A Fuzzy Approach to 2-D Shape Recognition
Beatrice Lazzerini and Francesco Marcelloni

Abstract—This paper describes a method for fuzzy classification and recognition of two-dimensional (2-D) shapes, such as handwritten characters, image contours, etc. A fuzzy model is derived for each considered shape from a fuzzy description of a set of instances of this shape. A fuzzy description of a shape instance, in its turn, exploits appropriate fuzzy partitions of the two dimensions of the shape. These fuzzy partitions allow us to identify, and automatically associate an importance degree with, the relevant shape zones for classification and recognition purposes.

Two significant applications of the method are described, namely, recognition of olfactory signals and recognition of isolated, handwritten characters. In the former case, results are shown concerning the recognition of three different types of waste waters, collected in three different dilutions. In the latter case, results are shown concerning the application of the method to a NIST database, containing the segmented handwritten characters of 500 writers.

Index Terms—Fuzzy recognition, handwriting recognition, olfactory signal recognition, shape recognition.

I. INTRODUCTION

PATTERN recognition plays a crucial role in a large number of applications, including printed and handwritten text recognition [1]–[3], speech recognition [4], human face recognition [5], etc. A great variety of conventional and computational intelligence techniques for pattern classification and recognition can be found in [6] and [7].

Typically, each object involved in classification is represented as a multidimensional feature vector. The number and kind of original features depend on the specific application domain, and their choice (feature nomination) is usually made by a domain expert. Then, for efficiency reasons, feature selection is often required to reduce the dimension of the original feature space. Feature selection consists of choosing the features that are actually responsible for correct class discrimination. The quality of a set of features is generally assessed based on the error rate (i.e., the percentage of incorrectly recognized objects) achieved by the classifier on an appropriate test set. Alternatively, depending on the application domain, all the nominated features may be retained and appropriately weighted (feature weighting). Actually, a feature selection algorithm can be considered as a feature weighting algorithm that uses binary weights (i.e., 1 or 0) to either keep or discard features [8].

In this paper, we present a new general method for two-dimensional (2-D) shape recognition. We aim to model as faithfully as possible the way human beings recognize shapes. When you look at an unknown shape, you mentally compare it, as a whole, with known shapes and recognize it based on its general appearance. In the comparison, you exploit the presence and the location of features, maybe imprecise and vague, that resemble, more or less, a reference model (possibly vague and imprecise, in its turn) you have in your mind. Of course, the specific application domain determines the kind and the position of key features for shape recognition.

Based on these considerations, we aimed to develop a general method able to automatically build a fuzzy model of generic 2-D shapes. The basic idea underlying our method is the following. Suppose you want to describe the typical shape of a (2-D) object. You may consider a set of significant individual instances of that object and describe each of them in such a way that the corresponding descriptions are as similar as possible. These descriptions should take into account some key parts of the shape that are responsible for shape recognition. The more similar the individual descriptions are, the easier to derive an abstract, general description of the shape. Basically, our method does not make an a priori choice of the relevant features, rather it tries to automatically associate an importance degree, for modeling purposes, with the constituent parts of a shape, based on individual instances of that shape. Intuitively, the more frequently a constituent part appears in specific descriptions, the higher the importance of that part in modeling the shape of interest. Stated another way, within a shape, each single “point” has a more or less important role, depending on its position.

We are concerned with the following problem: the automatic recognition of the shape of unknown 2-D objects. We assume that the shapes under consideration are correctly positioned with respect to the coordinate axes. This means that possible image preprocessing actions, such as physical rotation and/or shifting for slope and skew correction, are external to our method. We first build a set of reference fuzzy models of the typical shapes we are interested in. A reference fuzzy model is built for each shape based on a fuzzy description of a set of instances of that shape. This description exploits appropriate fuzzy partitions of the two dimensions of the shape. In particular, for each shape instance, we consider the bounding box, which contains the instance, and partition the horizontal and vertical dimensions of the box into $T$ and $K$ (possibly different-sized) subintervals (in general, $T \neq K$). Then, for each interval extreme, a triangular fuzzy set is built. The fuzzy set modal value, i.e., the point at which the membership degree is 1, coincides with the interval extreme and the fuzzy set support covers two adjacent intervals, with the exception of the first and last fuzzy sets, which cover one interval. A label is assigned to each fuzzy set. A linguistic meaning might be associated with fuzzy set labels, thus producing linguistic partitions. The contribution of a shape point to...
the classification task is directly proportional to the closeness of this point to the modal values of the triangular fuzzy sets in the fuzzy partitions. It is obvious that the number and the position of these modal values identify the relevant points for shape classification and recognition. Actually, the modal values determine the fuzzy partitions and, consequently, they are responsible for identifying and appropriately weighing the key zones of the shapes. Such zones represent our features. The number and the position of the modal values in the two dimensions are optimized by means of a genetic algorithm whose fitness function is the recognition rate (i.e., the percentage of correctly recognized shapes) on the training set.

When an unknown shape has to be recognized we build a fuzzy description of the unknown shape, utilizing the same fuzzy partitions of the shape dimensions as in the reference fuzzy models. Then, we use a specific similarity measure based on a weighted distance to obtain a similarity degree between the fuzzy description of the unknown shape and each fuzzy reference model. Higher weights are associated with more relevant shape zones. Weights are automatically computed during the training phase. The similarity degree indicates how much the unknown shape is similar to the shape represented by the reference fuzzy model. The unknown shape is recognized as the reference fuzzy model with the highest similarity degree.

We describe two significant applications of our method: recognition of olfactory signals and recognition of handwritten characters. Note that, although signals are not really 2-D shapes, a simplified version of our method can be used for signal recognition. In the former case, our method has been used to develop the pattern recognition system of an electronic nose employing a conducting polymer sensor array. The pattern recognition system is responsible for recognizing the signals produced by the sensors when they are exposed to odorants. In the latter case, our method has been used to implement the character recognizer within a system for automatic off-line recognition of handwritten text [9]. The recognized characters are then processed by a context analysis module, which makes use of a dictionary and a syntax analyzer.

II. THE METHOD

In this section, we describe our approach without referring to a specific application. Let us assume that our problem is to recognize $C$ different classes of shapes in a 2-D space. We first represent the shape instances in terms of fuzzy sets. Then, for each class $c$, $c = 1 \ldots C$, of shapes we build a fuzzy model of $c$. When an unknown shape $\mathbf{u}$ has to be recognized, we compare the fuzzy representation of $\mathbf{u}$ with each fuzzy model $c$, $c = 1 \ldots C$, by means of a purposely defined similarity measure. The unknown shape is recognized as belonging to the class with the highest similarity value.

A. Modeling Shape Instances

For each shape instance $i$, we consider the bounding box (i.e., the minimum rectangle), parallel to the coordinate axes, that contains $i$. This provides a normalization that makes the shapes comparable with each other. Let the horizontal and vertical spaces denote, respectively, the horizontal and vertical dimensions of the bounding box containing $i$. Shape instances are modeled by performing the following two steps.

1) Fuzzy partition of the horizontal and vertical spaces: The horizontal and vertical spaces are divided, respectively, into $T$ and $K$ (possibly different-sized) subintervals (in general, $T \neq K$). The values for $T$ and $K$, and the size of the subintervals are optimized so as to highlight the regions of a shape that are most important to model the shape itself. A genetic algorithm is used for optimization (see Section III). In both spaces, a triangular fuzzy set is built for each subinterval extreme, which is the fuzzy set modal value. Each fuzzy set covers two adjacent subintervals, with the exception of the first and last fuzzy sets that cover one subinterval. A label is assigned to each fuzzy set.

Fig. 1 shows a generic 2-D shape, and the fuzzy partition of the horizontal and vertical spaces. In the following, we will use $H_t$, $t = 1 \ldots T + 1$, and $V_k$, $k = 1 \ldots K + 1$, to label the $t$th and $k$th fuzzy set in the horizontal and vertical space, respectively.

For the sake of simplicity, let us refer to a fuzzy partition on one dimension, e.g., the horizontal dimension, only. The contribution of a shape point $p = (h_p, v_p)$ to the shape model construction is as much higher as the coordinate $h_p$ is closer to a fuzzy set modal value. The maximum importance is assigned to the points whose coordinate $h_p$ is the same as the horizontal coordinate of a modal value. Therefore, the fuzzy set modal values have to be chosen so as to assign importance values to the shape points based on the closeness of the horizontal coordinate of these points to modal values. As a consequence, the number and the position of the fuzzy set modal values must be optimized in such a way that they really give more importance to shape points that are key points for shape classification. Of course, considering the horizontal partition only, does not allow us to discriminate between shape points having the same horizontal coordinate and different vertical coordinates. For this reason, we also use a fuzzy partition on the vertical dimension. By appropriately combining the two partitions we are able to study the relative importance, for modeling purposes, of the shape regions induced by the partitions. As shown in Fig. 2, given two fuzzy sets $H_t$ and $V_k$ on the horizontal and vertical spaces, respectively, the Cartesian product of their supports defines a region $R_{t,k}$ in the
space of the shape instance \(i\). A point \(p\) of instance \(i\) falling inside \(R_{t,k}\) contributes to modeling \(i\) based on the location of \(p\) with respect to the modal values of \(H_t\) and \(V_k\), respectively.

2) Fuzzy representation: For each fuzzy set \(V_k\), \(k = 1 \ldots K + 1\), we associate a horizontal importance value \(h_{\mu_{V_k}}(t_k)\) with each region \(R_{t,k}\), \(t = 1 \ldots T + 1\), as follows:

\[
h_{\mu_{V_k}}(t_k) = \frac{\sum_{p \in P_{t,k}} \mu_{H_t}(h_p)\mu_{V_k}(v_p)}{\sum_{p \in P_{t,k}} \mu_{V_k}(v_p)}
\]

where \(P_{t,k}\) is the set of points of the shape instance \(i\) contained in \(R_{t,k}\). \(h_p\) and \(v_p\) are the horizontal and vertical coordinates of the \(p\)th point, respectively. \(\mu_{H_t}\) and \(\mu_{V_k}\) are the triangular membership functions of \(H_t\) and \(V_k\), respectively. The rationale for this computation is that the contribution to \(h_{\mu_{V_k}}(t_k)\) of a point close to the modal value of a fuzzy set on the horizontal or vertical space must be greater than the contribution of points with a low membership degree.

The horizontal importance value \(h_{\mu_{V_k}}(t_k)\) varies in the interval \([0, 1]\). \(h_{\mu_{V_k}}(t_k) = 0\) when the set \(P_{t,k}\) is empty. \(h_{\mu_{V_k}}(t_k) = 1\) if all the points in \(P_{t,k}\) have the horizontal coordinate coincident with the modal value of \(H_t\). If the whole region \(R_{t,k}\) is black, it can be proved that \(h_{\mu_{V_k}}(t_k)\) approaches 0.5 as the number of points in the support of \(V_k\) increases.

For each fuzzy set \(V_k\), we define a fuzzy set \(X_{t,k} = \{h_{\mu_{V_k}}(t_k)/H_t + \cdots \} H_{T+1}/H_{T+1}\} \) in the space of the labels \(H_t\), the horizontal importance value \(h_{\mu_{X_k}}(t_k)\) being the membership degree of \(H_t\). In the adopted notation, the slash is employed to link the elements \(H_t\) of the universe of the labels with their grades of membership in \(X_{t,k}\), and the plus sign indicates that the listed pairs of elements and membership grades collectively form \(X_{t,k}\). We call \(X_{t,k}\) a horizontal fuzzy strip of instance \(i\). Similarly, for each fuzzy set \(H_t\), we define a fuzzy set \(V_{t,k} = \{h_{\mu_{V_k}}(t_k)/V_{t,k} + \cdots \} V_{K+1}/V_{K+1}\} \) in the space of the labels \(V_k\), the vertical importance value \(v_{\mu_{V_k}}(t_k)\) being the membership degree of \(V_k\). We call \(V_{t,k}\) a vertical fuzzy strip of instance \(i\).

The instance fuzzy representation of the shape instance \(i\) is the crisp set \(I_i = \{H_{t_1}, V_{t_1}, \ldots, H_{t_T}, V_{t_T+1}\} \) and \(V_{t_k} = \{V_{t_1}, V_{t_2}, \ldots, V_{t_K+1}\} \) are the horizontal and vertical fuzzy representations of \(i\), respectively. While the former associates the horizontal strips \(X_{t,k}\) with the vertical fuzzy set labels \(V_k\), the latter associates the vertical strips \(Y_{t,k}\) with the horizontal fuzzy set labels \(H_t\).

B. Modeling Shape Classes

For each shape class \(c, c = 1 \ldots C\), we build a fuzzy model called class fuzzy representation. The class fuzzy representation is obtained by considering a set of sample instances of class \(c\) (training set). As previously stated, for each instance in the training set, the size of the shape space is chosen to be equal to the minimum bounding box containing the instance. The horizontal and vertical spaces of all these instances are divided, respectively, into the same numbers \(T\) and \(K\) of subintervals. Within each space, the subintervals may have different sizes. The optimal number and position of the interval extremes (or, equivalently, of the fuzzy set modal values) in the horizontal and vertical spaces are application-dependent parameters. We will discuss a genetic algorithm-based approach to determine the optimal modal values in Section III.

Let us assume that, for each shape class \(c\), the training set contains \(N\) instances. For each class \(c\) and each fuzzy set \(V_k\), \(k = 1 \ldots K + 1\), we define the horizontal fuzzy strip \(X_{c,k}\) of class \(c\) as

\[
X_{c,k} = \left\{ \frac{\mu_{X_{c,1,k}}/H_1 + \cdots + \mu_{X_{c,T+1,k}}/H_{T+1}}{\sum_{t=1}^{T+1}} \right\}
\]

where \(\mu_{X_{c,t,k}}/H_1 + \cdots + \mu_{X_{c,T+1,k}}/H_{T+1}\} \) is the average of the horizontal importance values associated with the region \(R_{t,k}\) in the \(N\) instances of class \(c\). Similarly, for each class \(c\) and each fuzzy set \(H_t, t = 1 \ldots T + 1\), we define the vertical fuzzy strip \(Y_{c,t}\) of class \(c\) as

\[
Y_{c,t} = \left\{ \frac{\mu_{Y_{c,1,t}}/V_1 + \cdots + \mu_{Y_{c,T+1,t}}/V_{K+1}}{\sum_{t=1}^{T+1}} \right\}
\]

where \(\mu_{Y_{c,t,k}}/V_1 + \cdots + \mu_{Y_{c,T+1,t}}/V_{K+1}\} \) is the average of the vertical importance values associated with the region \(R_{t,k}\) in the \(N\) instances of class \(c\). Then, the class fuzzy representation that describes the class \(c\) is the crisp set \(C_c = \{HC_c, VC_c\} \).
where the crisp set \( HC_c = \{ \overline{X}_c, \ldots, \overline{X}_{c,K+1} \} \) is the horizontal fuzzy representation of \( c \), and the crisp set \( VC_c = \{ \overline{Y}_c, \ldots, \overline{Y}_{c,T+1} \} \) is the vertical fuzzy representation of \( c \).

### C. Recognition

The class fuzzy representations \( C_c, c = 1 \ldots C \), built during the training phase are used to classify unknown shapes. When an unidentified shape \( \eta \) is presented, its fuzzy representation is produced. More precisely, we calculate the instance fuzzy representation \( I_\eta \) using the same horizontal and vertical space partitions adopted in building the class fuzzy representations \( C_c \). Then, \( I_\eta \) is compared with each class fuzzy representation \( C_c \) to produce a similarity value \( S_c \).

For each class \( c \), we compute the similarity \( S_c(I_\eta, C_c) \) between the instance fuzzy representation \( I_\eta \) and the class fuzzy representation \( C_c \) as follows:

\[
S_c(I_\eta, C_c) = w_1 S(HI_\eta, HC_c) + w_2 S(VI_\eta, VC_c)
\]

(1)

where \( S(HI_\eta, HC_c) \) is the similarity between the horizontal fuzzy representations \( HI_\eta \) and \( HC_c \). \( S(VI_\eta, VC_c) \) is the similarity between the vertical fuzzy representations \( VI_\eta \) and \( VC_c \), the weights \( w_1 \) and \( w_2 \) are real numbers in the interval \([0,1]\) such that \( w_1 + w_2 = 1 \).

The two similarities \( S(HI_\eta, HC_c) \) and \( S(VI_\eta, VC_c) \) in (1) are convex sums of the similarities between pairs of corresponding fuzzy strips in the horizontal and vertical fuzzy representations:

\[
S(HI_\eta, HC_c) = \sum_{k=1}^{K+1} s_{c,k} S(X_{\eta,k}, \overline{X}_{c,k})
\]

(2)

\[
S(VI_\eta, VC_c) = \sum_{t=1}^{T+1} z_{c,t} S(Y_{\eta,t}, \overline{Y}_{c,t})
\]

(3)

where the weights \( s_{c,k}, z_{c,t} \in [0,1], \sum_{k=1}^{K+1} s_{c,k} = 1, \sum_{t=1}^{T+1} z_{c,t} = 1 \). \( S(X_{\eta,k}, \overline{X}_{c,k}) \) is the similarity between the horizontal fuzzy strips \( X_{\eta,k} \) and \( \overline{X}_{c,k} \). \( S(Y_{\eta,t}, \overline{Y}_{c,t}) \) is the similarity between the vertical fuzzy strips \( Y_{\eta,t} \) and \( \overline{Y}_{c,t} \).

The similarity between two fuzzy strips is computed using a weighted Hamming distance as follows:

\[
S(X_{\eta,k}, \overline{X}_{c,k}) = \sum_{t=1}^{T+1} q_{t}(1 - |\overline{h_{\eta,k,t}} - \overline{h_{c,t,k}}|)
\]

(4)

\[
S(Y_{\eta,t}, \overline{Y}_{c,t}) = \sum_{k=1}^{K+1} r_{k}(1 - |\overline{v_{\eta,k}} - \overline{v_{c,t,k}}|)
\]

(5)

where the weights \( q_{t}, r_{k} \in [0,1], \sum_{t=1}^{T+1} q_{t} = 1, \sum_{k=1}^{K+1} r_{k} = 1, \) and \( | \cdot | \) is the absolute value.

At each step in the computation of the similarity \( S_c(I_\eta, C_c) \), specific weights are used. The weights are generally application-dependent and may be chosen according to different criteria for each step. This takes into account the fact that, for different application domains, the various regions of a shape space might have different importance degrees in characterizing the shape itself. We can, therefore, associate appropriate weights with the regions under consideration so as to increase (decrease) the contribution of each region to the computation of the similarity measure.

Typically, we can consider that the contributions of the horizontal and vertical fuzzy representations to the similarity value have the same importance. Thus, in (1), \( w_1 = w_2 = 1/2 \).

In (2), the weights \( s_{c,k} \) are chosen so as to give more importance to the horizontal fuzzy strips \( \overline{X}_{c,k} \) which better characterize the class \( c \). A horizontal fuzzy strip \( \overline{X}_{c,k} \) provides a good characterization of a class \( c \) if it is very similar to the \( k \)th horizontal fuzzy strip of all the instances of class \( c \) in the training set. The more important the contribution of the horizontal strip \( \overline{X}_{c,k} \) in characterizing a shape, the higher the value of \( s_{c,k} \). The weights \( s_{c,k} \) are a measure of the relevance of the horizontal fuzzy strip \( \overline{X}_{c,k} \) with respect to the other horizontal fuzzy strips. Similar consideration can be made for the vertical fuzzy strips.

In our experiments, we used

\[
s_{c,k} = \frac{\sum_{i=1}^{N} S(X_{n,k}, \overline{X}_{c,k})}{\sum_{i=1}^{K+1} \sum_{j=1}^{N} S(X_{n,j}, \overline{X}_{c,j})}
\]

(6)

where \( S(X_{n,k}, \overline{X}_{c,k}) \) is the similarity between the \( k \)th horizontal fuzzy representation \( X_{n,k} \) of the \( n \)th instance of class \( c \) in the training set, and \( \overline{X}_{c,k} \).

The similarity \( S(X_{n,k}, \overline{X}_{c,k}) \) is computed using (4), where the weights \( q_{t} \) are chosen, based on considerations similar to the previous ones, so as to give more importance to the rectangles \( R_{t,k} \) that better contribute to model the shape class \( c \).

In our experiments, we adopted

\[
q_{t} = \frac{\sum_{n=1}^{N} (1 - |\overline{h_{n,t,k}} - \overline{h_{c,t,k}}|)}{\sum_{n=1}^{N} \sum_{j=1}^{K+1} (1 - |\overline{h_{n,j,k}} - \overline{h_{c,j,k}}|)}
\]

(7)

The unknown shape is recognized as belonging to the class \( c \) that produces the largest similarity value.

### III. Optimizing the Partitions

The performance of the method strongly depends on the number and the dimension of the subintervals, which partition the horizontal and vertical spaces, respectively. To optimize these partitions with respect to the recognition rate, we adopt a genetic algorithm (GA) [10]. GAs are search procedures that mimic the principles of natural selection. The potential solutions to a problem are represented as a population of chromosomes (typically, binary or real strings), which evolves using crossover and mutation. Generally, GAs start with a randomly generated initial population. In this paper, a chromosome codifies two real variables \( v_1 \) and \( v_2 \), representing the sequence of the modal values of the fuzzy sets defined on the horizontal and vertical spaces, respectively. Let \( v_1 = m_1^H \ldots m_{K+1}^H \) and \( v_2 = m_1^Y \ldots m_{T+1}^Y \) be the sequences of modal values, then the chromosome is \( ch_1 = m_1^H \ldots m_{K+1}^H, m_1^Y \ldots m_{T+1}^Y \). Both variable \( v_1 \) and \( v_2 \) may have any number of modal values under consideration.
between 2 and an application-dependent value. Consequently, chromosomes may have variable sizes.

Let $\Pi = \{\tau^c\}$ be the set of all the training sets $\tau^c$, where $\tau^c$ is the training set relative to the class $c$. For the sake of simplicity, we assume that each training set $\tau^c$ is composed of the same number $N$ of shape instances. The approach presented here, however, is easily extensible to training sets composed of different numbers of shape instances. The fitness $f(ch)$ of a given chromosome $ch$ is defined as the percentage of shapes in $\Pi$ that are correctly recognized. The fuzzy representations $I_c^h$, $n = 1\ldots N$, of the instances of class $c$ in $\Pi$ are computed using the horizontal and vertical space partitions induced by $ch$. For each class $c$, the fuzzy representation $C_c$ is built according to the procedure described in Section II-B taking the modal values determined by $ch$ into account. The aim of the genetic algorithm is to find the sequences of modal values that maximize the fitness function.

We adopt one point crossover for each variable $v_i$, $i = 1\ldots 2$, separately. More precisely, let

$$ch_A = \prod_{i=1}^{H} m_{A_i}^{H} \cdot \prod_{i=1}^{V} m_{A_i}^{V} \cdot \prod_{i=1}^{K} m_{A_i}^{K}$$

and

$$ch_B = \prod_{i=1}^{H} m_{B_i}^{H} \cdot \prod_{i=1}^{V} m_{B_i}^{V} \cdot \prod_{i=1}^{K} m_{B_i}^{K}$$

be two mating chromosomes. We randomly choose two cut points $h$ and $k$ in $v_1$ and $v_2$, respectively. Let $m_{A_h}^{H}$ and $m_{A_k}^{V}$ be the two modal values corresponding to the two cut points in the chromosome $ch_A$. The offspring $ch'_A$ and $ch'_B$ are the following:

$$ch'_A = \prod_{i=1}^{h-1} m_{A_i}^{H} \cdot \prod_{i=h}^{H} m_{A_i}^{H} \cdot \prod_{i=1}^{k} m_{A_i}^{V} \cdot \prod_{i=k+1}^{V} m_{A_i}^{V} \cdot \prod_{i=1}^{K} m_{A_i}^{K}$$

and

$$ch'_B = \prod_{i=1}^{h-1} m_{B_i}^{H} \cdot \prod_{i=h}^{H} m_{B_i}^{H} \cdot \prod_{i=1}^{k-1} m_{B_i}^{V} \cdot \prod_{i=k}^{V} m_{B_i}^{V} \cdot \prod_{i=1}^{K} m_{B_i}^{K}$$

where $m_{A_h}^{H}$ and $m_{A_k}^{V}$ are the first horizontal and vertical modal values of $ch_B$ greater than $m_{A_h}^{H}$ and $m_{A_k}^{V}$, respectively. If the modal values $m_{A_h}^{H}$ and $m_{B_h}^{H}$ are closer to each other than a fixed threshold value, then they are merged into $m_{A_h}^{H}$. Similar actions are performed for the pairs $m_{A_l}^{V}$ and $m_{B_l}^{V}$, $m_{A_h}^{H}$, and $m_{B_h}^{H}$, $m_{A_l}^{V}$, and $m_{B_l}^{V}$, respectively. Note that, in this way, the correct ordering of the modal values is maintained in both variables of the chromosomes.

The mutation operator randomly selects two modal values $m_h^V$ and $m_l^V$. Then

$$m_h^V$$

and

$$m_l^V$$

are transformed into

$$m_h^V = \frac{m_h^{V-1} + m_h^{V+1}}{2}$$

and

$$m_l^V = \frac{m_l^{V-1} + m_l^{V+1}}{2}$$

respectively.

We start with a randomly initialized population, typically consisting of 50 chromosomes. For each chromosome $j$, $j = 1\ldots 50$, first we generate the random numbers $T_j \in 2\ldots \bar{T}$ and $K_j \in 2\ldots \bar{K}$ with $\bar{T}$ and $\bar{K}$ application-dependent values. So the variables $v_1$ and $v_2$, contain, respectively, $T_j$ and $K_j$ randomly generated genes that represent modal values. At each generation, chromosomes are selected for reproduction with a probability directly proportional to their fitness value. More precisely, an intermediate population is generated by applying the stochastic sampling with replacement method [10]. The probabilities of crossover and mutation are 0.9 and $10^{-2}$, respectively.

The values of the GA parameters were chosen based on experimental observations. Regarding the population, we tried several sizes in the range 20–60. The upper limit was determined by considering the complexity involved in computing the fitness function. The best population size out of those tried appeared to be 50. For each population size, some different values for both crossover and mutation probabilities were used. The probabilities of crossover and mutation were held fixed during each run. From experimental observations we noticed that higher values (close to 1) for crossover probability gave better results. On the other hand, it appeared that the mutation probability could be kept much lower than the crossover probability. Notice that the mutation operator actually “perturbs” the set of modal values, represented by a chromosome, by replacing a modal value (of a certain fuzzy set) on each coordinate axis with a new modal value for the same fuzzy set.

Only 60% of the new population is composed of offspring, whereas 40% consists of the best chromosomes of the previous population. This acceptance mechanism gets a twofold result. First of all, accepting offspring (possibly) worse than their parents reduces the risk of premature convergence of the algorithm on local maxima. Second, transplanting the best individuals of the previous generation to the new generation guarantees that the results achieved so far in the evolution process are not lost. When the average of the fitness values of all the individuals in the population is greater than 97% of the fitness value of the best individual, or when a predetermined number of generations is reached, the GA is considered to have converged.

### IV. RECOGNITION OF Olfactory SIGNALS

In the last few years, a growing interest in mimicking the mammalian olfactory system has arisen with the aim of developing a computer-based system that is able to identify odorant samples and perhaps to measure odor intensity and nature, e.g., sweet, pungent, minty, tainted, etc. Such a system is called an electronic nose [11], [12].

Typically, electronic noses integrate an array of a few sensors with partially overlapping sensitivities to odors, and a pattern recognition system [13]. The physical and chemical reactions produced by the sensors when stimulated by an odorant are appropriately transduced into electrical signals. Each sensor responds differently to different odorants, and therefore, the output patterns from the sensor array can be used by the pattern recognition system for discriminating different kinds of odorants [14].
The encouraging results obtained by using nonlinear techniques, such as artificial neural networks \[15\], \[16\] and fuzzy neural networks \[17\], to recognize olfactory signals stimulated us to experiment our method in this field. Of course, signals are not really 2-D shapes. This is taken into account by considering shape models consisting of vertical fuzzy strips only. The method was employed to classify signals produced by an array of \(Q\) conducting polymer sensors \[18\]. The sensors were exposed to different odorants and the percentage change in resistance in output from each sensor was recorded for classification purposes. Fig. 3 illustrates the responses of one sensor to a given odorant in three repeated experiments. It can be seen that the signals are noisy and that there is a strong drift in their amplitudes. Nevertheless, the general shape of the signals does not change and can be used as the main classification feature. Therefore, our method seems to be particularly suited to this problem.

Typically, noisy signals are preprocessed for noise reduction and/or filtering. But having a closer look at the noise that affects the sensor responses, two types of noise can be identified: chemical and electronic. The former derives from the chemical reactions of the polymers, the latter is generated by the electronic devices that transform these reactions into digital signals. Of course, electronic noise can be eliminated by using appropriate electronic devices. Regarding chemical noise, instead, one may suppose that this noise is, in a sense, useful information and could be profitably exploited in modeling the sensor responses. So, we decided to model signal shape without any previous noise filtering.

**A. Application of the Method**

Let us consider a single response of sensor \(s, s = 1 \ldots Q\), to odorant \(o, o = 1 \ldots E\), as shown in Fig. 4. The response is transformed into an instance fuzzy representation \(I_{s,o}\) as described in Section II-A. Here, the horizontal and vertical spaces denote, respectively, the discrete interval during which samples are taken and the real interval to which the amplitude values of the signal belong. For each signal, the size of the signal space is chosen to be equal to the dynamic range of the signal itself.

As previously stated, we computed the instance fuzzy representation as \(I_{s,o} = V I_{s,o}\). Horizontal strips are not taken into account just because we only have one shape point on each vertical line. The fuzzy representation of the response of the sensor array to the odorant \(o\) can be expressed as

\[
A_o = \{I_{1,o}, \ldots, I_{Q,o}\}
\]

We call \(A_o\) the array fuzzy representation.

For each pair sensor/odorant \((s, o)\), \(s = 1 \ldots Q, o = 1 \ldots E\), we build a class fuzzy representation \(C_{s,o}\) as described in Section II-B. The class fuzzy representation is obtained by considering the responses of the sensor \(s\) to the odorant \(o\) in \(N\) repeated experiments (training set). The horizontal and vertical spaces of all these responses are divided, respectively, into the same numbers \(T\) and \(K\) of subintervals. The optimal position and size of these subintervals are determined using the genetic algorithm described in Section III. The generic response \(O_o\) of the electronic nose to the odorant \(o\) is then defined by the class fuzzy representations relative to the \(Q\) sensors of the sensor array, i.e., \(O_o = \{C_{1,o}, \ldots, C_{Q,o}\}\). \(O_o\) is called odorant fuzzy representation.

When an unidentified odorant \(u\) is presented, the fuzzy representation of the response of the sensor array is generated. More precisely, we calculate the array fuzzy representation \(A_u = \{I_{1,u}, \ldots, I_{Q,u}\}\), where each \(I_{s,u}, s = 1 \ldots Q\), uses the same horizontal and vertical space partitions adopted in building the odorant fuzzy representations \(O_o, o = 1 \ldots O\). Then, \(A_u\) is compared with each odorant fuzzy representation \(O_o\) to produce the similarity value \(S_o(A_u, O_o)\). For each odorant \(o\), the similarity value between \(A_u\) and \(O_o\) is

\[
S_o(A_u, O_o) = \sum_{s=1}^{Q} b_{s,o} S(I_{s,u}, C_{s,o})
\]

where the weights \(b_{s,o} \in [0, 1]\), \(\sum_{s=1}^{Q} b_{s,o} = 1\), and \(S(I_{s,u}, C_{s,o})\) is the similarity between the instance fuzzy representation \(I_{s,u}\) and the class fuzzy representation \(C_{s,o}\). \(S(I_{s,u}, C_{s,o})\) is computed using (1) with \(w_1 = 0\) as the horizontal fuzzy representations are not considered.

The weight \(b_{s,o}\) takes into account the confidence of the sensor \(s\) in recognizing the odorant \(o\). In general, the closer the signals generated by a sensor in repeated experiments with the same odorant, the better the sensor (with respect to that odorant). A measure of confidence of sensor \(s\) with respect to odorant \(o\) can be obtained by evaluating the similarity between
the class fuzzy representation \( C_{s,o} \) and each instance fuzzy representation \( I_{s,n}, n = 1 \ldots N \), of the responses to odorant \( o \) in the training set. The more the instance fuzzy representations \( I_{s,n} \) are similar to the class fuzzy representation \( C_{s,o} \), the more the sensor \( s \) is important in characterizing odorant \( o \). Weights \( b_{s,o} \) are therefore computed as

\[
b_{s,o} = \frac{\sum_{n=1}^{N} S(I_{s,n}, C_{s,o})}{\sum_{i=1}^{Q} \sum_{n=1}^{N} S(I_{s,n}, C_{i,o})}.
\]

When an unknown odorant is “sniffed,” it is recognized as the odorant \( o \) associated with the highest similarity value.

**B. An Example**

We have applied our method to an environmental problem: to recognize three different dilutions of three different types of waste waters. This was one of the targets of the European Communities Industrial RTD Project 25 254–INTESA. Samples were collected from the Italian ACQUARNO plant, one of the greatest in Europe, which treats waste waters produced by industrial activities of leather tanneries or by municipal sewers. The samples come from three basins, containing, respectively, untreated industrial waste waters (IND), partially treated industrial waste waters (PTI), and untreated municipal waste waters (MUN). In the experiments, the analysis of the headspace formed over liquid samples was preferred to the analysis of air samples because of the higher concentrations of gases and vapors. To simulate the collection of air samples at different distances from the odor source, one third of the samples collected from the three basins in ACQUARNO were diluted two times by adding 25 cm\(^3\) of pure water to 25 cm\(^3\) of waste water. Further, half of the remaining samples were diluted five times by adding 40 cm\(^3\) of pure water to 10 cm\(^3\) of waste water.

The prototype of the electronic nose includes eight conducting polymer sensors. The experimental protocol consists of three phases: sensor stabilization, absorption, and desorption of the odorant. The sampling frequency was 2 Hz. The samples in the three phases were 100, 400, and 200, respectively. The experiments were repeated 20 times for each dilution. The testing was performed using 10-fold cross validation. The initial data set was segmented to provide a training and a test set of 162 and 18 patterns, respectively. While the training set was generated by taking 18 patterns of each dilution, the test set was created by taking the remaining two patterns of each dilution. Each pattern consists of eight sensor responses composed of 700 samples. The patterns composing the training and test sets were rotated so that in each fold we had a different training and test set.

For each fold, we computed the optimal number and position of the modal values in the horizontal and vertical spaces by applying the genetic algorithm described in Section III. To speed up the computation, we adopted the following two solutions: signal compression and reduction of the training set. We compressed the signals using different compression factors. We observed that the recognition rate does not sensibly degrade with the increasing of the compression factor up to 10. So, in our experiments, we used compressed signals with compression factor equal to 10. Then, we reduced the number of instances in the training set. The reduction process is based on the following consideration: sensor responses vary over a long time period due to such causes as environmental changes, poisoning, or aging of the sensors. Anyway, sensor responses to the same odorant remain quite similar over a short time period. Therefore, we sorted the training instances according to their acquisition time and for each class we considered alternate instances.

We started with a randomly initialized population consisting of 50 chromosomes. The probabilities of crossover and mutation were 0.9 and 10\(^{-2}\), respectively. Further, along the horizontal space, modal values at a distance of less than four time samples were merged. Finally, we fixed the variation ranges for \( T \) and \( K \) to \([5 \ldots 18]\) and \([5 \ldots 20]\), respectively. These ranges for \( T \) and \( K \) are extracted from previous work of the same authors in which uniform partitions of the horizontal and vertical spaces were used [19].

The algorithm terminates either when the average of the fitness values of all the individuals in the population is greater than 97% of the fitness value of the best individual, or after 100 generations. We experimentally verified that after 100 generations the chromosome with the highest fitness is generally a good approximation of the optimal solution.

The average execution time to build an odorant fuzzy representation was 1.05 s on a 266-MHz Pentium II PC with a Microsoft Windows NT operating system. The average execution time needed to build the array fuzzy representation of the unknown odorant, and to compute the similarity values between this representation and the odorant fuzzy representations was 0.04 s. The average success rate and the standard deviation in the training and test set were 0.93 and 0.014, and 0.87 and 0.15, respectively. This result is comparable with results obtained in similar application domains (see, for instance [17], where the use of fuzzy neural networks for the classification of six different tainted waters achieved 85% recognition rate).

Table I shows the results obtained by applying the method to the 10 folds. In the table, the second and third columns represent the number of recognized waste waters in the training and test sets, respectively.

<table>
<thead>
<tr>
<th>Table</th>
<th>Dilution</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>D0</td>
<td>D1</td>
<td>D2</td>
</tr>
</tbody>
</table>
| Success | 92 | 91 | 93 | 90 | 94 | 92 | 91 | 93 | 90 | 94 | 92 | 92.0%
| Avg. Error | 8 | 9 | 7 | 10 | 8 | 5 | 11 | 7 | 12 | 6 | 9 | 9.0%

Note that most misclassifications involve different waste waters at the same dilution. This indicates that the electronic nose is particularly sensitive to the intensity of the odorants, although it
TABLE I
RESULTS OF ANALYZING THE ODORANT DATA. 162 AND 18 PATTERNS WERE USED FOR THE TRAINING AND TESTING PHASE, RESPECTIVELY

<table>
<thead>
<tr>
<th>Fold</th>
<th>Patterns recognized in the Training Set</th>
<th>Patterns recognized in the Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>148</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>154</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>151</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>152</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>151</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>152</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>155</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>148</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>149</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>151</td>
<td>17</td>
</tr>
<tr>
<td>Average</td>
<td>151.1</td>
<td>15.7</td>
</tr>
</tbody>
</table>

TABLE II
AVERAGE SUCCESS PERCENTAGE FOR EACH DILUTION IN THE TRAINING SET

<table>
<thead>
<tr>
<th></th>
<th>D0</th>
<th>D2</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND</td>
<td>95%</td>
<td>91.1%</td>
<td>80.5%</td>
</tr>
<tr>
<td>PTI</td>
<td>95%</td>
<td>95%</td>
<td>93.9%</td>
</tr>
<tr>
<td>MUN</td>
<td>94.4%</td>
<td>95%</td>
<td>99.4%</td>
</tr>
</tbody>
</table>

TABLE III
AVERAGE SUCCESS PERCENTAGE FOR EACH DILUTION IN THE TEST SET

<table>
<thead>
<tr>
<th></th>
<th>D0</th>
<th>D2</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND</td>
<td>95%</td>
<td>70%</td>
<td>75%</td>
</tr>
<tr>
<td>PTI</td>
<td>95%</td>
<td>90%</td>
<td>85%</td>
</tr>
<tr>
<td>MUN</td>
<td>95%</td>
<td>90%</td>
<td>90%</td>
</tr>
</tbody>
</table>

is however able to discriminate between different types of odorants. Furthermore, it is particularly interesting to note that the highest values of misclassification are due to IND D2 and IND D5, which are incorrectly classified as PTI D2 and PTI D5, respectively. Since PTI D2 and PTI D5 are obtained by partially treating IND D2 and IND D5, respectively, the high misclassification value seems to be caused by a substance (contained in IND D2 and IND D5), which is only partially modified during the treating phase. Finally, we can observe that, at higher dilutions, the signals corresponding to IND and PTI are spread over a wider area, whereas those corresponding to MUN are much closer to each other. In particular, IND tends to become much more similar to PTI, PTI tends to become more similar to MUN, whereas MUN tends to lower its similarity with PTI (see Tables IV and V). This helps explain why in Table II dilution decreases recognition for IND and PTI, but increases recognition for MUN. Similarly, in Table III, dilution decreases recognition significantly for IND, a lot for PTI, a little for MUN.

A further interesting use of our method in the field of electronic noses is sensor selection. The weight associated with each sensor during the training phase provides a useful indication of the importance of that sensor in recognition. Intuitively, the closer the signals generated by a sensor in repeated experiments with the same odorant, the better the sensor (with respect to that odorant). Fig. 5 graphically shows the weight $b_{s,o}$ associated with each sensor $s$, $s = 1 \ldots 8$, in one of the experiments carried out during the 10-fold cross validation. It can be observed that the importance of a sensor depends on the specific odorant being considered. For instance, sensor 5 seems to be more effective in recognizing IND D2 than sensor 6. Further, sensor 5 performs better in recognizing IND D2 than in recognizing IND D5.

V. RECOGNITION OF HANDWRITTEN CHARACTERS

Handwriting recognition is undoubtedly one of the most challenging areas of pattern recognition. It is extremely useful in a wide range of real-world practical problems, including document analysis, mailing address interpretation, bank check processing, signature verification, document verification, and many others [20].

Several pattern recognition approaches have been applied to both on-line and off-line handwriting recognition, including statistical methods, structural and syntactic methods, and neural networks. Some reading systems identify strokes, others try to identify characters, groups of characters, or entire words [21]. Recently, as an alternative to traditional handwriting recognition systems, there has been a growing interest in approaches to handwriting recognition based upon soft decisions, for a more faithful emulation of the human reading process [22], [23]. It is in this context that fuzzy logic and fuzzy sets have proved to be powerful tools to represent vague, uncertain, or imprecise character patterns [24]–[26].

The field of automatic recognition of handwritten isolated characters is the longest established branch of handwriting recognition. Although a large variety of writing styles exists, the general shape of characters can be described by fuzzy representations. Thus, we considered the recognition of isolated handwritten characters as an interesting test for our fuzzy method. We focused on lower-case characters and used a partition of the NIST Special Database 19 (SD19).

A. Application of the Method

The input to the fuzzy recognizer is a B/W bit-map large enough to contain each character. The characters are assumed to be “centered” within the bitmaps. This means that, for each character class $c$, a key point in the character body (i.e., in the part of the character not containing ascenders or descenders) should be identified. We call this key point the “body center” of the character. The body center does not necessarily coincide with the geometrical center of the character, but with a given position, unique to each character. The body center can be used as a reference point for positioning all instances of $c$ inside their bit-maps. All body centers are assumed to coincide with the bit-map center.

Fig. 6 shows how the bit-map containing the character is partitioned. For each lower-case character $c$, $c \in \{a, \ldots, z\}$, we
build a class fuzzy representation \( C_c \) called character fuzzy representation, as described in Section II-B.

### B. An Example

We applied our method to the NIST database hsf_4 (containing the segmented handprinted characters of 500 writers). hsf_4 is a partition of the NIST SD19.

The input to the fuzzy recognizer is a 300 dpi B/W bit-map. Actually, the original NIST 128 \( \times \) 128 bit-maps are not “centered” in the sense explained above. Therefore, before applying our method, first character body centers were roughly identified by visually inspecting the NIST bit-maps, then, a 64 \( \times \) 82 bit-map around each center was automatically extracted from the NIST bit-maps. The reduced bit-map is large enough to contain each character.

![Sensor Weights](image)

**Fig. 5.** Sensor selection in one of the experiments carried out during the 10-fold cross validation.

---

### TABLE IV
**Total Misclassifications in the 10 Repeated Experiments for the Training Set**

<table>
<thead>
<tr>
<th></th>
<th>IND D0</th>
<th>IND D2</th>
<th>IND D5</th>
<th>PTI D0</th>
<th>PTI D2</th>
<th>PTI D5</th>
<th>MUN D0</th>
<th>MUN D2</th>
<th>MUN D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND D0</td>
<td>-</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IND D2</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IND D5</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D2</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>MUN D0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MUN D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>MUN D5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

### TABLE V
**Total Misclassifications in the 10 Repeated Experiments for the Test Set**

<table>
<thead>
<tr>
<th></th>
<th>IND D0</th>
<th>IND D2</th>
<th>IND D5</th>
<th>PTI D0</th>
<th>PTI D2</th>
<th>PTI D5</th>
<th>MUN D0</th>
<th>MUN D2</th>
<th>MUN D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND D0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IND D2</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>IND D5</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PTI D5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MUN D0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MUN D2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MUN D5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
We performed 10-fold cross validation, i.e., the handprinted lower-case letters of the 500 writers were partitioned into blocks from 50 writers. The characters of 90% of the writers were used as training set and the characters of the remaining 10% as test set. The alphabets composing the training and test sets were randomly chosen. We repeated the experiment 10 times.

For each fold, we computed the optimal horizontal and vertical partitions. We applied the genetic algorithm described in Section III. To speed up the computation, we reduced the training set by randomly selecting 50 instances for each character. We started with a randomly initialized population consisting of 50 chromosomes. The probabilities of crossover and mutation were 0.9 and $10^{-2}$, respectively. Modal values at a distance of less than three pixels were merged. Finally, we fixed the variation ranges for $T$ and $K$ to $[5 \ldots 12]$ and $[5 \ldots 15]$, respectively. These ranges for $T$ and $K$ are extracted from a previous work of the same authors in which uniform partitions of the horizontal and vertical spaces were used [27]. The algorithm terminates either when the average of the fitness values of all the individuals in the population is greater than 97% of the fitness value of the best individual, or after 100 generations.

Table VI shows the results obtained by applying our method to the 10 folds. In the table, the second and third columns represent the number of recognized characters in the training and test sets, respectively. The average success rate and the standard deviation in the training and test sets were 79.57 and 0.97%, and 75.86 and 1.02%, respectively. Table VI shows that the recognition rate does not depend on the particular fold being considered.

Comparing our results to those obtained in [28], it can be noted that we achieved a recognition rate at the character level slightly lower than that obtained in [28] for the same experiment on the same database. In [28], a 10-fold cross validation of the NIST database hsf_4 achieves 19.2% recognition error rate for lower-case letters. The reason for the slightly worse results of our method stems from the fact that no application-specific improvements were taken into account. For instance, we adopted only one fuzzy model for each character, we performed the character centering empirically, we chose to keep the complexity of the GA low by a heavy reduction of the training set. On the other hand, the main purpose of this application was to show the basic characteristics of our method. Our results, however, are comparable to results obtained in similar application domains by applying soft computing techniques (see, for instance, [29]).

The average execution time to build a character fuzzy representation (considering all the 450 writers in the training set) was 20.2 s on a 266-MHz Pentium II PC with a Microsoft Windows NT operating system. The average execution time in the recognition phase was 0.04 s, i.e., 0.04 s were required to build the instance fuzzy representation of the unknown character and to compare this representation with all the character fuzzy representations constructed in the training phase. Table VII shows the average success percentage for each character in the training and test sets.

From Table VII, it can be seen that different results are obtained depending on the character. Actually, our system partitions the 2-D shape space with fuzzy sets, then computes the similarity measure between a shape instance and a shape model using weighted averages of importance values associated with rectangles $R_{i,k}$. This means that two characters, though different, may result to be “similar” in terms of these weighted averages. This is the case, e.g., for characters “e” and “c,” “g” and “q,” etc. However, the groups of characters mistaken one for another mostly coincide with characters that cannot be distinguished by human readers either. In fact, acceptable recognition performance for a given application can only be achieved by using contextual information, such as dictionary, syntactic and grammar rules [30].

Finally, we recall that our method was adopted to implement the character recognizer within a system for automatic recognition of handwritten text. It is well known that when a specific character recognizer is intended for use within a higher-level system, e.g., a word recognizer or a sentence recognizer, the character recognition rate might not be a relevant performance index [2], [31]. In such a case, it becomes crucial that the char-

![Fig. 6. Fuzzy partition of the horizontal and vertical spaces.](image)

**Table VI**

<table>
<thead>
<tr>
<th>Fold</th>
<th>Characters recognized in the Training Set</th>
<th>Characters recognized in the Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78.27%</td>
<td>75.50%</td>
</tr>
<tr>
<td>2</td>
<td>78.86%</td>
<td>76.58%</td>
</tr>
<tr>
<td>3</td>
<td>79.38%</td>
<td>77.38%</td>
</tr>
<tr>
<td>4</td>
<td>79.20%</td>
<td>77.43%</td>
</tr>
<tr>
<td>5</td>
<td>81.19%</td>
<td>75.13%</td>
</tr>
<tr>
<td>6</td>
<td>79.07%</td>
<td>75.17%</td>
</tr>
<tr>
<td>7</td>
<td>79.93%</td>
<td>74.52%</td>
</tr>
<tr>
<td>8</td>
<td>80.96%</td>
<td>74.96%</td>
</tr>
<tr>
<td>9</td>
<td>80.14%</td>
<td>75.55%</td>
</tr>
<tr>
<td>10</td>
<td>78.67%</td>
<td>76.37%</td>
</tr>
<tr>
<td>Average</td>
<td>79.57%</td>
<td>75.86%</td>
</tr>
</tbody>
</table>


TABLE VII
THE AVERAGE SUCCESS PERCENTAGE FOR EACH CHARACTER IN THE TRAINING AND TEST SETS

<table>
<thead>
<tr>
<th>Character</th>
<th>Average success percentage in training set</th>
<th>Average success percentage in test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>80.27%</td>
<td>77.96%</td>
</tr>
<tr>
<td>b</td>
<td>82.02%</td>
<td>80.26%</td>
</tr>
<tr>
<td>c</td>
<td>79.26%</td>
<td>76.11%</td>
</tr>
<tr>
<td>d</td>
<td>80.79%</td>
<td>79.73%</td>
</tr>
<tr>
<td>e</td>
<td>72.33%</td>
<td>64.15%</td>
</tr>
<tr>
<td>f</td>
<td>81.58%</td>
<td>79.27%</td>
</tr>
<tr>
<td>g</td>
<td>72.06%</td>
<td>63.84%</td>
</tr>
<tr>
<td>h</td>
<td>68.45%</td>
<td>65.28%</td>
</tr>
<tr>
<td>i</td>
<td>83.55%</td>
<td>79.40%</td>
</tr>
<tr>
<td>j</td>
<td>82.18%</td>
<td>79.86%</td>
</tr>
<tr>
<td>k</td>
<td>83.32%</td>
<td>80.51%</td>
</tr>
<tr>
<td>l</td>
<td>78.00%</td>
<td>73.25%</td>
</tr>
<tr>
<td>m</td>
<td>89.36%</td>
<td>82.11%</td>
</tr>
<tr>
<td>n</td>
<td>77.54%</td>
<td>73.91%</td>
</tr>
<tr>
<td>o</td>
<td>88.82%</td>
<td>83.92%</td>
</tr>
<tr>
<td>p</td>
<td>76.84%</td>
<td>74.22%</td>
</tr>
<tr>
<td>q</td>
<td>67.51%</td>
<td>64.32%</td>
</tr>
<tr>
<td>r</td>
<td>87.51%</td>
<td>80.04%</td>
</tr>
<tr>
<td>s</td>
<td>79.65%</td>
<td>77.17%</td>
</tr>
<tr>
<td>t</td>
<td>73.76%</td>
<td>68.89%</td>
</tr>
<tr>
<td>u</td>
<td>76.96%</td>
<td>75.79%</td>
</tr>
<tr>
<td>v</td>
<td>82.27%</td>
<td>77.86%</td>
</tr>
<tr>
<td>w</td>
<td>81.24%</td>
<td>80.00%</td>
</tr>
<tr>
<td>x</td>
<td>81.21%</td>
<td>76.27%</td>
</tr>
<tr>
<td>y</td>
<td>83.29%</td>
<td>80.62%</td>
</tr>
<tr>
<td>z</td>
<td>79.01%</td>
<td>77.56%</td>
</tr>
</tbody>
</table>
method, as an example of a new fuzzy recognition technology; ii) to show the strength of our method as a modeling tool, which is able to faithfully reproduce the pattern recognition process of human brain.

We are currently working to enhance our method to classify and recognize shapes in two steps. The first step is that described in the paper. In the second step, we refine the classification by building a new fuzzy partition in specific regions of the shape space. This is equivalent to say that first we build a coarse fuzzy partition on key regions of the shape; this allows us to distinguish between classes of similar shapes. Then we build one or more finer-grain fuzzy partitions on key regions of the shape; this allows us to distinguish between similar shapes belonging to the same class.

REFERENCES


Beatrice Lazzerini is a Full Professor at the Faculty of Engineering of the University of Pisa, Italy. She teaches Knowledge Engineering and Expert Systems. Her main research interests lie in the area of knowledge engineering, with particular emphasis on fuzzy systems, neural networks, and evolutionary computation. She has co-authored six books and has published more than 80 papers in international journals and conferences. She is co-editor of two books. She serves as an Associate Editor of the International Journal of Knowledge-Based Intelligent Engineering Systems.

Francesco Marcelloni received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the University of Pisa, Pisa, Italy, in 1991 and 1996, respectively. His Ph.D. dissertation dealt with object-oriented models and fuzzy-logic based methods in software development. Currently, he is an Assistant Professor with the Faculty of Engineering of the University of Pisa. His current research interests include object-oriented software development, object-oriented models, approximate reasoning, fuzzy rule-based systems, and pattern recognition. He is an author or coauthor of more than 40 papers in international journals and conferences.