Expert Systems with Applications 40 (2013) 2069–2080

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Texture-based descriptors for writer identification and verification

D. Bertolini [a], L.S. Oliveira [a,*], E. Justino [b], R. Sabourin [c]

[a] Federal University of Parana (UFPR), R. Rua Cel. Francisco H. dos Santos, 100, Curitiba, PR 81531-990, Brazil
[b] Pontifical Catholic University of Parana (PUCPR), R. Imaculada Conceição, 1155, Curitiba, PR 80215-901, Brazil
[c] Ecole de Technologie Superieure, 1100 rue Notre Dame Ouest, Montreal, Quebec, Canada

ARTICLE INFO

Keywords:
Writer identification
Writer verification
Pattern recognition
Texture

ABSTRACT

In this work, we discuss the use of texture descriptors to perform writer verification and identification. We use a classification scheme based on dissimilarity representation, which has been successfully applied to verification problems. Besides assessing two texture descriptors (local binary patterns and local phase quantization), we also address important issues related to the dissimilarity representation, such as the impact of the number of references used for verification and identification, how the framework performs on the problem of writer identification, and how the dissimilarity-based approach compares to other feature-based strategies. In order to meet these objectives, we carry out experiments on two different datasets, the Brazilian forensic letters database and the IAM database. Through a series of comprehensive experiments, we show that both LBP- and LPQ-based classifiers are able to surpass previous results reported in the literature for the verification problem by about 5 percentage points. For the identification problem, the proposed approach using LPQ features is able to achieve accuracies of 96.7% and 99.2% on the BFL and IAM and databases respectively.

1. Introduction

In the last decade, several researchers have dedicated a considerable amount of effort to solving the problems of writer identification and verification. The former task concerns the retrieval of handwritten samples from a database using the handwritten sample under study as a graphical query. It provides a subset of relevant candidate documents on which complementary analysis will be performed by the expert. The latter task must, on its own, yield a decision as to whether or not two samples of handwriting were produced by the same writer (Bensefia, Paquet, & Heutte, 2005).

To cope with these problems, two different groups of approaches have been proposed in the literature. The first groups consists of local approaches, as they are based on specific features of the writing, and, in general, involve a segmentation process. Very often these features are localized, extracted from characters or allographs. Some examples of this strategy can be found in Bensefia et al. (2005), Bulacu, Schomaker, and Vuurpijl (2003), Kirli and Gulmezoglu (2011), Marti, Messerli, and Bunke (2001), Siddiqi and Vincent (2010), Srihari, Cha, Arora, and Lee (2002).

One of the main bottlenecks encountered in the local approaches occurs at the segmentation stage. As in many other pattern recognition problems, when the segmentation fails, most of the subsequent tasks, such as feature extraction and classification, are compromised. In order to avoid this outcome, some authors proposed a second group of approaches, referred to as global approaches. In this case, the methods identify the writer of a document based on the overall look and feel of the writing. In other words, a good way of avoiding segmentation is to look at handwriting as a texture. This strategy was first tried by Said, Tan, and Baker (2000), where the texture was represented by Gabor filters and the gray level co-occurrence matrix (GLCM). A similar strategy was employed by Bush, Boles, and Sridharan (2005) for script identification. More recently, Hanusiak, Oliveira, Justino, and Sabourin (2011) discussed the use of GLCM for author verification. They demonstrate in their experiments that texture-based features are a good alternative to author verification.

To the best of our knowledge, the best results using the global approach have been achieved using the well-known GLCM and its descriptors, which were proposed by Haralick, Shanmugan, and Dunstein (1973) almost 40 years ago. Since then, other texture descriptors have been developed and successfully applied in various areas. Two of them, local binary patterns (LBP) and local phase quantization (LPQ), have attracted a great deal of attention because of their performance in a number of applications. The concept of the LBP was first proposed by Ojala, Pietikäinen, and Harwook (1996) as a simple approach, robust in terms of grayscale variations, which proved its ability to efficiently discriminate among a wide range of rotated textures. Later, they extended their work (Ojala, Pietikäinen, & Mäenpää, 2002) to produce a grayscale and rotation invariant texture operator. The concept of LPQ was

* Corresponding author. Tel./fax: +55 41 33613655.
E-mail address: lesoliveira@inf.ufpr.br (L.S. Oliveira).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.10.016

originally proposed by Ojansivu and Heikkila (2008), and has been shown to be robust in terms of blur, and to outperform LBP in texture classification (Ojansivu, Rahtu, & Heikkila, 2008).

With this in mind, we consider here both LBP and LPQ as texture descriptors to perform writer verification and identification. We apply the same classification scheme used in Bertolini, Oliveira, Justino, and Sabourin (2010), Hanusiak et al. (2011), Rivard, Granger, and Sabourin (2011), based on the dissimilarity representation. This scheme, also known as the writer-independent (WI) approach, takes into account a dichotomy transformation which makes it possible to transform any *n*-class pattern recognition problem into a 2-class problem. This property allows us to design a verification/identification system, even when a limited number of samples from a large number of users is available. The underlying hypothesis of the dissimilarity-based approach is that the learning set is representative of the entire population of legitimate users enrolled in the verification/identification system (Rivard et al., 2011; Srihari et al., 2002). This means that it is not necessary to retrain the dissimilarity model each time an unknown writer is presented to the system.

In addition to exploiting various texture descriptors, we address the following issues that the work of Hanusiak et al. (2011) has left as open questions: (i) what is the impact of the number of references and the fusion rules used for verification and identification? (ii) how does the dissimilarity-based approach perform on the problem of writer identification? (iii) how does the dissimilarity-based approach compare to the feature-based approaches?

To answer these questions, we have carried out experiments on two different databases, the Brazilian forensic letter (BFL) database (Freitas, Oliveira, Sabourin, & Bortolozzi, 2008) and the IAM database (Marti & Bunke, 2002). First, we addressed the verification problem. Our results using LBP and LPQ surpass, by a considerable margin, those of the GLCM-based classifier introduced in Hanusiak et al. (2011) on the BFL database. In the case of the IAM database, the two texture descriptors also achieved remarkable results with error rates below 0.5%.

For the identification problem, we highlight, through a series of detailed experiments, the importance of the number of references available to correctly identify a given writer using the dissimilarity framework along with the texture features. To show the efficiency of this approach, we compare it to other two feature-based classification strategies. Our results show that the proposed approach is able to achieve accuracies of 94.7% on the BFL and 94.5% on the IAM using LPB features, and 99.2% on the BFL database and 96.7% on the IAM database using LPQ features. These results compare favorably with those of the state of the art.

This work is structured as follows: Section 2 describes the databases considered in our experiments. Section 3 describes the dissimilarity framework and how the dissimilarity feature vectors are created. Section 4 introduces the proposed texture-based feature sets and classification methods, and our experimental results and a discussion are presented in Section 5. Finally, Section 6 concludes this work and indicates some future directions for our work.

2. Databases

Two databases were considered in this work, the Brazilian forensic letter (BFL) database (Freitas et al., 2008) and the IAM database (Marti & Bunke, 2002). Both are described in the following subsections.

2.1. BFL database

This database is composed of 315 writers, with three samples per writer, for a total of 945 images. The samples were provided by undergraduate students in three different sessions over a one month period. The texts were collected on a sheet of white A4 paper with no pen-draw baseline, and then scanned in gray levels at 300 dpi (3760×2448 pixels). Each writer was allowed to use his/her own pen, which means that numerous different pens were used. The text is concise (131 words in Portuguese), and complete in the sense that it contains all the characters (letters and numerals) and certain character combinations of interest. This makes it suitable for text-dependent writer identification as well. Fig. 1a shows the sample content.

In order to validate the main hypothesis of the dissimilarity-based approach, i.e. that the writers used for training are representative of the entire population, we divided the database into two corpora: one composed of 200 writers for training, and the other of 115 writers for testing. Four different partitions for training were considered: 25, 50, 100, and 200 writers, the idea being to analyze the impact of the number of writers used for training on the overall performance. The remaining 115 writers were used for testing. Fig. 1 depicts a sample of the BFL database.

Using the algorithm described in Section 4, we extracted nine blocks of texture (256×256 pixels) from each image. Since we have three samples per writer, this adds up to 8505 texture images. Nine is the maximum number of fragments that we could extract from the writers for whom we had less handwriting information: although the number of words was the same, the characters were very small. In most cases, we could have extracted more fragments, but we decided to fix the number of fragments to 9, so that all the authors would be equally represented.

2.2. IAM database

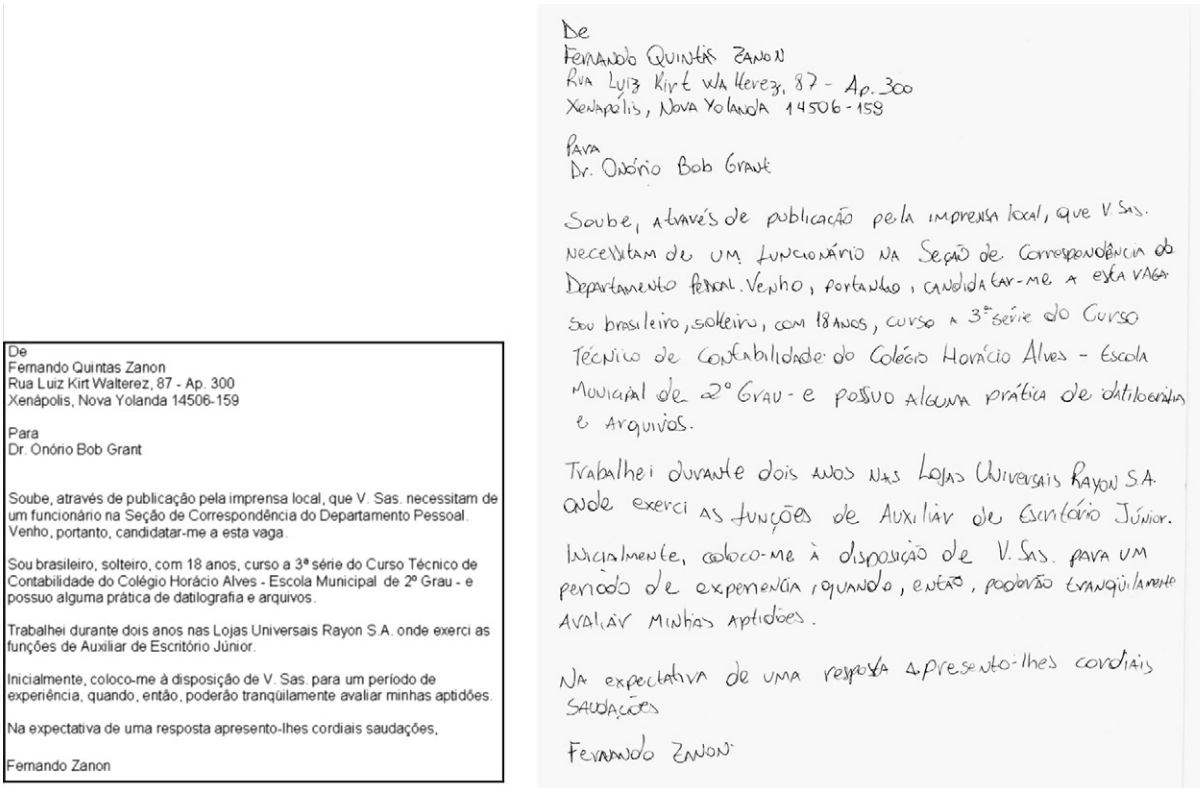
The IAM dataset (Marti & Bunke, 2002) is one of the best known and widely used databases in problems such as handwriting recognition and writer identification. It comprises forms with handwritten English text of variable content. The images have been scanned at 300 dpi, 8 bits/pixel, in gray-scale. A total of 650 writers have contributed to the dataset, with 350 writers having only one page, 300 writers with at least two pages, and 125 writers with at least four pages. Fig. 2a shows the distribution of the IAM database.

This database was divided in the same proportions as the BFL database, which means that the specimens contributed by the 650 writers represented in the database were divided into training and testing. As with the BFL database, four different partitions for training were considered: 50, 100, 205, and 410 writers, again, the idea being to analyze the impact of the number of writers used for training on the overall performance. The remaining 240 writers were used for testing. Fig. 2b shows an example of the IAM database.

Since some samples in this database contain only a few lines of handwriting, it was not possible to create 9 fragments of 256×256 pixels, which was desirable to facilitate the analysis of the experimental results. However, we were able to create 9 blocks of 256×128 pixels.

3. The dissimilarity framework

In this work, we have adopted the framework used by Hanusiak et al. (2011), which is based on a dichotomy transformation (Cha & Srihari, 2002) that makes it possible to reduce any insurmountable pattern recognition problem to a 2-class problem. Writer identification is an example of such a problem. Given a queried handwritten document and a reference handwritten document, the aim is to determine whether or not the two documents were produced by the same writer. Let V_i and Q_i be two vectors in the feature space, labeled l_V and l_Q respectively. Let Z_i be the dissimilarity feature



De
Fernando Quintas Zanon
Rua Luiz Kirt Walterez, 87 - Ap. 300
Xenápolis, Nova Yolanda 14506-159

Para
Dr. Onório Bob Grant

Soube, através de publicação pela imprensa local, que V. Sas. necessitam de um funcionário na Seção de Correspondência do Departamento Pessoal. Venho, portanto, candidatar-me a esta vaga.

Sou brasileiro, solteiro, com 18 anos, curso a 3ª série do Curso Técnico de Contabilidade do Colégio Horácio Alves - Escola Municipal de 2º Grau - e possuo alguma prática de datilografia e arquivos.

Trabalhei durante dois anos nas Lojas Universais Rayon S.A. onde exerci as funções de Auxiliar de Escritório Júnior.

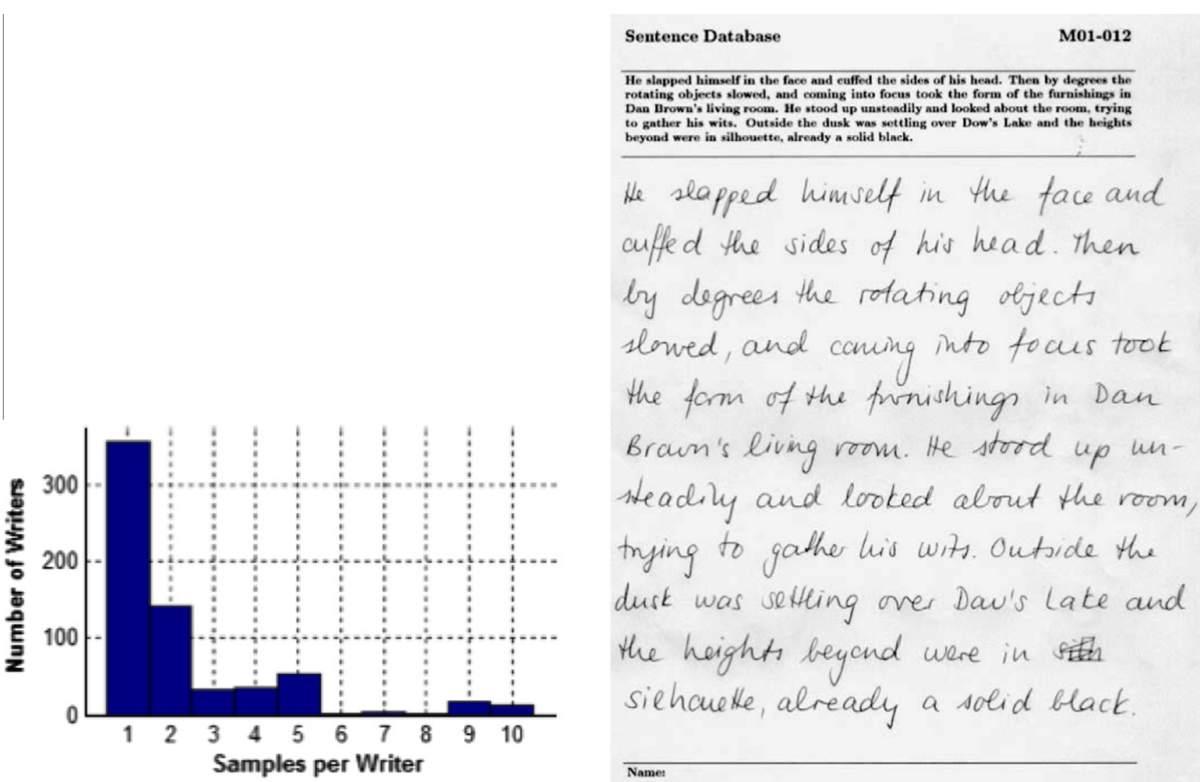
Inicialmente, coloco-me à disposição de V. Sas. para um período de experiência, quando, então, poderão tranqüilamente avaliar minhas aptidões.

Na expectativa de uma resposta apresento-lhes cordiais saudações,

Fernando Zanon

(a) (b)

Fig. 1. BFL database: (a) the contents of the Brazilian forensic letter and (b) a sample of the database.



(a) (b)

Fig. 2. IAM database (Marti & Bunke, 2002): (a) distribution and (b) sample of the database.

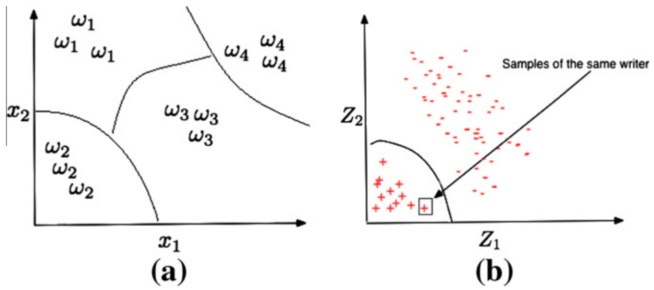


Fig. 3. Dichotomy transformation: (a) samples in the feature space and (b) samples in the dissimilarity space where (+) stands for the vectors associated to the within class and (−) stands for the vectors associated to the between class.

vector resulting from the dichotomy transformation $Z_i = |V_i - Q_i|$, where $|\cdot|$ is the absolute value. This dissimilarity feature vector has the same dimensionality as V_i and Q_i.

In the dissimilarity space, there are two classes that are independent of the number of writers: the within class (+) and the between class (−). The dissimilarity vector Z_i is assigned the label l_Z, according to Rivard et al. (2011):

$$l_Z = \begin{cases} + & \text{if } l_V = l_Q, \\ - & \text{otherwise}, \end{cases} \tag{1}$$

Fig. 3 illustrates this transformation. Suppose there are four writers, $\{\omega_1, \ldots, \omega_4\}$, and each one of them provides three samples. The feature extraction process extracts a vector (X_1, X_2) from each sample, and these are shown in Fig. 3a. Then, a dichotomy transformation takes place and computes the dissimilarity between the features of each pair of samples to form vectors (Z_1, Z_2). These vectors, which we call dissimilarity feature vectors, are shown in Fig. 3b.

We can see in Fig. 3 that the dichotomy transformation affects the geometry of the distribution. In the feature space, multiple boundaries are needed to separate all the writers. In the dissimilarity space, by contrast, only one boundary is necessary, since the problem is reduced to a 2-class classification problem. The number of samples in the dissimilarity space is larger, because these samples are made up of every pair of feature vectors. We can also see in Fig. 3 that, if both samples come from the same writer (genuine), then all the components of such a vector should be close to 0, otherwise they come from different writers (a forgery), in which case the components should be far from 0. This is true under favorable conditions. However, as in any other feature representation, the dissimilarity feature vector can be affected by intra-writer variability. This variability could generate values that are far from zero, even when the dissimilarity between the samples produced by the same writer is measured.

As mentioned earlier, one advantage of this approach is that even writers whose specimens were not used for training can be identified by the system. This characteristic is quite attractive, since it obviates the need to train a new model every time a new writer is introduced. In our experiments, we emphasize this feature by using disjoint sets of writers for training and testing.

The framework underpinning this work is depicted in Fig. 4. Initially, a handwritten document is converted to a texture image. Then, the texture is split into n equal parts, R_i ($i = 1, 2, \ldots, n$), which are sent to the feature extraction module. The resulting feature vectors, V_i, are stored in a database. The actual feature extraction process is discussed in Section 4. When a queried handwritten document is presented to the system, it is also converted to a texture and split into m equal parts, S_i($i = 1, 2, \ldots, m$). These m textures undergo the same feature extraction process, and so creating the feature vectors Q_i. Then, the dissimilarity feature vectors $Z_i = |V_i - Q_i|$ are computed and sent to the SVM classifier, which yields a decision on each dissimilarity feature vector. The final decision, D, is based on combining these partial decisions, and is obtained by means of a fusion rule. Section 5 discusses how this combination is achieved.

3.1. Dissimilarity feature vectors

The dissimilarity framework requires the classifiers to discriminate between genuine (positive) samples and forgeries (negative). To generate the positive samples, we computed the dissimilarity vectors among R genuine samples (references) of each writer (one segment of texture extracted from each letter), which resulted in $\binom{R}{2}$ different combinations. The same number of negative samples is generated by computing the dissimilarity between one reference of one author against one reference of other authors picked at random.

Considering, for example, 50 writers and three texture segments ($R = 3$) for training, we would have 150 (3×50) positive samples and 150 (3×50) negative samples. Fig. 5 exemplifies this process.

In Fig. 5a, V_a, V_b, and V_c are the reference feature vectors extracted from the reference images (e.g. texture segments) for a given writer. Based on these three vectors, three dissimilarity vectors (Z_1,Z_2, and Z_3) are computed. These are positive (genuine) dissimilarity vectors, which are expected to have components close to 0. A similar process is depicted in Fig. 5b to create the negative (forgery) dissimilarity vectors. In this case, the reference feature vectors are compared with the feature vectors of other authors picked at random, and it is expected that they will have components that are far from 0.

4. Building textures and extracting features

The texture blocks were built using the algorithm described in Hanusiak et al. (2011). To make this work self-contained, we describe the main steps of such an algorithm below.

First, the image is binarized using the Otsu algorithm and then scanned, top-down and left–right, to detect all the connected components of the image. The 8-adjacency was considered in this work. Small components, such as periods, commas, strokes, and noise, are discarded at this time. The bounding box of the

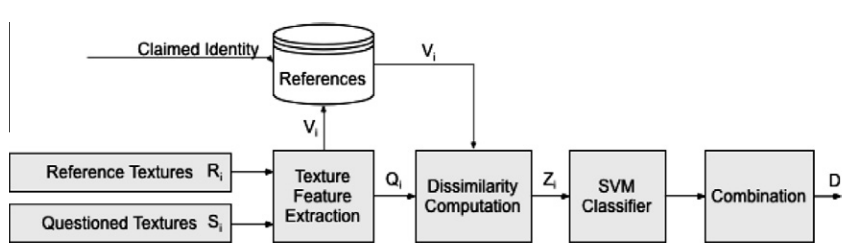


Fig. 4. The dissimilarity framework used for writer identification/verification (Hanusiak et al., 2011).

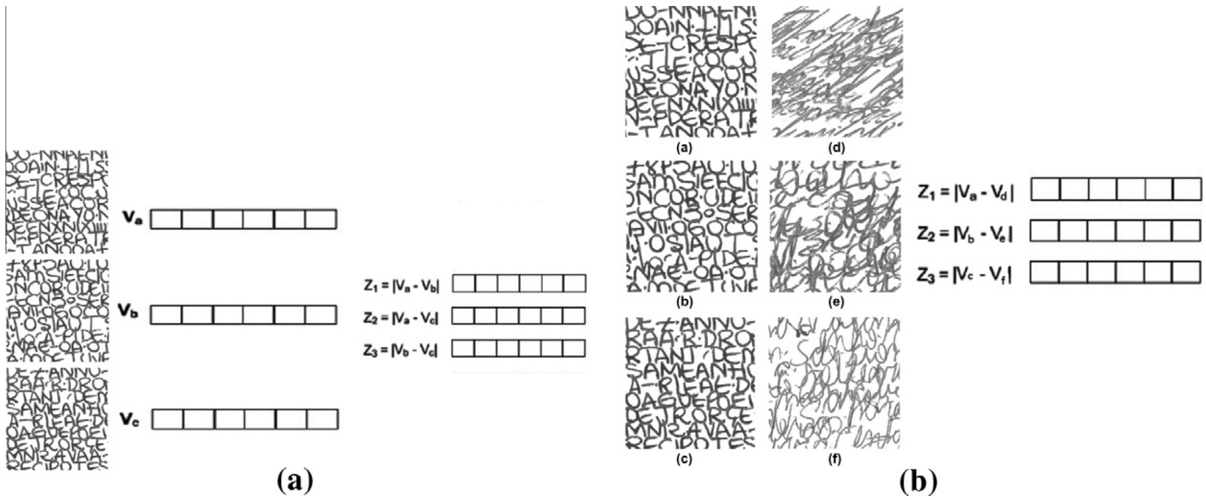


Fig. 5. Dissimilarities (a) among genuine samples of the same writer to generate the positive samples and (b) among genuine samples from different writers to generate the negative samples.

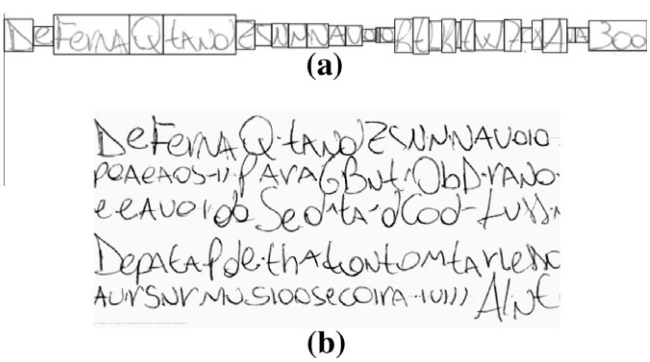


Fig. 6. The texture generation process (a) filling a line and (b) spaced texture.

remaining components is then used to extract the original components of the gray level image. The components in gray levels are then aligned with the new image using the center of mass of the bounding box. This process is depicted in Fig. 6a.

After filling the first line we compute the average height of all the connected components used in such a process. This value is used to define the y-coordinate of the next line, which is given by

$$new\_y = previous\_y + \frac{h}{2} \quad (2)$$

where the *previous_y* is the y-coordinate used to fill the previous line (in the case of the first line, a constant, $k = 150$, was used for both databases, BFL and IAM) and h is the average height of all the connected components used to fill the previous line. Reducing the gap between the lines by dividing h by two allows us to build more representative textures, otherwise the texture will contain too many blank spots, as in Fig. 6b. This denominator was found empirically. Figs. 7 and 8 show an example of the texture created from the original gray level letter for BFL and IAM respectively. The final texture image representing the writer's handwriting is finally segmented into blocks. In Section 5, we discuss the impact of using different block sizes.

This segmentation scheme differs from those presented in the literature (Bush et al., 2005; Said et al., 2000), in the sense that no preprocessing step, such as slant correction, is needed for line segmentation. Besides making the segmentation simpler, the proposed texture generation method also keeps some features, such as skew and slant.

4.1. Local binary patterns

Ojala et al. (2002) present a model to describe texture, called local binary patterns (LBP). In this model, each pixel C contains a set of neighbors P, equally spaced at a distance of R and C.

A histogram h is defined by the texture intensity differences between C and its neighbors, P. When the neighbors do not correspond to an image pixel integer value, that value is obtained by interpolation. An important characteristic of this descriptor is its invariance to changes in the value of the average intensity of the central pixels, when comparing it with its neighbors.

Considering the resulting sign of the difference between C and each neighbor P, by definition, we assign a result of 1 to a positive sign, and 0 otherwise. This makes it possible to obtain the invariance of the intensity value of pixels in gray scale format. With this information, the LBP value can be obtained by multiplying the binary elements for a binomial coefficient. So, a value $0 \leqslant C' \leqslant 2^P$ is generated, which is the value of the feature vector.

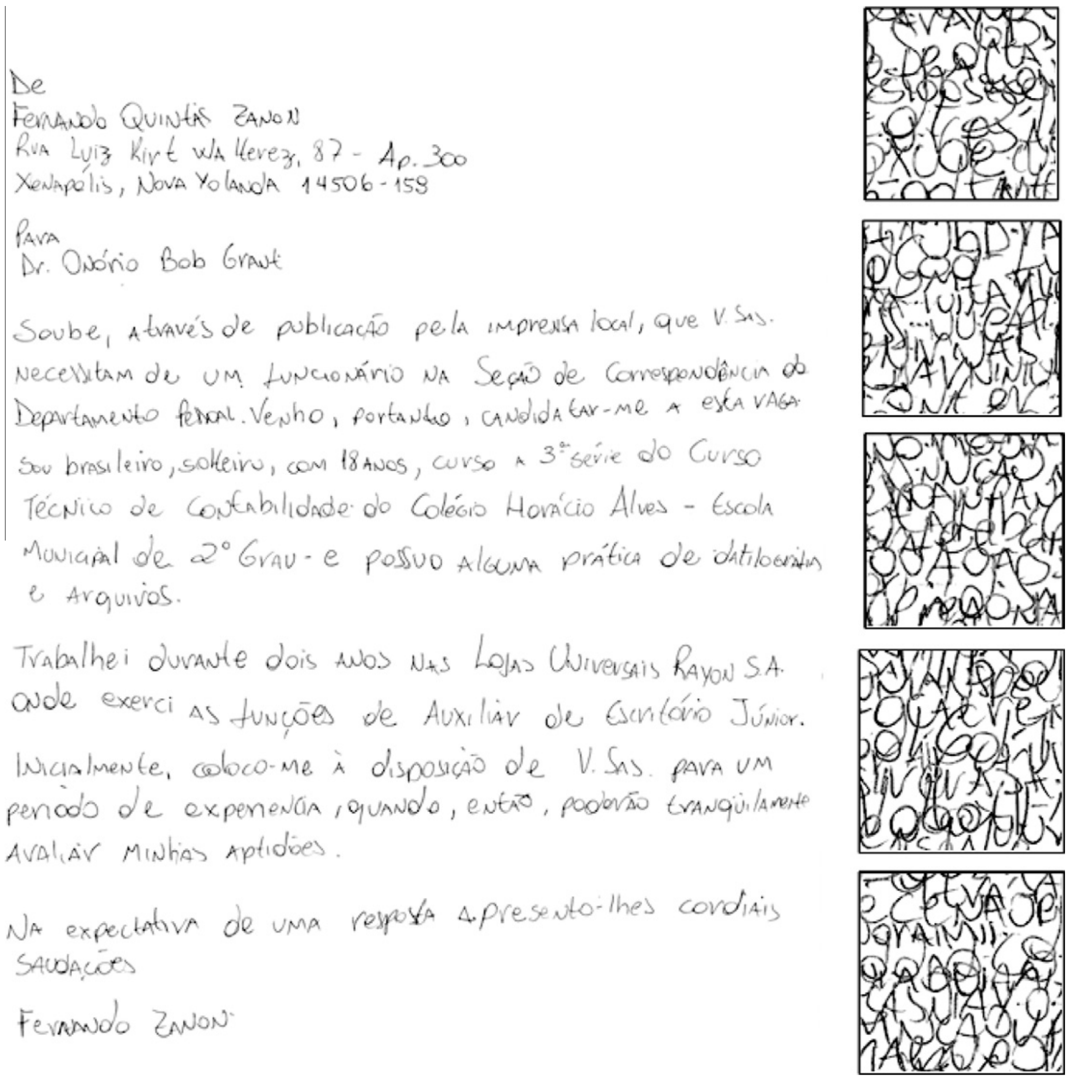
Observing the non uniformity of the vector obtained, Ojala et al. (2002) introduced a concept based on the transition between 0 s and 1 s in the LBP image. They explained that a binary LBP code is considered uniform if the number of transitions is less than or equal to 2, also considering that the code is seen as a circular list. That is, the code 00100100 is not considered uniform, because it contains four transitions, while the code 00100000, because it only has two transitions, is characterized as uniform. Fig. 9 illustrates this idea.

So, instead of using the whole histogram, the size of which is 2^P, we can use only the uniform values, which constitute a smaller feature vector. This version of the descriptor is called "$u2$", a label that accompanies the values of the radius R and the neighborhood size P, and so the definition of LBP is as follows: $LBP^{label}_{P,R}$.

A version of the descriptor rotation invariant is also defined, called "riu2'. But, in this work, the best results were provided by the uniform descriptor "u". Furthermore, we observed during the experiments that the feature extraction with $LBP^{u2}_{8,2}$ is fast and accurate enough for the proposed application. Then, we set $P = 8$ and $R = 2$ for the experiments described in this paper. This produces a feature vector of 59 components, which was normalized using the min–max rule.

4.2. Local phase quantization

The local phase quantization (LPQ) (Ojansivu & Heikkila, 2008) is based on the blur invariance property of the Fourier


De
Fernando Quintas Zanon
Rua Luiz Kirt Wallerez, 87 - Ap. 300
Xenapolis, Nova Yolanda 14506-158

Para
Dr. Onório Bob Grant

Soube, através de publicação pela imprensa local, que V. Sas. necessitam de um funcionário na Seção de Correspondência do Departamento Pessoal. Venho, portanto, candidatar-me a esta vaga. Sou brasileiro, solteiro, com 18 anos, curso a 3ª série do Curso Técnico de Contabilidade do Colégio Horácio Alves - Escola Municipal de 2º Grau - e possuo alguma prática de datilografia e arquivos.

Trabalhei durante dois anos nas Lojas Universais Rayon S.A. onde exerci as funções de Auxiliar de Escritório Júnior. Inicialmente, coloco-me à disposição de V. Sas. para um período de experiência, quando, então, poderão tranquilamente avaliar minhas aptidões.

Na expectativa de uma resposta apresento-lhes cordiais saudações

Fernando Zanon

Fig. 7. Original letter and texture blocks – sample from the BFL database.

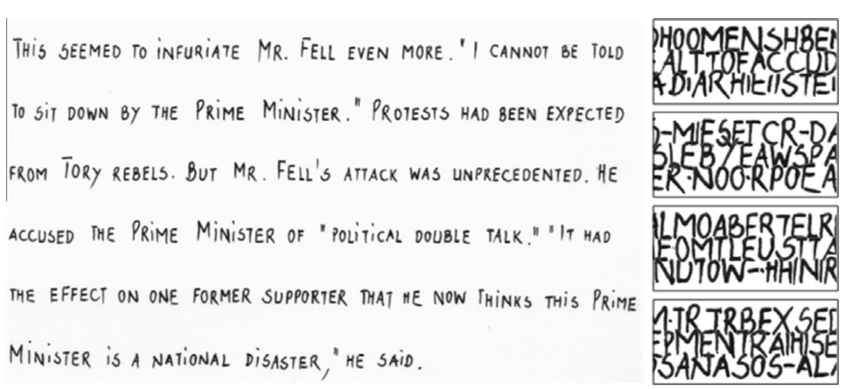

THIS SEEMED TO INFURIATE MR. FELL EVEN MORE. "I CANNOT BE TOLD TO SIT DOWN BY THE PRIME MINISTER." PROTESTS HAD BEEN EXPECTED FROM TORY REBELS. BUT MR. FELL'S ATTACK WAS UNPRECEDENTED. HE ACCUSED THE PRIME MINISTER OF "POLITICAL DOUBLE TALK." "IT HAD THE EFFECT ON ONE FORMER SUPPORTER THAT HE NOW THINKS THIS PRIME MINISTER IS A NATIONAL DISASTER," HE SAID.

Fig. 8. Original letter and texture blocks – sample from the IAM database.

phase spectrum. The local phase information of an $N \times N$ image $f(x)$ is extracted by the 2D DFT (short-term Fourier transform (STFT))

$$\hat{f}_{\mathbf{u}_i}(\mathbf{x}) = (f \times \Phi_{\mathbf{u}_i})\mathbf{x} \tag{3}$$

The filter $\Phi_{\mathbf{u}_i}$ is a complex valued $m \times m$ mask, defined in the discrete domain by

$$\Phi_{\mathbf{u}_i} = \left\{ e^{-j2\pi \mathbf{u}_i^T \mathbf{y}} | \mathbf{y} \in \mathbb{Z}^2; \|\mathbf{y}\|\infty \leqslant r \right\} \tag{4}$$

where $r = (m-1)/2$, and $\mathbf{u}_i$ is a 2D frequency vector. In LPQ only four complex coefficients are considered, corresponding to 2D frequencies $\mathbf{u}_1 = [a,0]^T$, $\mathbf{u}_2 = [0,a]^T$, $\mathbf{u}_3 = [a,a]^T$, and $\mathbf{u}_4 = [a, -a]^T$, where $a = 1/m$. For the sake of convenience, the STFT presented in Eq. (3) is expressed using the vector notation presented in Eq. (5)

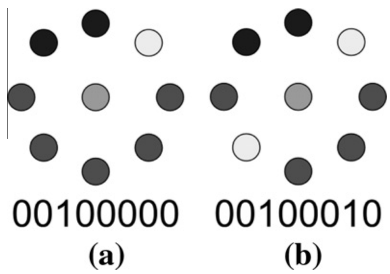


Fig. 9. LBP uniform pattern (Ojala et al., 2002). (a) The two transitions showed identifies the pattern as uniform, and (b) with four transitions, it is not considered a uniform pattern.

$$\hat{f}_{\mathbf{u}_i}(\mathbf{x}) = \mathbf{w}_{\mathbf{u}_i}^T \mathbf{f}(\mathbf{x}) \tag{5}$$

where $\mathbf{w_u}$ is the basis vector of the STFT at frequency $\mathbf{u}$ and $\mathbf{f}(\mathbf{x})$ is a vector of length m^2 containing the image pixel values from the $m \times m$ neighborhood of $\mathbf{x}$. Let

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2), \ldots, \mathbf{f}(\mathbf{x}_{N^2})] \tag{6}$$

denote an $m^2 \times N^2$ matrix that comprises the neighborhoods for all the pixels in the image and let

$$\mathbf{w} = [\mathbf{w}_R, \mathbf{w}_I]^T \tag{7}$$

where $\mathbf{w}_R = Re[\mathbf{w}_{\mathbf{u}_1}, \mathbf{w}_{\mathbf{u}_2}, \mathbf{w}_{\mathbf{u}_3}, \mathbf{w}_{\mathbf{u}_4}]$ and $\mathbf{w}_I = Im[\mathbf{w}_{\mathbf{u}_1}, \mathbf{w}_{\mathbf{u}_2}, \mathbf{w}_{\mathbf{u}_3}, \mathbf{w}_{\mathbf{u}_4}]$. In this case, $Re\{\cdot\}$ and $Im\{\cdot\}$ return the real and imaginary parts of a complex number, respectively.

The corresponding $8 \times N^2$ transformation matrix is given by

$$\widehat{F} = \mathbf{wF} \tag{8}$$

In Ojansivu and Heikkila (2008), the authors assume that the image function $f(x)$ is a result of a first order Markov process, where the correlation coefficient between two pixels x_i and x_j is exponentially related to their L^2 distance. Without a loss of generality, they define each pixel to have unit variance. For the vector $\mathbf{f}$, this leads to a $m^2 \times m^2$ covariance matrix C with elements given by

$$c_{i,j} = \sigma^{\|x_i - x_j\|} \tag{9}$$

where $\|\cdot\|$ stands for the L_2 norm. The covariance matrix of the Fourier coefficients can be obtained from

$$D = \mathbf{w}C\mathbf{w}^T \tag{10}$$

Since D is not a diagonal matrix, i.e., the coefficients are correlated, they can be decorrelated by using the whitening transformation $E = V^T \widehat{F}$ where V is an orthogonal matrix derived from the singular value decomposition (SVD) of the matrix D that is

$$D' = V^T D V \tag{11}$$

The whitened coefficients are then quantized using

$$q_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \geqslant 0, \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $e_{i,j}$ are the components of E. The quantized coefficients are represented as integer values from 0 to 255 using binary coding

$$b_j = \sum_{i=0}^{7} q_{i,j} 2^i \tag{13}$$

Finally, a histogram of these integer values from all the image positions is composed and used as a 256-dimensional feature vector in classification.

5. Experiments and discussion

Our experiments are divided into two parts. Those in the first part deal with verification, and those in the second part focus on the problem of writer identification. In all the experiments, support vector machines (SVM) were used as classifiers. The free parameters of the system and for SVM training were chosen using 5-fold cross validation. Various kernels were tried, and the best results were achieved using a Gaussian kernel. Parameters C and γ were determined through a grid search. The overall error rate that we used for evaluation purposes in this work is given by Eq. (14). This rate is always computed on the testing set.

$$\text{Overall error rate} = \frac{FP + FN}{TP + TN + FP + FN} \tag{14}$$

where FP, FN, TP, and TN stand for False Positive, False Negative, True Positive, and True Negative, respectively. These statistics are defined in the 2×2 confusion matrix depicted in Fig. 10.

One of the limitations of SVMs is that they do not work in a probabilistic framework. There are several situations where it would be very useful to have a classifier which produces a posterior probability $P(class|input)$. In our case, as depicted in Fig. 4, we are interested in estimating probabilities because we want to try different fusion strategies, like Sum, Max, Min, Average, and Median. Due to the benefits of having classifiers estimating probabilities, many researchers have been working on the problem of estimating probabilities with SVM classifiers (Milgram, Cheriet, & Sabourin, 2005; Platt, 1999; Sollich, 2002). In this work, we have adopted the strategy proposed by Platt (1999).

5.1. Writer verification

Writer verification is the task of determining whether or not a handwritten text has been written by a certain person. It is, by nature, a binary problem. Given an input feature vector x extracted from a text S and a claimed identity I, determine whether or not (I, x) belongs to class ω_1 or ω_2. If it belongs to ω_1, the claim is true, i.e. the text has been written by author I, and if it belongs to ω_2, the claim is false, i.e. the text was produced by an impostor. Unlike the identification problem, where the task consists of identifying I among all the writers enrolled in the system, the verification task performs a 1:1 comparison.

To establish some basis for comparison, we reproduced the best experiment reported in Hanusiak et al. (2011), but replaced the GLCM descriptors by LBP and LPQ. To this end, we used the BFL database with 100 and 115 writers for training and testing respectively. Five texture images per author were used as references ($R = 5$) to generate positive and negative samples, and five texture images ($S = 5$) were used for verification. The fusion rule applied to combine the classifier's output was the Sum rule, which performs best, according to Hanusiak et al. (2011). Now, from Table 1, it is easy to see that both LBP and LPQ produce significantly better results than the best results reported in Hanusiak et al. (2011).

As mentioned earlier, we divided the IAM database in the same way that we divided the BFL database, i.e. we fixed a subset for testing (240 writers) and different subsets for training (50, 100,

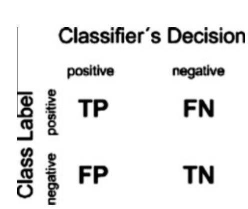


Fig. 10. 2×2 Confusion matrix.

Table 1
Comparison between GLCM (Hanusiak et al., 2011), LBP, and LPQ ($R = S = 5$).

Descriptor	Overall error rate (%)
GLCM (entropy) (Hanusiak et al., 2011)	5.0
LBP	1.3
LPQ	1.3

Table 2
Performance of the texture descriptor on the IAM database for different number of writers in the training set.

Descriptor	Fusion rule	Overall error rate (%)			
		Number of writers			
		25	100	205	410
LBP	Max	4.8	2.5	1.3	0.4
	Majority Vote	2.7	1.5	1.5	0.6
	Sum	1.5	**0.4**	0.4	0.4
LPQ	Max	1.4	0.4	2.5	1.2
	Majority Vote	0.8	1.2	1.5	1.5
	Sum	**0.4**	0.6	1.2	0.4

205, and 410 writers). As described in Section 2.2, some letters in the IAM database contain only a few lines of text, which does not allow us to generate fragments of 256 × 256 pixels. In order to have the same number of texture images considered in the BFL database, we used images of 256 × 128 pixels. The results for writer verification on the IAM database using different fusion rules and varying the number of writers on the training set are reported in Table 2.

In this experiment, we note that the behavior of LBP and LPQ differ slightly. While LBP achieves lower error rates as the number of writers in the training set increases, LPQ is able to achieve lower error rates using fewer writers for training. Table 2 shows that the classifier trained with LPQ is capable of achieving an error rate of 0.4% using no more than 25 writers for training.

In the light of the remarkable results on the verification problem for the two databases using texture images of 256 × 256 pixels (BFL) and 256 × 128 pixels (IAM), we wondered if we could reduce the size of the texture images without sacrificing performance. Smaller images have some advantages, such as processing smaller samples (e.g. with few lines of text), faster feature extraction, and the capacity to generate more pieces of texture, and hence more references. As we discuss later in this paper, a larger number of references is important when dealing with writer identification. These experiments were performed on the BFL database, which allows us to create different fragment sizes. In this case, 100 and 115 writers were considered for training and testing respectively. Table 3 shows the impact of using different sizes of texture image.

We can draw two conclusions from Table 3. First, very small texture images, such as those 64 × 64 pixels, are not suitable for either LBP or LPQ. Second, larger images, those larger than 256 × 128 pixels, can yield some reduction in the overall error rates; however, it appears that, beyond a certain point, larger images contain more variability, which does not help in reducing the error rates. This to some extent corroborates the findings presented in Hanusiak et al. (2011), where the authors show that texture images larger than 256 × 256 pixels do not bring about any improvement. It is worth noting, though, that the descriptors considered in that case were based on GLCM.

5.2. Writer identification

According to the definition provided in Jain, Ross, and Prabhakar (2004), the identification problem consists of identifying writer I among all the writers enrolled in the system. Given an input feature vector x from a texture image S, we determine the identity I_c, $c \in 1,2,\ldots,N$, where N is the number of writers enrolled in the system. Hence, $S \in I_c$, if $\max_c\{D_{model}(x, R_c)\}$, where D_{model} is the dissimilarity model trained to return an estimation of posterior probability, which indicates that S and the reference R_c belong to the same writer.

However, the identification system can also provide a list of documents that are similar to the queried document. The size of this list, also known as the hit list, can vary, e.g. 1, 5, or 10. The results are then expressed in terms of TOP-1, TOP-5, or TOP-10 writer identification performance. This means that a hit list will be considered correct if at least one version of the queried specimen appears on it.

Our baseline experiment for identification used the same protocol as we applied for verification, i.e. $R = S = 5$, and texture images of 256 × 256 pixels for the BFL database and 256 × 128 pixels for the IAM database. Tables 4 and 5 show the TOP-1 performance for the BFL and the IAM database respectively, using different fusion rules and four different training sets.

A look at the first part of Tables 4 and 5 reveals that the LBP-based classifier produces lower error rates as we increase the number of writers in the training set. For the IAM database, this reduction is more compelling. The error rate drops from 82% to 15.5% when the Median is used as the fusion rule. The second part of Tables 4 and 5 present the performance of the LPQ-based classifier.

Table 3
Performance of the texture descriptors on writer verification for different texture image sizes – BFL database ($R = S = 5$).

Size ($W \times H$)	Overall error rate (%)	
	LBP	LPQ
64 × 64	12.6	5.6
182 × 128	1.7	1.3
209 × 128	2.2	1.3
256 × 128	1.3	2.6
329 × 128	0.4	2.1
460 × 128	1.7	2.1
768 × 128	2.6	1.7
209 × 256	1.7	3.0
256 × 256	1.3	1.3
329 × 256	0.4	1.3
460 × 256	0.8	1.7
768 × 256	0.8	1.3

Table 4
Error rates (%) of the texture descriptors on the BFL database ($R = S = 5$).

Fusion rule	LBP Writers for training				LPQ Writers for training			
	25	50	100	200	25	50	100	200
Sum	30.4	27.0	17.9	13.1	11.3	13.0	7.8	6.0
Max	61.7	52.5	46.6	33.1	20.0	6.9	5.3	9.5
Product	31.3	27.0	18.3	13.1	12.1	15.6	10.4	7.8
Median	37.8	24.8	20.0	**12.8**	13.0	8.7	**5.2**	6.9

Table 5
Error rates (%) of the texture descriptors on the IAM database ($R = S = 5$).

Fusion rule	LBP Writers for training				LPQ Writers for training			
	50	100	205	410	50	100	205	410
Sum	80.4	77.5	31.3	20.0	25.4	20.8	19.5	17.5
Max	85.0	82.1	62.9	31.3	44.1	30.0	21.6	20.4
Product	80.2	79.2	33.0	22.5	27.9	22.0	21.6	20.0
Median	82.1	80.5	31.5	**15.5**	26.2	18.3	15.4	**12.0**

Table 6
Error rates (%) for different number of writes and training references (R) – BFL database ($S = 5$).

Number of references R	LBP Writers for training				LPQ Writers for training			
	25	50	100	200	25	50	100	200
3	52.8	18.3	18.5	18.3	11.3	9.5	5.2	6.0
5	37.8	24.8	20.8	12.8	13.0	8.7	5.2	6.9
7	52.2	32.2	18.3	8.7	7.8	7.8	7.8	6.9
9	31.3	19.2	13.9	7.9	12.1	4.3	7.8	6.5

Table 7
Error rates (%) for different number of writes and training references (R) – IAM database ($S = 5$).

Number of references R	LBP Writers for training				LPQ Writers for training			
	50	100	205	410	50	100	205	410
3	42.5	74.1	30.8	20.8	14.1	17.5	17.0	13.75
5	82.1	80.5	31.5	15.5	26.2	18.3	15.4	12.0
7	45.8	48.3	26.3	15.5	20.4	19.5	13.3	11.6
9	57.5	52.9	22.9	12.0	12.5	12.0	13.7	10.4

Table 8
Evolution of the number of texture images for identification – BFL database ($R = 9$).

Fusion rule	LBP Number of references				LPQ Number of references			
	$S = 3$	$S = 5$	$S = 7$	$S = 9$	$S = 3$	$S = 5$	$S = 7$	$S = 9$
Sum	21.8	7.8	3.5	6.1	11.3	7.8	0.8	0.8
Max	28.7	25.2	28.7	27.0	10.4	7.8	5.2	6.9
Product	22.6	10.5	6.1	9.6	11.3	9.5	4.3	3.4
Median	23.5	9.6	5.4	5.3	13.9	9.5	2.6	0.8
TOP-5	2.7	0.8	0.8	0.8	4.3	3.4	0.0	0.8
TOP-10	0.8	0.8	0.8	0.8	1.7	0.1	0.0	0.0

For the IAM database, we observe the same behavior, i.e. the larger the number of writers in the training set, the smaller the error, but, for the BLF database the LPQ-based classifier is able to achieve the best performance without using the largest training set. In this case, the lower error rate (5.2%) was achieved with 100 writers in the training set.

So far, we have used five texture images ($R = 5$) to generate the dissimilarity feature vectors and five texture images ($S = 5$) for identification. The fusion rules are then used to produce a final decision. One aspect worth investigating is the impact of the number of references per writer used for training and identification. By increasing R, we increase the number of positive and negative samples in the training set. By increasing S, we can rely on more data to produce a final decision.

Since, in the previous experiments, the Median was the best fusion rule, we decided to adopt it for the subsequent experiments. Tables 6 and 7 show the evolution of the number of training references (R) for both the BFL and the IAM databases. It is easy to see from these tables that increasing R reduces the overall error rates.

In spite of the size of R, Tables 6 and 7 are similar to Tables 4 and 5 respectively, in the sense that the LPQ-based classifier is able to achieve lower error rates with fewer writers in the training set.

By analyzing the errors and the hit lists produced by the classifiers, we note that, in most cases, the correct writer was not very far from the classifier's TOP-1 choice. With this in mind, we propose increasing the number of texture images for identification (S). The rationale behind this is that, if we could count on more data to make a decision, we would profit from the information available on the hit list. Tables 8 and 9 show the evolution of S for the BFL and the IAM database respectively. In both cases, we used the largest training set available. TOP-5 and TOP-10 are related to the Median rule.

By adding more texture images for identification, we are able to reduce the overall error rates considerably. Our best results were achieved with the LPQ-based classifier in both databases, 0.8% and 3.3% for BFL and IAM respectively. Compared with our baseline results reported in Tables 4 and 5, the error rates were reduced by 4.4 and 8.7 percentage points for BFL and IAM respectively.

Fig. 11 shows the cumulative match characteristic (CMC) curve Bolle, Connell, Pankanti, Ratha, and Senior (2005), which plots the probability of identification against the 1:N candidate list size returned. It shows the probability that a given user will appear on any of the candidate lists. The faster the CMC curve approaches 1, which indicates that the user always appears on a particular size of candidate list, the better the matching algorithm. In these figures, we have plotted the results achieved with the median fusion rule.

It is important to bear in mind that, in general, in this kind of system, a recognition rate of 100% on a TOP-1 hit list is not necessary, since a domain specialist can make a final decision based on a TOP-5 or a TOP-10 hit list. However, to be able to use the TOP-5 or the TOP-10 list efficiently, it is very important that these lists achieve a high performance. In our case, we can see from the CMC curves that we were able to reach performance above 99% in both databases, i.e. the correct answer is always on the TOP-5 hit list.

Table 10 summarizes several works on writer verification/identification reported in the literature. Comparing these results is not a straightforward task, since in some cases the database used is not publicly available. In the case of IAM, a more direct comparison is possible, since the dataset is publicly available. However, in several cases, the authors used only a subset of it. In our case, we divided the original dataset into two corpora of different writers, in order to better assess the dissimilarity framework. We believe that it is

Table 9
Evolution of the number of texture images for identification – IAM database ($R = 9$).

Fusion rule	LBP Number of references				LPQ Number of references			
	$S = 3$	$S = 5$	$S = 7$	$S = 9$	$S = 3$	$S = 5$	$S = 7$	$S = 9$
Sum	30.8	16.7	9.6	7.1	31.2	16.2	8.7	9.1
Max	27.5	30.8	24.2	22.5	23.7	19.5	14.5	12.9
Product	31.3	19.8	12.5	9.6	31.6	19.1	12.5	10.4
Median	31.3	16.3	11.7	**5.4**	32.0	10.4	6.2	**3.3**
TOP-5	5.9	1.7	0.0	0.4	8.7	1.6	0.8	0.0
TOP-10	2.1	0.8	0.0	0.0	2.5	1.2	0.8	0.0

not fair to have the same writers in both the training and testing groups when using a dissimilarity-based system. In spite of the different databases, Table 10 still gives us a good basis for comparison.

5.3. Impacts of the number of texture images used for identification

From the experiments reported so far, we can assert the importance of the number of texture images (S and R), as well as the fusion rules. In our experiments, we observed that, in general, the Median rule provides the best results. However, we wonder what could be gained by combining several decisions instead of relying on a single one. In other words, would it be better to use just one large texture image for identification, instead of m small ones?

The idea behind using several small texture images is to better represent the writer's variability, hoping that these images will provide a certain degree of complementarity which can be exploited by the fusion rules. In spite of the fact that the texture images are extracted from the same handwritten letter, it has been observed that one writer may use different writing styles in different parts of a single sample.

To show the importance of using multiple texture images for identification in the proposed approach, we designed an experiment on the BFL database using an LPQ-based classifier. Instead of splitting the large texture area created from the questionable handwritten sample into m pieces, we used it to perform an identification, i.e. $S = 1$ with a texture image of 2304×256 pixels. Since $S = 1$, the fusion rule is no longer necessary. Such a strategy produces an elevated error rate of about 44%. This experiment can be compared with the second part of Table 4. As we can see, in that case, our worst result, an error rate of 13.9%, was achieved using three references ($S = 3$).

The weak performance of this experiment can be explained by the fact that the large texture area contains all the variability of

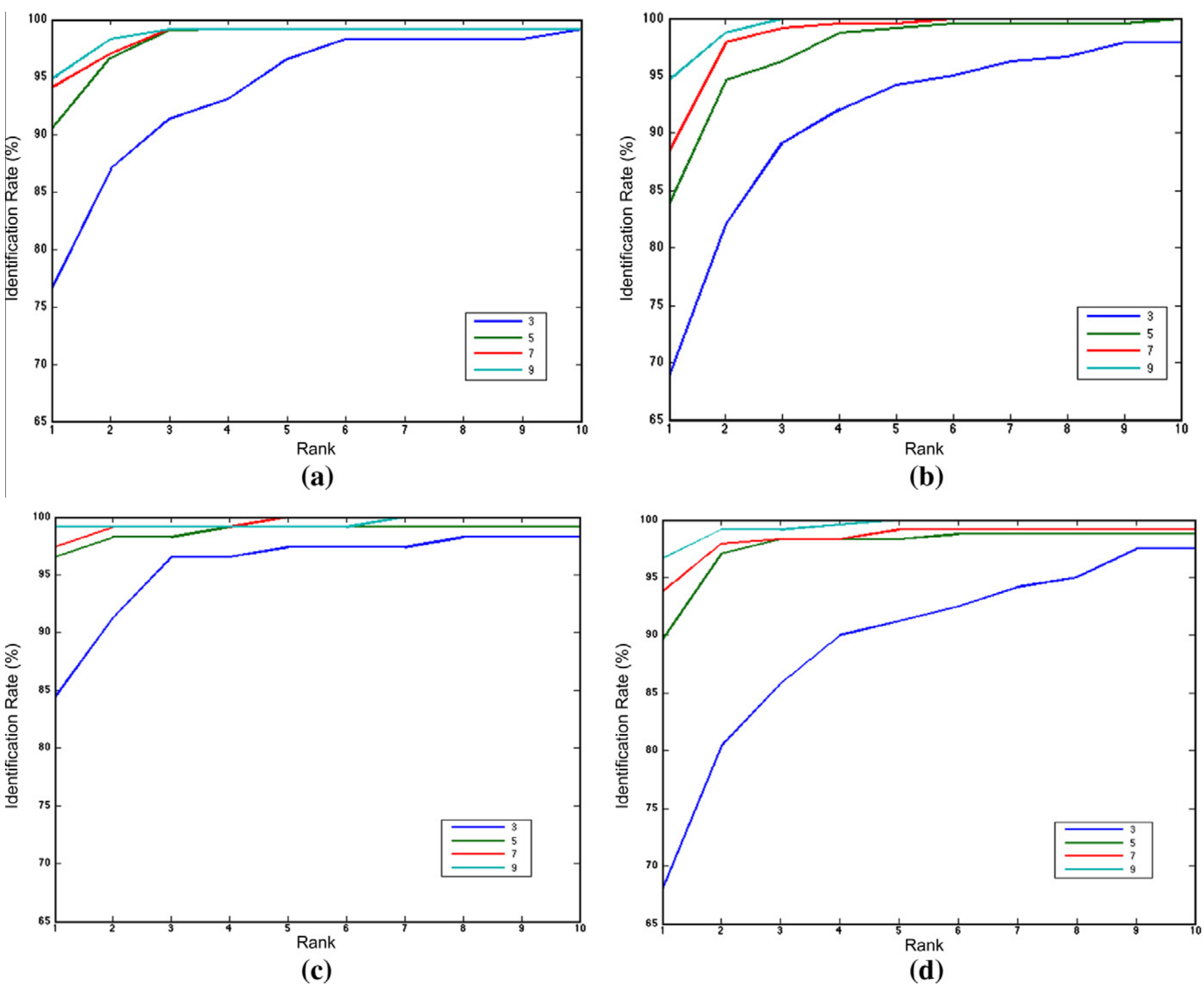


Fig. 11. The CMC curves for (a) LPB-based classifier on BFL database, (b) LPB-based classifier on IAM database, (c) LPQ-based classifier on BFL database, and (d) LPQ-based classifier on IAM database.

Table 10
Summary of the state of the art on author verification and identification.

Ref.	Data	Year	Features	Writers	Classifier	Performance (%)	
						Verification	Identification
Hanusiak et al. (2011)	BFL	2010	Texture	315	SVM	96.1	–
Marti et al. (2001)	IAM	2001	Structural	20	k-NN	–	90.7
Schlapbach and Bunke (2004)	IAM	2004	Geometric	120	HMM	97.5	96.5
Bensefia et al. (2005)	IAM	2005	Graphemes	150	VSM	96.0	86.0
Schomaker and Bulacu (2007)	IAM	2007	Graphenes	650	Dist. Hamming	97.2	89.0
Imdad et al. (2007)	IAM	2007	Directional	30	SVM	–	83.0
Schlapbach and Bunke (2007)	IAM	2007	Geometric	100	HMM	97.5	96.0
Schlapbach and Bunke (2008)	IAM	2008	Geometric	100	GMM	–	97.8
Siddiqi and Vincent (2010)	IAM	2010	Global and Local	650	Dist. X2	97.7	91.0
Kirli and Gulmezoglu (2011)	IAM	2011	Global and Local	93	NDDF	–	98.7
Said et al. (2000)	–	1998	Gabor e GLCM	40	WED	–	96.0
Zois and Anastassopoulos (2000)	–	1999	Morphological	50	MLP	–	96.5
Cha and Srihari (2000)	–	2002	Micro and Macro	1000	k-NN	–	81.0
Shen et al. (2002)	–	2002	Texture	50	k-NN	–	97.6
He and Tang (2004)	–	2004	Gabor	50	WED	–	97.0
Ubul et al. (2009)	–	2009	Gabor and ICA	55	k-NN	–	92.5
Ours	BFL	2012	Texture (LPQ)	315	SVM	99.4	99.2
Ours	IAM	2012	Texture (LPQ)	650	SVM	99.6	96.7

the queried document, but, after feature extraction, all that variability is lumped into the same feature vector.

5.4. Comparing with writer-dependent approaches

At this point, we may ask if the good performance reported in this work is due to: (i) the texture descriptors we used, (ii) the dissimilarity framework we used, or (iii) combining the dissimilarity framework with texture features. To address these points, we trained two other writer-dependent (WD) approaches. The writer-dependent or personal model is a feature-based approach that considers one model per author. Usually, it yields good results, but its drawback is that, for each new author, a new model should be built. Another important issue in this strategy is that a considerable amount of data is generally necessary to train a reliable model. In our case, the number of samples available for learning is small (9 fragments of texture per writer).

The first WD model we implemented was a multi-class SVM using a pairwise approach. In this strategy, the number of classifiers that should be trained is $q\,(q-1)/2$, where q is the number of classes (writers in our case). This approach shows its limitations as the number of writers increases.

The second strategy was one-against-others decomposition, which works by constructing an SVM ω_i for each class q which first separates that class from all the others. Compared to the pairwise approach, the one-against-others strategy is more suitable for our application, because only one new model must be trained each time a new writer is enrolled in the system.

In order to keep the same protocol, the number of classes (q) used in these experiments is the number of writers in the testing set, i.e. 115 for BFL and 240 for IAM. Unlike the dissimilarity protocol, both approaches, pairwise and one-against-others, need samples of the same writer in the training and testing sets. In the case of the BFL database, all the authors have three handwritten letters, and so we were able to split them into two samples (18 texture images of 256×256 pixels) for training and one letter for testing (9 texture images of 256×256 pixels). In the IAM, by contrast, some authors have contributed only two samples, which allows us to divide them into one sample for training (9 texture images of 256×128 pixels) and the other for testing (9 texture images of 256×128 pixels). Table 11 summarizes the results.

Regarding the questions raised at the beginning of this section, these results show us that the dissimilarity-based approach combined with the texture feature offer a robust framework for writer identification. In the case of the BFL database, where more samples are available for training, the dissimilarity approach achieved slightly better results. A considerably larger difference, though, can be observed for the IAM database, where the training set is smaller. The possibility of generating positive and negative samples using the dichotomy transformation makes the dissimilarity approach suitable, even when only a few samples per writer are available.

Table 11
Error rates (%) of different strategies of classification using the LPQ features.

Strategy	BFL	IAM
Dissimilarity	0.80	3.3
Pairwise	1.74	14.8
One-against-others	2.61	11.7

6. Conclusion

In this work, we have addressed the problems of writer verification and identification using the same framework as proposed by Hanusiak et al. (2011), in which explicit segmentation is avoided by generating a texture using the writer's handwriting. Thereafter, features are extracted from the texture, and dissimilarity feature vectors are used to train an SVM classifier. We have demonstrated that both LBP and LPQ are interesting alternatives for describing this kind of texture. As in Ojansivu et al. (2008), we have observed in our work that the classification accuracy of LPQ is higher than with the well-known LBP. However, both LPQ and LBP surpass by a considerable margin the results achieved by GLCM descriptors in writer verification.

Our experimental results show that the dissimilarity-based approach that we have successfully applied to verification problems is also a viable strategy for identification problems, in that it achieves a performance comparable to the state of the art. We have shown the importance of a larger number of references for testing in this approach, and of limiting that number of references (S and R) to nine, so that all the writers are equally represented. We also show that the dissimilarity approach compares favorably with classic classification approaches, such as the pairwise and one-against-others methods.

On aspect worth investigating is the upper limit to the number of references used for testing. We plan to extend the current protocol to use more references for testing when they are available. Based on the results reported in this work, we believe that we could reduce the errors for those writers even more if more handwriting were available. In future work, we also plan to investigate whether or not all the available writers in the training set are required, in order to build a good dissimilarity model.

Acknowledgement

This research has been supported by The National Council for Scientific and Technological Development (CNPq) Grant 301653/2011-9.

References

Bensefia, A., Paquet, T., & Heutte, L. (2005). A writer identification and verification system. *Pattern Recognition Letters, 26*(13), 2080–2092.

Bertolini, D., Oliveira, L. S., Justino, E., & Sabourin, R. (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition, 43*(1), 387–396.

Bolle, R. M., Connell, J. H., Pankanti, S., Ratha, N. K., & Senior, A. W. (2005). The relation between the ROC curve and the CMC. In *4th Workshop automatic identification advanced technologies* (pp. 15–20).

Bulacu, M., Schomaker, L., & Vuurpijl, L. (2003). Writer identification using edge-based directional features. In *8th International conference on document analysis and recognition, Edinburgh, Scotland* (pp. 937–941).

Bush, A., Boles, W., & Sridharan, S. (2005). On measuring the distance between histograms. *IEEE Transactions on Pattern Analysis and Machine Inteligence, 27*, 1721–1732.

Cha, S.-H., & Srihari, S. (2000). Multiple feature integration for writer verification. In *7th International workshop on frontiers on handwriting recognition* (pp. 333–342).

Cha, S.-H., & Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition, 35*, 1355–1370.

Freitas, C., Oliveira, L. S., Sabourin, R., & Bortolozzi, F. (2008). Brazilian forensic letter database. In *11th International workshop on frontiers on handwriting recognition, Montreal, Canada*.

Hanusiak, R., Oliveira, L. S., Justino, E., & Sabourin, R. (2011). Writer verification using texture-based features. *International Journal on Document Analysis and Recognition, 15*(3), 213–226.

Haralick, R. M., Shanmugan, K. S., & Dunstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, 3*(6), 610–621.

He, Z. Y., & Tang, Y. Y. (2004). Chinese handwriting-based writer identification by texture analysis. In *2004 International conference on machine learning and cybernetics* (Vol. 6, pp. 3488–3491).

Imdad, A., Bres, S., Eglin, V., Rivero-Moreno, C., & Emptoz, H. (2007). Writer identification using steered hermite features and svm. In *9th International conference on document analysis and recognition* (pp. 839–843).

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 14*, 4–20.

Kirli, O., & Gulmezoglu, M. (2011). Automatic writer identification from text line images. *International Journal on Document Analysis and Recognition*, 1–15.

Marti, U. V., & Bunke, H. (2002). The IAM-database: An english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition, 5*(1), 39–46.

Marti, U. V., Messerli, R., & Bunke, H. (2001). Writer identification using text line based features. In *8th International conference on document analysis and recognition, Seattle, USA* (pp. 101–105).

Milgram, J., Cheriet, M., & Sabourin, R. (2005). Estimating accurate multi-class probabilities with support vector machines. In *International joint conference on neural networks, Montreal, Canada* (pp. 1906–1911).

Ojala, T., Pietikäinen, M., & Harwook, D. (1996). Comparative study of texture measures with classification based on feature distributions. *Pattern Recognition, 29*, 51–59.

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(7).

Ojansivu, V., & Heikkila, J. (2008). Blur insensitive texture classification using local phase quantization. In *Proceedings of image and signal processing (ICISP 2008)* (pp. 236–243).

Ojansivu, V., Rahtu, E., & Heikkila, J. (2008). Rotation invariant local phase quantization for blur insensitive texture analysis. In *International conference on pattern recognition*.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola et al. (Eds.), *Advances in large margin classifiers* (pp. 61–74). MIT Press.

Rivard, D., Granger, E., & Sabourin, R. (2011). Multi-feature extraction and selection in writer-independent offline signature verification. *International Journal on Document Analysis and Recognition*.

Said, H. E. S., Tan, T. N., & Baker, K. D. (2000). Personal identification based on handwriting. *Pattern Recognition, 33*, 149–160.

Schlapbach, A., & Bunke, H. (2004). Using hmm-based recognizers for writer identification and verification. In *Proceedings of 9th international workshop on frontiers in handwriting recognition* (pp. 167–172).

Schlapbach, A., & Bunke, H. (2007). A writer identification and verification system using hmm based recognizers. *Pattern Analysis and Applications, 10*, 33–43.

Schlapbach, A., & Bunke, H. (2008). Off-line writer identification and verification using gaussian mixture models. In S. Marinai & H. Fujisawa (Eds.), *Machine learning in document analysis and recognition*. *Studies in computational intelligence* (Vol. 90, pp. 409–428). Berlin/Heidelberg: Springer.

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

Shen, C., Ruan, X. G., & Mao, T. L. (2002). Writer identification using gabor wavelet. In *4th World congress on intelligent control and automation* (Vol. 3, pp. 2061–2064).

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.

Sollich, P. (2002). Bayesian methods for support vecotr machines: Evidence and predictive class probabilities. *Machine Learning, 46*(1–3), 21–52.

Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2002). Individuality of handwriting. *Journal of Forensic Sciences*, 47.

Ubul, K., Tursun, D., Hamdulla, A., & Aysa, A. (2009). A feature selection and extraction method for uyghur handwriting-based writer identification. In *2009 International conference on computational intelligence and natural computing* (pp. 345–348).

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