally important. To see the effect of these two parameters on the performance, we have done extensive experimentation varying these two parameters utilizing grid method. The size of dictionary is varied from 50 to 500 by an interval of 50 while the weight parameter μ is varied from 0 to 1 with a step size of 0.05. The performance of the system using both the average and max pooling is depicted in Figure 5. As shown in the figure, the average pooling outperforms the

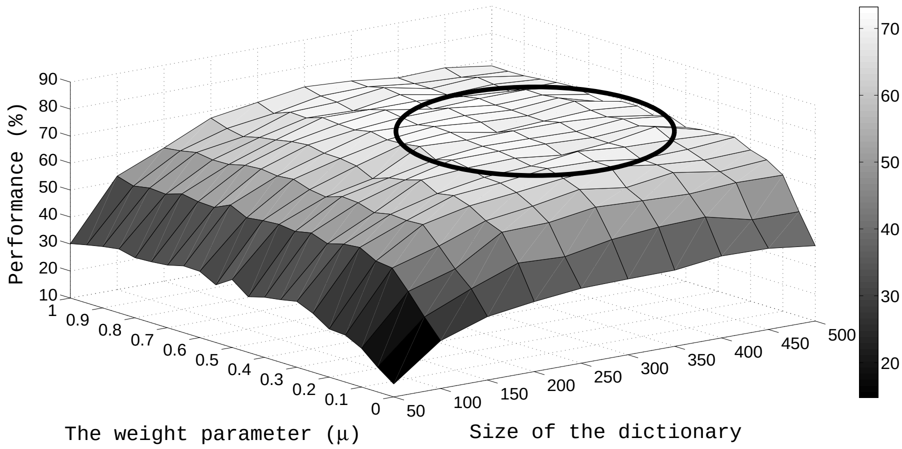
Table 1: Performance of the proposed method with the no. of partition used for dictionary learning

No. of parts used for		Performance		
dictionary learning and as training data	as test data	Top1	Top5	Top10
One	Nine	40.79	62.68	72.60
Two	Eight	62.40	81.89	88.13
Three	Seven	75.80	89.95	95.43
Four	Six	81.13	93.46	96.96
Five	Five	85.54	95.74	98.02
Six	Four	88.43	97.26	99.24
Seven	Three	86.30	95.28	98.33
Eight	Two	83.11	94.22	97.72
Nine	One	67.12	87.82	91.32

max pooling method. The region marked with the circle is the region where the performance is optimum. In the case of average pooling, performance of the system with a dictionary size 100 to 250 and the value of μ between 0.1 and 0.5 falls in high performance region. On the other hand, in case of max pooling, a dictionary size of 200 to 500 and μ=0.15 to 0.75 gives the best results. However, the maximum accuracy of 88.43% is achieved for average pooling (μ=0.15, dictionary size=200) and the maximum accuracy of 73.21% for max pooling (μ=0.75, dictionary size=300). The range of μ in both plots of Figure 5 suggests that the value of μ is not very critical and can be selected from a wide interval. However, for a better understanding of the dependency of performance on dictionary size, a plot of performance versus size of dictionary at $\mu = 0.15$ is shown in Figure 6. This figure also shows the fact that the performance of the system gets stable using a dictionary



(a)



(b)

Figure 5: Performance of the system with varying size of dictionary and weight parameter μ for (a) average pooling and (b) max pooling. The circled area depicts the region of maximum performance.

size 200 or more for average pooling.

The regularization parameter λ gives a trade-off between the reconstruction error and the coefficients of the basis vectors. The optimization of reg-

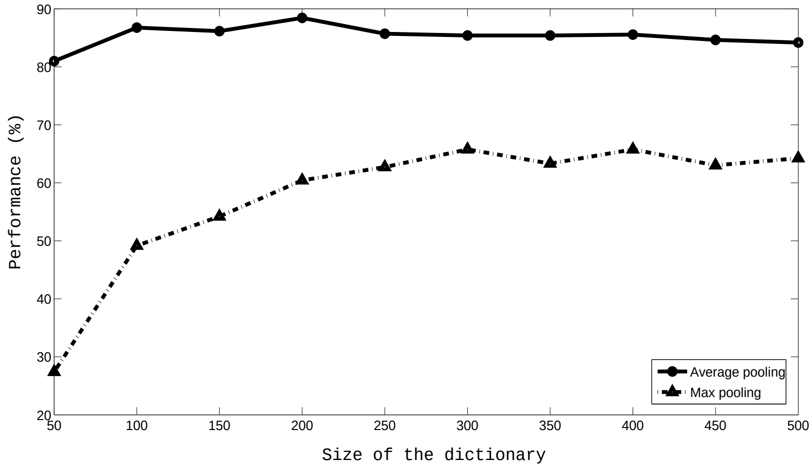


Figure 6: Variation in performance with dictionary size.

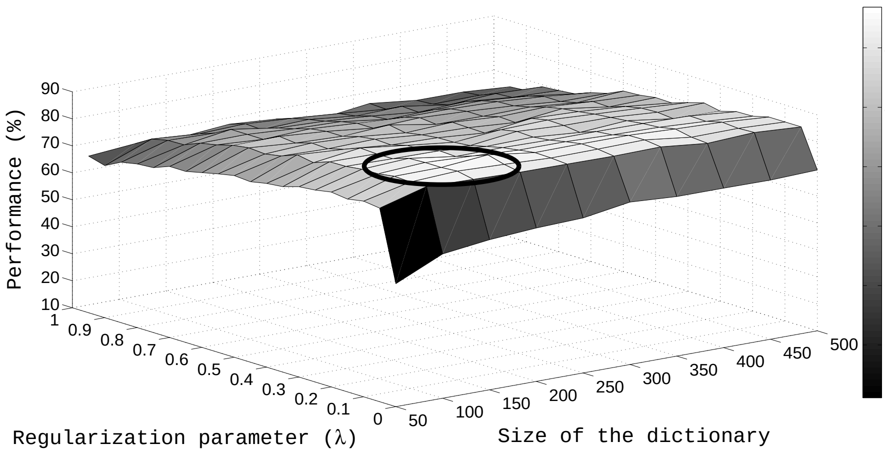


Figure 7: Performance of the system with varying size of dictionary and regularization parameter λ. Average pooling has been considered for describing the writer (document). The circled area depicts the region of maximum performance.

ularization parameter may be considered as a crucial step. In an effort to optimize the value of λ, we do some experiments on a fraction (P) of the data while varying λ (from 0 to 1 with step size 0.5) and size of the dictionary

Table 2: Comparison of performance of grapheme based methods reported on IAM database.

System	No. of writers	Size of the dictionary	Performance	
			Top 1	Top 10
Bensefia et al. (2005b)	150	***[1]	86.00	97.00
Bulacu and Schomaker (2007)	650	400	80.00	94.00
Siddiqi and Vincent (2010)	650	100	84.00	96.00
Proposed Method	650	200	88.43	99.24
		100	86.75	98.63

(from $K = 50$ to 500 with a step size 50). Since average pooling has given better performance than the max pooling, we adopt only average pooling for these experiments. Figure 7 depicts its outcomes. From the figure, one can see a general trend that as both λ and K increases there is a decrease in performance. However, the region of optimum performance is between $\lambda = 0.1$ to 0.5 and $K = 100$ to 250. The area is marked with a circle. Since $\lambda = 0.15$ with dictionary size 200 gives the best performance, we take $\lambda = 0.15$ throughout the experiments, in this paper.

Comparison of the proposed system with the state-of-the-art methods available is a usual practice to establish the efficacy of the new method. Often, it is not easy to provide a fair comparison between the systems. It is mostly due to lack of a common reference database and the large variability in the experiments carried out with different choice of parameters. In Table 2, we show a comparison of only grapheme based methods reported for

[1]Not apparent from the paper.

IAM database. For comparison purposes, the proposed method is tested for size 100 and 200, and the accuracy is measured as both Top 1 and Top 10 outcomes. From the table, one can see that the proposed system performs better than the other three systems, with a recognition accuracy of 88.43% to select the top most candidate and 99.24% for Top 10.

Computational complexity is always be an area of serious concern for a pattern recognition system. Besides knowing the fact that sparse coding based dictionary learning with SPAMS is much faster than a clustering based dictionary learning, the size of dictionary is one of the crucial parameters for dictionary based writer identification system. This directly influences the computational cost of the system and is needed to be explored. Keeping other parameters fixed, lesser the size of the dictionary be, faster is the system. In Figure 6, we have seen that the proposed system is mostly insensitive to the dictionary of size more than 100 and there is only a marginal variation in the performance of the system beyond this size. In Table 2, comparison of performance with the size of the dictionary is also noted. From the table, one can see that the proposed system is superior to the other ones also in terms of computational complexity (i.e., size of the dictionary). Even with a dictionary size 100, the performance of the proposed system is the best among the refereed systems.

Although, the proposed system utilizes whole IAM database for the experimentation purposes, it would be interesting to see how the proposed system behaves while varying the number of writers in the database. To see the performance against the number of writers, we select n number of writers out of 650 writers randomly and evaluate the performance of the system. This experiment is repeated 50 times and an average is noted as % accuracy. In Figure 8, we have shown this result varying n from 5 to 650. From the figure, one can see the performance of the system decreases as the number of writer increases. The decrease in the performance is obvious as the nearest neighbor classifier has to be compared with the larger number of writers where inter-writer distinction reduces progressively. Although rate of decrease in performance is high till 150 writers and it goes down as the number of writers increases.

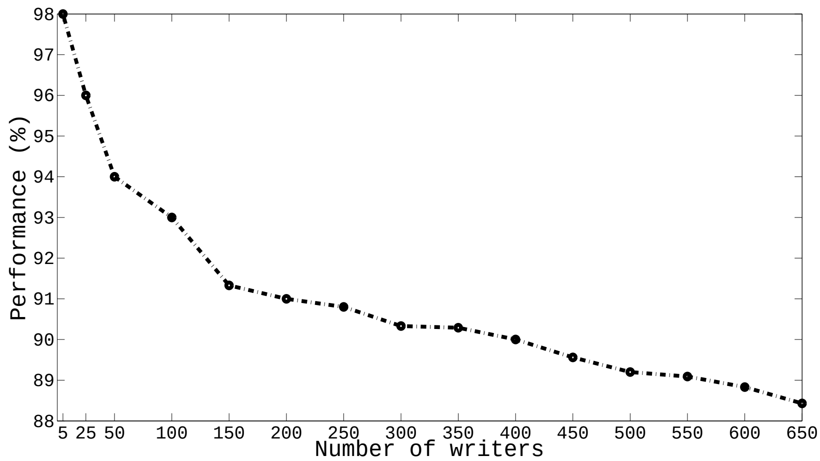


Figure 8: Performance of the proposed system with different number of writers. The performance is shown as identification rate, as Top 1 candidate.

6. Conclusion

In this paper, we have proposed a novel sparse coding based forensic writer identification. The proposed algorithm is different from earlier reported dictionary based writer identification techniques in the sense that the former use sparse coding to represent a grapheme as a combination of codewords present in the dictionary. Secondly, the earlier methods represent the graphemes in terms of contour in spatial domain and generate the codebook by clustering. On the other hand, the proposed method represents the graphemes in terms of Fourier and wavelet descriptors and learns dictionary through sparse modeling system. Results show that the proposed method provides a better representation of graphemes and achieves better identification of a writer. Even in the case, where the number of grapheme is very low, the proposed method appears to be useful. Extension of the proposed method for a larger database and using some other dictionary learning techniques are some useful future works.

Acknowledgements

Authors are thankful to the anonymous reviewers for their constructive suggestions which helped to improve this paper. The first author would like to thank Directorate of Forensic Science Services, Ministry of Home Affairs, Govt. of India for the financial support.

References

Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I., 1992. Image coding using wavelet transform. IEEE Transactions on Image Processing 1 (2), 205–220.

Bensefia, A., Paquet, T., L.Heutte, August 2005a. Handwritten Document Analysis for Automatic Writer Recognition. Electronic Letters on Computer Vision and Image Analysis 5 (2), 72–86.

Bensefia, A., Paquet, T., L.Heutte, Oct. 2005b. A Writer Identification and Verification System. Pattern Recognition Letters 26 (10), 2080–2092.

Bulacu, M., Schomaker, L., 2007. Text-independent writer identification and verification using textural and allographic features. IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (4), 701–717.

Chang, T., Kuo, C. C. J., 1993. Texture analysis and classification with tree-structured wavelet transform. IEEE Transactions on Image Processing 2 (4), 429–441.

Coates, A., Ng, A. Y., 2011. The importance of encoding versus training with sparse coding and vector quantization. In: Proceedings of ICML.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Annals of statistics 32 (2), 407–499.

Franke, K., Koppen, M., 2001. A computer-based system to support forensic studies on handwritten documents. IJDAR 3 (4), 218–231.

Gonzalez, R. C., Woods, R. E., 2008. Digital image processing. Prentice Hall.

Hertel, C., Bunke, H., 2003. A set of novel features for writer identification. In: AVBPA. Guildford, UK, pp. 679–687.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2010. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research 11, 19–60.

Marti, U., Bunke, H., 2002. The iam-database: An english sentence database for off-line handwriting recognition. Internation Journal on Document Analysis and Recognition 5, 39–46.

Marti, U.-V., Messerli, R., Bunke, H., 2001. Writer identification using text line based features. In: Proceedings of 6th International Conference on Document Analysis and Recognition. Seattle, WA, USA, pp. 101–105.

Olshausen, B. A., Field, D. J., 1996. Emergence of simple cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609.

Osborne, A. S., 1929. Questioned Documents. Boyd Printing Co., New York.

Otsu, N., 1979. A Threshold Selection Method from Gray-level Histogram. IEEE Transaction on System, Man and Cybernatics 6, 62–66.

Plamondon, R., G.Lorrete, 1989. Automatic signature verification and writer identification -the state of art. Pattern Recognition 22 (2), 107–131.

Plamondon, R., Srihari, S. N., Jan. 2000. On-line and Off-line Handwriting Recognition:A Comprehensive Survey. IEEE Transaction on Pattern Analysis and Machine Intelligence 22 (1), 63–84.

Said, H. E. S., Peake, G. S., Tan, T. N., Baker, K. D., 1998. Writer identification from non-uniformly skewed handwriting images. In: Proceedings of the British Machine Vision Conference. Southampton, UK, pp. 478–487.

Schomaker, L., Bulacu, M., 2004. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (6), 787–798.

Siddiqi, I., Vincent, N., 2010. Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. Pattern Recognition 43, 3853–3865.

SPAMS, 2011. SPArse modeling software.
URL `http://spams-devel.gforge.inria.fr/`

Srihari, S. N., Cha, S. H., Arora, H., Lee, S., 2002. Individuality of handwriting. Journal of Forensic Sciences 47 (4), 1–17.

Tomai, C. I., Zhang, B., S.N.Srihari, 2004. Discriminatory power of handwritten words for writer recognition. In: Seventeenth International Conference on Pattern Recognition, Cambridge. pp. 638–641.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Thomas S, H., Yan, S., 2010. Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE 98 (6), 1031–1044.

Yang, J., Yu, K., Gong, Y., Huang, T. S., 2009. Linear spatial pyramid matching using sparse coding for image classification. In: CVPR. pp. 1794–1801.

Zhang, B., Srihari, S. N., 2003. Analysis of handwriting individuality using word features. In: Seventh International Conference on Document Analysis and Recognition, Edinburgh. pp. 1142–1146.

Zhang, D., Lu, G., June 1-9 2001. A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. In: Proceedings of International Conference on Intelligent Multimedia and Distance Education (ICIMADE01). Fargo, ND, USA, pp. 1–9.

Zois, E. N., Anastassopoulos, V., 2000. Morphological waveform coding for writer identification. Pattern Recognition 33 (3), 385–398.