Expert Systems with Applications 40 (2013) 3813–3827

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Feature representation selection based on Classifier Projection Space and Oracle analysis

Rafael M.O. Cruz [a,b], George D.C. Cavalcanti [a,*], Ing Ren Tsang [a], Robert Sabourin [b]

[a] Federal University of Pernambuco (UFPE), Center for Informatics (CIn), Av. Jornalista Anibal Fernandes s/n, Cidade Universitária, 50740-560 Recife, PE, Brazil
[b] Département de Génie de la Production Automatisée, École de Technologie Supérieure, 1100 rue Notre-Dame Ouest, Montréal, QC, Canada H3C 1K3

ARTICLE INFO

Keywords:
Feature representation selection
Multiple classifier system
Classifier Projection Space
Handwritten recognition

ABSTRACT

One of the main problems in pattern recognition is obtaining the best set of features to represent the data. In recent years, several feature extraction algorithms have been proposed. However, due to the high degree of variability of the patterns, it is difficult to design a single representation that can capture the complex structure of the data. One possible solution to this problem is to use a multiple-classifier system (MCS) based on multiple feature representations. Unfortunately, still missing in the literature is a methodology for comparing and selecting feature extraction techniques based on the dissimilarity of the feature representations. In this paper, we propose a framework based on dissimilarity metrics and the intersection of errors, in order to analyze the relationships among feature representations. Each representation is used to train a classifier, and the results are compared by means of a dissimilarity metric. Then, with the aid of Multidimensional Scaling, visual representations are obtained of each of the dissimilarities and used as a guide to identify those that are either complementary or redundant. We applied the proposed framework to the problem of handwritten character and digit recognition. The analysis is followed by the use of an MCS built on the assumption that combining dissimilar feature representations can greatly improve the performance of the system. Experimental results demonstrate that a significant improvement in classification accuracy is achieved due to the complementary nature of the representations. Moreover, the proposed MCS obtained the best results to date for both the MNIST handwritten digit dataset and the Cursive Character Challenge (C-Cube) dataset.

1. Introduction

The selection of the feature extraction algorithm is known to be an important factor in the performance of any recognition system (Trier, Jain, & Taxt, 1995). However, designing a single feature extraction algorithm in a complex recognition problem that can recognize every kind of pattern is unlikely, because of the high degree of variability of the data. Some features might present a better result for a predetermined class of patterns. For instance, in the problem of handwritten character recognition, one feature extraction algorithm might represent lowercase letters better, while another is a more robust performer for uppercase letters. Moreover, every feature extraction technique represents a different aspect of the image, such as concavities (Oliveira, Sabourin, Bortolozzi, & Suen, 2002), character structure (Kavallieratou, Sgarbas, Fakotakis, & Kokkinakis, 2003), edges (Chim, Kassim, & Ibrahim (1998)), projections (Chim et al., 1998), and directional information based on the gradient (Ping & Lihui, 2002).

In our opinion, the information captured by different feature extraction techniques can be complementary, and a multiple-classifier system (MCS) developed using multiple feature representations achieves higher classification performance. Unfortunately, there is no framework in the literature for comparing and analyzing the relationships among feature representations. Feature extraction techniques are only compared based on classification accuracy, and none analyzes the diversity among them.

In the MCS context, the system can only perform better than the best individual classifier when there is diversity among the classifiers (Shipp & Kuncheva, 2002), so that they achieve different solutions. In other words, we seek significantly different representations because they produce different solutions – combining techniques that perform identically is not useful.

In this paper, we propose a novel framework to study the relationships among the various feature representations. Each feature extraction technique is used to train a classifier. Their results are evaluated based on dissimilarity/diversity measures (Giacinto & Roli, 2001; Shipp & Kuncheva, 2002). Then, the relationships obtained are used to project each representation onto a space

* Corresponding author. Tel.: +55 81 2126 8430x4346; fax: +55 81 2126 8438.
E-mail addresses: rmoc@cin.ufpe.br (R.M.O. Cruz), gdcc@cin.ufpe.br (G.D.C. Cavalcanti), tir@cin.ufpe.br (I.R. Tsang), robert.sabourin@etsmtl.ca (R. Sabourin).
URL: http://www.cin.ufpe.br/~viisar (G.D.C. Cavalcanti).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.096

(Classifier Projection Space Pekalska, Duin, & Skurichina, 2002). Each feature extraction technique is represented by a point, and the distance between two points corresponds to the difference between them. In this way, a spatial relationship is achieved between different representations. Feature representations that are close to one another produce similar results, and so may be redundant. Combining them is unlikely to improve the accuracy of the system, and they could be removed from the system without a significant loss in performance. Those that are far apart are able to correctly recognize different classes of images, and should be considered for an MCS.

The purpose of our proposed framework in this context is twofold: to perform an analysis of the complementarity within a subset of feature extraction methods, and to serve as a methodology for identifying and removing feature representations that produce similar results. In this way, a more efficient MCS is achieved.

We apply this framework to the problem of handwritten character and digit recognition. This is an important area in the field of pattern recognition, because of the many practical applications that exist, such as mail sorting, bank check analysis, and form processing, all of which depend on quality feature extraction techniques. Pattern recognition in handwritten documents is a major challenge, owing to the diversity of handwriting styles. A writer can, for example, change his writing style as a result of a change in his neurological status, the type of pen he uses, and his hand position (Schomaker & Bulacu, 2004), especially if the shapes of the characters are complex (Srihari, Tomai, Zhang, & Lee, 2003).

A total of nine feature extraction techniques for handwritten recognition are evaluated here. Two of them, Modified Edge-Maps and Multi-Zoning, are based on classical algorithms. We selected techniques that capture different views of the image, such as concavities and projections, as well as techniques that capture the same type of information, such as directional information based on the gradient, but are extracted using different algorithms. Our analysis enables us to answer the following questions:

1. Do different feature extraction techniques present complementary information (i.e. are they able to correctly classify different images)?
2. Are feature extraction techniques that use a similar approach (e.g. different methods for extracting the gradient) less complementary than techniques that use different characteristics (e.g. edges, concavities)?
3. Can the proposed framework be used to select a subset of feature representations?

We perform an analysis of feature representations, which serves as the basis on which we propose a novel MCS for handwritten recognition. The proposed system is applied to two different handwritten recognition tasks: digit recognition, and cursive character recognition. For the handwritten digit recognition problem, we use the MNIST database, which is a very well-known benchmark. For cursive character recognition, we use the Cursive Character Challenge database (C-Cube). We carry out a sensitivity analysis for both cases, and demonstrate that the use of complementary feature representations greatly improves recognition performance. We also show that a scheme that includes a Multi-Layer Perceptron (MLP) neural network trained to combine the classifiers presented the highest accuracy rates in both cases, and these rates are also the best results obtained for these databases to date.

This paper is organized as follows. The framework for feature representation analysis is introduced in Section 2. Section 3 describes the nine feature extraction techniques studied in this paper. The evaluation of each feature extraction algorithm and the sensitivity analysis of these algorithms are shown in Section 4. Section 5 shows the performance of the system when an MCS is designed based on different feature representations. Finally, our conclusion is presented in the last section.

2. Feature representation analysis

This section describes the feature representation selection scheme shown in Fig. 1. The first step in this approach is to extract m different feature representations, $F_1,\ldots,F_m$, of the patterns from the data (DB). These feature representations are used to train m classifiers, $C_1,\ldots,C_m$, separately. Then, the dissimilarity matrix D (Section 2.1) and its projection onto the classifier space $\tilde{D}$ (Section 2.2) are computed. The matrix $\tilde{D}$ contains the spatial relationship between the classifiers that is equivalent to their dissimilarities (matrix D). That spatial relationship is used to perform the sensitivity analysis (Section 2.3), from which redundant representations can be identified. Finally, a subset $m' \subset m$ of the feature representations is selected.

2.1. Dissimilarity matrix

The matrix D is an $m \times m$ symmetrical matrix, where each member $d(i,j)$ represents the dissimilarity between the classifiers C_i and C_j. In order to compute D, we first need to select an appropriate metric that measures the difference between feature representations. There are many diversity measures in the literature

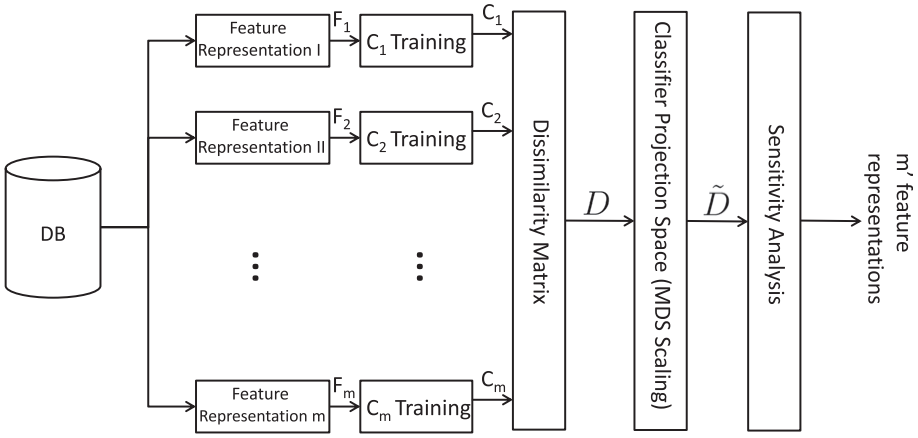

Fig. 1. Overview of the proposed feature representation selection scheme. Each F_i represents one feature representation, and it is used to train the classifier C_i. Based on the output of each pair (feature representation, classifier), we compute the dissimilarity matrix D that is used to perform the classifier projection $\tilde{D}$ through multidimensional scaling (MDS). The matrix $\tilde{D}$ is used to analyze the complementarity of the feature representations and perform the selection.

(Shipp & Kuncheva, 2002). We selected the Double Fault (Giacinto & Roli, 2001), because it has already been demonstrated that this measure presents a positive correlation with ensemble accuracy (Kuncheva & Whitaker, 2003). Eq. (1) shows the Double Fault measure between a pair of classifiers C_i and C_j.

$$d(i,j) = \frac{N^{00}}{N^{00} + N^{01} + N^{10} + N^{11}} \quad (1)$$

where N^{ij} is the number of examples correctly classified (1) or misclassified (0) for the classifiers C_i and C_j respectively. In other words, the Double Fault measures the probability that the same pattern is misclassified by both classifiers.

2.2. Classifier Projection Space

After the dissimilarity matrix D has been obtained, the next step is to project each feature representation onto the Classifier Projection Space (CPS). The CPS is a $\mathbb{R}^{\kappa}$ space, where each classifier is represented as a point, and the Euclidean distance between two classifiers represents their dissimilarity (Pekalska et al., 2002). Classifiers that are similar are closer together in the CPS, while those that are less similar are further apart. In this way, it is possible using CPS to obtain the spatial representations of all the classifiers. This spatial representation provides a better understanding of the relationships among the classifiers than when only the value of the diversity measure is used. The diversity measure only describes the relationship between a pair of classifiers, while the CPS shows the relationships among all the classifiers. A two-dimensional CPS is used for better visualization. In order to obtain a two-dimensional classifier projection, a dimensionality reduction of the data is required. This can be achieved using Multidimensional Scaling (MDS) (Cox & Cox, 2000; Pekalska et al., 2002), which refers to a group of methods used to visualize high-dimensional data mapped to a lower dimensional space (Pekalska & Duin, 2002).

Given the dissimilarity matrix D, a configuration X of m points in $\mathbb{R}^k$, $(k \leqslant m)$ is computed using a linear mapping, called classical scaling (Cox & Cox, 2000). The process is performed through rotation and translation, such that the distances after dimensionality reduction are preserved. The projection X is computed as follows: first, a matrix of the inner products is obtained by the square distances $B = -\frac{1}{2}JD^2J$, where $J = I - \frac{1}{m}UU^T$, and I and U are the identity matrix and unit matrix respectively. J is used as a normalization matrix, so that the mean of the data is zero. The eigendecomposition of B is then obtained, $B = Q\Lambda Q^T$, where Λ is a diagonal matrix containing the eigenvalues (in decreasing order) and Q is the matrix of the corresponding eigenvectors. The configuration of points in the reduced space is determined by the k largest eigenvalues. Therefore, X is uncorrelated in the space $\mathbb{R}^k$, $X = Q_k\sqrt{\Lambda_k}$. In our case, $k = 2$.

MDS is obtained by applying the Sammon mapping over X. The Sammon mapping is a nonlinear projection that preserves the distances between the points (Cox & Cox, 2000; Pekalska et al., 2002). The mapping is performed by defining a function, called stress function $\mathcal{S}$ (Eq. (2)), which measures the difference between the original dissimilarity matrix D and the distance matrix of the projected configuration, $\tilde{D}$, where $\tilde{d}(i,j)$ is the distance between the classifiers i and j in the projection X, as defined in (2):

$$\mathcal{S} = \frac{1}{\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} d(i,j)^2} \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} (d(i,j) - \tilde{d}(i,j)) \quad (2)$$

In other words, the objective of $\mathcal{S}$ is to minimize the difference between D and $\tilde{D}$, and so the projection onto the CPS is found in iterative fashion. The algorithm starts with an initial representation of points in Euclidean space (the configuration of points in X with its corresponding distance matrix $\tilde{D}$). Then, the configuration of the points is adjusted to minimize $\mathcal{S}$. A scaled gradient algorithm (Pekalska et al., 2002) is used for this purpose. In the end, the distances between the classifiers correspond to an approximation of their original dissimilarity.

Fig. 2 shows an example of the CPS space for different feature representations extracted from the Iris dataset.[1] This dataset consists of four features. In order to simulate different feature representations, we use random combinations of two and three features. Ten different representations were generated: FS I to FS VI are combinations of two features, FS VII to FS IX are combinations of three features, and FS X is a representation consisting of all four features. A Perceptron was used as the classifier for each feature representation.

2.3. Sensitivity analysis

The first step in the sensitivity analysis is to use the CPS as a visual tool to group feature representations based on the spatial information provided by the CPS. In Fig. 2, we can observe that there are three feature representations that are really close together: FS V, FS IX, and FS X. Consequently, they are probably redundant. In contrast, some feature representations, such as FS II and FS VI, are far apart, and can be considered to be from different groups. We can see that the CPS is used to identify groups of representations that perform in similar fashion.

The second step is to analyze the performance of some combinations of feature representations. This is achieved using the concept of the Oracle, which produces the best possible result of any combination of classifiers (Kuncheva, 2002b). It considers that the ensemble obtains the correct classification if at least one classifier produces the true label. So, based on the analysis of the error performed by Oracle, it is possible to know whether or not individual classifiers are able to correctly recognize different patterns.

From the analysis in Fig. 2, we constructed two diagrams. The first is composed of feature representations that are close together: FS V, FS IX, and FS X (Fig. 3(a)). The second is composed of representations that are far apart, and can be considered to belong to different groups: FS II, FS III, and FS VI (Fig. 3(b)). The number inside each circle indicates the number of errors committed by each classifier. The area where the classifiers intersect represents the errors committed by all of them, and can be viewed as the error obtained by the Oracle combination (i.e. (FS V $\cap$ FS IX) is the number of patterns misclassified by the Oracle combination).

The total number of errors obtained using FS X is 28 (Fig. 3(a)). However, none of these errors was committed by this feature representation alone (i.e. an individual error). The majority of the errors lie at the intersection of the three feature spaces. In other words, 23 patterns are misclassified in the three feature subspaces, while 4 and 2 are common errors obtained by the intersections (FS X $\cap$ FS IX) and (FS X $\cap$ FS V), respectively. So, errors committed by classifiers in the same group are likely to occur in the same patterns, which means that combining them is unlikely to improve recognition performance.

In contrast, when representations that are far apart are combined (i.e. they belong to different groups, such as FS II, FS III, and FS VI) (Fig. 3(b)), we can observe that the intersection of the errors produces a lower value. For instance, from the 19 errors committed by FS III, 16 occur individually. The intersections (FS III $\cap$ FS II) and (FS III $\cap$ FS VI) produce errors of 1 and 2, respectively. Moreover, looking at the intersection of the three techniques, we note that no pattern was misclassified by all the techniques, as opposed to 23 in Fig. 3(a). Therefore, these feature representations can be considered complementary, since they can correctly recognize different patterns.

[1] archive.ics.uci.edu/ml/.

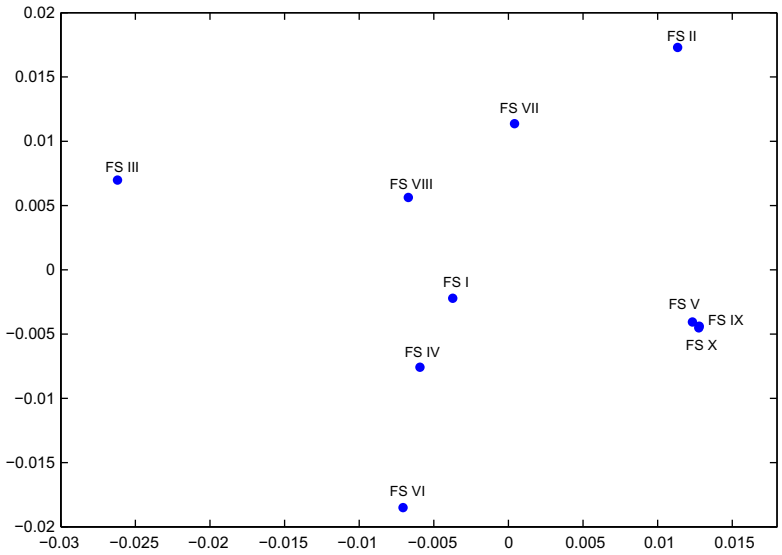

Fig. 2. Example of a two-dimensional CPS plot for different feature representations extracted from the iris dataset.

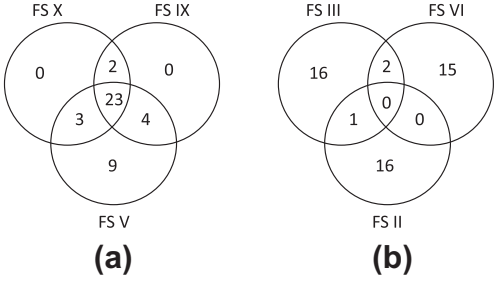

Fig. 3. Oracle error analysis for different feature representations trained on the Iris dataset. (a) Representations that belong to the same cluster. (b) Representations that belong to different clusters.

Consequently, we can identify representations that are redundant using the sensitivity analysis. From the point of view of an MCS, there is no advantage to combining FS V with FS IX and FS X for the Iris dataset, as they behave almost identically. So, instead of using m representations, we can use the sensitivity analysis to select a more efficient set $m' = 3$ (FS II, FS III, and FS VI) that consists only of dissimilar representations. We expect that the subset m' produces results that are better than, or at least comparable to, the whole set m. Applying this methodology in the context of feature representations, it is possible to compare different representations and select only the most dissimilar ones. The selected feature representation is used to construct a more robust MCS for pattern recognition problems.

3. Feature extraction methods

Feature extraction can be defined as a means for obtaining the most relevant information to be used in the classification procedure (Devijver & Kittler, 1982). There are several feature extraction techniques, and choosing a technique can be considered the most important factor in the achievement of high accuracy rates in a pattern recognition problem (Trier et al., 1995). A total of nine feature extraction algorithms are summarized below. Feature sets I to VII have been proposed by others (Camastra, 2007; Chim et al., 1998; Kavallieratou et al., 2003; Oliveira et al., 2002; Zhang, Bui, & Suen, 2007), and feature sets VIII and IX are new contributions.

3.1. Feature set I: Structural Characteristics

This feature set is obtained by combining projections and profiles in a single feature vector. First, the input image is scaled to a 32×32 matrix. Then, three types of histogram (horizontal, vertical, and radial) and two types of profile (radial in-out and radial out-in) are computed.

The horizontal and vertical histograms (Fig. 4(b) and (c)) are calculated by summing the number of black pixels in each line and column respectively. So, 32 features are generated for each histogram.

The radial histogram (Fig. 4(d)) is computed as the number of black pixels in 72 directions at 5° intervals. The process progresses from the centroid of the image to its border, and 72 features are generated.

Radial in-out and radial out-in profiles are defined by the position of the first and last black pixel respectively, from a search that progresses from the centroid of the image to its border in 72 directions at 5° intervals. In this way, each profile generates 72 features. These features form a 280-dimensional feature vector (32 horizontal projections + 32 vertical projections + 72 radial projections + 72 in-out profiles + 72 out-in profiles). Details of this technique are described in Kavallieratou et al. (2003).

3.2. Feature set II: Image Projections

This method consists of extracting the radial and diagonal projections. The diagonal projections are computed by grouping the pixels in two diagonal lines (45° and −45°). A total of 32 features are obtained for each diagonal.

To extract the radial projections, the image must first be divided into four quadrants: top, bottom, right, and left. The quadrants are used to remove rotational invariance, which is an undesirable characteristic in handwritten recognition, since it makes it impossible to distinguish between some digits (e.g. digits 6 and 9).

For each quadrant, the radial projections are obtained by grouping pixels from its radial distance to the centroid of the image. The values of each projection are normalized to a $[0-1]$ range. The normalized features are combined into a single vector containing 128 features (16 for each radial projection and 32 for each diagonal projection). More details about this procedure are described in Chim et al. (1998).

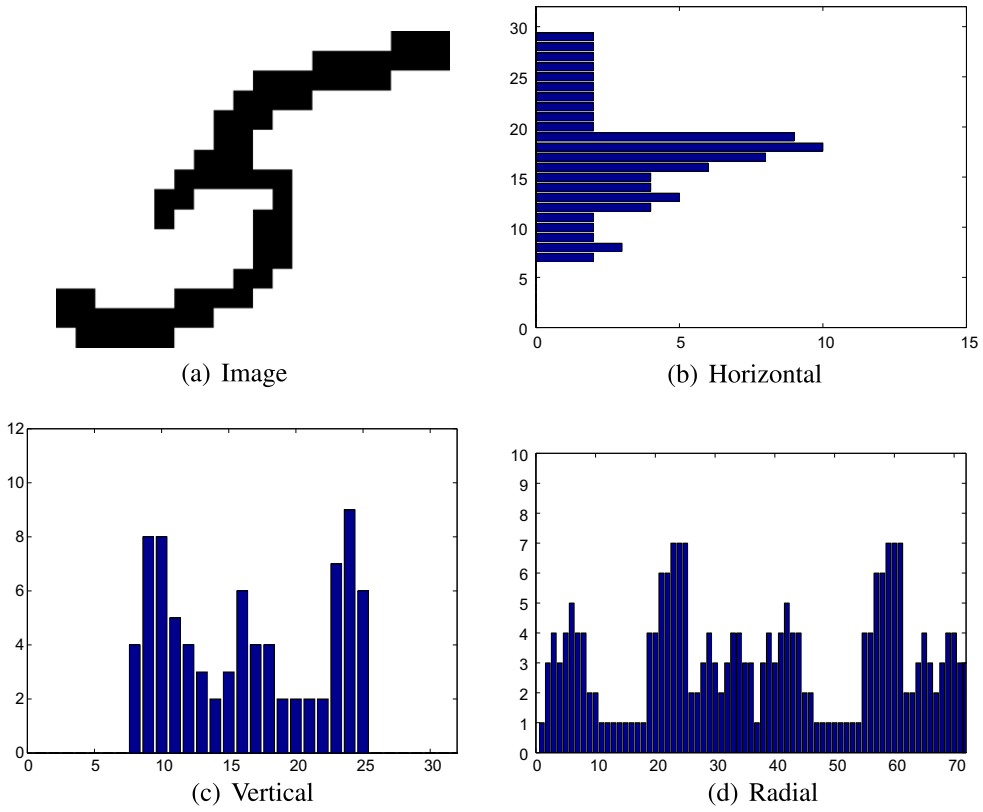

Fig. 4. Feature set I: example of a 5 digit projections.

3.3. Feature set III: Concavity Measurement

These features are obtained using the following steps: The image (Fig. 5(c)) is scaled to a matrix 18×15, and divided into six zones. Each part contains its own 13-dimensional feature vector, and the position of each feature vector corresponds to one of the 13 possible configurations (Fig. 5(d)).

For each white pixel (background), the algorithm conducts a search, starting from that pixel and moving in each of the four "main directions" (Fig. 5(a)). The search continues until a black

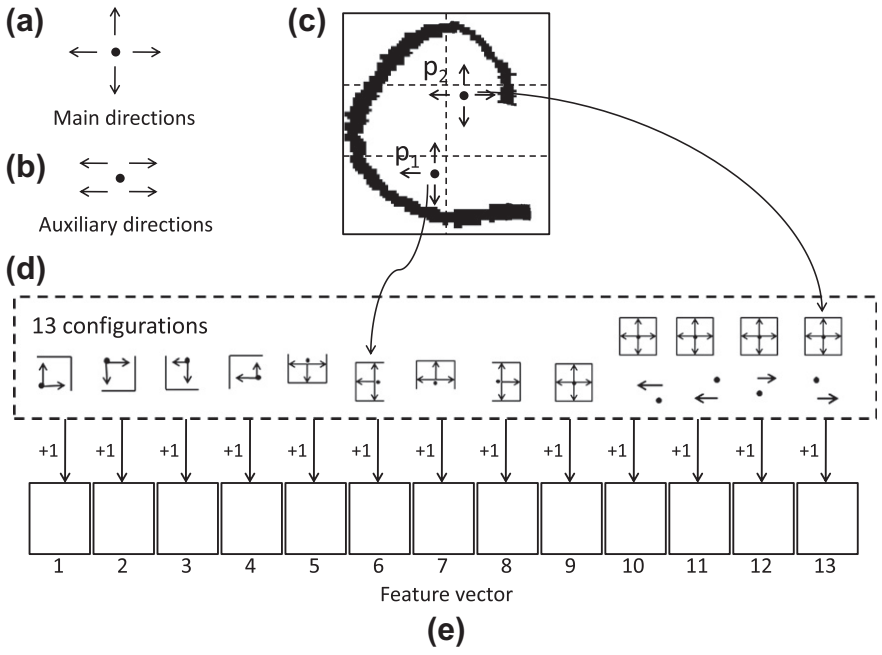

Fig. 5. Feature set V: the Concavity Measurement procedure. (a) Main directions. (b) Auxiliary directions. (c) Query image. (d) The thirteen possible configurations. (e) Feature vector.

pixel (foreground) is found, or when the end of the image is reached. Finally, the number of directions ending with a black pixel is computed, as are the directions themselves, each of which corresponds to one of the 13 possible configurations (Fig. 5(d)). So, the configuration in the feature vector corresponding to the result of the search is incremented.

However, in some cases, the search may find a black pixel in the four "main directions", but that pixel is not in a closed area. In order to guarantee that the white pixel is in a closed region, a new search is performed using the "auxiliary directions" (Fig. 5(b)). If the search using one of the auxiliary directions reaches the end of the image without finding a black pixel, the correct configuration (from the 10th to the 13th) is incremented. Otherwise, the point is in a closed region (9th position of the feature vector).

To better understand the method, we analyze two cases. In the case of P_1, the search finds a black pixel in three directions: top, bottom, and left. So, the configuration corresponds to the 6th position of the vector (Fig. 5(e)), and this position is incremented. In the case of P_2, the search in the four main directions finds a black pixel. However, using the auxiliary directions, the search also finds that the point is not in a closed region (no black pixel was found in the bottom right auxiliary direction). Therefore, P_2 corresponds to the 13th configuration.

These steps are computed for the six zones separately. At the end of the process, the feature vectors of each zone are combined into a single vector with 78 (13×6) features. A detailed description of the algorithm is presented by Oliveira et al. (2002).

3.4. Feature set IV: MAT-based Directional Gradient

This algorithm computes the gradient components of a grayscale image. So, the first step in this procedure is to transform a binary image into a pseudo-grayscale image using the Medial Axial Transformation (MAT) algorithm. The Sobel operators in the horizontal S_x and vertical S_y directions are applied to the pseudo-grayscale image I_m, generating the X-gradient image I_{m_x} and the Y-gradient image I_{m_y}. These are defined as:

$$I_{m_x} = I_m * S_x \tag{3}$$

$$I_{m_y} = I_m * S_y \tag{4}$$

For each pixel, the magnitude $r(i,j)$ and the phase $\Theta(i,j)$ are defined as:

$$r(i,j) = \sqrt{I^2_{m_x}(i,j) + I^2_{m_y}(i,j)} \tag{5}$$

$$\Theta(i,j) = \tan^{-1} \frac{I^2_{m_y}(i,j)}{I^2_{m_x}(i,j)} \tag{6}$$

In order to generate a fixed number of features, the phase of each pixel $\Theta(i,j)$ is quantized into eight directions at $\pi/4$ intervals each. Then, the image is divided into 16 equally spaced sub images, and, for each sub image, the number of pixels in each of the eight directions is used as a feature. So, the feature vector size is equal to 128 (16 sub images $\times$ 8 directions). Details of this feature extraction algorithm can be found in Zhang et al. (2007).

3.5. Feature set V: Binary Directional Gradient

This algorithm computes the gradient components of a binary image. The gradient is computed using the same procedure as that of a MAT-based Directional Gradient, defined in Section 3.4, except that no MAT transform is needed, because a binary image is used instead of a grayscale one. A total of 128 features are extracted per image.

3.6. Feature set VI: Median Gradient

In this technique, the image is first enhanced using a median filter to remove noise. Next, the Robert operators (Gonzalez & Woods, 2006) in the horizontal R_x and vertical R_y directions are applied to the filtered image to generate the X-gradient image I_{m_x} and the Y-gradient image I_{m_y}.

$$I_{m_x} = I_m * R_x \tag{7}$$

$$I_{m_y} = I_m * R_y \tag{8}$$

The gradient is computed using the same procedure as described in Section 3.4 section, generating 128 features. This method is described in detail by Zhang et al. (2007).

3.7. Feature set VII: Camastra 34D

This feature extraction algorithm was proposed by Camastra (2007). The image is divided into 16 sub images (cells), forming a 4×4 grid with a small overlap between them. Two operators are computed for each cell. The first is similar to the Zoning algorithm, and computes the number of black pixels (foreground) relative to the total number of black pixels in the whole image. The difference is that, in the Zoning algorithm, the number of black pixels is computed relative to the number of pixels in each zone. The second is a directional operator, which estimates the directions of the pixels. The method defines N equally spaced lines in the selected direction, after which the number of black pixels in each line is computed. The same steps are performed for the orthogonal direction. The difference between the selected direction and the orthogonal direction is used as a feature. The direction selected in this implementation was 0°, having the orthogonal direction of 90°. This results in a feature vector with 32 values. Two additional pieces of information were used as global features: The width/height ratio and the portion of the character that is below the baseline. The final vector consists of 34 features (16×2 local features + 2 global features).

3.8. Proposed feature extraction algorithms

3.8.1. Feature set VIII: Multi-Zoning

The idea behind using multiple configurations of zones simultaneously is to compute information from the image at different levels of detail. Using larger zones, global information about the shape of the character can be computed. In smaller zones, the focus is on local details, which are important for distinguishing between characters with similar shapes (e.g. digits 2 and 3). As a result, both global and local information is extracted at the same time.

This algorithm works as follows: an $M \times N$ image is divided into several sub images, and the percentage of black pixels in each sub image is used as feature. To achieve better recognition performance, many different divisions (Fig. 6) are selected and grouped to form the feature vector. A total of thirteen different configurations ($3 \times 1, 1 \times 3, 2 \times 3, 3 \times 2, 3 \times 3, 1 \times 4, 4 \times 1, 4 \times 4, 6 \times 1, 1 \times 6, 6 \times 2, 2 \times 6$, and 6×6) were chosen, resulting in 123 (3 + 3 + 6 + 6 + 9 + 4 + 4 + 16 + 6 + 6 + 18 + 18 + 36) features.

The Multi-Zoning technique differs from previous zoning techniques, such as the one described by Impedovo, Lucchese, and Pirlo (2006), in that the latter use only one zoning configuration. In the proposed method, instead of searching for an optimal division, we use multiple divisions, in order to have a representation of the image at different levels of detail. Moreover, we expect to achieve a better result using multiple configurations, since it is difficult to find a single configuration that can deal with the high degree of variability among handwriting styles.

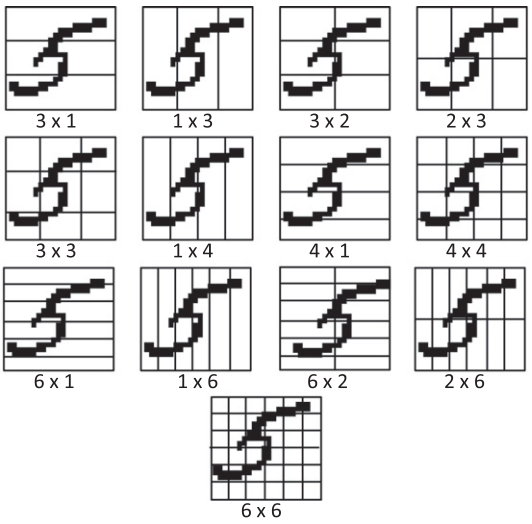

Fig. 6. Feature set VIII: thirteen configurations used in the Multi-Zoning technique.

3.8.2. Feature set IX: Modified Edge Maps

This algorithm is a modified version of the Edge Maps algorithm of Chim et al. (1998). An $M \times N$ image is first thinned using the Zhang and Suen algorithm (1984) and scaled to a 25×25 matrix. Then, the Sobel operators Gonzalez and Woods (2006) are used to extract four distinct edge maps: one horizontal, one vertical, and two diagonal (45° and −45°). Fig. 7 shows the four edge maps and the image after the thinning process has been performed.

The four edge maps and the thinned image are then divided into 25 sub images of 5×5 pixels each. The features are obtained through the computation of the percentage of black pixels in each sub image (25 features for each map). They are then combined to form a single feature vector containing 125 (25×5) features. The original algorithm, the Edge Maps algorithm of Chim et al. (1998), does not compute the percentage of black pixels per sub image, but instead uses the value of each pixel in greyscale as features.

4. Empirical evaluation of feature extraction techniques

The analysis of the feature extraction techniques was performed by conducting experiments using two different handwritten recognition problems: digit recognition and cursive character recognition. In the latter experiment, the Cursive Character Challenge database was used, while the handwritten digit recognition experiment was performed using the MNIST database. Both databases are publicly accessible, and both have been widely used as benchmarks.

4.1. C-Cube database

C-Cube is a public database available on the Cursive Character Challenge website Camastra, Spinetti, and Vinciarelli (2006). It consists of 57,293 images, including both uppercase and lowercase letters, manually extracted from the CEDAR and United States Postal Service (USPS) databases. As reported by Camastra et al. (2006), there are three advantages to using this database:

1. It is already divided into training sets and test sets, and so the results of different researchers can be rigorously compared.
2. It contains not only images, but also their feature vectors extracted using the algorithm proposed by Camastra (2007).
3. The results obtained using the state-of-the-art methods still leave room for significant improvement.

The database is divided into 38,160 (22,274 lowercase and 15,886 uppercase) images for training, and 19,133 (11,161 lowercase and 7972 uppercase) images for testing. All the images are binary and variable in size. For each image, four additional pieces of information are provided as global features: the distance between the base and the upper line, the distance between the upper extremity and the baseline, the distance between the lower

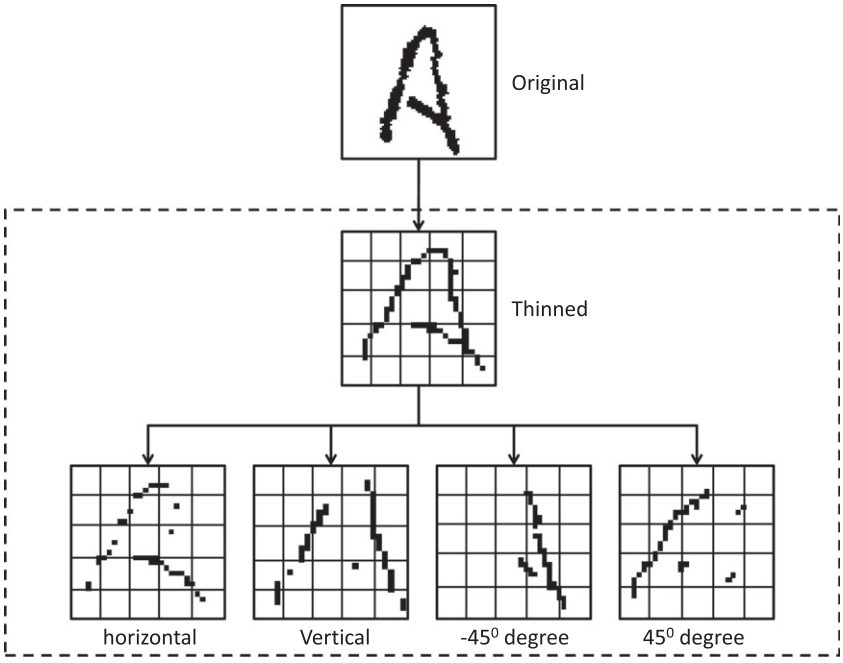

Fig. 7. Feature set IX: example of the process for obtaining the features for a character "A".

extremity and the baseline, and the width/height ratio. The samples, which varied in number per class, were selected based on their frequency of occurrence in the documents extracted from the CEDAR and USPS datasets. Figs. 8 and 9 show the distribution of the lowercase and uppercase letters respectively.

Thornton, Blumenstein, Nguyen, and Hine (2009) observed that the image files (*test.chr* and *training.chr*) available on the C-Cube website do not match the feature vectors (*test.vec* and *training.vec*). For this reason, they labeled the dataset with the feature vectors as *Split A* and the dataset with the image files as *Split B*. In this work, only *Split B* is used, since the image files of the *Split A* are not available.

4.2. MNIST database

MNIST is a well-known handwritten digit recognition database. It contains 60,000 images for training and 10,000 images for testing. All the images in the dataset are size-normalized and centered to a 28×28 image.

The advantage of using this database is twofold. First, the images are already preprocessed. Second, the database is already divided into a test set and a training set. This makes it easy to compare the results obtained by different researchers.

4.3. Experimental protocol

All the experiments were conducted using a three layer MLP, trained with the *Resilient Backpropagation* (RPROP) (Riedmiller & Braun, 1993) algorithm. This algorithm was chosen because it features a faster convergence rate and produces a better result than the conventional *Backpropagation* (Cruz, Cavalcanti, & Tsang, 2010a, 2010b).

The training set was divided into two parts: 80% for training, and 20% for validation. In addition, the division was performed maintaining the distribution of each class, so the MLP network is capable of estimating the Bayesian *a posteriori probability* (Richard & Lippmann, 1991). Consequently, their results can be combined through a probabilistic framework.

In every experiment, the number of nodes in the hidden layer was selected by means of the *crossvalidation* method using the training data. The search was performed by varying the number of nodes from 150 to 600 at 10-point intervals. Then, we replicated the configuration that achieved the best Results 10 times to obtain the average result. The weights of the neural networks were randomly initialized before each execution.

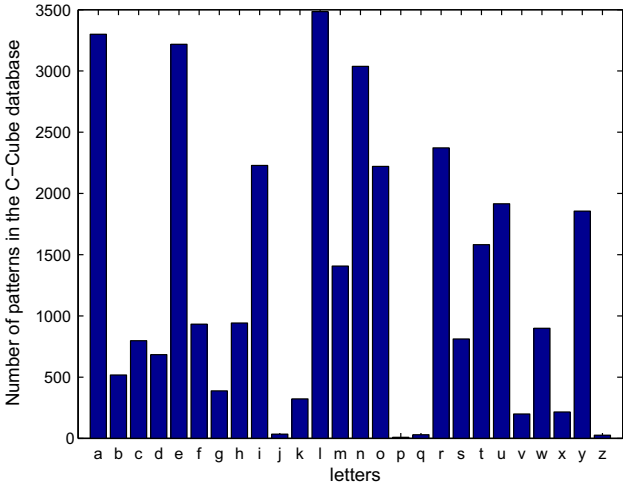

Fig. 8. Distribution of lowercase letters in the C-Cube database.

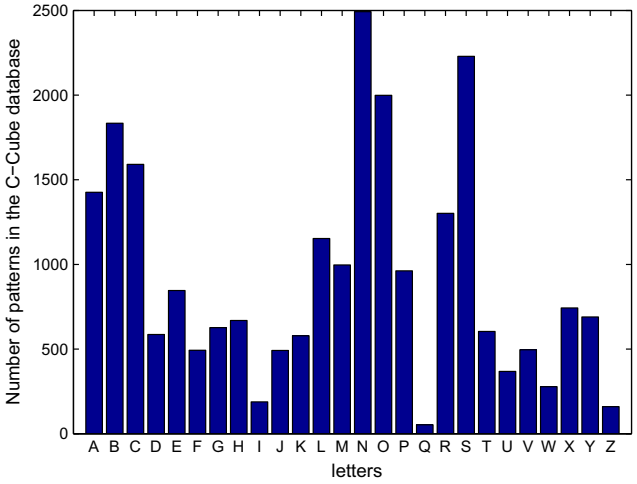

Fig. 9. Distribution of uppercase letters in the C-Cube database.

4.4. Results for the C-Cube database

For each feature set, the global information provided by the database (width/height ratio, distance between the baseline and the upper line, distance from the baseline to the upper extremity, and distance from the baseline to the lower extremity) were included in the feature vector. These features contributed to an average increase of two percentile points in the recognition rate.

Two different experiments were performed. The first was conducted to evaluate the performance of the technique for the uppercase and lowercase letters separately (split case). It is important to do this, since some applications need to recognize either uppercase or lowercase letters specifically. The second experiment was conducted using both; however, characters that present similar shapes in the two cases were combined in a single class (joint case). An analysis to verify whether or not the uppercase and lowercase forms of the same letters are similar in shape was performed in Camastra (2007). The letters *(C, X, O, W, Y, Z, M, K, J, U, N, F, V)* presented the greatest similarity between the two cases and were combined in a single class. This resulted in 39 classes in the second experiment.

The results for the split and joint cases are shown in Tables 1 and 2, respectively. The results are ordered by the recognition rates. The proposed Modified Edge Maps algorithm presented the best result overall.

Most feature sets presented better accuracy for the upper case letters. The exceptions are Image Projections, Concavity Measurement, and Camastra 34D. This fact supports the claim that it is difficult to design a feature extraction method that can deal with the variability of the patterns. In addition, the aim is to recognize both uppercase and lowercase letters, and so it is an advantage to combine techniques that are expert in each task.

4.5. Results for the MNIST database

For the Modified Edge Maps and Directional Gradient methods, the number of nodes in the hidden layer is 300. For the Zoning, Structural Characteristics, Concavity Measurement, and Image Projection techniques, the number of nodes in the hidden layer is 360, 340, 175, and 330, respectively. Table 3 shows the results for each feature set.

Table 3 shows that some feature sets have better discriminative power for certain classes of digits. A clear example of this occurs in digits with complex shapes, such as 8 and 9, where the difference between the largest and smallest values can be more than six per-

Table 1
Recognition rate by feature set for the C-Cube database. Uppercase and lowercase letters. # Nodes is the number of nodes in the hidden layer, and Mean is the average performance considering both uppercase and lowercase letters.

Method	# Nodes	Upper case (%)	Lower case (%)	Mean (%)
Modified Edge Maps	490	**86.52**	81.13	**83.55** ± 0.27
Binary Grad.	490	86.35	79.89	82.58 ± 0.18
MAT Grad.	300	85.77	79.22	81.95 ± 0.19
Median Grad.	360	85.10	79.48	81.81 ± 0.21
Camastra 34D	400	79.63	**84.37**	81.74 ± 0.35
Zoning	450	84.46	78.07	80.74 ± 0.41
Structural	320	81.94	77.70	79.53 ± 0.56
Concavities	530	73.35	81.89	76.90 ± 0.16
Projections	500	71.73	79.90	75.10 ± 0.39

Table 2
Feature set results for the C-Cube database (Joint case). # Nodes is the number of nodes in the hidden layer.

Method	# Nodes	Recognition rate (%)
Modified Edge Maps	490	**82.49 ± 0.27**
Binary Grad.	490	81.46 ± 0.18
MAT Grad.	300	80.83 ± 0.19
Median Grad.	360	79.96 ± 0.21
Camastra 34D	400	79.97 ± 0.35
Zoning	450	78.60 ± 0.41
Structural	320	77.07 ± 0.56
Concavities	530	74.90 ± 0.16
Projections	500	73.85 ± 0.39

centile points. For the digits 0, 1, 4, 6, 7, 8, and 9, the Structural Characteristics method achieved the best results, while for the digits 2, 3, and 5, the proposed Multi-Zoning technique obtained a better recognition rate.

The techniques that presented the best results for the MNIST database, Structural Characteristics and Multi-Zoning, are among the worst performers for the C-Cube database (Tables 1 and 2). The proposed Modified Edge Maps presented the best accuracy for the C-Cube database, and the second worst result for the MNIST database. This is another reason to use multiple feature extraction techniques.

4.6. Sensitivity analysis

The validation dataset was used to compute the dissimilarity matrix D and its projection onto the two-dimensional CPS $\widetilde{D}$. Fig. 10 shows the CPS for the C-Cube database. Based on visual analysis, four groups of feature representation can be observed. The Modified Edge Maps and Image Projection techniques are a long way from every other point, and can be considered an atomic cluster. The Structural Characteristics and Concavity Measurement techniques make up another group. The last cluster is composed of the gradient methods (MAT Gradient, Binary Gradient, and Median Gradient), as well as the Camastra 34D and Zoning techniques. The fact that the three gradient methods are close to one another is an interesting finding, and the reason for their proximity is that the gradient-based techniques extract similar information (directional), with a slight difference in the preprocessing of the image. The Camastra 34D method also computes directional information.

Fig. 11 presents the Oracle error analysis for the C-Cube database. We compare feature representations that are next to one another against representations that are far apart. Fig. 11(a) shows the Oracle error analysis for three techniques that are close together in the CPS and can be considered to be from the same group. In this case, the three methods that extract information based on gradients (MAT-based Gradient, Binary Gradient, and Median Gradient) are compared. The total number of errors committed using the MAT-based Gradient representation is 3668. Approximately 20% of the errors (692 images) are misclassified by this feature representation alone. The intersection between the MAT-based Gradient and the Binary Gradient methods shows that 2113 images are misclassified, while 1849 images are misclassified when MAT-based and Median Gradient representations are used. In addition, 986 images are misclassified based on the intersection of the three techniques. Therefore, as the majority of errors of these three techniques occur in the same images, combining them is unlikely to result in improved performance.

In contrast, Fig. 11(b) shows the Oracle error analysis for the MAT-based Gradient with two representations that are far apart in the CPS: Projections, and Edge-Maps. In this case, we can easily see that the majority of errors committed by the MAT-based Gradient, 2058 happens only individually. Both the pair-wise intersections and the intersection of the three techniques produce a much lower number of errors, and only 252 images are misclassified considering these three feature representations. This number is approximately 10 times less than the number of images that are misclassified when only the MAT-based Gradient is considered (2058). So, the errors made by the three techniques occurred in distinct patterns, and therefore can be considered complementary, since they are able to correctly classify different images.

Fig. 12 shows the CPS plot for the MNIST database. We can identify three feature representations that are far away from all the others: Concavities, Zoning, and Structural Characteristics. As with the C-Cube experiment, the results of the gradient-based methods and the Camastra 34D representation are close together, forming a group of similar feature representations. The Modified Edge Maps representation results lie between the Zoning and Projections representation results, and these methods can also be considered to belong to a distinct group.

Fig. 13(a) shows the Oracle error analysis among three methods: Structural Characteristics, Multi-Zoning, and Concavity Measurement. Only nine images were misclassified by the three methods used simultaneously. Moreover, the pairwise intersection of the three techniques also reduces the number of errors. As a result, these three techniques together are able to correctly classify different images. Fig. 13(b) shows the intersection of three techniques that are closer together on the CPS plot (Structural Characteristics, Edge Maps, and MAT-based Gradient). In this case, the intersection of errors shows that it is possible to reduce the individual errors, since they present complementary information.

So, based on the proposed framework, we can answer two of the questions posed in this paper: Do different feature extraction techniques present complementary information? We demonstrate that different feature extraction techniques are indeed complementary. The majority of the techniques are far apart in the CPS for both datasets. Furthermore, combining them using the Oracle analysis can reduce the individual error by a factor of as high as 10 for the C-Cube dataset (Fig. 11(b)), and can result in a very low error rate for the MNIST dataset (Fig. 11(a)).

Table 3
Results for each feature extraction method for the MNIST database.

Digit	Structural	Edge Maps	Projections	Multi-Zoning	Concavity	MAT Grad.	Binary Grad.	Median Grad.	Camastra 34D
0	**98.88**	97.86	98.17	**98.88**	96.13	97.96	98.46	98.06	98.46
1	**99.12**	98.15	98.42	98.95	98.33	98.68	99.03	99.11	99.03
2	96.03	95.26	95.26	**96.23**	95.66	95.16	96.22	96.31	96.22
3	96.14	94.76	94.76	**96.84**	91.69	94.46	96.23	96.23	96.23
4	**97.25**	92.15	96.33	97.05	92.98	96.94	98.16	97.45	98.16
5	95.63	94.73	93.61	**96.96**	95.56	96.30	95.62	95.96	95.62
6	**97.81**	96.66	97.18	97.08	96.35	97.39	96.45	96.76	96.45
7	**96.89**	93.77	95.43	95.62	94.38	95.04	95.81	94.94	95.81
8	**96.00**	93.54	93.74	95.90	89.64	95.54	93.83	93.42	93.83
9	**95.60**	90.58	93.85	95.16	92.11	92.66	94.44	95.14	94.44
Mean	**96.95 ± 0.29**	94.78 ± 0.15	95.72 ± 0.13	96.84 ± 0.18	94.31 ± 0.25	95.83 ± 0.13	96.47 ± 0.12	96.38 ± 0.12	96.47 ± 0.32

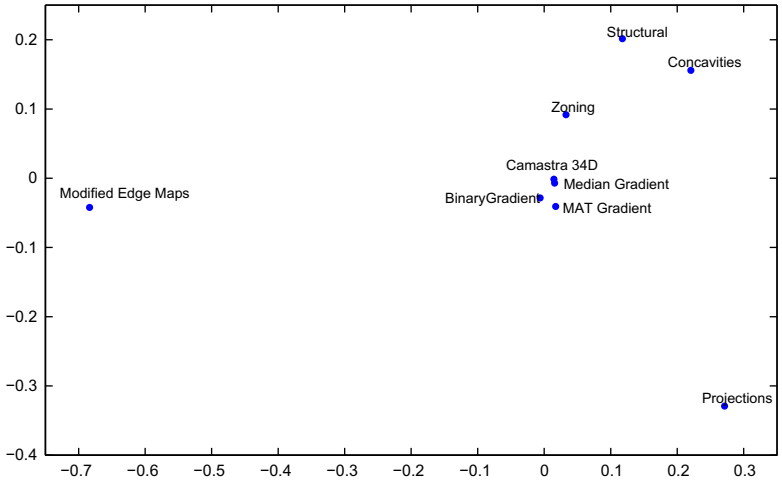

Fig. 10. Classifier Projection Space for the C-Cube dataset. The axes of the CPS plot have no significance, and only the distances between the points are important.

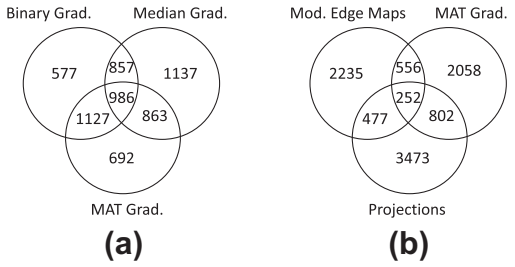

Fig. 11. Oracle error analysis for the C-Cube dataset. (a) Comparison of feature representations that belong to the same cluster. (b) Comparison of feature representations that are far apart.

The exceptions are the representations that extract the gradients of the images. Therefore, in answer to the second question: Are feature extraction techniques that use a similar approach (e.g. different methods to extract gradients) less complementary than techniques that use different characteristics (e.g. edges, concavities)? According to Figs. 10 and 12, the gradient-based methods based (MAT-based Gradient, Binary Gradient, and Median Gradient) are really close to each other, creating a cloud of points. In addition, the results of the Oracle analysis (Fig. 11(b)) demonstrate that the number of errors that are common to the three techniques is higher than the number of individual errors. From this, we can conclude that techniques using similar approaches are less complementary, and are likely to misclassify the same images.

5. Multiple-classifier systems

MCS have been widely studied as an alternative means of increasing efficiency and accuracy in pattern recognition systems (Kittler, Hatef, Duin, & Matas, 1998; Kuncheva, 2002a, 2004). The main motivation for using classifier ensembles comes from the observation that errors committed by classifiers trained with different feature extraction methods do not overlap. Another reason to use them is based on the divide-and-conquer paradigm: instead of using a single set consisting of all feature sets, the idea is to use each feature extraction method separately and combine their results. There are many examples in the literature that show the efficiency of an ensemble of classifiers in various tasks, such as signature verification (Batista, Granger, & Sabourin, 2010), pedestrian detection (Xu, Cao, & Qiao, 2011), and image labeling (Singh & Singh, 2005).

The advantage of combining classifiers that deal with distinct feature sets is that they represent different transformations of the image into the feature space. Suppose, for example, that a pattern is located near the decision boundary. The recognition of this pattern is a difficult task in the feature space used. It is still difficult when multiple classifiers are applied over the same feature space. However, if different feature spaces are used, this pattern might be close to the decision boundary in one feature space, but the same pattern might be far from the decision boundary of another feature space, as its transformation is completely different. In this way, the pattern can be easily recognized.

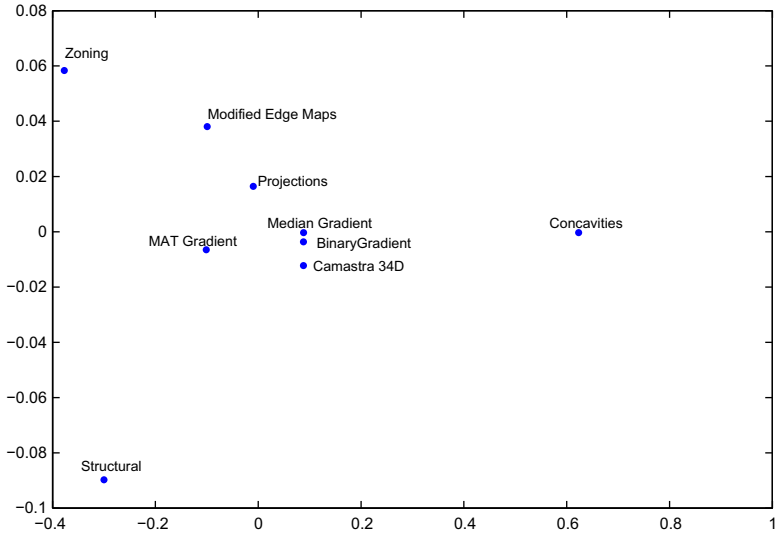

Fig. 12. The Classifier Projection Space of the MNIST dataset.

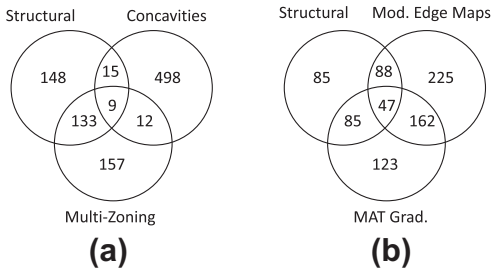

Fig. 13. Oracle error analysis for the MNIST dataset. (a) comparison of feature representations that are far apart. (b) comparison of feature representations that form a cloud of points.

5.1. Trained combiner

Duin (2002) concluded that fixed combination rules only achieve the best results in very strict conditions. Normally, these results are suboptimal, and the performance of these rules falls far short of the performance of the Oracle. For instance, the Product rule is known to fail if one of the classifiers' estimates is close to zero, or is accidentally zero. So, if one feature set is not suitable for the query image, the system is likely to fail. The majority vote rule only produces the correct classification if at least half the classifiers predict the correct class. However, there are certain images that are correctly classified in only one or two of the nine feature sets, and so we cannot achieve a performance close to the Oracle using this combination rule either.

Consequently, we decided to use a trained combiner in order to achieve a more robust combination of classifiers. Trained combiners usually perform better, since the combiner can adapt to the classification problem (Duin, 2002). In this methodology, the outputs of the base classifiers are used as input features for a new classifier that is trained to aggregate the results. During the training phase, the combiner learns how to deal with difficult situations, such as, for example, when a small subset of the base classifiers produces the correct answer.

In the experimental study, the trained combiner is an MLP network with one hidden layer. Neural networks are good candidates for use as trained combiners, because they are robust to noise. This means that the MLP combiner can still predict the correct output, even when the majority of the base classifiers present errors.

5.2. Experimental protocol

In this section, the results obtained by combining the feature extraction techniques are presented. For the combination module, the MLP combiner is compared to well-known fixed combination rules. The fixed rules considered are Sum, Product, Maximum, Median, Voting, and Oracle. The theoretical framework for the fixed combination rules is described in Kittler et al. (1998) and Kuncheva (2002b).

The experiment was conducted using 10 iterations, in order to obtain the mean and standard deviation for the results. For each iteration, the base classifiers were retrained following the protocol described in Section 4.3. This replication is important, since the results are sensitive to the initial weight configuration of the base classifiers.

For each image in the training set, the *a posteriori probability* for each feature set is estimated and used as an input feature to train the MLP combiner. Two experiments were conducted using this combiner: MLP_{all}, which consists of the nine feature representations, and $MLP_{selection}$, which consists of a subset using feature representations selected based on the sensitivity analysis.

For the $MLP_{selection}$ configuration, the MAT-based Gradient, Binary Gradient, Median Gradient, and Camastra 34D techniques are considered redundant for both datasets (Section 4.6). As we use only the Binary Gradient to represent this group of techniques, because it achieved the highest accuracy, the configuration $MLP_{selection}$ consists of only 6 feature representations: Modified Edge Maps, Concavity Measurement, Multi-Zoning, Structural Characteristics, Binary Gradient, and Image Projections.

In every experiment, combiner training is accomplished using the *Resilient Backpropagation* algorithm. The number of nodes in the hidden layer of the MLP combiner was selected using the *cross-validation* method with the training data. The search was conducted by varying the number of nodes from 10 to 300 at 10-point intervals. The number of nodes in the hidden layer of the MLP combiner for the C-Cube and MNIST datasets were 300 and 50, respectively.

5.3. Results for the C-Cube dataset

Tables 4 and 5 show the results of the combination for the C-Cube database. A Kruskal–Wallis non parametric statistical test (95% confidence level) applied to the difference in accuracy rates showed that the results with the combination rules are statistically

Table 4
Results of each combination method for the C-Cube database. Uppercase and lowercase letters.

Method	Upper case (%)	Lower case (%)	Mean (%)
Sum	91.21	86.94	88.92
Product	85.92	79.52	82.37
Maximum	89.83	85.22	87.14
Median	91.00	87.33	88.86
Maj. Vote	90.99	87.44	88.92
MLP_{all}	**91.39**	**88.45**	**89.67**
$MLP_{selection}$	**90.89**	**88.25**	**88.85**
Oracle	96.87	97.24	97.09

Table 5
Results of each combination rule for the C-Cube database (Joint case).

Method	Best (%)	Mean (%)
Sum	88.51	88.22 ± 0.19
Product	86.99	85.52 ± 0.89
Maximum	85.48	85.73 ± 0.67
Median	88.84	88.04 ± 0.53
Maj. Vote	89.22	88.00 ± 0.81
MLP_{all}	**89.65**	**89.28 ± 0.22**
$MLP_{selection}$	**89.54**	**88.98 ± 0.50**
Oracle	97.78	97.25 ± 1.72

significant when compared to the classifiers trained using a single feature extraction technique. This can be explained by the fact that the feature extraction techniques considered in this analysis present complementary information; the majority of them are far apart in the CPS (Fig. 10). This means that the recognition performance could be improved any combination rule.

The only exception was the Product rule. Its results for the separate case were not statistically better than those of the Modified Edge Maps technique. This might be explained by the fact that there was a large difference in the accuracy of the feature representations.

Fig. 14 shows the box plot with the results for the combination rules for the C-Cube database. The gain in recognition performance for the MLP combiner is statistically significant when compared with that of the fixed combination rules. The MLP_{all} combiner presented the best mean result. However, based on the Kruskal–Wallis test, the results were not statistically better than those of the reduced combination, $MLP_{selection}$.

5.4. Results for the MNIST database

Table 6 shows the results obtained by the combination methods for the MNIST database. The recognition performance of all the combination rules was a great improvement over all the (feature extraction, classifier) pairs shown in Table 3. The Kruskal–Wallis non parametric statistical test with a 95% confidence level was also used, and the result obtained by every combination rule was statistically better. Once again, the gain in performance is explained by the fact that the majority of feature representations considered presents complementary information.

As with the C-Cube dataset, the trained combiner outperformed the other combination rules. The $MLP_{selection}$ combination achieved an accuracy rate close to the Oracle performance (which was 100%). This is due to the ability of the network to learn how to perform the best combination using the training set. In addition, the standard deviation of the trained combiner is 0.04%, which is approximately six times less than the standard deviation for the Maximum rule. Even when one or more feature sets produce a very inaccurate result, the trained combiner is still able to predict the correct output. Fig. 15 shows the box plot for the combination rules. The median result of both MLP_{all} and $MLP_{selection}$ achieved a lower error rate than the best results of the other combination rules.

Furthermore, the result of the MLP combiner is followed by the Maximum rule that also presented a high recognition rate. This is because of the ability that some of the feature extraction methods have to recognize certain types of digits.

In both experiments, the results using all the feature representations (MLP_{all}) and the configuration following the sensitivity analysis ($MLP_{selection}$) are statistically equivalent. Nevertheless, for the C-Cube dataset, the $MLP_{selection}$ achieved a Result 0.04 percentile points higher than that of MLP_{all}. This is an interesting finding, since $MLP_{selection}$ is composed of a small number of feature representations. The redundant nature of MLP_{all} might interfere with

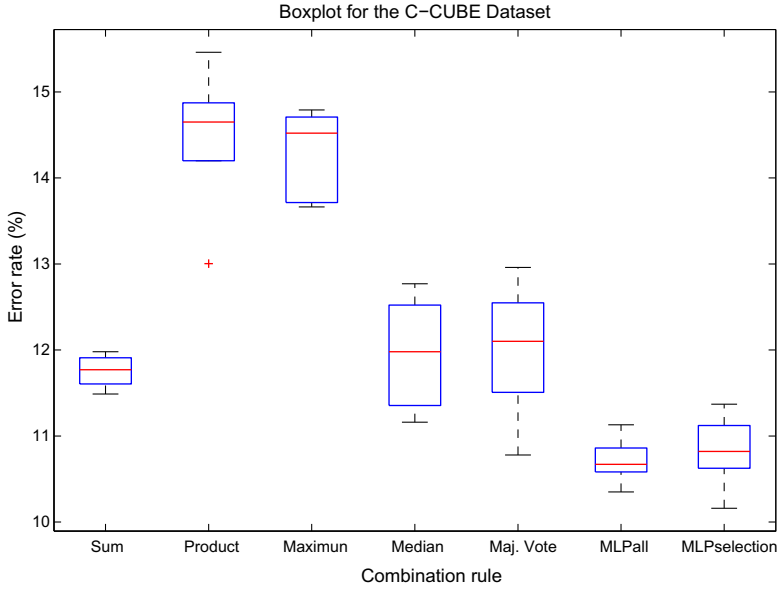

Fig. 14. Boxplot diagram comparing the combination rules for the CCUBE database. MLP_{all} and $MLP_{selection}$ are the MLP combiner for the experiment using every feature representation and the reduced feature representation set respectively.

Table 6
Results of each combination rule for the MNIST database.

Method	Best (%)	Mean ± std dev (%)
Sum	99.23	98.96 ± 0.42
Product	99.55	99.27 ± 0.34
Maximum	99.58	99.43 ± 0.23
Median	99.12	98.85 ± 0.25
Maj. Vote	98.98	98.63 ± 0.49
MLP_{all}	**99.72**	**99.70 ± 0.01**
$MLP_{selection}$	**99.76**	**99.72 ± 0.04**
Oracle	100	100 ± 0.00

Table 7
Comparative results for the C-Cube database. RBF = Radial Basis Network with 5120 centers, HVQ = Hierarchical Vector Quantization, MDF = Modified Directional Features, SVM = SVM with Radial basis Kernel.

Algorithm	Recognition rate (%)
HVQ-32 (Thornton et al., 2008)	84.72
HVQ-16 (Thornton et al., 2008)	85.58
MDF-RBF (Thornton et al., 2009)	80.92
34D-RBF (Thornton et al., 2009)	84.27
MDF-SVM (Thornton et al., 2009)	83.60
34D-SVM + Neural GAS (Camastra, 2007)	86.20
34D-MLP (Camastra, 2007)	71.42
Proposed	**89.28 ± 0.22**

the performance of the combination. This means that we can answer the third question posed in the introduction, as follows: The proposed framework selects feature representations that can be used to construct an efficient MCS in terms of accuracy rates.

5.5. Computational time

Analyzing the proposed system when the trained combiner is used, the average computational time per image is 4 ms for the MNIST and 9 ms for the C-Cube dataset. The application was developed using C++ running on a 2.40 Ghz machine with four cores.

We measured the difference in computational cost of the MLP combiner and the fixed combination rule. That difference is measured in microseconds, and does not affect the overall computational time of the system. This was expected, since the MLP combiner has a total of 30,000 connections (60 inputs $\times$ 50 hidden nodes $\times$ 10 output nodes), while the network trained with the Structural Characteristics feature set has 952,000 connections (280 inputs $\times$ 340 hidden nodes $\times$ 10 output nodes). In other words, the cost of computing the combination is approximately 31 times less than the cost of computing a single feature set.

5.6. Comparison with the state of the art

The best results obtained for the C-Cube database are shown in Table 7. To the best of our knowledge, the proposed combination scheme outperforms all the previous results in the *Split B* of this database. Furthermore, it is important to observe that the past best results are based on Support Vector Machines (SVM) using the one-versus-the-rest approach (Scholkopf & Smola, 2001). This method trains one specific classifier for each class. For this problem, a large number of classifiers is required, which is one of the drawbacks of these approaches. As far as we know, the proposed system is the first to show high accuracy using only MLPs.

The best results obtained for the MNIST database are shown in Table 8. The proposed combination scheme outperformed all the previous results for this database. It is also important to observe that many of the best results (Ciresan, Meier, Gambardella, & Schmidhuber, 2010; Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009; Lauer, Suen, & Bloch, 2007; LeCun, Bottou, Bengio, & Haffner, 1998; Simard, Steinkraus, & Platt, 2003; Ranzato, Boureau, & LeCun, 2008) are based on large neural networks, such as Convolutional Neural Networks or Deep Neural Networks. In addition, the techniques used previously need to expand the training data by creating new images through distortions (Ciresan et al., 2010; Kussul, Baidyk, Wunsch, Makeyev, & Martin, 2006; Lauer et al., 2007; LeCun et al., 1998; Simard et al., 2003; Ranzato et al., 2008). Our approach to achieving high performance in handwritten recognition is different, in that no additional training data is required.

6. Conclusion

We have proposed a new framework for analyzing the relationship between different feature representations. Each representation is used to train a single classifier, and the dissimilarities

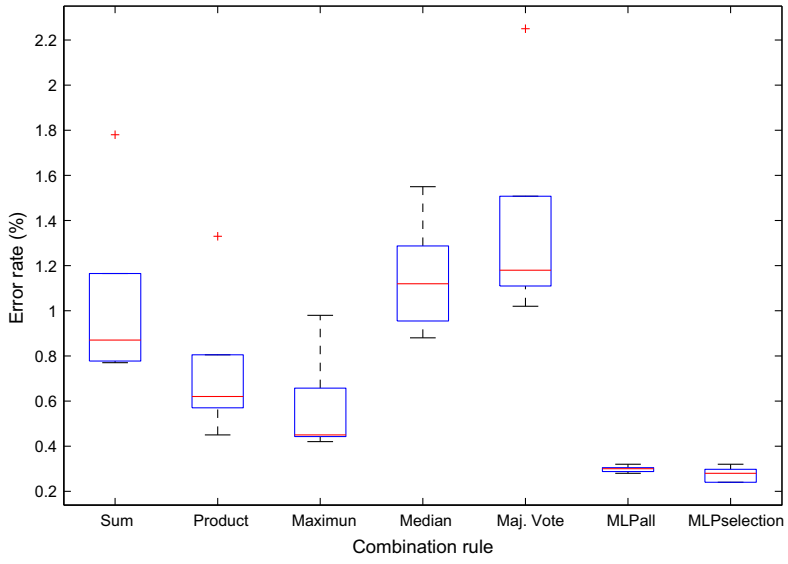

Fig. 15. Boxplot diagram comparing the combination rules for the MNIST database. MLP_{all} and $MLP_{selection}$ are the MLP combiner for the experiment using every feature representation and the reduced feature representation set respectively.

Table 8
Comparative Results for the MNIST Database.

Method	Distortions	Recognition rate (%)
Boosted LeNet-4 (LeCun et al., 1998)	Affine	99.30
unsupervised sparse features + SVM (Labusch et al., 2008)	–	99.41
Trainable feature extractor + SVM (Lauer et al., 2007)	Affine	99.46
Large convolutional network + unsupervised pretraining (Jarrett et al., 2009)	–	99.47
PNCN classifier (Kussul et al., 2006)	Skewing	99.56
Cascade ensemble classifier (without rejection) (Zhang, 2006)	–	99.59
Convolutional neural networks (Simard et al., 2003)	Elastic	99.60
Large convolutional network + unsupervised pretraining (Ranzato et al., 2008)	Elastic	99.61
6 Layers MLP 841-2500-2000-1500-1000-500-10 (Ciresan et al., 2010)	Elastic	99.65
Proposed	–	**99.72 ± 0.04**

between them are computed to generate a dissimilarity matrix. Through the Multidimensional Scaling method (Sammon Mapping), this dissimilarity matrix is embedded in a two-dimensional space (CPS) where the Euclidean distance between two feature representations reflects their dissimilarity. Based on this two-dimensional plot, a sensitivity analysis is performed in order to determine whether the representations are complementary or redundant.

We have applied the proposed framework to two handwritten recognition datasets: the Cursive Character Challenge (C-Cube) for handwritten letters, and the MNIST dataset for handwritten digits. The results demonstrate that feature representations using distinct approaches (edges, projections, gradient, and concavities) extract information that is dissimilar. Consequently, they are complementary. Techniques that use the same observations, using a different rule to compute the features (e.g. the MAT-based Gradient, Median Gradient, and Binary Gradient) perform in a similar fashion. They appear close to each other in both experiments and are likely to commit errors on the same images. As a result, they can be considered redundant.

A multiple-classifier system using distinct feature extraction techniques was designed based on the feature representation analysis. As the majority of techniques considered present complementary information, the results of every combination rule outperform the best individual classifier for both datasets. With the aim of searching for the optimal combination rule, we used a neural network as a combiner. The results show that the proposed approach presents better accuracy when compared with state-of-the-art techniques.

The two experiments that were performed: one using all the feature representations, and the other a reduced set of representations based on a sensitivity analysis, demonstrate that the strategies are statistically equivalent. In some cases, the reduced set of representations can even achieve higher performance, as redundant classifiers can negatively affect performance. This shows that our framework can also be used to perform feature representation selection. In this paper, however, we use the empirical analysis of the CPS and the Oracle error analysis manually, in order to make this selection. An algorithm designed to perform the selection automatically using our framework is currently being developed.

Acknowledgments

This work was supported in part by the Brazilian governmental agencies CNPq and FACEPE.

References

Batista, L., Granger, E., & Sabourin, R. (2010). Improving performance of HMM-based off-line signature verification systems through a multi-hypothesis approach. *International Journal of Document Analysis and Recognition, 13*, 33–47.

Camastra, F. (2007). A SVM-based cursive character recognizer. *Pattern Recognition, 40*, 3721–3727.

Camastra, F., Spinetti, M., & Vinciarelli, A. (2006). Offline cursive character challenge: A new benchmark for machine learning and pattern recognition algorithms. *International Conference on Pattern Recognition*, 913–916.

Chim, Y., Kassim, A. A., & Ibrahim, Y. (1998). Dual classifier system for handprinted alphanumeric character recognition. *Pattern Analysis and Application, 4*, 155–162.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep big simple neural nets excel on handwritten digit recognition. *Computer Research Repository, abs/1003.0358*.

Cox, T. F., & Cox, M. A. A. (2000). *Multidimensional scaling* (2nd ed.). Chapman and Hall.

Cruz, R. M. O., Cavalcanti, G. D. C., & Tsang, I. R. (2010a). An ensemble classifier for offline cursive character recognition using multiple feature extraction techniques. In *International joint conference on neural networks* (pp. 744–751).

Cruz, R. M. O., Cavalcanti, G. D. C., & Tsang, I. R. (2010b). Handwritten digit recognition using state-of-the-art techniques. In *International conference on systems, signals and image processing* (pp. 215–218).

Devijver, P., & Kittler, J. (1982). *Pattern recognition: A statistical approach*. Londres: Prentice-Hall.

Duin, R. P. W. (2002). The combining classifier: To train or not to train? In *International conference on pattern recognition* (Vol. 2, pp. 765–770).

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing, 19*, 699–707.

Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing* (3rd ed.). Upper Saddle River, NJ, USA: Prentice-Hall.

Impedovo, S., Lucchese, M. G., & Pirlo, G. (2006). Optimal zoning design by genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 36*, 833–846.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *International conference on computer vision* (pp. 2146–2153).

Kavallieratou, E., Sgarbas, K., Fakotakis, N., & Kokkinakis, G. (2003). Handwritten word recognition based on structural characteristics and lexical support. In *International conference on document analysis and recognition* (pp. 562–567).

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*, 226–239.

Kuncheva, L. I. (2002a). Switching between selection and fusion in combining classifiers: An experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 32*, 146–156.

Kuncheva, L. I. (2002b). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 281–286.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley-Interscience.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning, 51*, 181–207.

Kussul, E., Baidyk, T., Wunsch, D., Makeyev, O., & Martin, A. (2006). Permutation coding technique for image recognition systems. *IEEE Transactions on Neural Networks, 17*, 1566–1579.

Labusch, K., Barth, E., & Martinetz, T. (2008). Simple method for high-performance digit recognition based on sparse coding. *IEEE Transactions on Neural Networks, 19*, 1985–1989.

Lauer, F., Suen, C. Y., & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. *Pattern Recognition, 40*, 1816–1824.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*, 2278–2324.

Oliveira, L. S., Sabourin, R., Bortolozzi, F., & Suen, C. Y. (2002). Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 1438–1454.

Pekalska, E., & Duin, R. P. W. (2002). Spatial representation of dissimilarity data via lower-complexity linear and nonlinear mappings. In *Proceedings of the joint IAPR international workshop on structural, syntactic, and statistical pattern recognition* (pp. 488–497). Springer-Verlag.

Pekalska, E., Duin, R. P. W., & Skurichina, M. (2002). A discussion on the classifier projection space for classifier combining. In *International workshop on multiple classifier systems* (pp. 137–148). London, UK.

Ping, Z., & Lihui, C. (2002). A novel feature extraction method and hybrid tree classification for handwritten numeral recognition. *Pattern Recognition Letters, 23*, 45–56.

Ranzato, M., Boureau, Y.-L., & LeCun, Y. (2008). Sparse feature learning for deep belief networks. *Advances in Neural Information Processing Systems*, 1185–1192.

Richard, M. D., & Lippmann, R. P. (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation, 3*, 461–483.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE international conference on neural networks* (pp. 586–591).

Scholkopf, B., & Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press.

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

Shipp, C. A., & Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion, 3*, 135–148.

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *International conference on document analysis and recognition* (Vol. 2, pp. 958–963).

Singh, S., & Singh, M. (2005). A dynamic classifier selection and combination approach to image region labelling. *Signal Processing: Image Communication, 20*, 219–231.

Srihari, S. N., Tomai, C. I., Zhang, B., & Lee, S. (2003). Individuality of numerals. In *International conference on document analysis and recognition* (pp. 1086–1091).

Thornton, J., Faichney, J., Blumenstein, M., & Hine, T. (2008). Character recognition using hierarchical vector quantization and temporal pooling. In *Australasian joint conference on artificial intelligence* (pp. 562–572).

Thornton, J., Blumenstein, M., Nguyen, V., & Hine, T. (2009). Offline cursive character recognition: A state-of-the-art comparison. In *14th Conference of the international graphonomics society* (pp. 148–152).

Trier, O. D., Jain, A. K., & Taxt, T. (1995). Feature extraction methods for character recognition: A survey. *Pattern Recognition, 29*, 641–662.

Xu, Y., Cao, X., & Qiao, H. (2011). An efficient tree classifier ensemble-based approach for pedestrian detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 41*, 107–117.

Zhang, P. (2006). Reliable recognition of handwritten digits using a cascade ensemble classifier system and hybrid features. Ph.D. thesis, Concordia University Montreal, P.Q., Canada.

Zhang, P., Bui, T. D., & Suen, C. Y. (2007). A novel cascade ensemble classifier system with a high recognition performance on handwritten digits. *Pattern Recognition, 40*, 3415–3429.

Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM, 27*, 236–239.