Pattern Recognition 43 (2010) 2807–2816

Contents lists available at ScienceDirect

Pattern Recognition

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

Handwritten character recognition through two-stage foreground sub-sampling

Georgios Vamvakas *, Basilis Gatos, Stavros J. Perantonis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", 153 10 Athens, Greece

ARTICLE INFO

Article history:
Received 10 February 2009
Received in revised form
11 February 2010
Accepted 23 February 2010

Keywords:
Handwritten character/digit recognition
Feature extraction
Two-stage classification

ABSTRACT

In this paper, we present a methodology for off-line handwritten character recognition. The proposed methodology relies on a new feature extraction technique based on recursive subdivisions of the character image so that the resulting sub-images at each iteration have balanced (approximately equal) numbers of foreground pixels, as far as this is possible. Feature extraction is followed by a two-stage classification scheme based on the level of granularity of the feature extraction method. Classes with high values in the confusion matrix are merged at a certain level and for each group of merged classes, granularity features from the level that best distinguishes them are employed. Two handwritten character databases (CEDAR and CIL) as well as two handwritten digit databases (MNIST and CEDAR) were used in order to demonstrate the effectiveness of the proposed technique. The recognition result achieved, in comparison to the ones reported in the literature, is the highest for the well-known CEDAR Character Database (94.73%) and among the best for the MNIST Database (99.03%)

1. Introduction

Optical Character Recognition (OCR) is a field of research in pattern recognition, artificial intelligence and machine vision. It refers to the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text. Nowadays, the accurate recognition of machine printed characters is considered largely a solved problem. However, handwritten character recognition is comparatively difficult, as different people have different handwriting styles. So, handwritten OCR is still a subject of active research.

A widely used approach in OCR systems is to follow a two step schema: (a) represent the character as a vector of features and (b) classify the feature vector into classes [1]. Selection of a feature extraction method is important in achieving high recognition performance. A feature extraction algorithm must be robust enough so that for a variety of instances of the same symbol, similar feature sets are generated, thereby making the subsequent classification task less difficult [2]. On the other hand, Vapnik et al. [3] have suggested that powerful classification algorithms suffice even when given features are just sufficiently discriminative. The choice of classifier, however, is not an easy task since the classifier depends on many factors such as available training set, number of free parameters, etc. Classification methods based on learning from examples have been applied to character recognition mainly since the 1990s. These methods include statistical methods based on Bayes decision rule, artificial neural networks (ANNs), Kernel methods including Support Vector Machine (SVM) and multiple classifier combination [4–7]. So, taking into account all the above, we can state that feature extraction techniques, classification methods and architectures interact in complex ways.

Feature extraction methods for handwritten characters and digits have been based mainly on two types of features: (a) statistical derived from statistical distribution of points and (b) structural. The most common statistical features used for character representation are: (a) zoning, where the character is divided into several zones and features are extracted from the densities in each zone [8] or from measuring the direction of the contour of the character by computing histograms of chain codes in each zone [9], (b) projections [10] and (c) crossings, that consist of the number of transitions from foreground to background pixels along horizontal and vertical lines and distances, that rely on the calculation of the distance of the first foreground pixel detected from the upper/lower (left/right) boundaries of the image along vertical (horizontal) lines [11]. Structural features are based on topological and geometrical properties of the character while encoding some knowledge of the structure of the character or of what sort of components is made up, such as maxima and minima, reference lines, ascenders, descenders, cusps above and below a threshold, strokes and their direction between two points, horizontal curves at top or bottom, cross points, end points, branch points, etc. [12]. Many feature extraction techniques along

* Corresponding author. Tel.: +30 210 6503218; fax: +30 210 6532175.
E-mail addresses: gbam@iit.demokritos.gr (G. Vamvakas), bgat@iit.demokritos.gr (B. Gatos), sper@iit.demokritos.gr (S.J. Perantonis).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.02.018

the above lines of research have been described in the literature. For example, in Blumenstein et al. [13], a feature extraction technique that extracts direction information from the structure of the character contours and uses two neural networks based classifiers is investigated, while Camastra and Vinciarelli [14], present an OCR methodology that relies on local features derived from zoning and global ones such as the character's aspect ration followed by a recognition procedure that combines neural gas, an unsupervised version of vector quantization where no topology of a fixed dimensionality is imposed on the network, and learning vector quantization. Singh and Hewitt [15] propose a modified Hough Transform method. Character images are divided into uniform regions that are searched for vertical, horizontal and diagonal segments. The total number of such segments is fed to the classifier. Kimura et al. [16] present a feature extraction technique calculating histograms based on chain code information followed by neural and statistical classifiers. Gader et al. [17] suggest a feature extraction scheme based on the calculation of transitions from foreground to background pixels in both vertical and horizontal directions using neural networks with back-propagation for the recognition procedure. A survey on feature extraction methods can be found in [18].

There have been quite a number of successes in determination of invariant features in handwriting and a wide range of classification methods have been extensively researched. However, as mentioned in [19], most character recognition techniques use a 'one model fits all' approach, i.e. a set of features and a classification method are developed and every test pattern is subjected to the same process regardless of the constraints present in the problem domain. It is shown that approaches which employ a hierarchical treatment of patterns can have considerable advantages compared to the 'one model fits all' approaches, not only improving the recognition accuracy but also reducing the computational cost as well. In Park et al. [19], a dynamic character recognizer is presented. The recognizer begins with features extracted in a coarse resolution and focuses on smaller sub-images of the character on each recursive pass, thus working with a finer resolution of a sub-image each time, till classification meets acceptance criteria. By employing an approach called *gaze planning*, a means of expanding only some of the nodes in a tree structure similar to quad trees [20], not all of the sub-images are subjected to further subdivision but only those where it is believed that features of interest are present. So, a feature vector is extracted for each character that has more information from those sub-images that are deemed to be more important than others. The feature vector is generated by combing all features extracted in each sub-image. These features are based on histogram of gradient and moment-based projections. In [21] the character image is subdivided recursively into smaller sub-images based on the quad tree rule. The input image is then represented by fractal codes obtained at each iteration by encoding algorithm. In [22] a feature extraction technique relied on recursive subdivisions of the image for the recognition of mathematical glyphs is introduced. Each split is based on the centre of gravity of the corresponding sub-image. The initial splitting is vertical and each level of splitting then alternates between horizontal and vertical. For each rectangular region a four dimensional feature vector is extracted consisting of the vertical or horizontal component of the centroid and the three second order central moments.

Moreover, other approaches focus on measuring the similarity/dissimilarity between shapes by mapping one character onto another [23,24]. In Belongie et al. [23] the *shape context* is presented. Each shape is represented by a set of points extracted from the contour. For each shape, a descriptor is introduced, the shape context, which is the log-polar histogram of the point. Corresponding points on two similar shapes are supposed to have the same shape context thus resulting in a bipartite graph matching problem. In [24] two characters are matched by deforming the contour of one to fit the edge strengths of the other, and a dissimilarity measure is derived from the amount of deformation needed, the goodness of fit of the edges and the interior overlap between the deformed shapes.

Most classification strategies in OCR deal with a large number of classes trying to find the best discrimination among them. However, such approaches are vulnerable to classification errors when classes of similar shapes are present since they are not easily distinguished. In [25] a two-stage classification approach is presented to detect and solve possible conflicts between characters such as 'A' and 'H' or 'U' and 'V'. During the first stage, a single classifier or ensemble of classifiers detect potential conflicts. The second processing stage becomes active only when a decision on the difficult cases must be taken. A comparative study between three different two-stage hierarchical learning architectures can be found in [26].

In our work, the idea of recursive subdivisions of the character image as in [19,22] is used as a starting point. We focus on a novel feature extraction method based on different levels of granularity. At each level, features are extracted based on the point, at the intersection of the horizontal and vertical lines, which divides the character image into four sub-images that approximately consist of the same amount of foreground pixels. Even though the feature extraction method itself is quite efficient when a specific level of granularity is used, there is more to be gained in classification accuracy by exploiting the intrinsically recursive nature of the method. This is achieved by appropriately combining the results from different granularity levels using a two-stage hierarchical approach. Initially, the level at which the highest recognition rate is achieved is used to perform a preliminary discrimination, whereas the procedure is iterated once more in order to find the level at which patterns of similar shapes, confused at the first step of the classification procedure, are best distinguished. The remainder of this paper is organized as follows. In Sections 2 and 3 the proposed OCR methodology is presented while experimental results are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. The proposed OCR methodology

The proposed OCR methodology follows a two step schema: first a feature extraction method is applied to obtain feature vectors and then a two-stage classification scheme is performed.

2.1. Preprocessing

Before employing the proposed feature extraction technique all character images must be black and white (b/w) and normalized to an $N \times N$ matrix. In case of character/digit images that are already b/w just the size normalization step is performed under the condition that the aspect ratio is preserved. On the other hand, the character/digit images that are gray scale have to pass through a binarization step beforehand. The well-known Niblack's approach [27] was used for this step.

2.2. Feature extraction

In this section a new feature extraction method for handwritten character recognition is presented. This method is based on structural features extracted directly from the character image that provide a good representation of the character at different levels of granularity.

Let *im*(x,y) be the character image array having 1s for foreground and 0s for background pixels and x_{max} and y_{max} be the width and the height of the character image. Our feature extraction method relies on iterative subdivisions of the character image, so that the resulting sub-images at each iteration have balanced (approximately equal) numbers of foreground pixels, as far as this is possible. At the first iteration step (zero level of granularity, that is $L=0$) the character image is subdivided into four rectangular sub-images using a vertical and a horizontal divider line as follows: firstly, a vertical line is drawn that minimizes the absolute difference of the number of foreground pixels in the two sub-images to its left and to its right. Subsequently, a horizontal line is drawn that minimizes the absolute difference of the number of the foreground pixels in the two sub-images above and below. An important point is that the above dividing lines are determined taking into account that each split results to either two disjoint sub-images or two sub-images that share equally the foreground pixels on the division line as explained below in more formal detail. The pixel at the intersection of the two lines is referred to as the *division point* (DP). At further iteration steps (levels of granularity $L=1$, 2, 3…), each sub-image obtained at the previous step is further divided into four sub-images using the same procedure as above.

More formally, the co-ordinates (x_0, y_0) of the DP of the initial character image are calculated as follows: Let V_0 [x_{max}] be the vertical projection array of the initial image (Fig. 1). Create V_1 [$2*x_{max}$] array by inserting a '0' before each element of V_0 (Fig. 1c). Then, the element x_q in V_1, that minimizes the difference between the sum of the left partition [1, x_q) and the right partition (x_q, $2*x_{\max}$], is found. Finally, the horizontal co-ordinate x_0 is calculated from x_q divided by two. In order to achieve the minimum difference, with better accuracy, between the left and the right partitions while dividing a region, the foreground pixels of the x_0 column of the image array are either considered to belong equally to both the left and the right regions of the vertical division or just to the left one. So, the initial image is divided vertically into two rectangular sub-images depending on the value of x_q. If x_q mod 2=0 then the vertex co-ordinates of these two sub-images are: {(1, 1), (x_0, y_{max})} and {(x_0 , 1), (x_{max}, y_{max})}. Otherwise, if x_q mod 2=1, then the vertex co-ordinates are: {(1, 1), (x_0, y_{max})} and {(x_0+1, 1), (x_{max}, y_{max})}. Fig. 1 illustrates the vertical division of an image where the resulted sub-images share in common the foreground pixels on the division line. From V_1 of Fig. 1c $x_q=10$ thus $x_0=5$. Moreover, x_q mod 2=0 and so the vertex co-ordinates for the two sub-images are: {(1, 1), (5, 9)} and {(5, 1), (9, 9)} (Fig. 1a). Another example of an image array where x_q mod 2=1 is demonstrated in Fig. 2.

Likewise, the vertical co-ordinate y_0 is calculated thus resulting in the division of the initial image into four rectangular sub-images. The whole procedure is applied recursively for every sub-image (Fig. 3).

Let L be the current level of granularity. At this level the number of the sub-images is $4^{(L+1)}$. For example, when $L=0$ (Fig. 3b) the number of sub-images is four and when $L=1$ it is 16 (Fig. 3c). The number of DPs at level L equals to 4^L (Fig. 4). At level L, the co-ordinates (x_i, y_i) of all DPs are stored as features. So, for every L a $2*4^L$-dimensional feature vector is extracted. As Fig. 4 shows, the larger the L the better representation of the character is obtained. Up to here two questions rise as one can easily realize. First, at which level L of granularity the best recognition result is achieved and second, which is the maximum level of granularity that will be used. Both questions are answered in the next section.

After all feature vectors are extracted each feature is scaled to [0, 1]. Since each character is normalized to an $N \times N$ matrix all feature values f are in the range of [1, N]. Therefore, the value f_i of the ith feature of every feature vector is normalized according to Eq. (1).

$$f_i' = \frac{f_i}{N} \tag{1}$$

2.3. Classification

For the recognition procedure a two-stage classification scheme is employed. Since characters with similar structure i.e. 'ζ' and 'ξ' or 'φ' and 'ψ' from the Greek alphabet, are often mutually confused when using a certain granularity feature representation, we propose to merge the corresponding classes at this level of classification. At a next step, we distinguish those character classes by employing a feature vector extracted at another level of granularity where the misclassifications between

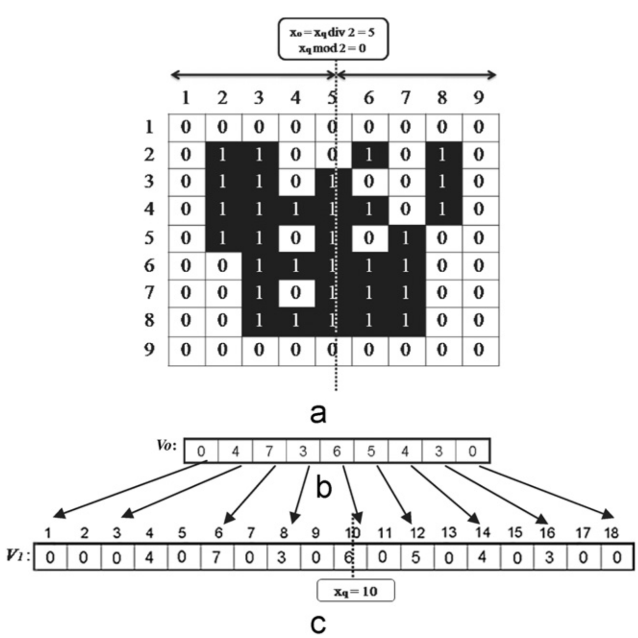

Fig. 1. (a) Example of a vertical division of an image array (x_{max}=9, y_{max}=9); (b) vertical projection (V_0) and (c) creation of V_1 array from V_0 and calculation of x_q.

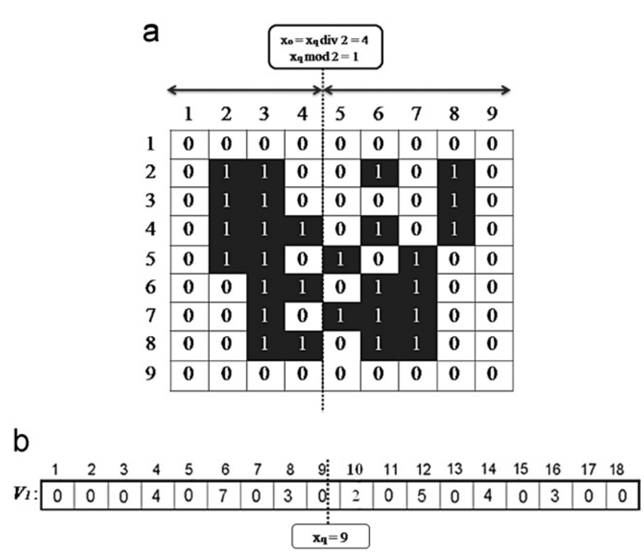

Fig. 2. Example of a vertical division of an image array when x_q mod 2=1.

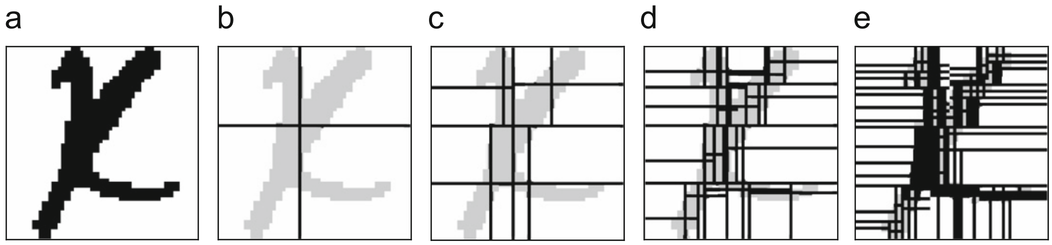

Fig. 3. Character image and sub-images based on DP: (a) original image, (b), (c), (d) and (e) subdivisions at levels 0, 1, 2 and 3, respectively.

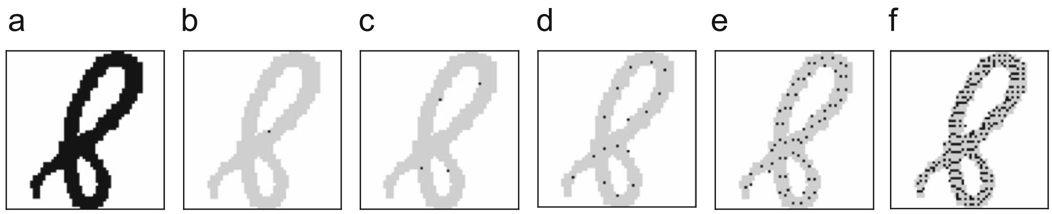

Fig. 4. Features based on DP: (a) original image, (b), (c), (d), (e) and (f) features at levels 0, 1, 2, 3 and 4, respectively.

them are the least possible. The proposed classification scheme has (a) a training and (b) a recognition phase.

2.3.1. *Training phase*

The training phase consists of three distinct steps: step 1 is used to determine the level with the highest recognition rate for the initial classification, step 2 to merge mutually misclassified classes at the level found in step 1 and step 3 to find the level at which each group of merged classes is distinguished the best and to train a new classifier for each one at this level. These steps are described below:

Step 1: Starting from level 1 and gradually proceeding to higher levels of granularity, features are extracted, the confusion matrix is created and the overall recognition rate is calculated, until the recognition rate stops increasing. The level at which the highest recognition rate (*Max_RR*) is achieved is considered to be the best performing granularity level (L_{best}) (Fig. 5). Alternatively, we could examine a large number of granularity levels and choose the one which corresponds to the highest recognition rate. However, after experimentations, we observed that as we proceed to higher levels of granularity, when the recognition rate starts decreasing it will never reach the already achieved maximum again. In addition to that, when using very high levels of granularity the extracted features tend to depend on the exact shape of each character causing more confusion rather than helping in distinguishing between classes since they do not take into account the different variations of a handwritten character.

Confusion matrices are created at each level from the training set using a K-fold cross-validation process. In K-fold cross-validation, the original training set is partitioned into K subsets. Of the K subsets, a single subset is retained as the validation data for testing the model, and the remaining K–1 subsets are used as training data. The cross-validation process is then repeated K times (the *folds*), with each of the K subsets used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and testing, and each observation is for validation exactly once. In our case K is set to 10.

Step 2: At L_{best} where the maximum recognition rate is obtained the corresponding confusion matrix is scanned and classes with high misclassification rates are merged. Class merging is performed using the *disjoint grouping scheme* presented in [26] which is similar to agglomerative clustering [28]. Let the confusion matrix for C classes be $A_{i,j}$, where $A_{i,j}$ (i, $j=1, 2 \ldots C$) is the number of samples that belongs to class i and is classified to class j. The similarity between classes i and j is defined according to Eq. (2).

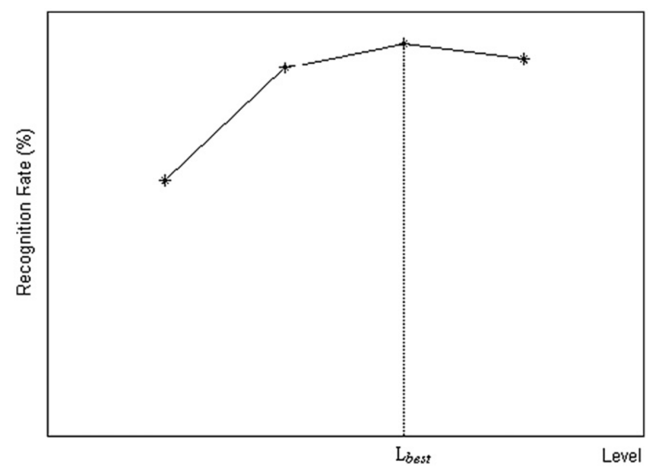

Fig. 5. Example of finding the level of granularity with the highest recognition rate (L_{best}).

$$N_{i,j} = A_{i,j} + A_{j,i}, \quad (i<j) \tag{2}$$

Suppose we have two groups of classes G_p and G_q having m and n classes, respectively. The similarity between these groups ($p<q$) is defined as

$$S_{p,q} = \min_{i<j} N_{i,j}, \quad (i = i_1 \ldots i_m,\ j = j_1 \ldots j_n) \tag{3}$$

Initially each class is a group. First two classes i and j with the highest $N_{i,j}$ value are found and merged into one group thus resulting in C–1 groups. Next, the most similar groups according to Eq. (3) are merged into one. The procedure is iterated until all similarity values between groups are equal to zero in order to find all possible misclassifications.

Step 3: Let G be the total number of groups found in step 2. For each group of classes i, where $i=1, 2 \ldots G$, the procedure described in step 1 is performed again and the best distinguishing granularity level (l_i) for its classes is found. Then, for every group i another classifier is trained with features extracted at its l_i in order to distinguish the merged classes at the next stage of the classification.

2.3.2. Recognition phase

Each pattern of the test set is fed to the initial classifier with features extracted at L_{best}. If the classifier decides that this pattern belongs to one of the non-group classes then its decision is taken into consideration and the unknown pattern is assumed to be classified. Else, if it is classified to one of the group classes then it is given to the group's corresponding classifier and this new classifier decides about the recognition result. Note that if a sample is wrongly classified to a non-group class then at the next stage it will remain wrong. However, if it is misclassified to a group-class then it is possible to be correctly classified in the second stage.

3. Classifier

In the particular recognition problem, classification step was performed using Support Vector Machine (SVM) [3,29] with Radial Basis Function (RBF).

The SVM is a machine learning method basically used for two-class recognition problems. Given a training set of instance-label pairs (x_i, y_i), $i=1\ldots m$, where $x_i \in R^n$ and $y_i \in \{1,-1\}^m$, the SVM selects the optimal hyperplane that maximizes the margin which results into solving Eq. (4).

$$\min_{w,b} \quad \frac{1}{2} w^T w \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1 \tag{4}$$

where w is the weight vector and b is the bias. When the training points are not linearly separable, the cost function is reformulated by introducing slack variables $\xi_i \geq 0, \quad i=1,2,\ldots,m$

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \quad \text{subject to} \quad \begin{matrix} y_i(w^T x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{matrix} \tag{5}$$

where $C > 0$ is the penalty parameter of the error term. However, when the decision function is nonlinear the above scheme cannot be used directly and the SVM requires the solution of the optimization problem to be defined as follows:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \quad \text{subject to} \quad \begin{matrix} y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{matrix} \tag{6}$$

Training vectors x_i are mapped into o higher dimensional space by the function $\phi(\cdot)$. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space.

SVM is used in conjunction with the Radial Basis Function (RBF) kernel, a popular, general-purpose yet powerful kernel, denoted as

$$K(x_i, x_j) \equiv \exp(-\gamma \|x_i - x_j\|^2) \tag{7}$$

RBF kernel nonlinearly maps each sample into a higher dimensional space, so it can handle the case when the relation between class labels and attributes is nonlinear [30].

Furthermore, a grid search was performed in order to find the optimal values for both the variance parameter γ of the RBF kernel and the cost parameter C of SVM using cross-validation. Basically pairs of (C, γ) are tried and the one with the best cross-validation accuracy is picked. For our experiments, the optimal values found for variance parameter γ and the cost parameter C are 0.3 and 100, respectively. The variance parameter γ was searched in the range of (0, 1] and the cost parameter C in the range of (0, 1000].

4. Experiments

4.1. Data setup

For our experiments two handwritten character databases: the CIL Database [31] and the CEDAR Character Database CD-ROM-1 [32] and two handwritten digit databases: the MNIST Database [33] and the CEDAR Digit Database CD-ROM-1 were used. The CIL Database comprises samples of 56 Greek handwritten characters written by 125 Greek writers. Every writer contributed five samples of each letter, thus resulting in a database of 625 variations per letter and an overall of 35 000 isolated and labeled characters. The CEDAR Database consists of samples of 52 English handwritten characters and handwritten digits. For character recognition, 19 145 characters were used for training and 2183 characters for testing. For digit recognition a data set of 24 270 digits were used for training and another one of 5631 for testing. Finally, the MNIST Database consists of 70 000 isolated and labeled handwritten digits. It is divided into a training set of 60 000 and a test set of 10 000 digits.

As mentioned in Section 2.1 all character/digit images pass through a preprocessing stage before the feature extraction method takes place. For the CIL Database and for both of the CEDAR Databases all images are already b/w, so just the size normalization step is performed. In our experiments N is set to 60 for these databases. On the other hand, the digit images in the MNIST Database are already normalized to a 28×28 matrix ($N=28$) but they are gray scale. So, before applying the feature extraction method these images have to pass through the binarization step.

For the CIL Database, after size normalization some characters such as the uppercase 'E' and the lowercase 'ε' are considered to be the same. So, we merged these two classes into one, by randomly selecting 625 characters from both classes as in [31]. This was done to a total of 10 pair of classes, as shown in Table 1 and concluded in having 46 classes of 28 750 characters. Moreover, $\frac{1}{5}$ of each class was used for testing, that is 5750 characters and the remaining $\frac{4}{5}$ for training (23 000 characters).

4.2. Recognition results

As described in Section 2, first features are extracted at different levels of granularity for all patterns in training set and the confusion matrices at each level are constructed, in order to find the best performing granularity level (L_{best}). For the CIL Database L_{best} is found to be three. This is also confirmed by the first column of Table 2 where the recognition accuracies for the

Table 1
Uppercase and lowercase characters with similar shapes that are merged in CIL Database.

CIL Database	
Uppercase	**Lowercase**
Ε	ε
Θ	θ
Κ	κ
Ο	ο
Π	π
Ρ	ρ
Τ	τ
Φ	φ
Χ	χ
Ψ	ψ

Table 2
Experimental results using the CIL Database (46 classes).

CIL Database (46 classes)			
One-stage classification		Confused classes at $L_{best}=3$	l_i
	Recognition rates (%)	ι, I	3
Level 1	64.22	β, B	2
Level 2	90.48	ζ, ξ, ς	4
Level 3	**92.53**	ν, υ, Y	2
Level 4	92.34	φ, ψ	4
		ε, ∑	2
		A, Λ, λ	4
		ω, ώ	4
		η, π, α	3
		έ, ί	4
		ό, ύ	4
		τ, Z, Ξ	4
		μ, M, H, N	3
		δ, σ, γ, Γ	4
		ά, ή	4
		θ, Δ, *o*	4
		ρ, Ω	4

Table 3
Comparison of the proposed OCR methodology using the CIL Database.

CIL Database	
HYB [34]	91.61%
STR [34]	88.62%
DIM [31]	92.05%
VAM [35]	93.21%
Proposed methodology (two-stage classification)	**95.63%**

Table 4
Experimental results using the CEDAR Character Database (52 classes).

CEDAR Character Database (52 classes)			
One-stage classification		Confused classes at $L_{best}=3$	li
	Recognition rates (%)	O, o	2
Level 1	55.24	s, S	2
Level 2	77.23	i, I	2
Level 3	**78.42**	c, C, G	2
Level 4	77.46	u, U	2
		x, X	1
		m, M	2
		w, W	1
		p, P	2
		y, Y	2
		v, V	1
		k, K	2
		H, N	3
		A, R	4
		f, F	4
		t, T, r	3
		B, D	4
		e, E	4
		l, L	1
		z, Z	2
		d, J	3
		b, h	3

Table 5
Recognition rates using the CEDAR Character Database (52 classes).

CEDAR Character Database (52 classes)	
Uppercase characters	86.17%
Lowercase characters	84.05%
Two-stage classification	**85.11**%

test set, when using different levels of granularity for one-stage classification, are shown.

Next, the results when the second stage of the classification approach is applied are presented. Since the best recognition for the CIL Database is achieved in level 3, features from this level are

Table 6
Uppercase and lowercase characters with similar shapes that are merged in CEDAR Character Database.

CEDAR Character Database	
Uppercase	**Lowercase**
O	o
S	s
I	i
C	c
U	u
X	x
M	m
W	w
P	p
Y	y
V	v
K	k
F	f
T	t
E	e
L	l
Z	z

Table 7
Experimental results using the CEDAR Character Database (35 classes).

CEDAR Character Database (35 classes)			
One-stage classification		Confused classes at $L_{best}=4$	li
	Recognition rates (%)	{i, I}, {l, L}, b	2
Level 1	62.62	u, v	2
Level 2	88.86	{o, O}, D	2
Level 3	89.96	{c, C}, {e, E}, {p, P}, B	4
Level 4	**90.70**	{m, M}, N	3
Level 5	82.54	A, R	4
		{f, F}, r	4
		{t, T}, {y, Y}, g	3
		{s, S}, J, d	3
		a, G	3
		{k, K}, {x, X}	4
		h, n	4
		{w, W}, H	2
Two-stage classification recognition rate = 94.73%			

Table 8
Experimental results using only uppercase characters from the CEDAR Character Database (26 classes).

CEDAR Character Database—Uppercase characters (26 classes)			
One-stage classification		Confused classes at $L_{best}=4$	li
	Recognition rates (%)	U, V	1
Level 1	68.32	D, O, B	4
Level 2	91.95	M, N, H	3
Level 3	93.48	A, R, Z	4
Level 4	**93.78**	J, S	3
Level 5	84.05	F, T	2
		K, X	3
		C, L	2
		P, Y	3
		E, I	3
Two-stage classification recognition rate = 95.90%			

used to train the initial SVM. Then, the confusion matrix at level 3 is scanned and classes with high misclassification rates are detected. Table 2 also shows the groups of classes which are confused the most. Each group of classes is then merged into one class. For every group i the granularity level that best distinguishes its classes (l_i) is found and a new SVM is trained with features from that level. As shown in the last row of Table 3, when the second stage of the classification scheme is used the overall recognition rate is improved (95.63%). In Table 3, we also present a comparison of this result with other state-of-the-art feature extraction methods for handwritten character recognition, that are to the best of our knowledge the only works in the literature that deal with Greek handwritten characters. These methods are the following:

(a) A hybrid feature extraction scheme based on zones and projections (HYB) [34].
(b) A scheme based on structural features based on projections and radial profiles (STR) [34].
(c) Features based on both statistical and structural methods with a dimensionality reduction scheme (DIM) [31].
(d) Previous work of the authors that results only to disjoint sub-images and does not include iteration of the first step of the classification procedure for each group of merged classes (VAM) [35].

Table 9
Experimental results using only lowercase characters from the CEDAR Character Database (26 classes).

CEDAR Character Database—Lowercase characters (26 classes)			
One-stage classification		Confused classes at $L_{best}=3$	*li*
	Recognition rate (%)	i, l, p	2
Level 1	70.09	u, v	2
Level 2	87.25	c, e, s, z	3
Level 3	**89.70**	a, o	3
Level 4	88.60	f, r, t	3
		h, n	4
		g, y	3
		k, x	4
		b, d	4
		m, w	2
Two-stage classification recognition rate = 93.50%			

Regarding the CEDAR Character Database the best performing granularity level, for 52 classes, is also three. Again, from Table 4 it is clear that the recognition rate for the test set is higher (78.42%) when features from this level are used. Then, misclassified classes are detected (Table 4) and for each group a new SVM is trained with features from the best distinguishing granularity level and when the second stage of the classification scheme is applied the overall recognition rate is improved (85.11%), as shown in Table 5. In Table 5, the recognition rates for the uppercase characters (A–Z) as well as for the lowercase characters (a–z) are also presented separately.

From the right column of Table 4, it is evident that in order to have meaningful results uppercase and lowercase characters with similar shapes should be merged, as also suggested in [14,15]. So, according to Table 4 we merge 17 of these pairs (Table 6), thus resulting in 35 classes. The best performing level, for this 35-class problem, is four (Table 7). Classes with high misclassification

Table 10
Comparison of the proposed OCR methodology using the CEDAR Character Database (52 classes).

CEDAR Character Database			
	Uppercase characters (%)	**Lowercase characters** (%)	**Overall recognition rate** (%)
YAM [36]	NA	NA	75.70
KIM [16]	NA	NA	73.25
GAD [17]	79.23	70.31	74.77
Proposed methodology	**86.17**	**84.05**	**85.11**

Table 11
Comparison of the proposed OCR methodology using the CEDAR Character Database for uppercase only and lowercase only characters.

CEDAR Character Database						
	Uppercase characters (26 classes)			**Lowercase characters (26 classes)**		
	No. of train patterns	**No. of test patterns**	**Recognition rate (%)**	**No. of train patterns**	**No. of test patterns**	**Recognition rate (%)**
BLU [13]	7175	939	81.58	18655	2240	71.52
Proposed methodology	11454	1367	**95.90**	7691	816	**93.50**

Table 12
Comparison of the proposed OCR methodology using the CEDAR Character Database after merging lowercase and uppercase characters with similar shapes.

CEDAR Character Database				
	Number of classes (all classes)	**Recognition rate (%)**	**Number of classes (after merging)**	**Recognition rate (%)**
SIN [15]	52	NA	36	67
CAM [14]	52	83.74	39	84.52
Proposed methodology	52	**85.11**	35	**94.73**

rates are detected and for each one a new SVM is trained with features from the level that distinguishes them the best, while the last row of Table 7 depicts the improved recognition rate after applying the second step of classification.

Finally, Tables 8 and 9 present the results of the proposed methodology when using only the uppercase characters (A–Z) or only the lowercase ones (a–z).

The CEDAR Database is widely used in the literature:

(a) In Yamada et al. (YAM) [36] a classifier was trained to output one of the 52 classes (a–z, A–Z). Their top recognition rate was 75.7%.
(b) Kimura et al. (KIM) [16] produced a recognition rate of 73.25% again for 52 classes.
(c) Singh and Hewitt (SIN) [15] propose merging uppercase and lowercase characters with similar shapes, such as ‘O’ and ‘o’, resulting in 36 classes. Their best score was around 67%.
(d) Gader et al. (GAD) [17] achieved a recognition rate of 79.23% for uppercase characters and 70.31% for lowercase characters, according to [29], for a 52-class classification problem.
(e) In Camastra and Vinciarelli (CAM) [14], the number of classes used is between 26 and 52 depending on how many uppercase and lowercase characters with similar shapes are merged. The recognition rate for 52 classes was 83.74%, while the best recognition rate that they report was 84.52% when using 39 classes.
(f) In Blumenstein et al. (BLU) [13], the top recognition rate for uppercase characters was 81.58% and for lowercase characters is 71.52%.

Comparisons of the above state-of-the-art techniques with the proposed methodology are shown in the tables below. Table 10 depicts the results when all 52 classes (a–z, A–Z) are fed to the classifier. The proposed methodology not only achieved the highest overall recognition rate but also performed better even when trying to distinguish the lowercase characters (a–z), or the

Table 13
Experimental results using the MNIST Digit Database (10 classes).

MNIST Digit Database (10 classes)			
One-stage classification		Confused classes at $L_{best}=4$	li
	Recognition rates (%)	4, 9	4
Level 1	78.01	3, 5, 8	4
Level 2	95.63	2, 7, 1	4
Level 3	97.58		
Level 4	**98.08**		
Level 5	97.43		
Two-stage classification recognition rate = 99.03%			

Table 14
Experimental results using the CEDAR Digit Database (10 classes).

CEDAR Digit Database (10 classes)			
One-stage classification		Confused classes at $L_{best}=3$	li
	Recognition rates (%)	4, 6	3
Level 1	86.91	3, 9, 2	3
Level 2	96.39	1, 7	4
Level 3	**97.17**	5, 8	3
Level 4	96.83		
Two-stage classification recognition rate = 98.66%			

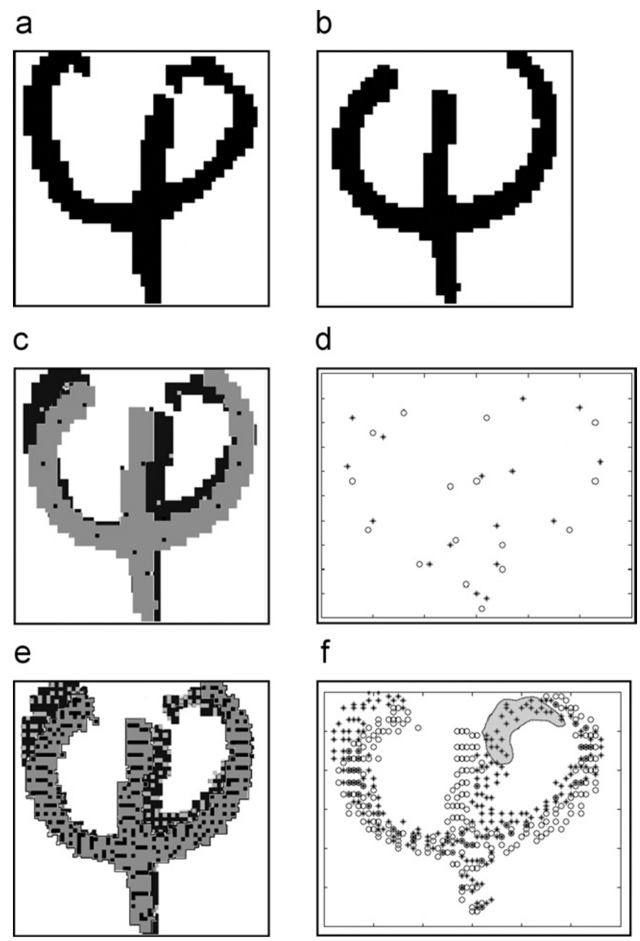

Fig. 7. (a), (b) Characters ‘φ’ and ‘ψ’. (c), (e) Features at granularity levels 2 and 4, respectively. (d) Representation in feature space at level 2: (*) for ‘φ’ and (o) for ‘ψ’. At this level such characters cannot be distinguished sufficiently. (f) Representation in feature space at level 4: (*) for ‘φ’ and (o) for ‘ψ’. Level 4 works fine for these characters mainly because of the features in the gray region.

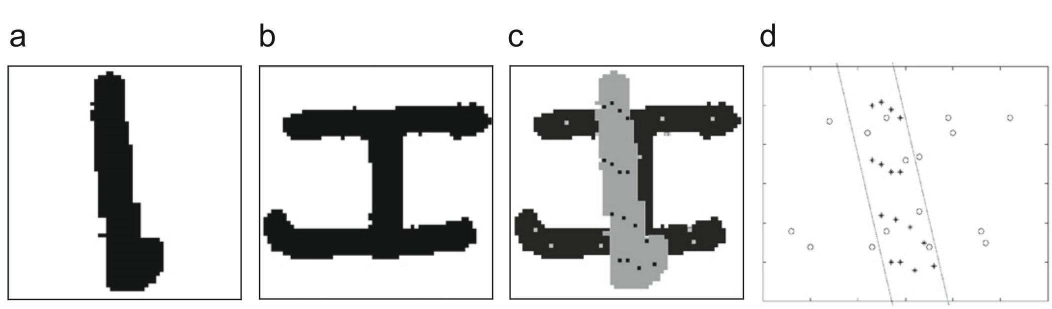

Fig. 6. (a), (b) Characters ‘ι’ and ‘I’. (c) Features at granularity level 2. (d) Representation in feature space: (*) for ‘ι’ and (o) for ‘I’. At this level such characters can be distinguished.

Table 15
Computational time of the proposed OCR methodology.

Database	No. of classes	No. of train samples	No. of test samples	Training phase	Recognition phase
CIL	46	23000	5750	33′ 42″	1′ 14″
CEDAR	52	19145	2183	34′ 08″	42″
CEDAR	35	19145	2183	54′ 34″	50″
CEDAR (uppercase)	26	11454	1367	72′ 35″	35″
CEDAR (lowercase)	26	7691	816	5′ 06″	15″
CEDAR (digits)	10	24270	5631	14′ 34″	21″
MNIST	10	60000	10000	122′ 53″	11′ 31″

uppercase ones (A–Z), among all 52 classes. Table 11 presents the recognition rates when dealing only with the lowercase characters (26 classes) or only with the uppercase characters (26 classes). Finally, in Table 12 it is obvious that the proposed methodology scored higher for 52 classes than Camastra et al. [14]. Moreover, their highest recognition rate (84.52%) that was achieved after merging uppercase and lowercase characters with similar shapes resulting to 39 classes is considerably less than the one achieved by the proposed methodology for 35 classes.

For the MNIST Database and the digits from the CEDAR Database the best performing granularity level is 4 and 3, respectively (Tables 13 and 14). Again, the best recognition rate is achieved when the second stage of the classification scheme is applied.

According to [33] the lowest recognition rate for the MNIST Database is 88% and the highest is 99.61%, while the best results available vary between 98.5% and 99.5%.

The best performing granularity level L_{best} for all experiments is 3 or 4 depending on the conflicts between characters from different classes but with similar structure. Level 3 or 4 is considered to be sufficient enough to perform a preliminary discrimination since the features extracted at these levels provide a good representation of the shape of the character. However, the granularity level at which merged classes are separated at the second stage of the classification procedure varies from level 1 to 4. As shown in the above experimental results features from low levels of granularity ($l=2$) are suitable for distinguishing a pair of characters such as ‘ι’ and ‘I’ (Fig. 6). On the other hand, at this level the misclassification rate between characters such as ‘φ’ and ‘ψ’ is high since discrimination is very difficult (Fig. 7d). For these characters features from higher levels ($l=4$) of granularity need to be employed (Fig. 7f).

All experiments were conducted on a Core 2 CPU 6400@2.13 GHz with 2.00 GB of RAM under 32-bit Windows XP operating system. In Table 15 the computational time for both training and recognition phase is presented for all databases used in our experiments. As one can observe, the training phase is time consuming depending on the number of classes, the number of train patterns, the maximum level of granularity that needs to be examined in order to find the best performing level for the initial step of the classification procedure, the number of groups of merged classes and the calculation of their best distinguishing levels. However, the recognition phase is very fast.

5. Concluding remarks

In this paper we propose an OCR methodology for handwritten characters that relies on a new feature extraction technique based on recursive subdivisions of the image as well as on calculation of the introduced *division point*. Even though the feature extraction method itself is quite efficient when a specific level of granularity is used, there is more to be gained in classification accuracy by exploiting the intrinsically recursive nature of the method. This is achieved by appropriately combining the results from different levels using a two-stage hierarchical approach. During the first stage a preliminary discrimination is performed at a certain level while at the next stage features from different levels of granularity help in distinguishing between characters of similar shapes that are confused at the first stage. As shown in the experimental results the recognition rates that we achieve are the highest, to the best of our knowledge, when dealing with handwritten characters from the CIL and the CEDAR Databases. Moreover, the proposed methodology, although focused on handwritten characters, works efficiently enough even for handwritten digits. Our future research is focused on applying the proposed features for word recognition as well as combine them with other feature extraction schemes in order to further improve the recognition performance.

References

[1] A.S. Brito, R. Sabourin, F. Bortolozzi, Foreground and background information in a HMM-based method for recognition of isolated characters and numeral strings, in: Proceedings of the Ninth International Workshop on Frontiers in Handwritten Recognition, 2004, pp. 371–376.

[2] J.A. Fitzgerald, F. Geiselbrechtinger, T. Kechadi, Application of fuzzy logic to online recognition of handwritten symbols, in: Proceedings of the Ninth International Workshop on Frontiers in Handwritten Recognition, 2004, pp. 395–400.

[3] C. Cortes, V. Vapnik, Support-vector network, Machine Learning 20 (1997) 273–297.

[4] C.L. Liu, H. Fujisawa, Classification and learning for character recognition: comparison of methods and remaining problems, in: International Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, 2005.

[5] F. Bortolozzi, A.S. Brito, L.S. Oliveira, M. Morita, Recent advances in handwritten recognition, in: U. Pal, S.K. Parui, B.B. Chaudhuri (Eds.), Document Analysis, 2008, pp. 1–30.

[6] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combing classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.

[7] L.I. Kuncheva, Combing Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, Hoboken, New Jersey, USA, 2004.

[8] L.S. Oliveira, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Transactions on Pattern Recognition and Machine Intelligence 24 (11) (2001) 1448–1456.

[9] K.M. Mohiuddin, J. Mao, A comprehensive study of different classifiers for hand-printed character recognition, Pattern Recognition, Practice IV (1994) 437–448.

[10] A.L. Koerich, Unconstrained handwritten character recognition using different classification strategies, in: International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR), 2003.

[11] J.H. Kim, K.K. Kim, C.Y. Suen, Hybrid schemes of homogeneous and heterogeneous classifiers for cursive word recognition, in: Proceedings of the Seventh International Workshop on Frontiers in Handwritten Recognition, Amsterdam, 2000, pp. 433–442.

[12] N. Arica, F. Yarman-Vural, An overview of character recognition focused on off-line handwriting, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31 (2) (2001) 216–233.

[13] M. Blumenstein, X.Y. Liu, B. Verma, A modified direction feature for cursive character recognition, in: IEEE International Joint Conference on Neural Networks, vol. 4, 2007, pp. 2983–2987.

[14] F. Camastra, A. Vinciarelli, Combining neural gas and learning vector quantization for cursive character recognition, Neurocomputing 51 (2003) 147–159.
[15] S. Singh, M. Hewitt, Cursive digit and character recognition on cedar database, in: International Conference on Pattern Recognition, (ICPR 2000), Barcelona, Spain, 2000, pp. 569–572.
[16] F. Kimura, N. Kayahara, Y. Miyake, M. Shridhar, Machine and human recognition of segmented characters from handwritten words, in: International Conference on Document Analysis and Recognition (ICDAR '97), Ulm, Germany, 1997, pp. 866–869.
[17] P.D. Gader, M. Mohamed, J.-H. Chiang, Handwritten word recognition with character and inter-character neural networks, IEEE Transactions on System, Man and Cybernetics—Part B: Cybernetics 27 (1997) 158–164.
[18] O.D. Trier, A.K. Jain, T. Taxt, Features extraction methods for character recognition—a survey, Pattern Recognition 29 (4) (1996) 641–662.
[19] J. Park, V. Govindaraju, S.N. Shrihari, OCR in hierarchical feature space, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (24) (2000) 400–408.
[20] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1990.
[21] S. Mozaffari, K. Faez, M. Ziaratban, Character representation and recognition using quadtree-based fractal encoding scheme, in: Proceedings of the Eighth International Conference on Document Analysis and Recognition, vol. 12, Seoul, Korea, 2005, pp. 819–823.
[22] A.P. Sexton, V. Sorge, Database-driven mathematical character recognition, graphics recognition, algorithms and applications (GREC), Lecture Notes in Computer Science (LNCS), Hong Kong, 2006, pp. 206–217.
[23] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.
[24] A.K. Jain, D. Zongker, Representation and recognition of handwritten digits using deformable templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (12) (1997) 1386–1391.
[25] L. Vuurpijl, L. Schomacker, M. van Erp, Architecture for detecting and solving conflicts: two-stage classification and support classifiers, International Journal on Document Analysis and Recognition (IJDAR) 5 (4) (2004) 213–223.
[26] T.K. Bhowmik, P. Ghanty, A. Roy, S.K. Parui, SVM-based hierarchical architectures for handwritten Bangla character recognition, International Journal of Document Analysis and Recognition (IJDAR) 12 (2) (2009) 97–108.
[27] W. Niblack, An Introduction to Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986, pp. 115–116.
[28] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1988.
[29] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[30] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, Technical Report, Taipei, 2003.
[31] G. Vamvakas, B. Gatos, S. Petridis, N. Stamatopoulos, An efficient feature extraction and dimensionality reduction scheme for isolated Greek handwritten character recognition, in: Proceedings of the Ninth International Conference on Document Analysis and Recognition, Curitiba, Brazil, 2007, pp. 1073–1077.
[32] J.J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.
[33] The MNIST Database, 〈http://yann.lecun.com/exdb/mnist/〉.
[34] G. Vamvakas, N. Stamatopoulos, B. Gatos, I. Pratikakis, S.J. Perantonis, Greek Handwritten character recognition, in: Proceedings of the Eleventh Panhellenic Conference in Informatics, vol. B, Patras, Greece, 18–20 May 2007, pp. 343–352.
[35] G. Vamvakas, B. Gatos, S.J. Perantonis, Hierarchical classification of handwritten characters based on novel structural features, in: Eleventh International Conference on Frontiers in Handwriting Recognition (ICFHR'08), Montreal, Canada, August 2008, pp. 535–539.
[36] H. Yamada, Y. Nakano, Cursive handwritten word recognition using multiple segmentation determined by contour analysis, IECIE Transactions on Information and System E 79-D (1996) 464–470.

About the Author—GEORGIOS VAMVAKAS was born in Athens in 1981. He graduated in 2005 from the Department of Informatics and Telecommunications of National and Kapodistrian University of Athens, and from January 2006 he is a Ph.D. candidate under scholarship at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos". His research interests are image processing and document image analysis, optical character recognition, processing and recognition of historical documents.

About the Author—BASILIOS GATOS received his Electrical Engineering Diploma in 1992 and his Ph.D. degree in 1998, both from the Electrical and Computer Engineering Department of Democritus University of Thrace, Xanthi, Greece. He is currently working as a researcher at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos", Athens, Greece. His main research interests are in image processing, pattern recognition, document image analysis, OCR, processing and recognition of historical documents.

About the Author—STAVROS J. PERANTONIS is the holder of a BS degree in Physics from the Department of Physics, University of Athens, an M.Sc. degree in Computer Science from the Department of Computer Science, University of Liverpool and a D.Phil. degree in Computational Physics from the Department of Physics, University of Oxford. Since 1992 he has been with the Institute of Informatics and Telecommunications, NCSR "Demokritos", where he currently holds the position of Senior Researcher and Head of the Computational Intelligence Laboratory. His main research interests are in image processing and document image analysis, OCR and pattern recognition.