Assessing Textural Features For Writer Identification on Different Writing Styles and Forgeries

Diego Bertolini[1], Luiz S. Oliveira[1], Edson Justino[2], Robert Sabourin[3]

[1]*Federal University of Parana, Department of Informatics, Curitiba, PR, Brazil*
[2]*Pontifical Catholic University of Parana, Curitiba, PR, Brazil*
[3]*Ecole de Technologie Superieure, Montreal, QC, Canada*

***Abstract*—In this study we assess the performance of textural descriptors for writer identification on different writing styles and also on forgeries. To do that, we have performed a series of experiments using the Firemaker database, which provides for the same writer texts written on three different writing styles and also copied forged text. Our experimental protocol is based on the dissimilarity framework and SVM classifiers, which were trained with LBP (Local Binary Pattern) and LPQ (Local Phase Quantization). The 250 writers of the database were divided into different configurations to observe the impacts of different sizes of the training set on the performance of the system. Our experimental results corroborates the fact that the texture is an interesting alternative for writer identification. The classifier trained with LPQ was able to produce error rates 23 percentage points smaller than those reported in the literature for upper-case and free writing styles. Regarding the forgeries, the LPQ-based classifier goes further reducing the error rate up to 44 percentage points depending on the writing style used for training.**

***Keywords*—Writer Identification; Texture;**

I. Introduction

Writer identification and verification has attracted a great deal of attention during the last decade where the main applications concern user assistance in querying large databases such as in forensic applications, digital libraries, and text retrieval from archives of scanned historical documents [1], [2], [3], [4].

The literature shows us two different approaches for writer identification. The first approach, also known as local approach, takes into account specific features of the handwriting and, in general, involves a segmentation algorithm. The features are extracted from characters or allographs [1], [2]. To avoid the complexity of segmentation, one can rely on the global approach, which tries to identify the writer of a document based on the look and feel of the writing. The best results of this approach were achieved by looking at the handwriting as a texture [5], [6], [7].

In spite of the good results provided by approaches based on texture, to the best of our knowledge, there is no work in the literature discussing the performance of this kind of approach on forgeries. To fill this gap, in this work we have performed a series of comprehensive experiments using the Firemaker database [8]. Besides assessing the performance on forgeries, we also evaluate textural features on three other writing styles available in the Firemaker database: copied, upper-case, and free writing styles. This study was inspired by the work of Schomaker et al. [3] where the writers characterise the performance of allograph-based features for different writing styles and forgeries.

To show the efficiency of the textural features on different writing styles and forgeries, we have used 250 writers from the Firemaker database considering 150 for testing and two different sizes for training (20 and 100 writers). The experimental protocol was adopted from [5], which is based on the dissimilarity [9], [10], [11] framework and SVM classifiers. Our experimental results show that the texture-based approach, using Local Binary Pattern (LPB) and Local Phase Quantization (LPQ) achieved similar performance to that reported in [3] for copied text and significantly better performance for the other handwriting styles. In the case of forgery, our system produced an error rates ranging from 6% to 16% depending on the writing style used for training. Considering a training set composed of a mix of different writing styles, the LPQ-based classifier achieved an error rate 36 percentage points lower than the error reported in [3].

II. The Dissimilarity Framework

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 and Q be two vectors in the feature space, labeled l_V and l_Q respectively. Let Z be the dissimilarity feature vector resulting from the dichotomy transformation $Z = |V - Q|$, where $|\cdot|$ is the absolute value. This dissimilarity feature vector has the same dimensionality as V and Q.

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 is assigned the label l_Z,

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

Figure 1 illustrates this transformation. Suppose there are three writers, $\{\omega_1, \omega_2, \omega_3\}$, and each one of them provides some samples. The feature extraction process extracts a vector from each sample, and these are shown in Figure 1a. Then, a dichotomy transformation takes place and computes the dissimilarity between the features of each pair of samples to form vectors. The distribution of such vectors, which we call dissimilarity feature vectors, are shown in Figure 1b.

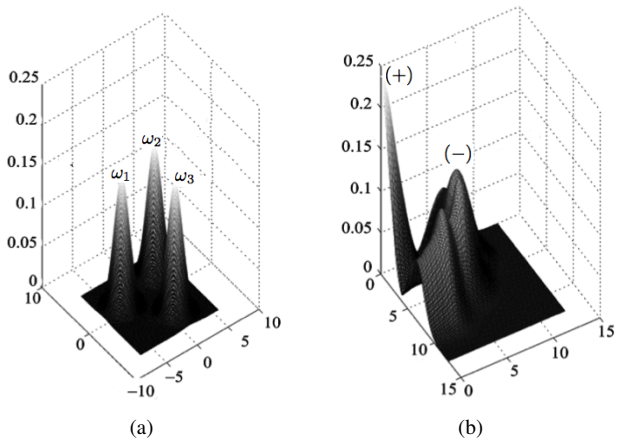


Figure 1. Dichotomy transformation: (a) samples in the feature space (b) samples in the dissimilarity space where (+) stands for the vectors associated to the within class and (-) stands for the vectors associated to the between class.

We can see in Figure 1 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 Figure 1 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 dissimilarity framework requires the classifiers to discriminate between genuine (positive) and forgeries (negative). To generate the positive samples to train the SVM classifier, we computed the dissimilarity vectors among the R genuine samples (references) of each writer 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 writer against one reference of other writers picked at random. Following the findings of our recent study [5], the best results were found using 9 references per writer.

III. Database

In the last few years different databases devoted for writer identification have been published in the literature [12], [13], [8]. However, in most of them it is impossible to assess the performance on forgery and different writing styles. One dataset that allows this kind of analysis is the Firemaker [8], which contains 250 writers and four pages per writer. Page 1 contains a copied text in natural writing style; Page 2 contains copied upper-case text; Page 3 contains copied forged text ("please write as if to impersonate another person") while Page 4 contains a self-generated description of a cartoon image in free writing style.

The text content and amount of written ink varies considerably per writer in this latter page. All pages were scanned at 300 dpi gray-scale, on lineated paper with a vanishing line color. The text to be copied has been designed in forensic praxis to cover a sufficient amount of different letters from the alphabet while remaining conveniently writable for the majority of suspects.

IV. Feature Extraction

In order to generate the texture, the document is binarized and scanned top-down and left-right to detect all the connected components of the image. Small components, such as periods, commas, strokes, and noise, are discarded. The bounding box of the 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 algorithm, described in details in [5], compacts the handwriting generating texture images. Then, the texture is segmented into nine 256×256 blocks. Figure 2 shows two examples of the handwriting texture produced by two different writers.

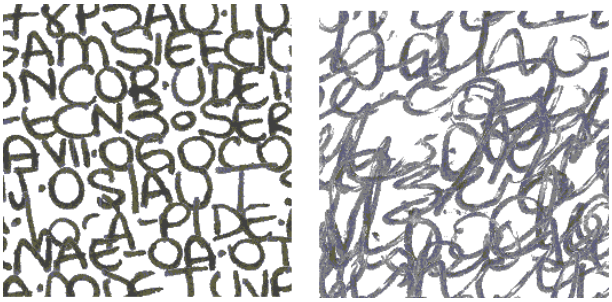


Figure 2. Examples of handwriting textures from two different writers.

After creating the textural fragments, the next step consists in dealing with representation. The literature shows us a long story of research in texture representation but recent works have shown that Local Binary Pattern (LBP) [14] and Local Phase Quantization (LPQ) [15] appear to be a very interesting alternatives to represent texture. They have been successfully applied to different problems achieving promising results. Besides, they are quite easy to implement. For these reasons we have adopted these two descriptors in our study. To make this paper self-contained, in the next subsection we describe briefly each one of them.

A. LBP

The original LBP proposed by Ojala et al. [14] labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value. Then, considering the results

as a binary number and the 256-bin histogram of the LBP labels computed over a region, they used this LBP as a texture descriptor. The LPB operator $\text{LBP}_{P,R}$ produces 2^P different binary patterns that can be formed by the P pixels in the neighbor set on a circle of radius R. However, certain bins contain more information than others, and so, it is possible to use only a subset of the 2^P LBPs. Those fundamental patterns are known as uniform patterns.

Accumulating the patterns that have more than two transitions into a single bin yields an LBP operator, denoted $\text{LBP}^{u2}_{P,R}$, with fewer than 2^P bins. For example, the number of labels for a neighborhood of 8 pixels is 256 for the standard LBP but 59 for LBP^{u2}. Then, a histogram of the frequency of the different labels produced by the LBP operator can be built [14]. In this work, the best results were achieved through the traditional configuration ($\text{LBP}^{u2}_{8,2}$), which generates a feature vector of 59 components.

B. LPQ

Proposed by Ojansivu e Heikkila [15], LPQ is based on quantized phase information of the Discrete Fourier Transform (DFT). It uses the local phase information extracted using the 2-D DFT or, more precisely, a Short-Term Fourier Transform (STFT) computed over a rectangular $M \times M$ neighborhood N_x at each pixel position x of the image $f(x)$ defined by

$$F(u,x) = \sum\nolimits_{y \in N_x} f(x-y) e^{-j2\pi u^T y} = w_u^T f_x \qquad (2)$$

where w_u is the basis vector of the 2-D DFT at frequency u, and f_x is another vector containing all M^2 image samples from N_x.

The STFT can be implemented using a 2-D convolutions $f(x)e^{-2\pi j u^T x}$ for all u. In LPQ only four complex coefficients are considered, corresponding to 2-D frequencies $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$, where a is a scalar frequency below the first zero crossing of the DFT $H(u)$. $H(u)$ is DFT of the point spread function of the blur, and u is a vector of coordinates $[u, v]^T$. More details about the LPQ formal definition can be found in [15], where Ojansivu e Heikkila introduced all mathematical formalism. At the end, we will have an 8-position resulting vector G_x for each pixel in the original image. These vectors G_x are quantized using a simple scalar quantizer (Eq. 3, and 4), where g_j is the jth component of G_x [15].

$$q_j = \begin{cases} 1, & \text{if } g_j \geqslant 0 \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

$$b = \sum_{j=1}^{8} q_j 2^{j-1}. \qquad (4)$$

The quantized coefficients are represented as integer values between 0-255 using binary coding (Eq. 4). These binary codes will be generated and accumulated in a 256-bin histogram, similar to the LBP method. The accumulated values in the histogram will be used as the LPQ 256-dimensional feature vector.

V. Experimental Results

In all 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 error rate (EER) used for evaluation purposes in this work is given by Equation 5, which is always computed on the testing set.

$$\text{Equal Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} \qquad (5)$$

where FP, FN, TP, and TN stand for False Positive, False Negative, True Positive, and True Negative, respectively.

As stated before, two different configurations of the database were considered. In the first case, we have used the number of writers for training and testing proposed in [3], i.e., 100 for training and 150 for testing. Thereafter, we reduced the number of writers in the training set to 20. The goal was to assess the impacts of a smaller training set in the performance of the system.

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.

In all experiments we have used $S = R = 9$, i.e., both reference and questioned documents were compacted and nine pieces of texture were extracted. The nine fragments are classified independently generating a partial decision and then a final decision is computed by combine all partial decisions. Different fusion rules were tried out but the Sum Rule produced the best results.

The structure of the experiments follows the same idea described in [3], which is divided into two parts. In the first part we assess the use of the different writing styles available in the database for training and testing the classifier, i.e., the classifier is trained and tested using only one writing style. Still in the first experiment, we increase the variability of the training set by mixing different writing styles while the testing was performed separately on copy, upper-case and free styles. In the second part, we have used the same idea, but the testing set is composed of forgeries only. The goal in this case is to verify, in the context of texture, the robustness of each writing style against forgeries.

A. Different Writing Styles

Tables I and II report the results on different writing styles for the two different configurations of the database (Training=100 and Training=20 writers). In the first part of both tables the classifiers were trained and tested using the same writing style. The best results were achieved using copied text in natural writing style, which features less variability. In

such a case, the best results were produced by the classifier trained with LPQ, which achieved an error rate of 2% in both database configurations. Comparing all the results in Tables I and II, however, we can notice the impacts of having a larger training set. The error rates for Upper case and Free writing styles drop considerably, especially for the LBP-based classifier. The LPQ-based classifier seems to be more robust to the smaller training set.

The results of Table I may be compared to those published in [3], since they use the same amount of users for testing and training. For the Copied text the results are quite similar. For the other two styles, where the writing variability is more prominent, the error rate reaches 6%. This is 24 percentage points smaller than the results published in [3], which corroborates to the argument that texture is a good alternative for writer identification [5]. For the sake of clarity, Figure 3 compares only the ROC curves produced by the classifiers trained with LPQ (Training=100, Testing=150), which have the best performance.

Table I. DIFFERENT WRITING STYLES. ERROR RATES FOR TRAINING = 100, TESTING =150

Writing Sytle		Descriptors		[3]
Training	Testing	LBP	LPQ	
Copy	Copy	2.0	**2.0**	3.0
Upper case	Upper case	9.0	6.0	30.0
Free	Free	7.0	7.0	30.0
Mix	Copy	10.0	5.0	-
Mix	U. case	21.0	13.0	-
Mix	Free	24.0	22.0	-

Table II. DIFFERENT WRITING STYLES. ERROR RATES FOR TRAINING = 20, TESTING =150

Writing Sytle		Descriptors	
Training	Testing	LBP	LPQ
Copy	Copy	4.0	2.0
Upper case	Upper case	21.0	9.0
Free	Free	13.0	11.0
Mix	Copy	5.0	6.0
Mix	U. case	23.0	14.0
Mix	Free	26.0	20.0

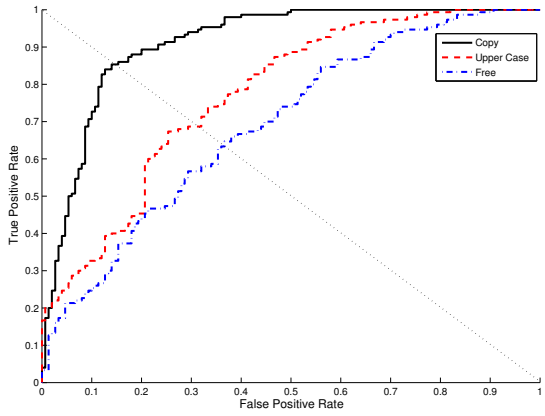


Figure 3. ROC Curves produced by the classifiers trained with LPQ

It is easy to observe from Figure 3 and Table I that the performance for upper-case and free writing styles are poorer, when compared to copied text. In the case of the upper case, this style produces textures with less discriminatory information, what explain the higher error rates. The Free writing style, in which the user must describe a cartoon image, features a large variability in the handwriting, what explains the lower performance.

One interesting aspect we have noticed in the Firemaker database, is that sometimes writers use a mix of writing styles, even when asked to employ a specific writing style. With this in mind, in the second part of this experiment we assessed the impact of using a mixture of different writing styles to train the models. The results are presented in the second part of Tables I and II. As we can see, using a training set with more variability (composed of different writing styles), does not increase the performance when the testing set contains a single style.

B. *Forged Text*

Regarding the experiments with forged text, we observed an opposite behaviour. The writing style with less variability, the copied text, achieves the worst performance in detecting forgeries. The free style, on the other hand, seems more suitable to detect forgeries in the context of textures. The smallest error rate, 6%, was achieved by the LPQ classifier trained with textures extracted from the free writing style. Similarly to the previous experiment, the use of all writing styles (mix) does not bring any benefit in terms of performance.

Tables III and IV report the results for the training set with 100 and 20 writers, respectively. The ROC curves for the LPQ-based classifiers are depicted in Figure 4. Similarly to the previous experiment, the results on Table III may be compared to those reported in [3], especially the last line where the training set is composed of a mix of writing styles. In such a case, the LPQ-based classifier was able to achieve an error rate 36 percentage points lower than that reported in [3].

Table III. EXPERIMENTS ON FORGERIES - ERROR RATES FOR TRAINING = 100, TESTING =150

Writing Sytle		Descriptors		[3]
Training	Testing	LBP	LPQ	
Copy	Forged	22.0	16.0	-
Upper case	Forged	9.0	9.0	-
Free	Forged	8.0	**6.0**	-
Mix	Forged	16.0	14.0	50.0

Table IV shows that reducing the training set impacts directly the performance of both classifiers. However, even in this scenario the error rates are still acceptable. Figure 5 compares the free style writing (a) and the forged text of the same writer. It exemplifies how difficult may be to identify a forged text in this database.

Table IV. EXPERIMENTS ON FORGERIES - ERROR RATES FOR TRAINING = 20, TESTING =150

Writing Sytle		Descriptors	
Training	Testing	LBP	LPQ
Copy	Forged	22.0	22.0
Upper case	Forged	16.0	14.0
Free	Forged	12.0	**10.0**
Mix	Forged	24.0	28.0

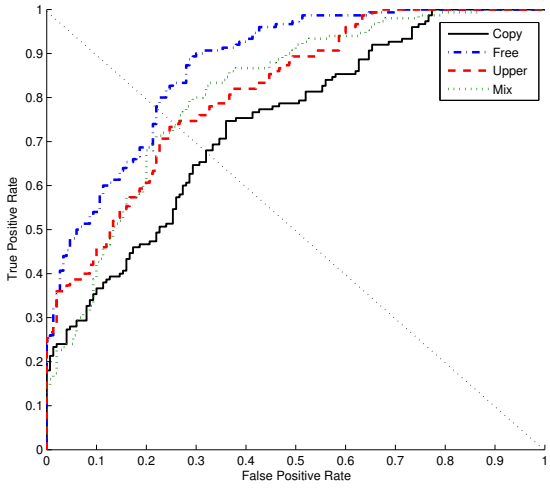
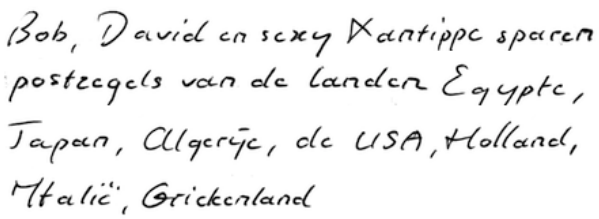
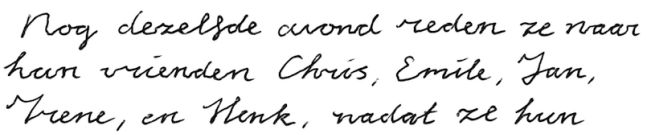


Figure 4. ROC Curves for forgeries



(a)



(b)

Figure 5. Two pieces of text from the same writer: (a) Free style (b) Forged.

VI. Conclusions

In this paper we have discussed the use of textural descriptors for writer identification. Our focus was to assess this concept on different writing styles and also on forgeries. All the experiments were carried out on the Firemaker database, which to the best of our knowledge is the only database for writer identification suitable for this task.

The results produced by the classifier trained with LPQ are very encouraging and compare favourably to other published methods [3]. In the case of forgery, our best result, an error rate of 6%, was achieved by a classifier trained with free writing style. Besides the compelling results, the use of texture to represent the writer identification problem offers the advantage of no manual measuring on text details since it is a segmentation-free method. It also can be easily applied to other, no-western scripts. Concerning future works, we will focus on the design of the ensemble of classifiers based on different representations and also on different techniques of dynamic selection of classifiers [16] to explore the huge variability of the handwriting styles.

References

[1] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2080–2092, 2005.

[2] L. Schomaker and M. Bulacu, "Text-independent writer identification and verification using textural and allographic features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 701–717, 2007.

[3] L. Schomaker, K. Franke, and M. Bulacu, "Using codebooks of fragmented connected-component contours in forensic and historic writer identification," *Pattern Recognition Letters*, vol. 28, pp. 719–727, 2007.

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

[5] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Texture-based descriptors for writer identification and verification," *Expert Systems With Applications*, vol. 40, pp. 2069–2080, 2013.

[6] R. Hanusiak, L. S. Oliveira, E. Justino, and R. Sabourin, "Writer verification using texture-based features," *International Journal on Document Analysis and Recognition*, vol. 15, pp. 213–226, 2012.

[7] H. E. S. Said, T. N. Tan, and K. D. Baker, "Personal identification based on handwriting," *Pattern Recognition*, vol. 33, pp. 149–160, 2000.

[8] L. Schomaker and L. Vuurpijl, "Forensic writer identification: A benchmark data set and a comparison of two systems," Nijmegen, Tech. Rep., February 2000.

[9] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, 2010.

[10] L. S. Oliveira, E. Justino, and R. Sabourin, "Off-line signature verification using writer-independent approach," in *International Joint Conference on Neural Networks*, 2007, pp. 2539–2544.

[11] S. Srihari, A. Xu, and M. Kalera, "Learning strategies and classification methods for offline signature verification," in *Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition*, 2004, pp. 161–166.

[12] C. Freitas, L. Oliveira, R. Sabourin, and F. Bortolozzi, "Brazilian forensic letter database," in *11th International Workshop on Frontiers on Handwriting Recognition*, Montreal, Canada, 2008.

[13] U. V. Marti and H. Bunke, "The IAM-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[14] T. Ojala, M. Pietikäinen, and D. Harwook, "Comparative studyof texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

[15] V. Ojansivu and J. Heikkila, "Blur insensitive texture classification using local phase quantization." in *Proc. Image and Signal Processing*, 2008, pp. 236–243.

[16] A. B. Jr, R. Sabourin, and L. S. Oliveira, "Dynamic selection of classifiers - a comprehensive review," *Pattern Recognition*, 2014.