IJDAR (2012) 15:213–226
DOI 10.1007/s10032-011-0166-4

ORIGINAL PAPER

Writer verification using texture-based features

R. K. Hanusiak · L. S. Oliveira · E. Justino · R. Sabourin

Received: 5 March 2010 / Revised: 6 May 2011 / Accepted: 11 May 2011 / Published online: 24 May 2011

Abstract In this work, we propose a writer verification system that takes into account texture-based features and dissimilarity representation. Textures of the handwritings are created based on the inherent properties of the writer. Independent of the writing style, the proposed method reduces the spaces between lines, words, and characters, producing a texture that keeps the main features thus avoiding the complexity of segmentation. We also address an important issue of verification system, i.e., the number of writers used for training. Our experiments show that the number of writers do not have an important impact on the overall error rate, but it has an important role in reducing the false acceptance of the verification system. We show that the false acceptance decreases as the number of writers increases. Finally, the ROC curves produced by different classifiers trained with different texture descriptors are combined using the maximum likelihood analysis, producing a ROC combined classifier. A set of experiments on a database composed of 315 writers show the efficiency of the texture-based features and the ROC combination scheme. Experimental results report an overall error rate of about 4%. This performance compares to the state of the art. Besides, the combination scheme is able to considerably reduce the false-positive rates while maintaining the same true-positive rates.

R. K. Hanusiak · E. Justino
Pontifical Catholic University of Parana (PUCPR), R. Imaculada Conceição, 1155, Curitiba, PR 80215-901, Brazil

L. S. Oliveira (✉)
Federal University of Parana (UFPR), R. Rua Cel. Francisco H. dos Santos, 100, Curitiba, PR 81531-990, Brazil
e-mail: lesoliveira@inf.ufpr.br

R. Sabourin
École de technologie supérieure, 1100 rue Notre Dame Ouest, Montreal, QC H3C 1K3, Canada
e-mail: robert.sabourin@etsmtl.ca

1 Introduction

The main goal of a writer verification system is to determine whether or not a handwritten text was written by a given writer. Similar to other verification systems, like signature verification, a failure is referred as type I error (false rejection), i.e., rejecting a genuine text. The system should cope also with a more challenging problem, i.e., avoiding the acceptance of forgeries as being authentic. This second error is referred as type II error (false acceptance).

The writer verification problem can be categorized into on-line and off-line. In general, on-line systems achieve better performance since they can use spatial and temporal information about the writing. Off-line systems, on the other hand, are difficult to design since the dynamic information about the writing is lost during the document acquisition. They also can be categorized into text-dependent and text-independent. As the name suggests, in the text-dependent methods, the writing samples contain a predefined text. These methods normally use the comparison between individual characters or words of known transcription and thus require the text to be recognized or segmented into characters or words prior to writer verification [28]. As pointed out by Siddiqi and Vincent [31], in this respect, text-dependent writer verification is similar to signature verification. The text-independent methods, on the other hand, use any text to establish the identity of the writer; hence, they feature less constraints and are more suitable for real applications.

The writer verification problem is a binary problem by nature. Given an input feature vector x extracted from a text t and a claimed identity I, determine whether (I, x) belongs to class ω_1 or ω_2. The class ω_1 indicates that the claim is true, i.e., the text has been written by the author I, and ω_2 indicates that the claim is false, i.e., the text is from an impostor. Differently from the identification problem, where the task consists in determining the identity I among all the writers enrolled in the system, the verification task performs an 1:1 comparison, which makes this kind of approach suitable even when a huge number of writers should be considered.

One can deal with the writer verification problem in two different ways. The standard approach, which is often used for signature verification, consists in building a specific model for each writer. In this context, different techniques of classification have been reported in the literature, such as hidden Markov models [28], distance measures [6,7], grapheme Clustering [1], dissimilarity [3,30], and Bayesian classifiers [31]. In these cases, some samples of a given writer are used to model ω_1, and some samples of other writers, chosen randomly, are used to model the forgery class ω_2. The main drawbacks of this approach are the need of learning the model each time a new writer should be included in the system and the great number of genuine samples necessary to build a reliable model. In real applications, usually a limited number of samples per writer is available to train a classifier, which leads the class statistics estimation errors to be significant, hence resulting in unsatisfactory verification performance.

An alternative approach that has been successfully applied to signature verification have been presented by Bertolini et al. in [2]. It is based on dissimilarity representation, and it allows the possibility of adding new writers into the system without retraining the models. This strategy is based on a forensic document examination approach and is called writer-independent approach as the number of models does not depend on the number of writers [33]. It has been demonstrated that through this strategy, it is possible to build robust verification systems even when few samples per writer are available.

Regarding the feature extraction methods used in writer verification systems, they can be classified into local and global. Local methods [4] are based on specific features which in general involve a segmentation process, whereas global techniques [25] are based on the overall look and feel of the writing. Combining both global and local strategies has been investigated by some authors [1,30,31], which have reported interesting results.

In general, local approaches are preferred from the forensic experts perspective, since they can be based on well-known graphometric features [16,20]. A fundamental problem with this approach lies in the difficulties of automatically segmenting the handwriting into lines, words, and characters for further feature extraction. To overcome the difficulties imposed by explicit segmentation strategies, in this work we propose a global approach for writer verification based on texture analysis. Such a texture is built based on the inherent properties of the writer. That is, regardless of the way the text has been drafted, all connected components are rearranged into a new space keeping the original slant but reducing the spaces between lines of text, words, and characters. This process creates a texture that keeps important features that enables us to use a global approach, avoiding the complexity of segmentation. Moreover, it is suitable for both text-dependent and text-independent verification systems.

Since our focus in this work is the texture generation rather than exploring different texture analysis methods, we decided to apply a widely used approach to texture analysis, i.e., the gray-level co-occurrence matrix (GLCM), which have been extensively validated in several different domains, including writer identification [5,10,25]. In light of this, several GLCM were tried out, and those with best performance were considered in this work. We also show that these results can be further improved by combining classifiers without the need for joint training. This combination is based on the maximum likelihood analysis of the receiver operating characteristics (ROC) curves of the classifiers.

Regarding the dissimilarity representation, one issue addressed in this work is the number of writers used to train the dissimilarity model, in our case, a support vector machine (SVM) trained to discriminate between genuine and forgery. We have shown that the size of the training set does not have an important impact on the overall error rate, but it has an important role in reducing the false acceptance (error type II) of the system. Through a set of experiments on a database composed of 315 writers, we demonstrate that the proposed approach is feasible. The overall error rate yielded by the system is about 4% after combining the classifiers. This performance compares to other methods reported in the literature. Besides, the combination scheme is able to considerably reduce the false-positive rates while maintaining the same true-positive rates.

The remaining of this paper is organized as follows: Section 2 presents the dissimilarity framework proposed in this work. Section 3 introduces the ROC combination algorithm used in this work. Section 4 describes the database considered in our experiments. Section 5 shows the proposed texture-based feature sets and classification methods, while the experimental results and discussions are reported in Sect. 6. Finally, Sect. 7 concludes this work and points out some future works.

2 The dissimilarity and writer identification framework

In this section, we describe the proposed framework based on dissimilarity for writer verification. Our approach is inspired on forensic experts who compare a questioned sample with some references to assert whether a piece of handwriting is genuine or forgery. During this comparison, the experts extract different features to compute the level of similarity between the samples being compared [20]. In fact, the work performed by forensic experts is quite more complex in the

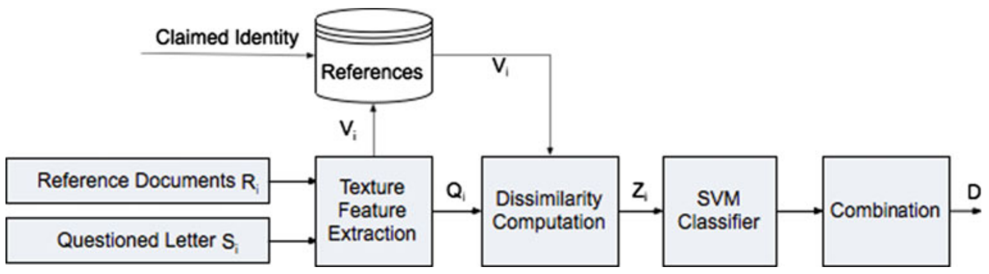


Fig. 1 The dissimilarity framework proposed for writer verification

sense they decide on the fly, based on their tacit knowledge, which features from the handwriting they should extract.

The concepts of similarity, dissimilarity, and proximity have been discussed in the literature from different perspectives [13,23,27]. Pekalska and Duin [23] introduce the idea of representing the relations between objects through dissimilarity, which they call dissimilarity representation. The seminal work using this concept in the field of author identification was presented by Cha and Srihari [7]. In this work, we use the idea of dissimilarity vectors presented by Bertolini et al. in [2], which combines feature-based description with the concept of dissimilarity. The idea is to extract the feature vectors from both questioned and reference samples and then compute what we call the dissimilarity feature vector. If both samples come from the same writer (genuine), then all the components of such a vector should be close to 0, otherwise (forgery), the components should be far from 0.

Of course, this is totally true under favorable conditions. As any other feature representation, the dissimilarity feature vector can be affected by the intra-writer variability. Such a variability could generate values far from zero when measuring the dissimilarity of genuine writers.

The proposed framework is depicted in Fig. 1. After a preprocessing step, composed basically of the Otsu thresholding algorithm [22], a set of n genuine handwriting samples, $R_i (i = 1, 2, 3, \ldots, n)$ is sent to the feature extraction module and the feature vectors V_i are stored into a database. Feature extraction is discussed in Sect. 5. When questioned samples S_i are presented to the system, they go through the same feature extraction process, thus generating the feature vectors Q_i. Then, the dissimilarity feature vectors $Z_i = |V_i - Q_i|$ are computed to train a binary classifier, which provides a decision for each dissimilarity feature vector. The final decision D depends on the combination of these partial decisions, which are obtained through a fusion rule. Section 6 discusses how such a combination is performed.

In the remaining of this work, we use several SVM classifiers trained on two features representations (directional and texture), employ traditional straightforward fusion rules, and also explore the use of a ROC-based classifier combination algorithm (described in Sect. 3) as an alternative combination method.

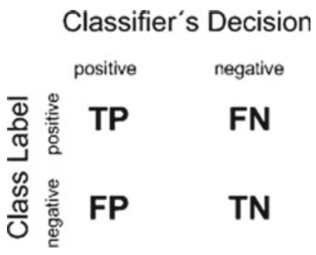

Classifier´s Decision
positive
negative
Class Label
positive
negative
TP
FN
FP
TN

Fig. 2 2×2 confusion matrix

3 Combining classifiers in the ROC space

ROC curves are two-dimensional graphs in which true-positive rate (TPR) is plotted on the y axis and false-positive rate (FPR) is plotted on the x axis. Given a decision list and an instance set, a 2×2 confusion matrix (Fig. 2) can be generated, representing the classification performance. From this matrix, four statistics are represented: tp, fp, fn, and tn, which stand for true-positive, false-positive, false-negative, and true-negative, respectively.

For a given classifier C, the ROC is a set of points $(fp_C(k_C), tp_C(k_C))$ where k_C is the parameter that governs the decision process. A ROC graph depicts relative trade-offs between benefits (tp) and costs (fp). Each classifier produces a (fp, tp) pair corresponding to a single point in the ROC space. For an extensive review of ROC, please refer to [9].

Besides being an important tool to analyze and compare classifiers, ROC also have been used for combining classifiers. In this work, we have used the algorithm introduced by Haker et al. in [14]. It is based on the calculation of a combined ROC using maximum likelihood analysis to determine a combination rule for each ROC operating point.

Let us consider the ROC for two different classifiers A and B and their respective parameters k_A and k_B. The performance of classifiers C_A and C_B are represented in the ROC space by the points, (fp_A, tp_A) and (fp_B, tp_B), respectively. Given an input pattern, both classifiers will produce an output either positive $(+)$ or negative $(-)$, giving us a total of 4 possible cases. For each case, we have an expression of the maximum likelihood estimation (MLE) of the unknown truth T (Table 1).

Table 1 Binary output for classifiers A and B and the maximum likelihood combination

C_A	C_B	Combination MLE for truth T
+	+	$P(A=1, B=1 \mid T=1) \geq P(A=1, B=1 \mid T=0)$
+	−	$P(A=1, B=0 \mid T=1) \geq P(A=1, B=0 \mid T=0)$
−	+	$P(A=0, B=1 \mid T=1) \geq P(A=0, B=1 \mid T=0)$
−	−	$P(A=0, B=0 \mid T=1) \geq P(A=0, B=0 \mid T=0)$

Table 2 Binary output for classifiers A and B and the maximum likelihood combination

C_A	C_B	Combination MLE for truth T
+	+	$tp_A tp_B \geq fp_A fp_B$
+	−	$tp_A(1-tp_B) \geq fp_A(1-fp_B)$
−	+	$(1-tp_A)tp_B \geq (1-fp_A)fp_B$
−	−	$(1-tp_A)(1-tp_B) \geq (1-fp_A)(1-fp_B)$

Table 3 Schemes for combining classifiers A and B

C_A	C_B	$S_{A\&B}$	S_A	S_B	$S_{A\mid B}$
+	+	+	+	+	+
+	−	−	+	−	+
−	+	−	−	+	+
−	−	−	−	−	−

Table 4 FPR and TPR for the combination schemes

Scheme	FPR	TPR
$S_{A\&B}$	$fp_A fp_B$	$tp_A tp_B$
S_A	fp_A	tp_A
S_B	fp_B	tp_B
$S_{A\mid B}$	$fp_A + fp_B - fp_A fp_B$	$tp_A + tp_B - tp_A tp_B$

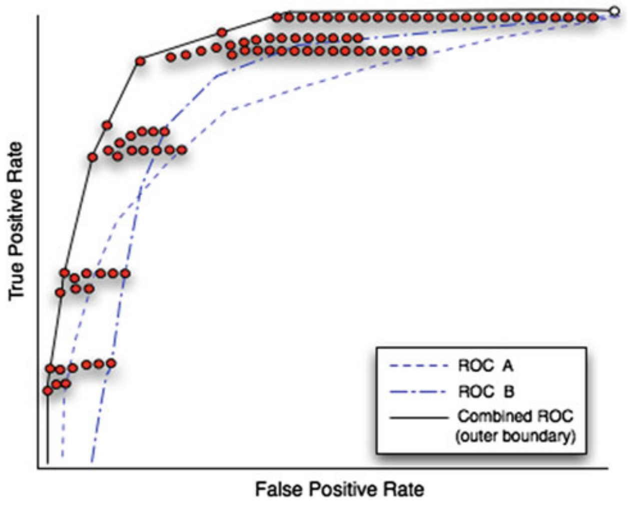


Fig. 3 Illustration of the algorithm proposed by Haker [14]

Each inequality (logical expression) in the rightmost column evaluates to either + or −, and the resulting value is the maximum likelihood estimate of the truth T. If conditional independence is assumed, then $P(A=1, B=1 \mid T=1) = P(A=1 \mid T=1)P(B=1 \mid T=1) = tp_A tp_B$. Proceeding similarly for the other terms in the rightmost column of Table 1, we get Table 2.

From the assumptions detailed above, $tp_A tp_B = fp_A fp_B$ and $(1-tp_A)(1-tp_B) = (1-fp_A)(1-fp_B)$, so whenever A and B are in agreement, their common output is the maximum likelihood estimate of T. Thus, the middle two rows of the Table 2 need to be determined, resulting in one of four possible MLE combination schemes, which are mnemonically named schemes "A AND B" ($S_{A\&B}$), "A" (S_A), "B" (S_B), and "A OR B" ($S_{A\mid B}$). Table 3 summarizes these schemes.

Using again the assumption of conditional independence, Table 4 shows how to calculate the false-positive and true-positive rates for these schemes. In practice, this means that decision processes can be combined without retraining, since there is no need to estimate joint distributions for the output of A and B, nor the need to know the distribution of the underlying truth T.

As stated before, the classifiers (ROC) A and B are governed by the parameters k_A and k_B, respectively. In other words, k_A and k_B are simple thresholds applied to the outputs s_A and s_B calculated as part of the A and B decision process. Thus, A returns the estimate $T=1$ if and only if $s_A > k_A$. The same holds for B. Therefore, a new decision rule is necessary for each scheme described so far. Let Υ be the combined classifier, created as described above. For a chosen operating point (fp, tp) on the ROC for Υ, we have associated thresholds k_A and k_B and an associated MLE combination rule. The new decision rules s for Υ are defined as function of s_A and s_B as follows: $\min(s_A - k_A, s_B - k_B)$, $s_A - k_A$, $s_B - k_B$, and $\max(s_A - k_A, s_B - k_B)$for the schemes $S_{A\&B}$, S_A, S_B, and $S_{A\mid B}$, respectively. The combined classifierΥ will assign "+" when $s > 0$; otherwise, it will assign "−". Algorithm 1 shows the pseudo-code for the combined ROC.

To combine classifiers A and B, we compute for each value of the parameter pair (k_A, k_B) and corresponding 4-tuple of FPR and TPR (fp_A, tp_A, fp_B, tp_B) the correct ML scheme to use according to Table 2 and the resulting combined rates (fp, tp) for that scheme using the formula described in Table 4. In practice, discrete values are assumed for k_A and k_B by sampling them evenly. Figure 3 shows an example of what would be a resulting set of points (fp, tp) for two example ROCs.

The set of points (fp, tp) represent possible operating points for the joint process. However, we do not need to consider all the points since for each point in the interior (the red

Algorithm 1 ROC Combination

```
1: FPComb {buffer for combined false positives}
2: TPComb {buffer for combined true positives}
3: for $j = 1$ to $length(FPR_A)$ do
4:   for $k = 1$ to $length(FPR_B)$ do
5:     $fp_A = FPR_A(j)$
6:     $tp_A = TPR_A(j)$
7:     $fp_B = FPR_B(k)$
8:     $tp_B = TPR_B(k)$
9:     if $(tp_A * tp_B >= fp_A * fp_B)$ AND $(\sim (tp_A * (1 - tp_B) >= fp_A * (1 - fp_B)))$ AND $(\sim ((1 - tp_A) * tp_B >= (1 - fp_A) * fp_B))$ AND $(\sim ((1 - tp_A) * (1 - tp_B) >= (1 - fp_A) * (1 - fp_B)))$ then
10:      FPComb(j,k) = $fp_A * fp_B$;
11:      TPComb(j,k) = $tp_A * tp_B$;
12:    else if $tp_A * tp_B >= fp_A * fp_B)$ AND $(tp_A * (1 - tp_B) >= fp_A * (1 - fp_B))$ AND $(\sim ((1 - tp_A) * tp_B >= (1 - fp_A) * fp_B))$ AND $(\sim ((1 - tp_A) * (1 - tp_B) >= (1 - fp_A) * (1 - fp_B)))$ then
13:      FPComb(j,k) = $fp_A$;
14:      TPComb(j,k) = $tp_A$;
15:    else if $(tp_A * tp_B >= fp_A * fp_B)$ AND $(\sim (tp_A * (1 - tp_B) >= fp_A * (1 - fp_B)))$ AND $((1 - tp_A) * tp_B >= (1 - fp_A) * fp_B)$ AND $(\sim ((1 - tp_A) * (1 - tp_B) >= (1 - fp_A) * (1 - fp_B)))$ then
16:      FPComb(j,k) = $fp_B$;
17:      TPComb(j,k) = $tp_B$;
18:    else if $(tp_A * tp_B >= fp_A * fp_B)$ AND $(tp_A * (1 - tp_B) >= fp_A * (1 - fp_B))$ AND $((1 - tp_A) * tp_B >= (1 - fp_A) * fp_B)$ AND $(\sim ((1 - tp_A) * (1 - tp_B) >= (1 - fp_A) * (1 - fp_B)))$ then
19:      FPComb(j,k) = $fp_A + fp_B - fp_A * fp_B$;
20:      TPComb(j,k) = $tp_A + tp_B - tp_A * tp_B$;
21:    end if
22:  end for
23: end for
```

ones in Fig. 3) there is a point on the outer boundary of the region which is superior, and thus a better operating point. For example, there is a point on the boundary which has the same FPR and a greater TPR. Such points form the combined ROC. In practice, the outer boundary ROC can be estimated by splitting the interval [0,1] into a number of sub-intervals i.e., bins, and within each bin finding the pair (fp, tp) having the largest value of tp [9].

4 Database

The database used in this investigation is the Brazilian Forensic Letter Database [11], which is composed of 315 writers, three samples per writer, summing up 945 images. The motivation for using such database is the growing interest of the Brazilian forensic experts as well as the Brazilian Federal Police in writer verification.

The samples were provided by undergraduate students in three different sessions during one month. The texts were collected on an A4 white sheet of paper with no pen-draw baseline and then scanned in gray level with 300 dpi (3760 $\times$ 2448). Each writer was allowed to use his/her own pen, which means that several different pens were used. The text is concise (131 words) and complete in the sense that it contains all characters (letters and numerals) and certain character combinations of interest. This makes it suitable for text-dependent writer identification as well. Figure 4 shows (a) the letter contents and (b) a image sample of the database. More details about this database[1] can be found in [11].

As discussed in the introduction, in this work we argue that it is possible to identify discriminant features from the texture created by concatenating the writer's handwriting. In this vein, a new database of image textures was build to support further experiments. More details about the technique used to build the textures are presented in Sect. 5. The texture database is composed of 15 (5 textures $\times$ 3 letters) images per writer, summing 4,725 texture images.

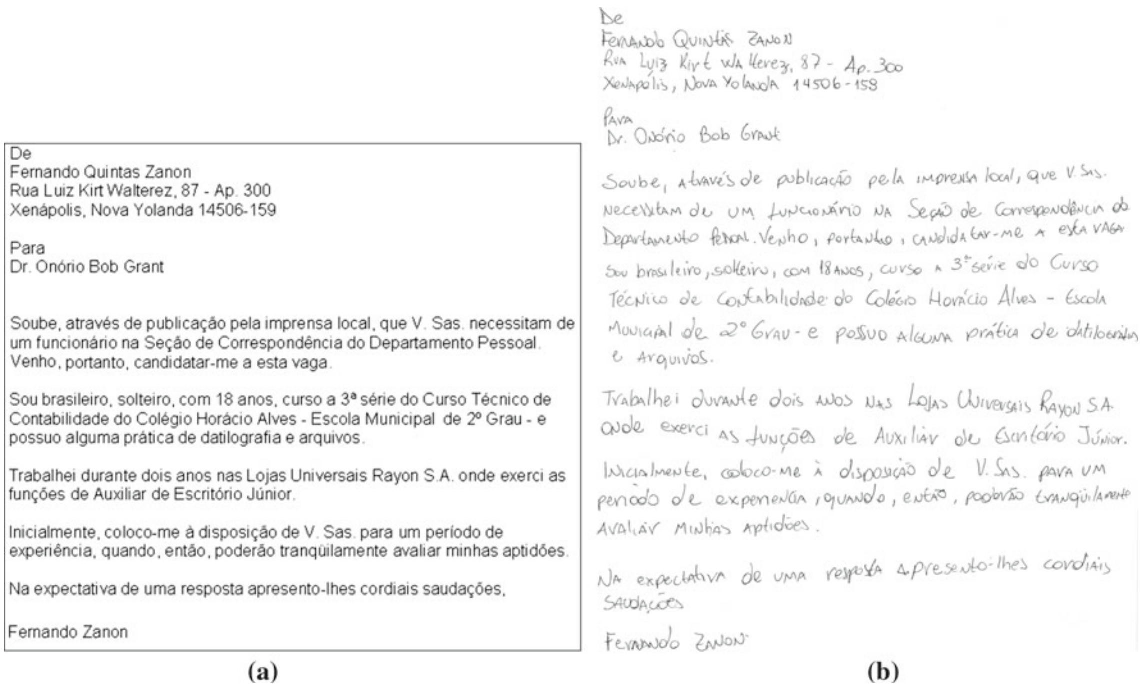
Considering the dissimilarity-based approach adopted in this work, the classifiers should be trained to discriminate between genuine (positive) and forgeries (negative). To generate the positive samples, we have computed the dissimilarity vectors among three genuine samples of each writer (one segment of texture extracted from each letter), which results into three different combinations. The negative samples were generated by computing the dissimilarity between one sample of one author against one sample of three other authors picked randomly. This also results into three different combinations.

The 315 writers of the database were divided into training and testing. Four different partitions for training were considered: 25, 50, 100, and 200 writers. The idea is to analyze the impact of the number of writers used for training in the overall performance. The remaining 115 were used for testing. It is important to remark that different writers were used for training and testing. Considering 50 writers and three textures segments for training, we would have 150 (3 $\times$ 50) positive samples and 150 (3 $\times$ 50) negative samples. Figure 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 feature vector of other authors picked randomly, and it is expected that they have components far from 0.

Regarding the testing set, we have picked 115 writers randomly to use as random forgeries. To build a balanced testing set, two letters from each author were selected randomly.

[1] The database is available under request for research purposes. To obtain a copy, please send an email to lesoliveira AT inf.ufpr.br.



(a)

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

(b)

De
Fernando Quintas Zanon
Rua Luiz Kirt Walterez, 87 - Ap. 300
Xenápolis, 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 tranqüilamente avaliar minhas aptidões.

Na expectativa de uma resposta apresento-lhes cordiais saudações

Fernando Zanon

Fig. 4 Brazilian Forensic letter database: **a** the letter contents and **b** an image sample

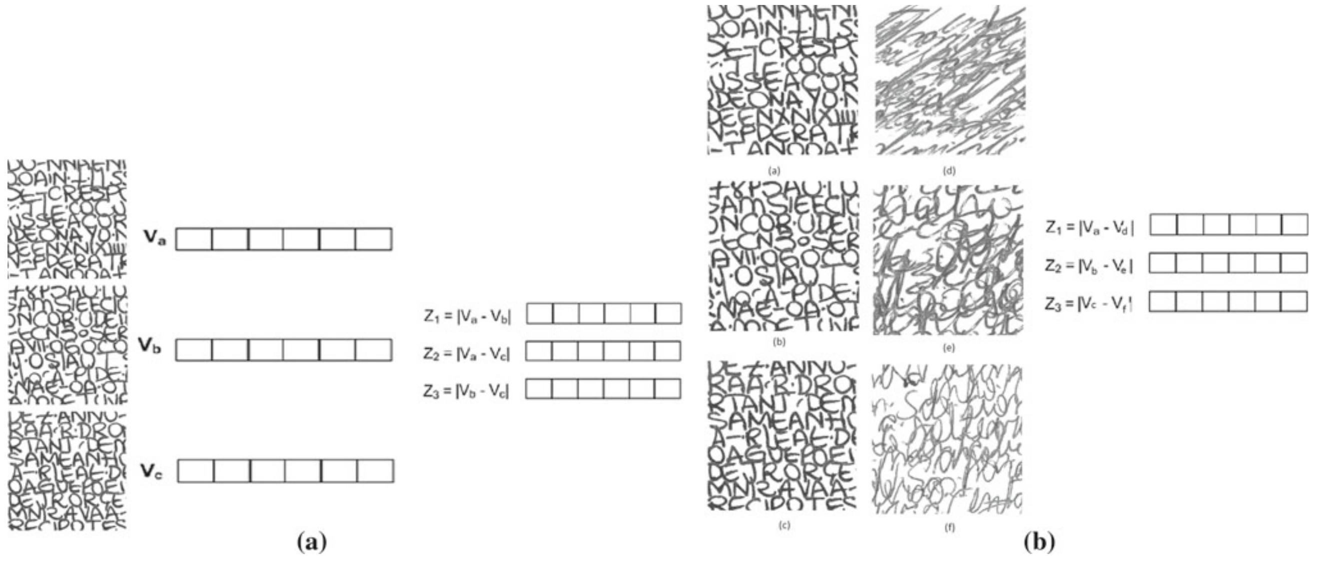


Fig. 5 Dissimilarities **a** among genuine samples of the same writer to generate the positive samples and **b** among genuine samples from different writers to generate the negative samples

Therefore, each writer of the testing set has one genuine letter, represented by 5 pieces of texture, and one random forgery, also represented by 5 pieces of texture.

5 Features

It is well known that the choice of the features is a crucial aspect for any pattern recognition problem. For writer verification, it is not different. However, when dealing with handwriting, one must rely on features that have well-established scientific basis. This is an important issue especially if the writer verification system is used to support a legal decision in a court of justice.

In light of this, forensic document examiners (FDE) usually rely on graphology to perform writer verification. Graphology can be defined as the study and analysis of handwriting especially in relation to human psychology. A branch of the graphology is the psychometrical graphology or graphometry. This is the term used to describe the technique of picking up psychic impressions about a person from a specimen of their handwriting [21]. The graphometrical features can be classified into genetic and generic. The genetic features are: minimal graphics (i dots, commas, cedillas, tildes, etc.), pressure, speed, entry/exit strokes, and movement. The generic features are: caliber, spacing between characters and words, proportion, slant, and alignment to baseline.

Gobineau and Perron [12] elaborated a theory of graphometry or, more exactly, a statistical method of the graphic elements. In their work, they propose more than 60 features but choose 14 which they deem essential and easy to extract.

Some of the concepts of graphology have been intrinsically used to build automatic signature verification systems by several different authors [17,18], but few works [26] have applied the graphology concepts to writer verification. This is because the extraction of graphometric features in the context of writer verification is not straightforward. The main problems are related to the difficulties of segmenting the handwriting into lines, words, and characters for further feature extraction. Consider the examples depicted in Fig. 6 where problems such as (a) overlapping and (b) skew are important issues for any segmentation algorithm. Besides, Fig. 6c shows a piece of a illegible text which is not so difficult to segment, but it is not so rich in terms of graphometric features.

To surpass these problems, in this work, we proposed a segmentation strategy based on the inherent properties of the writer. That is, regardless of the way the text has been drafted, the segmentation process will rearrange all connected components into a new space keeping the original slant but reducing the spaces between lines of text, words, and characters. In the end, the segmentation process will produce a texture image that still contains the main characteristics of the writing style.

The segmentation process is straightforward. First, the image is binarized using Otsu algorithm and then scanned top-down, 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 remaining components are then used to extract the original components of the gray-level image. The components in gray level are then aligned with the new image using the center of mass of the bounding box. Figure 7a exemplifies this process.

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

$$\text{new\_}y = \text{previous\_}y + \frac{h}{2} \quad (1)$$

where 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) and h is the average height of all 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 much blank spots, like in Fig. 7b. This denominator was found empirically. Figure 8 shows an example of the texture created from the original gray-level letter. As mentioned previously, the final texture image representing the writer's handwriting is finally segmented into equal blocks of 256×256 pixel.

The letters of the database used in this work allow us to generate up to nine blocks of texture. However, if smaller documents were considered, it will be impossible to get the same number of blocks. To make the proposed method less dependent of the size of the document, we decide to use only five blocks per letter.

This segmentation schemes differs from the ones presented in the literature [5,25] in the sense that there is no preprocessing such as slant correction which are necessary for line segmentation. Besides making the segmentation simpler, the proposed texture generation method also keeps some features such as skew and slant. Figure 9 shows that different textures can be created for different handwriting styles.

In Fig. 9a, different calibers, which is the relationship between height and width, can be observed. Words with smaller caliber produce less-overlapped texture than those with larger caliber. Figure 9b shows the progression where slow writers usually yield more legible texts than fast writers. In Fig. 9c, the proportion among ascenders, descenders, and the main body of the words is presented. Alike words with big caliber, high ascenders and descenders will create a more dense texture.

Pressure is related to the changing width of a line as pen pressure on paper varies. High pressure will produce darker textures, while low pressure will produce lighter textures, as depicted in Fig. 9d. Entry/End points are related to how a writer starts of finishes a character or word. This feature is known as characteristic gesture and usually very difficult to find by means of a computer program. In a texture (Fig. 9e), bigger entry/end points (strokes) may produce more white spaces. Finally, Fig. 9f shows different slants produced by different writers, which are clearly captured by the texture.

5.1 Feature extraction

After producing the textures from the handwritings, the next step is to choose a suitable descriptor that is capable of providing measures such as smoothness, coarseness, and regularity. As stated before, our choice was to apply a widely used approach to feature analysis, the GLCM. A GLMC is the joint probability occurrence of gray-level i and j within a defined spatial relation in an image. That spatial relation is defined in

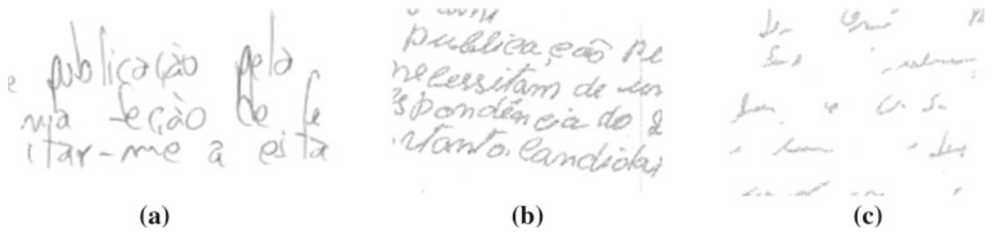


Fig. 6 Problems faced during feature extraction **a** overlapping, **b** skew, and **c** illegible text

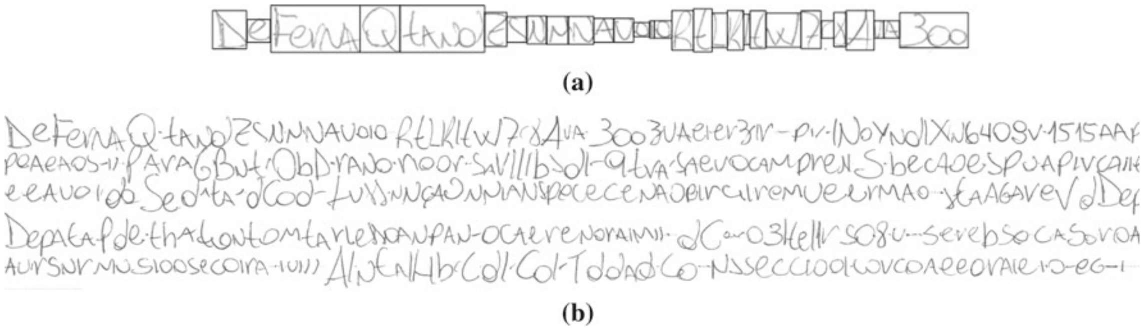
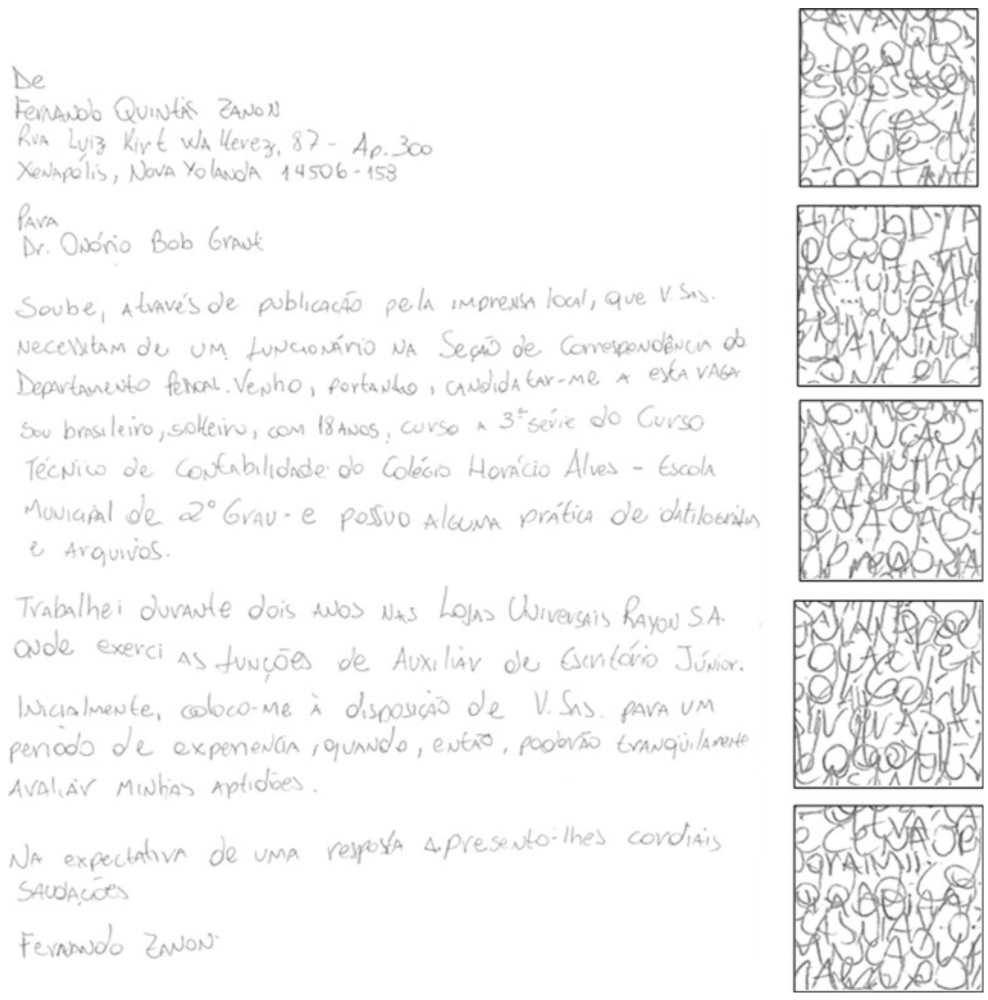


Fig. 7 The texture generation process. **a** Filling a line and **b** spaced texture



De
Fernando Quintas Zanon
Rua Luiz Kirt Wallerez, 87 - Ap. 300
Xenapólis, 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 2° 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

Fig. 8 Original letter and texture blocks

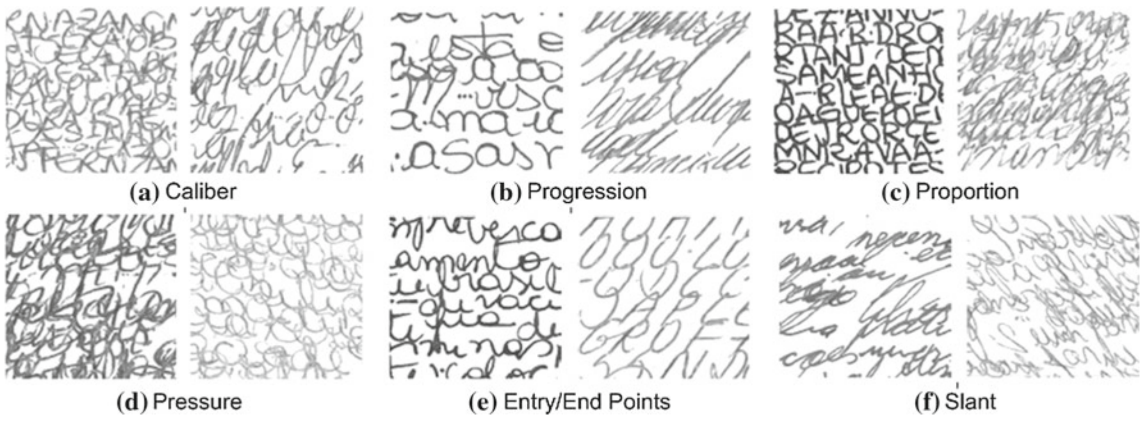


Fig. 9 Some observable features in the texture image

terms of a distance d and an angle θ. Given a GLMC, some statistical information can be extracted from it. The most common descriptors were proposed by Haralick [15] and have been successfully used in several application domains, including writing verification/identification [5,10,25]. In this work, we have used the following five descriptors, which provided the best preliminary results:

$$\text{Entropy} = -\sum_{i=0}^{n}\sum_{i=0}^{m} p(i,j) \times \log(p(i,j)) \tag{2}$$

$$\text{Homogeneity} = \sum_{i=0}^{n}\sum_{i=0}^{m} p(i,j)/(1+|i-j|) \tag{3}$$

$$\text{Dissimilarity} = \sum_{i=0}^{n}\sum_{i=0}^{m} p(i,j) \times |i-j| \tag{4}$$

$$\text{Inverse variance} = \sum_{i=0}^{n}\sum_{i=0}^{m} p(i,j)/(i-j)^2, i \neq j \tag{5}$$

$$\text{Energy} = \sqrt{\sum_{i=0}^{n}\sum_{i=0}^{m} p(i,j)^2} \tag{6}$$

In that case, $p(i,j)$ is the probability of co-occurrence of gray-level i and j observing consecutive pixels at distance d and angle θ.

Based on these measures, five different feature sets were extracted. Each of them contains 20 components, which are the combination of five distances ($d = 1, 2, 3, 4, 5$) and four angles ($\theta = 0, 45, 90, 135$). Longer distances and different angles were tried out during our experiments, but these values brought us the best results. The features are normalized between 0 and 1 using the min-max rule.

6 Experiments and discussion

Our experiments are divided into three parts. First, we explore the use of directional features extracted from both original and texture images. Then, we discuss different texture descriptors extracted from the GLCM and the impacts of the number of writers on the overall error rate. Finally, we report the experiments concerning the combination of classifiers using the ROC-based classifier combination strategy described in Sect. 3.

In all experiments, support vector machines (SVM) were used as classifiers. Training is done using fivefold cross-validation. Different kernels were tried, but the best results were achieved using a Gaussian kernel. Parameters C and γ were determined through a grid search. The overall error rate that has been used for evaluation purposes in this work is given by Eq. 7.

$$\text{Overall Error Rate} = \frac{fp+fn}{tp+tn+fp+fn} \tag{7}$$

One of the limitations with 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 producing a posterior probability $P(\text{class}|\text{input})$. In our case, as depicted in Fig. 1, we are interested in the estimation of 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 [24,32]. In this work, we have adopted the strategy proposed by Platt in [24].

6.1 Experiments with directional features

Before reporting the experiments using texture-based features, first we present some results using one of the traditional characteristics used by FDEs. According to the experts [16,20], a given writer takes some directions more frequently and more intensively than others. Therefore, the cumulative distribution of such directions is an important and discriminative feature of the writer. The directional features have been implemented as suggested by Crettez [8]. The feature

Table 5 Comparison between texture-based and directional features

Descriptor	Type I error (%)	Type II error (%)	Overall error rate (%)
Directional (original—5 blocks)	10.6	34.6	22.6
Directional (original—3 blocks)	10.3	34.1	22.2
Directional (texture)	6.9	12.1	9.5

vector is composed of 17 components containing a histogram of 17 directions ranging from 0 to 180 degrees.

The experimental protocol aforementioned, using 50 writers for training, was used into two different experiments. In the first case, the letters were segmented into 5 blocks of 800 × 600 pixels and then the features were extracted from these sub-images. In the second experiment, we extracted the features from five blocks of textures. The size 800 × 600 was the maximum size that allowed us to created five distinct pieces with handwriting information in all of them. In spite of the large size of the original image (3760 × 2448), some writers use just part of the sheet with the handwriting, which makes it impossible to extract more blocks of handwriting.

Using five blocks in both cases allows us to keep the same experimental protocol, i.e., combining five outputs to produce a final decision. However, it is fair to argue that this is not a complete honest comparison since the five original blocks of handwriting contain less information than the texture blocks. With this in mind, we increased the size of the original blocks to 1200 × 600 pixels, thus reducing the number of blocks to three. Experimental results show that using larger blocks has almost no impact in the overall error rate. Table 5 compares these experiments.

It is worth of noticing that when the directional features were extracted from the texture images, the overall error rate was about 13% points smaller than the features computed on the original images. This can be explained by the fact that the texture images (Fig. 10a) contain more information so more representative histograms can be created. From Fig. 10b, it is easy to observe that segmenting the original letter can produce images with several blank spots (Fig. 10a), thus producing less discriminative feature vectors.

6.2 Experiments with texture features

Concerning the experiments using the texture descriptors introduced in Sect. 5.1, the first (baseline) experiment we have performed used 50 writers for training and the entire testing set, i.e., 115 writers, for testing. The original texture images with 16 gray levels were considered. Regarding the fusion rule responsible for combining the results produced by the SVM on the three dissimilarity feature vectors, in our experiments the sum rule outperformed all the other methods. For this reason, it was used in all remaining experiments discussed in this section. Table 6 reports the results for the baseline system.

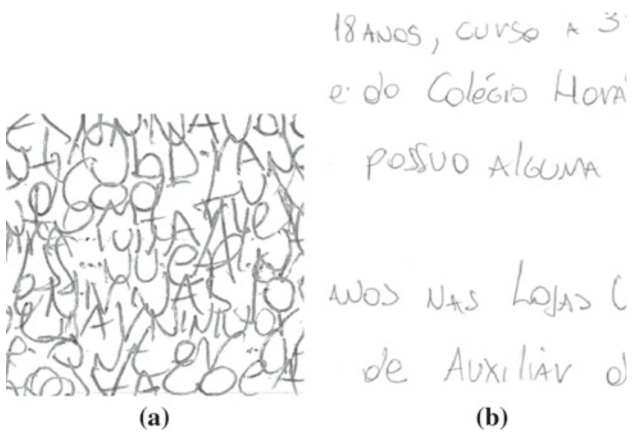


Fig. 10 Images used to compute directional histograms. **a** texture and **b** original

Table 6 Results achieved by the baseline system

Descriptor	Type I error (%)	Type II error (%)	Overall error rate (%)
Entropy	4.0	8.7	6.35
Homogeneity	5.7	9.2	7.5
Dissimilarity	12.7	41.9	27.3
Inverse variance	7.5	24.8	16.2
Energy	6.0	8.7	7.4

As we can observe from Table 6, those descriptors that look for the homogeneity of the texture such as entropy and homogeneity achieved smaller overall error rates. In all cases, though, the type II error is higher than type I Error, which is not interesting since in a verification system, it is desirable to reduce as much as possible the false acceptance.

After analyzing the baseline results and the co-occurrence matrices created with 16 gray-level images, we observed that the elevated number of gray levels did not bring enough discriminant information for the handwriting textures. With this in mind, we investigated the impact of reducing the number of gray levels until we get binary images. Table 7 shows the results of these experiments.

From Table 7, we can conclude that binary images are the best representation for handwriting textures when using co-occurrence matrices. In fact, the gray-level information contained in the letter images are not relevant. All the features discussed previously are available in the binary texture, which has been demonstrated through these experiments. Figure 11 shows the ROC curves for all the descriptors trained with binary images. As we can see, their performance is quite similar differing only for some operational points in the ROC space. The area under the curve (AUC), which is used very

Table 7 Impacts of reducing the number of gray levels

Descriptor	Overall error rate (%)			
	Gray-levels			
	16	8	4	2
Entropy	6.3	6.1	5.8	5.2
Homogeneity	7.5	8.4	7.2	6.7
Dissimilarity	27.3	27.0	6.9	5.9
Inverse variance	16.2	19.2	6.9	5.9
Energy	7.4	6.7	6.6	5.5

often to compare classifiers, is practically the same for all the classifiers (AUC $\simeq$ 0.98).

Another experiment we performed in this work regards the number of writers used in the training set. It is worth of remembering that in the dissimilarity approach the writers used for training are not used for testing. This is one of the benefits of this approach since we can add new writers into the system without retraining the classifier. This being said, this experiment aims at verifying the impacts of increasing the number of writers in the training set. In other words, is 50 writers enough to train a robust machine learning model based on dissimilarity? For these experiments, we have used the same protocol used so far but considering only the binary images, which achieved the best results in the previous experiments. Since we have 200 writers available in the training set, we doubled the number of writers for each experiment starting from 25 writers. Table 8 reports the results.

The experiments reported in Table 8 show that increasing the number of writers does not necessarily reduce the overall error rate. However, if we take a closer look, we will notice a trade-off between the number of writers used for training and the performance of error types I and II. By increasing the number of writers for training, we sure build more robust models against false acceptance. From the ROC depicted in Fig. 12, it is possible to notice that the classifier trained with 200 writers dominates all the others for low FPR, but after a certain level, it is surpassed by the others. Table 9 shows the trade-off between the size of the training set and error types I and II for the classifier trained with the entropy feature set. The same phenomenon was observed for all other feature sets.

6.3 Combining classifiers in the ROC space

A final experiment concerns the combination of classifiers using the strategy described in Sect. 3. The objective here is to improve the trade-off between false rejection and false acceptance by combining the classifiers through the maximum likelihood analysis of the ROC classifiers. We have done different experiments, but the setup that brought more improvement was the combination of classifiers trained with entropy and homogeneity descriptors. Figure 13 shows all the (fp, tp) pairs generated by Haker's algorithm and the combined ROC classifier as well.

As discussed in Sect. 3, the red dots in Fig. 13 represent all possible operating points for the joint process. As stated before, we do not need to consider all the points, since for each point in the interior there is a point on the outer boundary of the region which is superior and thus a better operating point.

It is clear from Fig. 13 that such a combination can considerably reduce the FPR while maintaining the same TPR. Using the combined ROC, for a TPR of 0.95 the corresponding FPR is 0.008. For the same TPR, entropy and homogeneity produce a FPR of 0.056 and 0.087, respectively.

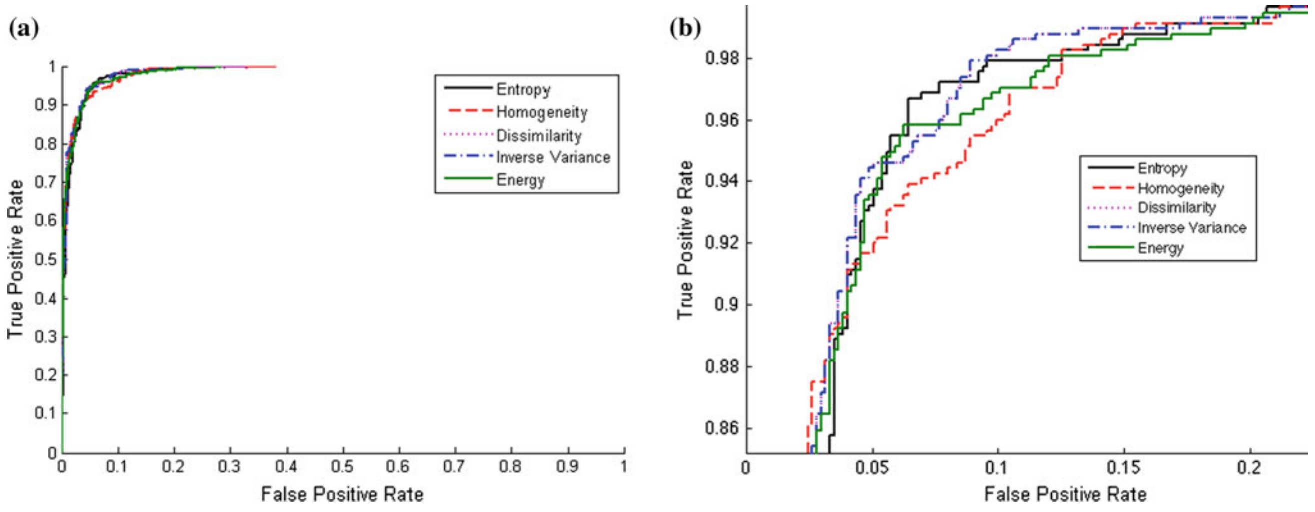


Fig. 11 Receiver operating characteristics comparing all descriptors **a** original **b** zoom on lower FPR

Table 8 Impacts of increasing the number of writers in the training set

Descriptor	Overall error rate (%)			
	Number of writers			
	25	50	100	200
Entropy	5.8	5.2	5.0	5.9
Homogeneity	6.3	6.7	5.9	6.6
Dissimilarity	5.8	5.9	6.0	5.5
Inverse variance	5.8	5.9	6.0	5.5
Energy	5.7	5.5	4.6	5.2

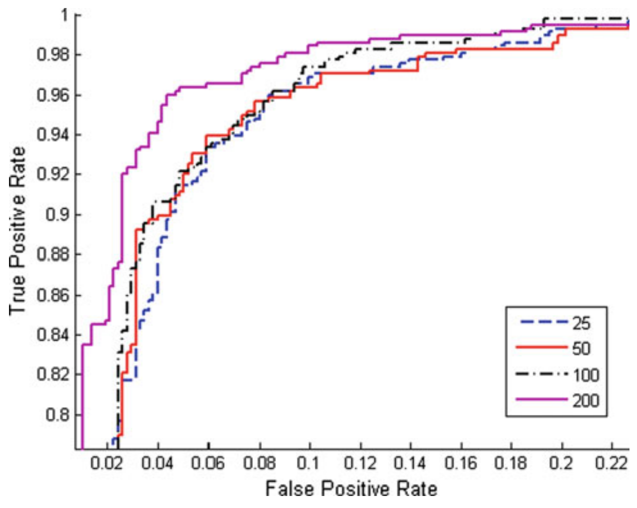


Fig. 12 ROC comparing different sizes of the training set

Table 9 Trade-off between the size of the training and error types I and II for the classifier trained with the entropy feature set

Number of writers in the training set	Type I error (%)	Type II error (%)	Overall error rate (%)
25	2.1	9.5	5.8
50	2.4	7.8	5.1
100	4.3	5.7	5.0
200	7.8	4.0	5.9

It is important to remember, though, that this algorithm relies on the assumption of conditional independence. In other words, the combined ROC is the upper-bound that could be achieved if both entropy and homogeneity classifiers were conditionally independents. In terms of recognition rate, the upper-bound would be 97.1%, since the pair (fp, tp) that maximizes performance is (0.008, 0.95). Considering that the testing set contains 575 (115 $\times$ 5 references) positive and 575 negative samples, $97.1\% = ((575 \times 0.95) + (575 - (575 \times 0.008))/1150)$

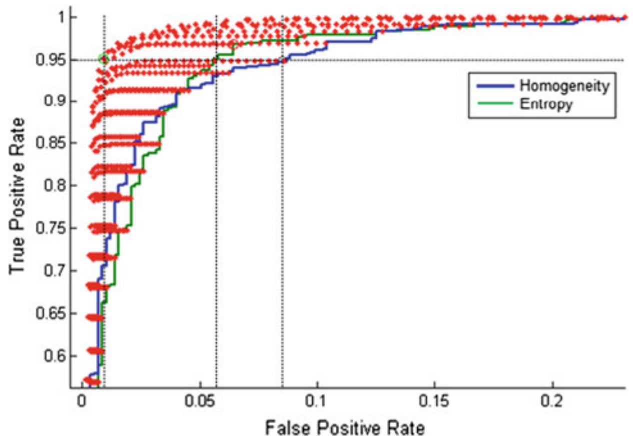


Fig. 13 Entropy and homogeneity combined in the ROC space

However, in practice, the performance observed during the experiments was 96.1%, i.e., an overall error rate of 3.9%. This corroborates to the fact that the classifiers we have used in this work do not hold the assumption of conditional independence. Nevertheless, the algorithm still brought about 1% point of improvement. This proves that it can be applied even when there is no guarantee of conditional independence.

Still, in the context of combining classifiers, we have used both classifiers (entropy and homogeneity) and combined them at the feature level. In this case, the SVM classifiers were trained with a feature vector composed of 40 texture descriptors, and the same experimental protocol described so far has been followed. The best overall error rate we got using this combination strategy was 5.2%.

To have a better insight into the performance of the results reported so far, Table 10 summarizes some works we found in the literature on writer verification/identification. Of course, a direct comparison is not possible since different databases, number of writers, features, classifiers, and samples per writer were considered. Our approach, for example, uses a full page of handwriting instead of segmented words and paragraphs. In spite of all that, it is possible to observe that the performance reported in this work compares to the state of the art.

7 Conclusion

In this work, we have proposed a writer verification system that takes into account texture-based features and a dissimilarity representation. The texture of the handwriting was created based on the inherent properties of the writer. Independent of the writing style, the proposed method reduces the spaces between lines, words, and characters, producing a texture that keeps important characteristics of the writing style. This enables us to use a global approach, avoiding the

Table 10 Performance of different methods reported in the literature

Reference	Writers	Data	Samples/ writer	Sample size	Classifier	Features	Perf. (%)
[25]	40		25	Text blocks	*k*NN	Gabor filters	95.3
[19]	20	IAM	5	Paragraph	*k*NN, Neural net.	Slant, Width, Height, etc.	90.7
[33]	100	CEDAR	3	156 words	Neural net.	Macro and micro features	94.0
	900	CEDAR	3	156 words			87.0
[1]	150	IAM	2	Paragraph	Hypothesis testing	Graphemes	86.0
	150	FSI	1	107 words			97.0
[29]	100	IAM	5	Paragraph	HMM	Global and local features	98.4
[30]	650	IAM	2	Paragraph	Distances	Graphemes and directional features	89.0
[31]	50	IAM	2	Paragraph	Distances	Global and local features	94

complexity of segmentation. The overall error rate achieved by the system trained with texture descriptors was about 13 percentage points smaller than the system trained with classical characteristics such as directional features. Besides, the handwriting texture can be used together with other features, e.g., directional features, improving their results.

We also demonstrate the impacts of increasing the number of writers for training in the proposed system. We have seen that the size of the training set does not have an important impact on the overall error rate, but it has an important role in reducing the false acceptance of the verification system. Based on these findings, it is reasonable to suggest a bigger training set when the requirements of the verification system demand small FPR. On the other hand, the number of writer in the training set can be relaxed, thus reducing the false rejection. Finally, we have demonstrated that it is possible to considerably reduce the FPR for a fixed TPR by combining the ROC produced by two different classifiers through the maximum likelihood analysis.

These results suggest that some kinds of writer selection process could be interesting to build better dissimilarity models for writer verification. Using those writers that feature excessive variability in his/her handwriting may not be helpful to build a stable machine learning model. Another aspect worth of investigation is the impact of the number of references used to perform the verification. Both issues will be subject of further investigation.

Acknowledgments This research has been supported by The National Council for Scientific and Technological Development (CNPq) grant 306358/2008-5.

References

1. Bensefia, A., Paquet, T., Heutte, L.: A writer identification and verification system. Pattern Recogn. Lett. **26**(13), 2080–2092 (2005)
2. Bertolini, D., Oliveira, L.S., Justino, E., Sabourin, R.: Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. Pattern Recogn. **43**(1), 387–396 (2010)
3. Brink, A., Schomaker, L., Bulacu, M.: Towards explainable writer verification and identification using vantage writers. In: 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp. 824–828 (2007)
4. Bulacu, M., Schomaker, L., Vuurpijl, L.: Writer identification using edge-based directional features. In 8th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, pp. 937–941 (2003)
5. Bush, A., Boles, W., Sridharan, S.: On measuring the distance between histograms. IEEE Trans. Pattern Anal. Mach. Intel. **27**, 1721–1732 (2005)
6. Cha, S.-H., Srihari, S.: Multiple feature integration for writer verification. In: 7th International Workshop on Frontiers on Handwriting Recognition, pp. 333–342 (2000)
7. Cha, S.N., Srihari, S.-H.: On measuring the distance between histograms. Pattern Recogn. **35**, 1355–1370 (2002)
8. Crettez, J.P.: A set of handwriting families: style recognition. In: 8th International Conference on Document Analysis and Recognition, Seoul, South Korea, pp. 489–494 (2005)
9. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **227**(8), 861–874 (2006)
10. Franke, K., Bunnemeyer, O., Sy T.: Ink texture analysis for writer identification. In: 8th International Workshop on Frontiers on Handwriting Recognition, pp. 268–273 (2002)
11. Freitas, C., Oliveira, L.S., Sabourin, R., Bortolozzi, F.: Brazilian forensic letter database. In: 11th International Workshop on Frontiers on Handwriting Recognition, Montreal, Canada (2008)
12. Gobineau, H., Perron, R.: Génétique de l'ecriture et étude de la personnalité: Essais de graphométrie. Delachaux and Niestlé, Lausanne (1954)

13. Goldfarb, L.: What is distance and why do we need the metric model for pattern learning. Pattern Recogn. **25**, 431–438 (1992)
14. Haker, S., Wells, W.M., Warfield, S.K., Talos, I., Bhagwat, J.G., Goldberg-Zimring, D., Mian, A., Ohno-Machado, L., Zou, K.H.: Combining classifiers using their receiver operating characteristics and maximum likelihood estimation. In: 8th International Conference on Medical Image Computing and Computer Assisted Intervention, pp. 506–514 (2005)
15. Haralick, R.M., Shanmugan, K.S., Dunstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **3**(6), 610–621 (1973)
16. Huder, R.A., Headrick, A.M.: Handwriting Identification: Facts and Fundaments. CRC Press, London (2000)
17. Impedovo, D., Pirlo, G.: Automatic signature verification: the state of the art. IEEE Trans. Syst. Man Cybern. C **38**(5), 609–635 (2008)
18. Leclerc, F., Plamondon, R.: Automatic signature verification: the state of the art 1989-1993. Int. J. Pattern Recogn. Artif. Intell. **8**(3), 643–660 (1994)
19. Marti, U.V., Messerli, R., Bunke, H.: Writer identification using text line based features. In: 8th International Conference on Document Analysis and Recognition, Seattle, USA, pp. 101–105 (2001)
20. Morris, N.: Forensic Handwriting Identification Fundamental Concepts and Principles. Academic Press, London (2000)
21. Oliveira, L.S., Justino, E., Freitas, C., Sabourin, R.: The graphology applied to signature verification. In: 12th Conference of the International Graphonomics Society, Salermo, Italy, pp. 286–290 (2005)
22. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Trans. Syst. Man Cybern. **8**, 62–66 (1978)
23. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recogn. **23**, 943–956 (2002)
24. Platt, J. et al.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Smola, A. (ed.) Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)
25. Said, H.E.S., Tan, T.N., Baker, K.D.: Personal identification based on handwriting. Pattern Recogn. **33**, 149–160 (2000)
26. Santana, O., Travieso, C.M., Alonso, J.B., Ferrer, M.A.: Writer identification based on graphology techniques. In: 42nd Annual IEEE International Carnahan Conference on Security Technology, pp. 167–173 (2008)
27. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 871–883 (1999)
28. Schlapbach, A.: Writer Identification and Verification. PhD thesis, Bern University (2007)
29. Schlapbach, A., Bunke, H.: A writer identification and verification system using hmm based recognizers. Pattern Anal. Appl. **10**, 33–43 (2007)
30. Schomaker, L., Bulacu, M.: Text-independent writer identification and verification using textural and allographic features. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 701–717 (2007)
31. Siddiqi, I., Vincent, N.: A set of chain code based features for writer recognition. In: 10th International Conference on Document Analysis and Recognition, Barcelona, Spain pp. 981–985 (2009)
32. Sollich, P.: Bayesian methods for support vecotr machines: evidence and predictive class probabilities. Mach. Learn. **46**(1–3), 21–52 (2002)
33. Srihari, S.N., Cha, S.-H., Arora, H., Lee, S.: Individuality of handwriting. J. Forens. Sci. **47**, 1–17 (2002)