Potential Improvement of Classifier Accuracy by Using Fuzzy Measures

Venu Govindaraju, *Senior Member, IEEE,* and Krassimir Ianakiev

***Abstract*—Typical digit recognizers classify an unknown digit pattern by computing its distance from the cluster centers in a feature space. The K-nearest neighbor (KNN) rule assigns the unknown pattern to the class belonging to the majority of its K neighbors. These and other traditional methods adopt a uniform rule irrespective of the "difficulty" of the unknown pattern. In this paper, we propose a methodology that has many salient aspects. First, the classification rule is dependent on the "difficulty" of the unknown sample. Samples "far" from the center, which tend to fall on the boundaries of classes are error prone and, hence, "difficult." An "overlapping zone" is defined in the feature space to identify such difficult samples. A table is precomputed to facilitate an efficient lookup of the class corresponding to all the points in the overlapping zone. The lookup function itself is defined by a modification of the KNN rule. By reassigning the error-prone samples, we actually redefine the class boundaries. A characteristic function defining the new boundaries is computed using the topology of the set of samples in the overlapping zones. Our two-pronged approach uses different classification schemes with the "difficult" and "easy" samples. This could enable potential improvement of a recognition system. We have tested this methodology on a large set (30 398) of handwritten digit images. The method described in this paper has improved the performance of the gradient structural concavity (GSC) digit recognizer desribed in [8]. The GSC is among the best digit recognizers described in the literature. The method described in this paper successfully reduced its error rate from 2.85% to 1.96%, i.e., by 0.89%, which is more than 30% of the initial error. We have tested our method on other available classifiers and have obtained similar results.**

***Index Terms*—Classifier, digit recognition, fuzzy measure, topology.**

I. INTRODUCTION

PATTERN classification consists of assigning a class label to a set of unclassified patterns. Classification typically involves two steps. First, the system is trained on a set of data (already labeled prototype patterns) and then this data set is used to classify a new set of unclassified patterns. When all possible classes are known in advance and the system is trained on a set of already classified patterns, the problem is known as supervised classification. Formally, given the pattern x (taken from the space of all unknown patterns X), a set of c class labels ($\omega_1 \ldots \omega_c$) and a set of labeled prototypes, the classifier assigns a label to x such that there is a minimum probability of misclassification. For the purpose of this paper, we will confine our discussions to the handwritten digit recognition problem where the number of classes is ten.

Manuscript received January 10, 2000; revised June 30, 2000.

The authors are with the Center of Excellence for Document Analysis and Recognition (CEDAR), Department of Computer Science, State University of New York at Buffalo, Amherst, NY 14228 USA (e-mail: govind@cedar.buffalo.edu).

Publisher Item Identifier S 1063-6706(00)10685-X.

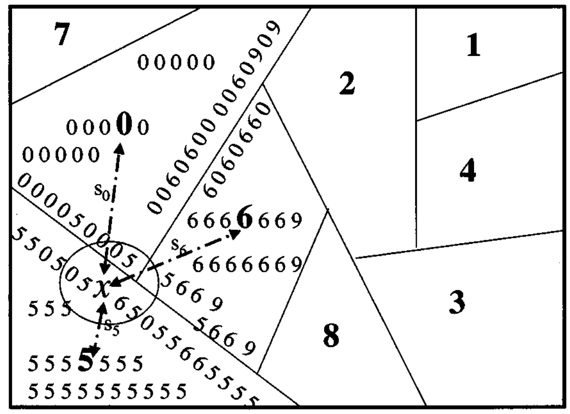

Fig. 1. Feature space of a typical digit recognizer. Samples that are close to the center of a class are strong representatives of the class and are usually correctly classified. Samples that are away from the center (close to the boundaries of classes) are weaker representatives and are prone to misclassification. The circle is essentially the sphere of the neighborhood of x. Typically, the confidence $\phi_i(x)$ is inversely proportional to the distance (s_i) of x from the centroid of cluster ω_i.

A typical digit recognizer returns scores corresponding to each class. The scores reflect the degree to which the unknown pattern x represents the prototype(s) of the class. The scores can be represented by the vector $\Phi(x)$, where $\Phi(x) = [\phi_0(x), \phi_1(x), \ldots, \phi_9(x)]^T$. Each $\phi_i(x)$ corresponds to class ω_i. $\phi_i : X \rightarrow [0, 1]$ is a function that maps each unknown pattern to a real number between zero and one. The performance of the digit recognizer is measured by relative frequency with which it assigns unknown patterns to their correct classes. In other words, the accuracy is the ratio of the number of correctly classified patterns and the number of all classified patterns. The higher the accuracy, more desirable is the recognizer in practical applications.

A. Issue with Traditional Techniques

Traditional classification techniques behave as follows. Given an input pattern x, a classifier typically computes the distance of x from the centers of each of the prototype classes in the feature space using some distance metric. The classes are then ranked in increasing order of the distances. In Fig. 1, x is "closest" to the center of the class ω_5 with a distance of s_5. The second and third classes are ω_6 and ω_0 at distances s_6 and s_0 respectively. It is to be noted that usually $\phi_i(x)$ is inversely proportional to s_i.

It is possible that the test pattern x falls within the boundary of one class but its distance is closest to the center of a neighboring class. Since x, in the example, falls close to the boundary of classes ω_5-ω_0 and ω_5-ω_6 there is more than an average chance of misclassification of x.

Fig. 1 illustrates the class boundaries among digit prototypes in a hypothetical feature space. We will assume for the purpose of this discussion that the classes are linearly separable. Samples that are "close" to the centroid of the classes are "stronger" representatives of the class while samples close to the boundaries are poorer representatives and at the same time prone to confusion with the neighboring classes.

So, it is natural for one to consider another level on top of the classifier in order to deal with the patterns that are more prone to misclassification. There are many different techniques that are applicable to this problem.

Using the K-nearest neighbor (KNN) rule (see [5] and [7]) is one way to address the problem stated above. Instead of simply relying on the distance of a sample from cluster centers, one labels x as the majority class in the set of its k nearest neighbors in feature space.

KNN has the advantage of not respecting fixed boundaries between classes. In the example of Fig. 1, the KNN classifier would label x as "5" with a score of $\frac{3}{6} = 0.5$, as four of the samples in the K-neighborhood ($k = 6$) are labeled ω_5 by the recognizer (some erroneously).

Since potentially, a test pattern can fall anywhere in a feature space, we can precompute the class that will be assigned to each point in the feature space. If the KNN rule is used, the boundaries between classes will have a complex shape. Further, the boundaries of classes will change based on the size of the neighborhood k.

Computation of such boundaries can be done ahead of time. For example applying Voronoi diagram to partition the space will require $O(k + \log n)$ time and $O(k^2 n)$ space, where n is the number of prototypes and k is the number of neighbors considered. When a test pattern falls at a particular point in the feature space a table lookup can determine its class. The table can prestore the class assignments for all points in feature space for various values of k.

A typical binary feature vector of size 512 requires a feature space of size 2^{512}. A table that specifies a class for each point in the feature space will need to be 2^{512} as well. Hence, while the idea is attractive for speed reasons, the tradeoff with space storage is unacceptable.

Using Bayes rule ([16]) is another way to deal with the problem, especially in the cases when a large database is available. This method is very attractive in view of recent empirical evaluations showing high accuracy ([9], [12]). Unfortunately, the learning efficiency suffers when the database is incomplete.

Another alternative is to employ a neural network pattern recognition system, such as a Fuzzy ARTMAP ([6]), which synthesizes fuzzy logic and adaptive resonance Theory by exploiting the formal similarity between the computation of fuzzy subsets and the dynamics of categoty choices, search, and learning. There are, however, some limitations to this approach: 1) it is order dependent; 2) it is severely affected by noise points; 3) it is time consuming; and 4) it lacks an objective function and stability.

The literature also describes some fuzzy K-nearest neighbor algorithms. An approach different from the one presented in this paper has been described by Keller *et al.* [11], [13]. However, there is one critical difference in the application of the two methods. Unlike our approach, Keller *et al.*requires information from the internal working of the classifiers, whereas our method treads all classifiers as black boxes.

II. Our Approach

We describe a method that is based on the following two ideas that appeal to common sense [4].

1) When the correctness of a classifier on a pattern x is in question, it is best to consider the performance of the same classifier on the patterns which are "similar" to x.
2) A classifier is usually accurate when the test pattern x falls close to the center of its class in feature space and prone to error when it falls near a class boundary.

KNN addresses the first observation quite adequately by evaluating the neighbors of a test pattern x. It is expected that in any small neighborhood all samples are approximately at the same distance from any cluster center and, hence, are all quite "similar." While, KNN does not oppose the second observation, we have described how it does not lend itself amenably to space storage efficiency when using a lookup table.

We propose the following two-pronged approach. For the case where a sample falls in "close" proximity of a cluster center (of a class to which it will be assigned), use the traditional method of distances to assign the class and compute its confidence score. For the case where the distances from several cluster centers are all approximately the same, we will adopt a modification of the KNN rule. Such an approach addresses both the observations made above and, at the same time, does not cause prohibitive computational issues. Since we expect only a small portion of the feature space to be near the boundaries between classes, the size of the lookup table will be reasonable.

A. Definitions

The main purpose of this paper is to describe a new approach for dealing with the patterns from the "overlapping" area. To this end we wish to consider how one defines the terms "class center," "close proximity," and "overlapping area."

There are many algorithms available for clustering given a set of prototypes and there are many different approaches for identifying the "overlapping" areas ([1], [7], [10], [14]). In this paper, we will choose simple and intuitive ways to describe these terms. Assuming that s_i is the maximum possible score corresponding to the class $\omega_i, i = 0, \ldots, 9$, the "class center" of class ω_A is defined by all prototypes y, such that $\phi_A(y) \geq \frac{3}{4} s_A$ and $\phi_B(y) \leq \frac{1}{10} s_B$ for all other classes ω_B. The "overlapping" area is defined as the part of the space of all patterns X, that contains patterns $y \in X$, such that $|\phi_A(y) - \phi_B(y)| \leq (s_A + s_B)/2$, where classes ω_A and ω_B are the top two choices for pattern y. Let us note that in some cases as in the example of Fig. 2, the "overlapping" area can be narrowed even further.

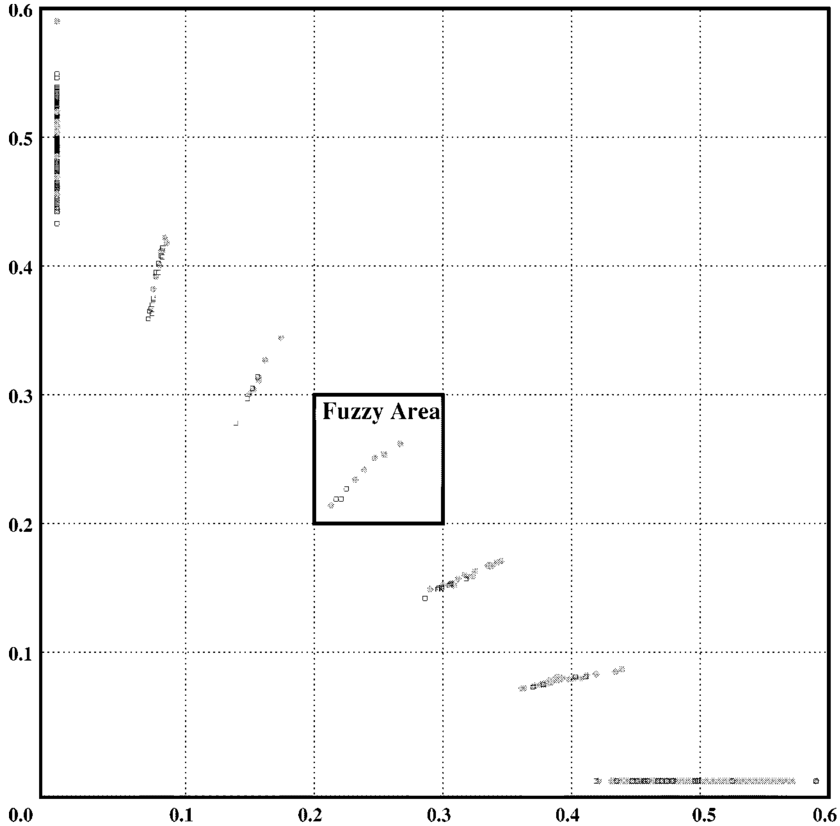

Fig. 2. This is a confidence space of two classes "0" and "6." The red dots represent the 0's and the blue dots represent the 6's. The GSC digit recognizer gives a confidence in the range of [0,0.6]. The truly "good" quality samples of "0" receive a score of 0.6 and the probability of misclassification is small as noted from the very few gray boxesmingled with open boxes at these confidence levels ($\phi_0 \approx 0.6$ and $\phi_6 \approx 0.0$). The reverse is also true. Maximum misclassifications occur when $\phi_0 \approx 0.3$ and $\phi_6 \approx 0.3$. This qualifies as the fuzzy area. These are the samples that we wish to reclassify.

B. Novelty

For most digit recognizers the correct class choice for an unknown pattern is almost always among the two classes with highest scores. This is to say that the top two choice corectness rate is almost 100%. Based on this observation the goal of our methodology is to overturn some of the results of the recognizer in order to promote the correct class with the second highest score to become the top choice of the recognizer. In this way the top choice correct rate will be very close to the original top two choices correct rate.

C. Gradient Structural Concavity

Although we have tested our method with other classifiers, we will discuss our experiments with the gradient structural concavity (GSC) recognizer which is described in [8]. GSC uses 512 symbolic multiresolutional features, which measure the image characteristics at local, intermediate, and large scales. The 192 gradient features measure edge curvature in a neighborhood of a pixel and provide information about stroke shape on a small scale. The 192 intermediate structural features measure short stroke types that span several pixels and give useful information about stroke trajectories. The 128 concavity features are used to detect stroke relationships at long distances, which can span across the image. Features at all the three levels, G, S, and C, are combined in a 512 binary feature vector and a weighted KNN algorithm is used for classification. If the top K prototypes $x_1, \ldots, x_k$ are chosen to be the closest to an unknown pattern x, a weighted vote for each class C is computed as follows:

$$\text{Weight}_C = \frac{1}{k} \sum_{i=1}^{k} c_i d(x, x_i).$$

Here, c_i is one if prototype x_i is labeled as class C and zero, otherwise, and $d(x, x_i) = (n_{11} + n_{00}/S)/512$, where n_{11} is the number of "1" bits that match between binary feature vectors of x and x_i and n_{00} is the number of "0" bits that match between binary feature vectors of x and x_i and S is a parameter in the interval [1, 5]. The classes C are then ranked according to the value of Weight$_C$. The maximum possible score s_i, corresponding to any class is equal to 0.6.

D. Methodology

Let us say the top two classes are ω_A and ω_B when presented with a test pattern x. A common practice [14] is to compute $|\phi_A(x) - \phi_B(x)|$. If $|\phi_A(x) - \phi_B(x)| \leq \tau$ (where τ is a predetermined threshold $\tau = 0.1$ for GSC), then the confusion between the two classes is deemed to be too high to return a "confident" top choice.

In our approach we consider two classes at a time, namely classes ω_A and ω_B. A sample is considered as "easy" if $|\phi_A(x) - \phi_B(x)| \ggg \tau$. Clearly, the recognizer is confident about the identity of x as either ω_A or ω_B, depending on

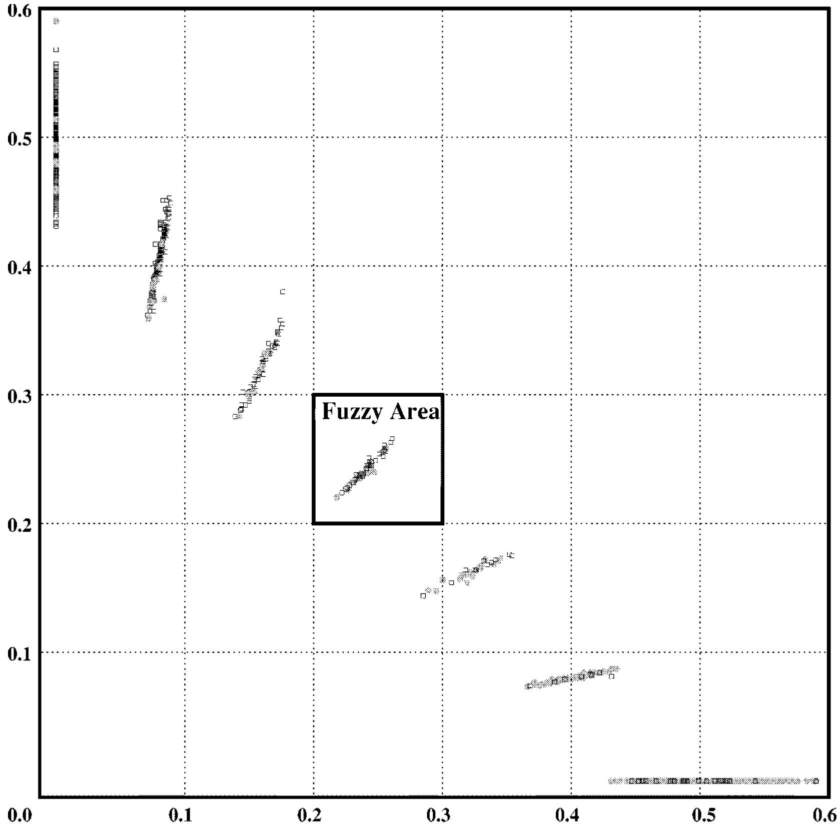

Fig. 3. This is a confidence space of two classes "0" and "1." The red dots represent the 0's and the blue dots represent the 1's. The GSC digit recognizer gives a confidence in the range of [0, 0.6]. The truly "good" quality samples of "0" receive a score of 0.6 and the probability of misclassification is small as noted from the very few gray boxes mingled with open boxes at these confidence levels ($\phi_0 \approx 0.6$ and $\phi_1 \approx 0.0$). The reverse is also true. Maximum misclassifications occur when $\phi_0 \approx 0.3$ and $\phi_1 \approx 0.3$. This qualifies as the fuzzy area. These are the samples that we wish to reclassify.

whichever is greater of the two. A sample x in the feature space falls in the "difficult" or the "overlapping" zone when $|\phi_A(x) - \phi_B(x)| \leq \tau$.

Using the confidence functions $\phi_A : X \rightarrow [0, 0.6]$ and $\phi_B : X \rightarrow [0, 0.6]$, we define a map $\Phi_{A,B} : X_{A,B} \rightarrow \mathbb{R}^2$, such that $\Phi_{A,B}(x) = (\phi_A(x), \phi_B(x))$, where $X_{A,B}$ is the set of all patterns y, for which ω_A and ω_B are the top two classes according to the classifier. Using $\Phi_{A,B}$ any pattern from $X_{A,B}$ is mapped onto a point of the square [0, 0.6]^2 and the intersection of the "overlapping" area and $X_{A,B}$ is projected onto an area in that square, we would like to call "fuzzy." Fig. 2 shows the example of two classes ω_0 and ω_6 with τ is 0.1 ($A \equiv 0$ and $B \equiv 6$). It is to be noted that when $[\phi_A(x) \ggg \phi_B(x)]$ OR $[\phi_B(x) \ggg \phi_A(x)]$ then the misclassifications of such patterns are rare. It is also interesting to note how the misclassifications increase as one approaches the fuzzy area from both extremes.

Another interesting insight we gain from Fig. 2 is the way the samples align in confidence space. When $\phi_6(x) \in [0.4 \ldots 0.6]$ AND $\phi_0(x) \in [0.0 \ldots 0.1]$ the samples in confidence space are all strongly leaning toward being a ω_6 (digit '6') by aligning parallel to the x axis. Similarly, when $\phi_0(x) \in [0.4 \ldots 0.6]$ AND $\phi_6(x) \in [0.0 \ldots 0.1]$ the samples in confidence space are all strongly leaning toward being a ω_0 (digit "0") by aligning parallel to the y-axis. When $\phi_0(x) \in [0.2 \ldots 0.3]$ AND $\phi_6(x) \in [0.2 \ldots 0.3]$ the samples in confidence space represent the fuzzy area (the area of interest). Their scatter pattern aligns at about 45°.

It is also clear from the way scores are computed that a sample x cannot receive simultaneously a very high or a very low score for both the classes. This actually helps us to narrow the fuzzy area. Hence, all patterns must fall in the diagonal area of the space spanning from point (0.0, 0.6) to point (0.6, 0.0) at an angle of 135°. Although, when we talk of cluster centers, we are refering to the confidence space, the notion of a cluster does hold meaning in the confidence space as well as can be seen in Fig. 2. The cluster center for the class ω_6 would be in the vicinity of point (0.6, 0.0) and the cluster center for ω_0 would in the vicinity of point (0.0, 0.6).

There are seven clusters shown in Fig. 2. One of the clusters is ambiguous about its class identity. This is the fuzzy area. The six other clusters either favor the class ω_0 (each cluster representing a different style of writing, perhaps) or the class ω_6. The "goodness" or quality of the clusters improves as they move away from the fuzzy area.

Figs. 3–5 are other examples. Fig. 3 involves classes ω_0 and ω_1 and shows the projection of $X_{0,1}$ onto the square $[0, 0.6]^2$. Fig. 4 involves classes ω_0 and ω_2 and shows the projection of $X_{0,2}$ onto the square $[0, 0.6]^2$. Fig. 5 involves classes ω_0 and ω_3 and shows the projection of $X_{0,3}$ onto the square $[0, 0.6]^2$.

E. Algorithm

1) *Identify* the two classes returned as the top two choices by a digit recognizer (say, ω_A and ω_B) on pattern x.

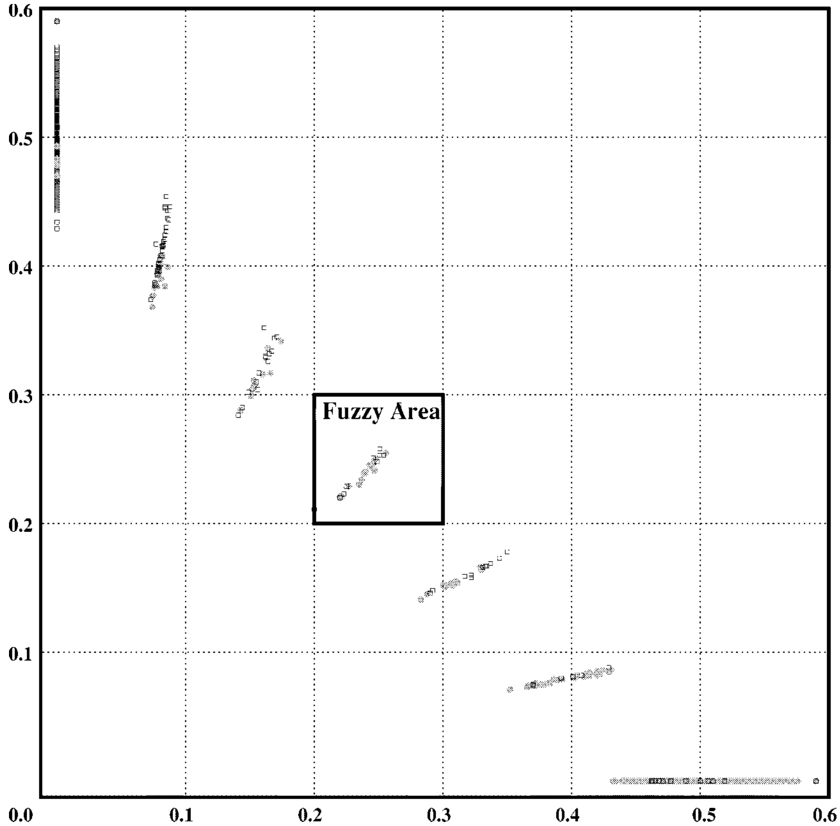

Fig. 4. This is a confidence space of two classes "0" and "2." The red dots represent the 0's and the blue dots represent the 2's. The GSC digit recognizer gives a confidence in the range of [0, 0.6]. The truly "good" quality samples of "0" receive a score of 0.6 and the probability of misclassification is small as noted from the very few gray boxes mingled with open boxes at these confidence levels ($\phi_0 \approx 0.6$ and $\phi_2 \approx 0.0$). The reverse is also true. Maximum misclassifications occur when $\phi_0 \approx 0.3$ and $\phi_2 \approx 0.3$. This qualifies as the fuzzy area. These are the samples that we wish to reclassify.

2) *If* the quantity $|\phi_A(x) - \phi_B(x)| \ggg \tau$
then accept the top choice as correct
else proceed to evaluate if the second choice should be promoted to the top by using a lookup table based on the modified KNN as described in the next section.

F. Analysis

Fig. 6 shows samples of images that will be confidently recognized as a "0" in the top choice by a traditional recognizer (such as the GSC recognizer [8]) and needs no adjustment to the classification.

Fig. 7 shows samples of images that will be confidently misclassified as a "6" in the top choice by a traditional recognizer (such as the GSC) and cannot be readily rectified.

Fig. 8 shows samples of images that will be confidently recognized as a "6" in the top choice by a traditional recognizer (such as the GSC) and needs no adjustment to the classification.

Fig. 9 shows samples of images that will be confidently misclassified as a "0" in the top choice by a traditional recognizer (such as the GSC) and cannot be readily rectified.

Fig. 10 shows samples of images that will be confusing to tell apart as a "0" or "6" by a traditional digit recognizer such as GSC. These are the cases which we wish to re-classify using our methodology. These are the samples that fall in the fuzzy area of the confidence space shown in Fig. 2.

Fig. 11 shows the new class boundary after re-classification based on the modified KNN for the samples in the fuzzy area (Fig. 2). A straight classification based on the value $\phi_i(x)$ being greater or less than a threshold (τ) would draw the class boundary as a straight line right through the middle of the fuzzy area at a 45° angle as shown. The modified KNN, as will be described in the next section, generates a complex boundary which does not eliminate all the classification errors, in some cases it even creates additional errors in the process of correcting some errors, nevertheless the method makes an overall improvement.

Figs. 12 and 13 show more examples. Fig. 12 involves classes ω_0 and ω_2 and shows a new class boundary between zeros and twos. Fig. 13 involves classes ω_0 and ω_3, and shows a new class boundary between zeros and threes.

III. TOPOLOGY

Let us start this section by repeating the assumption that the scores for each class returned by the recognizer are real numbers in interval [0, 1]. Let U be a "blind" set of digit patterns. We will assume that all classes are equally represented in U. Let V be a subset of U that contains all patterns in U that are classified correctly by the classifier. We will use the set V to simulate the fuzzy boundaries between classes. V is partitioned into a disjoint union of subsets $V_0, V_1, \ldots, V_9$, where each subset V_i contains those digit patterns in V, that represent class ω_i. The score, associated with class ω_i is the highest in the confidence vector returned by the classifier, i.e.,

$$V_i = \{x \in V : \phi_j(x) < \phi_i(x) \text{ for all } j \neq i\}.$$

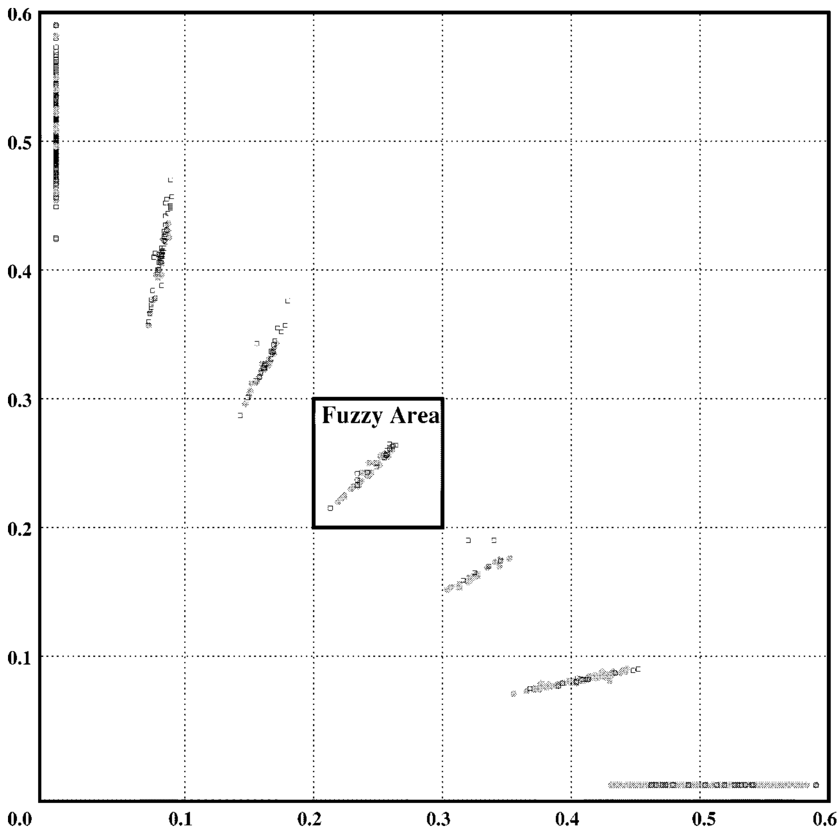

Fig. 5. This is a confidence space of two classes "0" and "3." The red dots represent the 0's and the blue dots represent the 3's. The GSC digit recognizer gives a confidence in the range of [0, 0.6]. The truly "good" quality samples of "0" receive a score of 0.6 and the probability of misclassification is small as noted from the very few gray boxes mingled with open boxes at these confidence levels ($\phi_0 \approx 0.6$ and $\phi_3 \approx 0.0$). The reverse is also true. Maximum misclassifications occur when $\phi_0 \approx 0.3$ and $\phi_3 \approx 0.3$. This qualifies as the fuzzy area. These are the samples that we wish to reclassify.

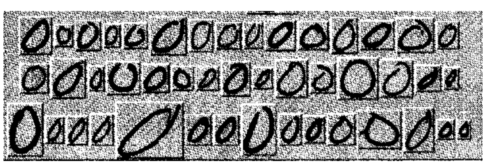

Fig. 6. A sampling of "good" quality 0's. These are samples that receive $\phi_0 \approx 0.6$ and $\phi_6 \approx 0.0$ (Fig. 2.)

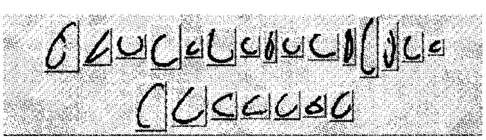

Fig. 7. A sampling of "poor" quality 0's. These are samples that receive $\phi_0 \approx 0.0$ and $\phi_6 \approx 0.6$ (Fig. 2).

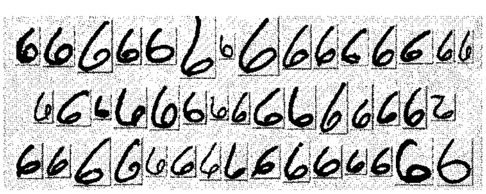

Fig. 8. A sampling of "good" quality 6's. These are samples that receive $\phi_6 \approx 0.6$ and $\phi_0 \approx 0.0$ (Fig. 1).

Let $f_i(x)$ be the characterictic function of the set V_i, i.e.,

$$f_i(x) = \begin{cases} & \text{if } x \in V_i \\ 0, & \text{otherwise.} \end{cases}$$

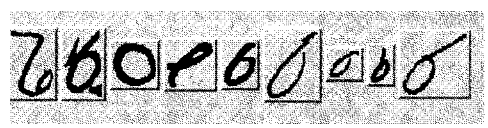

Fig. 9. A sampling of "poor" quality 6's. These are samples that receive $\phi_6 \approx 0.0$ and $\phi_0 \approx 0.6$ (Fig. 2).

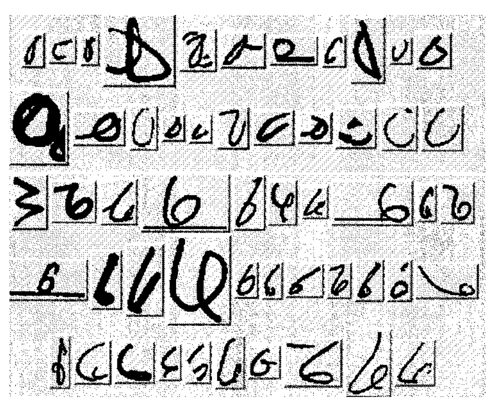

Fig. 10. A sampling of "difficult" quality 0's and 6's. These are samples that receive $\phi_6 \approx 0.3$ and $\phi_0 \approx 0.3$ (Fig. 2).

Let $W = [0,1]_0 \times [0,1]_1$ be a Cartesian product of two copies of the interval [0, 1]. Here $[0,1]_i$ denotes the ith-copy of the interval $[0,1]$. We can convert W into a topological space by introducing one of the following two topologies: finite or Euclidian.

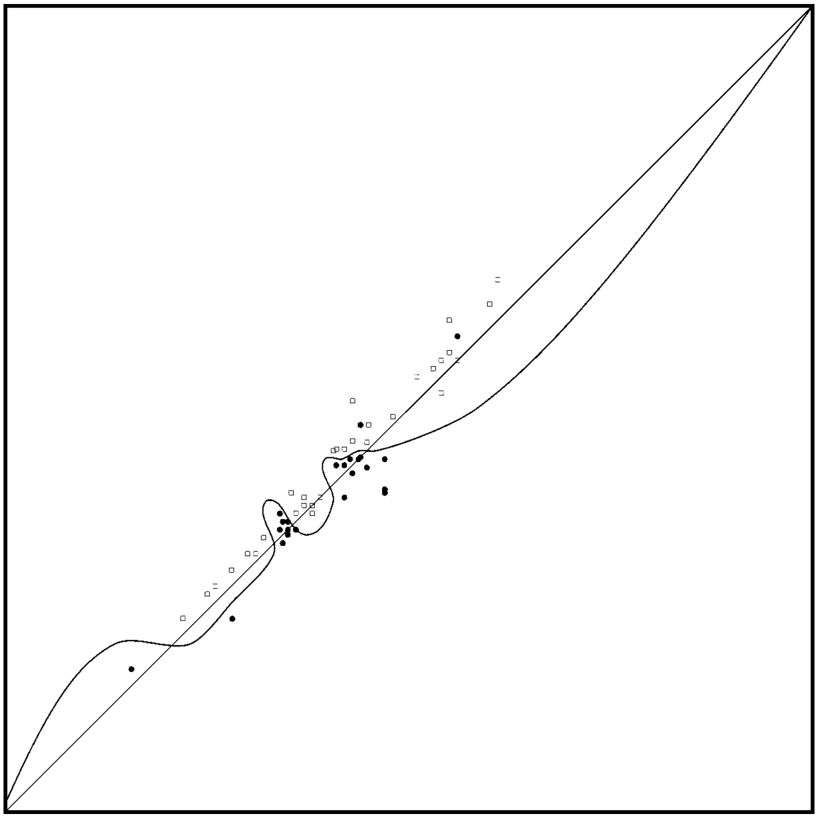

Fig. 11. Redefining the class boundary between 0's and 6's in the fuzzy area of Fig. 2.

1) *Finite Topology:* Assume that each interval $[0,1]_i$ is partitioned into finitely many disjoined subintervals $W_i^1, W_i^2, \ldots, W_i^n$. Then the set $\mathcal{B}$ of neighborhoods $W_0^{i_0} \times W_1^{i_1}$ forms a base of a topology in W. Let us enumerate the elements of $\mathcal{B}$, i.e., $\mathcal{B} = \{B_1, B_2, \ldots, B_M\}$, where $M = n^2$.

   Let $W(i_1, i_2, \ldots, i_j)$ be the union of the sets B_{i_1}, $B_{i_2}, \ldots, B_{i_j}$, i.e.,

$$W(i_1, i_2, \ldots, i_j) = B_{i_1} \cup B_{i_2} \cup \ldots \cup B_{i_j}.$$

   Let $\mathcal{M}$ be the family of all such sets $W(i_1, i_2, \ldots, i_j)$, i.e.

$$\mathcal{M} = \{W(i_1, i_2, \ldots, i_j) : j \in \{1, 2, \ldots, M\}, \quad 1 \leq i_1 < i_2 \ldots < i_j \leq M\}.$$

   Then $\mathcal{M}$ is a topology of W.

2) *Euclidean Topology:* In this case, we will consider W with the topology induced by the Euclidean topology of the space $\mathbb{R}^2$, i.e., the topology that is generated by the collection of all balls in the set W [15].

The next step is to construct a σ algebra, which includes the topology of the set W. We consider two cases.

In the finite case, it is apparent that $B_0 \cup B_1 \cup \ldots \cup B_M = W$. Hence, $W \in \mathcal{M}$. Moreover, since no two elements of the base $\mathcal{B}$ intersect, if the set $A \in \mathcal{M}$, then $A^c \in \mathcal{M}$, where A^c denotes the complement of A in W. Also, the union of any two sets from $\mathcal{M}$ is a set from $\mathcal{M}$. This, together with the fact that $\mathcal{M}$ is finite, implies that $\mathcal{M}$ is a σ algebra [15].

In the Euclidean case, we define $\mathcal{M}$ to be Borel σ algebra generated by the Euclidean topology.

Now, given two classes ω_i and ω_j ($i < j$), we will introduce a family of positive σ additive measures for the σ algebra $\mathcal{M}$. We define the function $\mu(\alpha, \beta) : \mathcal{M} \to \mathbb{R}$ as follows:

$$\mu(\alpha, \beta)(A) = \sum_{\{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A\}} \alpha f_i(x) + \beta f_j(x) \tag{1}$$

for all $A \in \mathcal{M}$. Here, $f_i(x)$ is the characteristic function of the set V_i; $f_j(x)$ is the characteristic function of the set V_j; and α and β are positive numbers. The values of the function $\mu(\alpha, \beta)$ are nonnegative. Moreover, a function thus defined has the property that if $A_1 \in \mathcal{M}$ and $A_2 \in \mathcal{M}$ such that $A_1 \cap A_2 = \emptyset$, then $A_1 \cup A_2 \in \mathcal{M}$ and

$$\mu_i(A_1 \cup A_2) = \mu_i(A_1) + \mu_i(A_2).$$

Indeed, if $x \in V$ is in $A_1 \cup A_2$, then $x \in A_1$ or $x \in A_2$ since $A_1 \cap A_2 = \emptyset$, implying

$$\begin{aligned}
\mu(\alpha, \beta)(A_1 \cup A_2) &= \sum_{\{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A_1 \cup A_2\}} \alpha f_i(x) + \beta f_j(x) \\
&= \sum_{\{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A_1\} \cup \{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A_2\}} \cdot \alpha f_i(x) + \beta f_j(x) \\
&= \sum_{\{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A_1\}} \alpha f_i(x) + \beta f_j(x) \\
&\quad + \sum_{\{x \in V_i \cup V_j : \Phi_{i,j}(x) \in A_2\}} \alpha f_i(x) + \beta f_j(x) \\
&= \mu(\alpha, \beta)(A_1) + \mu(\alpha, \beta)(A_2).
\end{aligned}$$

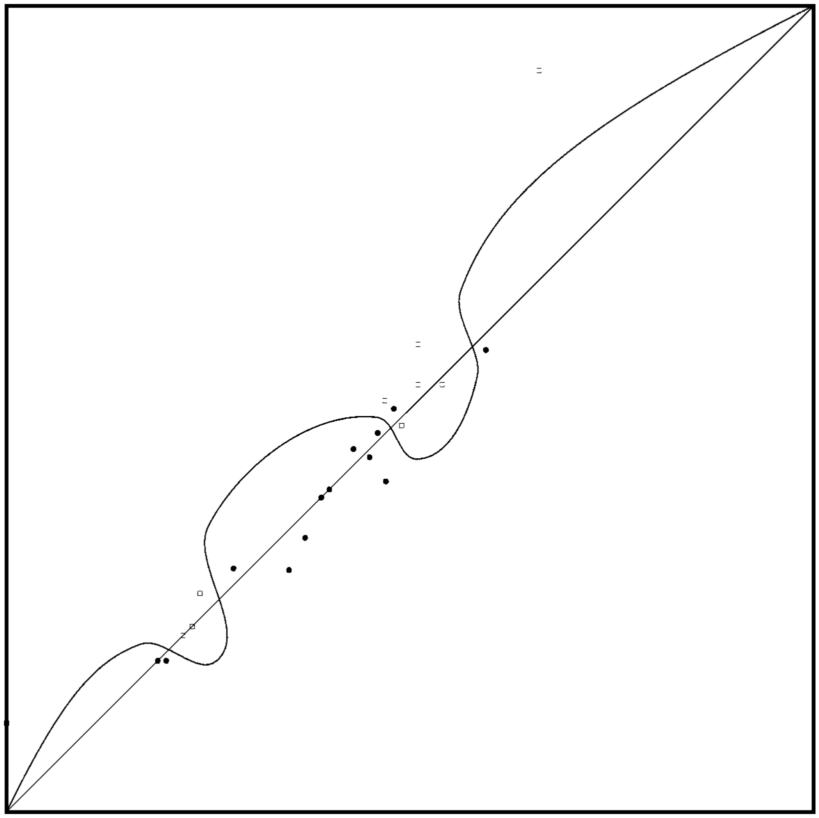

Fig. 12. Redefining the class boundary between 0's and 2's in the fuzzy area of Fig. 4.

So, we can conclude that $\mu(\alpha,\beta)$ is a family of positive σ additive measures [15]. According to [15, theorem 1.19, p. 16] the following properties are satisfied:

1) $\mu(\alpha,\beta)\,() = 0$;
2) if $A \subset B$, then $\mu(\alpha,\beta)\,(A) \leq \mu(\alpha,\beta)\,(B)$;
3) if $\{A_i\}_{i=1}^{\infty}$ is monotonic, then

$$\lim_{n\to\infty} \mu(\alpha,\beta)\,(A_i) = \mu(\alpha,\beta)\left(\bigcup_{i=1}^{\infty} A_i\right).$$

Note that the normalized measure $\mu(\alpha,\beta)/\mu(\alpha,\beta)\,(W)$ is a fuzzy measure [3].

Let us give special names of two members of this family. We introduce

$$\mu_i = \mu(1,0) \quad \text{and} \quad \mu_j = \mu(0,1).$$

IV. Fuzzy Sets

In the finite case, we have a set of of neighborhoods

$$\mathcal{B} = \{B_1, B_2, \ldots, B_M\}.$$

For every $y \in W$, there exists a unique neighborhood B_y, containing y. In the Euclidean case given a function $R: W \to \mathbb{R}$, the set of neighborhoods of y includes all balls of the forms

$$B_y = \{z \in W : dist(y,z) \leq R(y)\}.$$

For every $y \in W$, let $\chi_y : W \to \mathbb{R}$ be the characteristic function of its neighborhood B_y. Let us also define the family of functions

$$\begin{aligned}\varphi(\alpha,\beta)\,(y) &= \int_W \chi_y(z)\, d\mu(\alpha,\beta)\,(z)\\ &= \alpha|\{x \in V_i : \Phi_{i,j}(x) \in B_y\}|\\ &\quad + \beta|\{x \in V_j : \Phi_{i,j}(x) \in B_y\}|\end{aligned}$$

as well as the functions

$$\varphi_i(y) = \int_W \chi_y(z)\, d\mu_i(z) = |\{x \in V_i : \Phi_{i,j}(x) \in B_y\}|$$

and

$$\varphi_j(y) = \int_W \chi_y(z)\, d\mu_j(z) = |\{x \in V_j : \Phi_{i,j}(x) \in B_y\}|.$$

Let us observe, that all functions $\varphi(a,\beta)\,(y)$ and, in particular, $\varphi_i(y)$ and $\varphi_j(y)$, are nonnegative. Given a positive number α, let

$$\eta_i(y) = \begin{cases} \dfrac{\varphi_i(y)}{\varphi(1,\alpha)\,(y)}, & \text{if } \varphi(1,\alpha)\,(y) \neq 0\\ 0.5, & \text{otherwise}\end{cases}$$

and

$$\eta_j(y) = \begin{cases} \dfrac{\varphi_j(y)}{\varphi\left(\dfrac{1}{\alpha},1\right)\,(y)}, & \text{if } \varphi\left(\dfrac{1}{\alpha},1\right)\,(y) \neq 0\\ 0.5, & \text{otherwise.}\end{cases}$$

Let us observe, that for every $y \in W$, $0 \leq \eta_i(y) \leq 1$, $0 \leq \eta_i(y) \leq 1$, and $\eta_i(y) + \eta_j(y) = 1$.

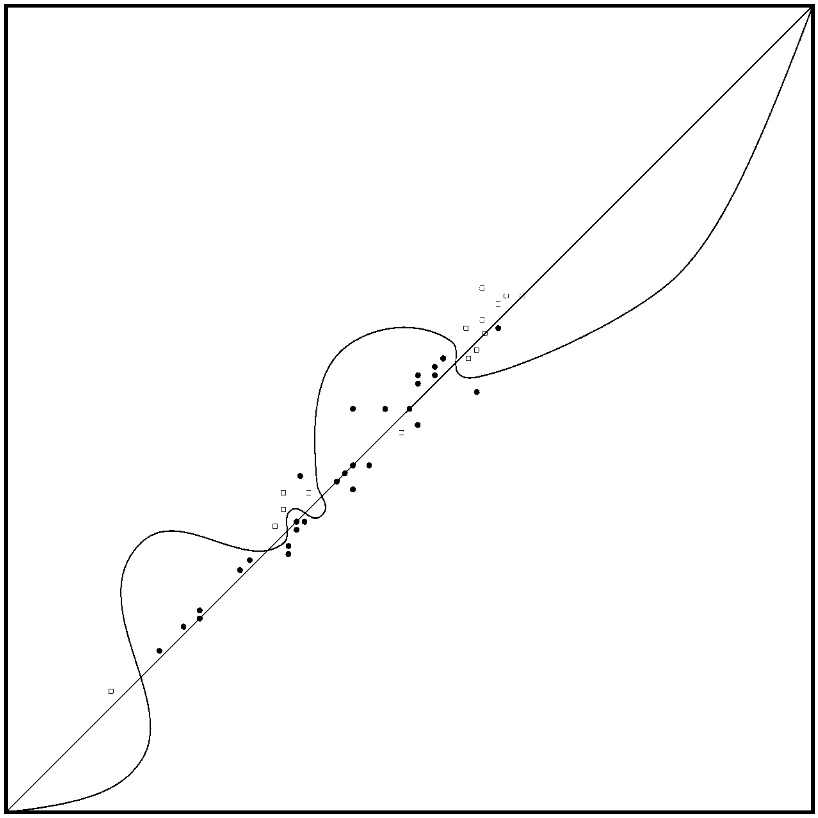

Fig. 13. Redefining the class boundary between 0's and 3's in the fuzzy area of Fig. 5.

Now, we are in a position to define the fuzzy sets corresponding to the different classes. Let x be an unknown pattern and classes ω_i and ω_j correspond to the two highest entries in the confidence vector

$$\boldsymbol{\Phi}(x) = (\phi_0(x), \phi_1(x), \ldots, \phi_9(x)).$$

Let τ be the threshold that determines the overlapping area. We can define the fuzzy membership functions as follows:

$$\xi_i(x) = \begin{cases} \eta_i(\Phi_{i,j}(x)), & \text{if } x \text{ is in the overlapping area} \\ \phi_i(x), & \text{otherwise} \end{cases}$$

$$\xi_j(x) = \begin{cases} \eta_j(\Phi_{i,j}(x)), & \text{if } x \text{ is in the overlapping area} \\ \phi_j(x), & \text{otherwise.} \end{cases}$$

$$\xi_i(x) = \begin{cases} 0, & \text{if } x \text{ is in the overlapping area} \\ \phi_i(x), & \text{otherwise.} \end{cases}$$

Each of these functions $\xi_i(x)$ is the membership function of the corresponding class ω_i. We say that the unknown pattern x belongs to the class ω_i, if $\xi_j(x) < \xi_i(x)$ for all $j \neq i$. Having defined our membership functions $\xi_i(x)$, we are able to partition the space W into class regions and boundaries, that separates them in the following way. The set

$$\Omega_i = \{x : \xi_i(x) > \xi_j(x) \text{ for all } j \neq i\}$$

$i = 0, 1, \ldots, 9$ is the region, all points of which are in class ω_i. The set

$$D_{ij} = \{x : \xi_i(x) = \xi_j(x) \text{ and } \xi_i(x) > \xi_k(x) \text{ for all } k \neq i, j\}$$

is the implicitly defined surface, that is a part of the boundary between classes ω_i and ω_j.

An illustration of these concepts (when only two classes are involved) is provided in Figs. 2–5. Initially, the recognizer uses a linear discriminant, namely the line $y = x$, in the projection space. Figs. 11, 12, etc., show the new nonlinear discriminant between the classes using the algorithm as described.

V. Bayesian View of the Membership Functions

Let x be a pattern in the overlapping area and ω_A and ω_B are the two classes with the highest and second highest scores in the confidence vector. Let s_A and s_B be the scores corresponding to ω_A and ω_B. Let us compute the *a posteriori* probabilities for the classes ω_A and ω_B being the true class for pattern x, given scores s_A and s_B of the two top class choices ω_A and ω_B

$$P(\omega_A/s_A, s_B) = \frac{P(s_A, s_B/\omega_A)P(\omega_A)}{P(s_A, s_B/\omega_A)P(\omega_A) + P(s_A, s_B/\omega_B)P(\omega_B)}.$$

Without any prior information, we can assume that both classes have equal chance of being the truth, so we can reasonably assume that $P(\omega_A) = P(\omega_B)$. Then

$$\begin{aligned} P(\omega_A/s_A, s_B) &= \frac{P(s_A, s_B/\omega_A)}{P(s_A, s_B/\omega_A) + P(s_A, s_B/\omega_B)} \\ &= \frac{P(s_A/\omega_A)P(s_A/s_B, \omega_A)}{P(s_A/\omega_A)P(s_A/s_B, \omega_A) + P(s_B/\omega_B)P(s_B/s_A, \omega_B)} \\ &= \frac{P(s_A/\omega_A)}{P(s_A/\omega_A) + \alpha P(s_B/\omega_B)} \end{aligned}$$

where $P(s_B/s_A,\omega_B)/P(s_A/s_B,\omega_A)$. In the same way we compute

$$P(\omega_B/s_A,s_B)=\frac{P(s_B/\omega_B)}{\frac{1}{\alpha}P(s_A/\omega_A)+P(s_B/\omega_B)}.$$

By definition, the probability records the frequency of the results of an event over many trials. However, given a pattern x, the classifier returns the same confidence vector over and over, no matter how many times the recognition process is repeated. In order to deal with this problem, i.e., to compute $P(s_A\omega_A)$, we will approximate the probability by using patterns y, which are "close" to pattern x in the "eyes" of the recognizer, i.e., $\Phi_{A,B}(y)$ falls in a given neighborhood B_x of $\Phi_{A,B}(x)$. Then,

$$\begin{aligned}P(s_A/\omega_A)&=\frac{\sum_{y\in V_A\cup V_B:\Phi_{A,B}(y)\in B_x} f_A(y)}{|V_A\cup V_B|}\\&=\frac{\mu_A(B_X)}{|V_A\cup V_B|}.\end{aligned}$$

Similarly

$$P(s_B/\omega_B)=\frac{\mu_B(B_X)}{|V_A\cup V_B|}$$

and, therefore

$$P(\omega_A/s_A,s_B)=\frac{\mu_A(B_X)}{\mu_A(B_X)+\alpha\mu_B(B_X)}=\xi_A(x)$$

and

$$P(\omega_B/s_A,s_B)=\frac{\mu_B(B_X)}{\mu_B(B_X)+\frac{1}{\alpha}\mu_A(B_X)}=\xi_B(x).$$

VI. Experiments

We have already described how the fuzzy boundaries change results only for those samples that fall in the proximity of class boundaries. Hence, if the the original recognizer has a high-recognition accuracy, our method can only marginally affect the results. GSC digit recognizer is such a recognizer with very high accuracy. The method described in this paper successfully reduced its error rate from 2.85% to 1.96%, i.e., by 0.89%, which is more than 30% from the initial error of 2.85% (Table I). To illustrate the significance of the improvement let us look at a simple application. Assuming we use GSC to recognize five digits in handwritten ZIP codes, taken from the United States (U.S.) mail stream. GSC recognizes correctly the entire ZIP code in 86.54% of the cases and the method, described in this paper will recognize correctly the entire ZIP code in 90.58% of the cases.

The training set U(50000 images) and the testing set(30398 images) were created using digit samples extracted from the U.S. mail stream. There are two reasons why we use our own database. First, GSC achieves almost 100% correct rate on databases available publicly such as National Institute of Standards and Technology (NIST). Second, all classes are

TABLE I
Improvements to GSC Recognizer Using Fuzzy Boundaries

test set	GSC		Fuzzy	
	correct	%	correct	%
30398	29532	97.15	29807	98.04

TABLE II
Improvements to GSC Recognizer Using Fuzzy Approach for Any of the Classes

class	test set	GSC		Fuzzy	
		correct	%	correct	%
0	6986	6833	97.81	6852	98.08
1	6554	6496	99.12	6496	99.12
2	3954	3759	95.07	3845	97.24
3	2238	2170	96.96	2186	97.68
4	3191	3070	96.21	3123	97.87
5	1556	1497	96.21	1513	97.24
6	1474	1442	97.83	1449	98.30
7	1445	1391	96.26	1412	97.72
8	2049	1945	94.93	2001	97.61
9	951	929	97.69	930	97.93

equally represented in the training set U, which is not the case with other databases.

In the experiment, we use $\tau=0.1$ to determine the overlapping area. Then the confidence space is partitioned as described in the previous sections into squares using $n=100$. Since the training is off-line, we precompute the measures of any of the squares in the fuzzy area and then store them in a lookup table. Once the recognizer returns the confidence vector $\Phi(x)$ of an unknown pattern, a check is performed to see if the pattern is in the "overlapping area." If it is in the overlapping area, the corresponding entries in the lookup table are used to determine the values of the fuzzy membership functions for the pattern and its classification is performed.

One way to evaluate the performance of two recognizers is to compare the corresponding percentage correctly classified samples (Tables I, II).

Actually, the best way to evaluate the performance of two classifiers is to compare the their graphs on a REJECT versus ERROR scale (Fig. 14).

Fig. 14 shows the graph of GSC recognizer and the fuzzy method on a REJECT vs ERROR scale in order to compare their performances. Table III, shows some selected points from the graphs of both recognizers (Fig. 14) in order to compare the correct rates of classifiers given the reject rate.

In the end, we have performed the similar experiment with other available recognizer and obtain similar results (Table IV).

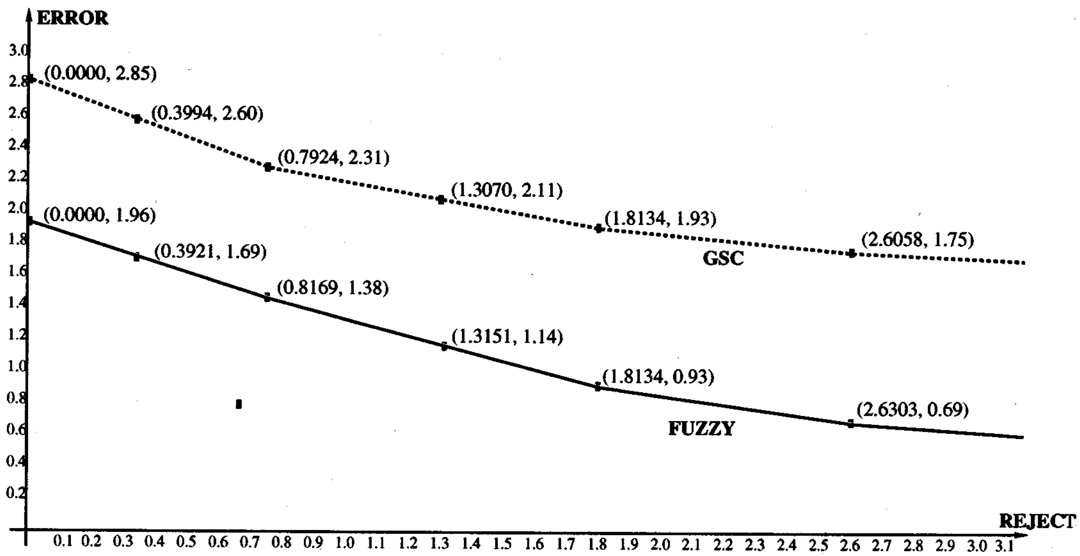

Fig. 14. This figure shows how the performances of GSC recognizer and the proposed fuzzy method are compared on a REJECT versus ERROR scale. Based on the fact the graph of the first fuzzy method is upon the graph of GSC, we can conclude that the overall performance of our method is better the one of GSC recognizer.

TABLE III
THIS TABLE SHOWS SOME SELECTED POINTS FROM THE GRAPHS OF BOTH RECOGNIZERS ON A REJECT VERSUS ERROR SCALE IN ORDER TO COMPARE THE CORRECT RATES OF CLASSIFIERS GIVEN A REJECT RATE AND SHOWS IMPROVEMENT OF THE FUZZY METHOD OVER GSC AT THOSE REJECT RATES

GSC		Fuzzy	
reject %	error %	reject %	error %
0.0000	2.85	0.0000	1.96
0.3994	2.60	0.3921	1.69
0.7924	2.31	0.8169	1.38
1.3070	2.11	1.3151	1.14
1.8134	1.93	1.8134	0.93
2.6303	1.75	2.6058	0.69

TABLE IV
IMPROVEMENTS TO OTHERS RECOGNIZER USING FUZZY METHOD

	Polynomial		Fuzzy	
	correct	%	correct	%
	27744	91.27	28155	92.62
test set	Gradient		Fuzzy	
	correct	%	correct	%
30398	28671	94.32	29039	95.53
	Curvature		Fuzzy	
	correct	%	correct	%
	29191	96.03	29513	97.09

VII. SUMMARY

First, let us discuss the implementation issues. In the case of finite topology, a lookup table of size $O(n^2)$ is created off line in $O(n|U|)$ time, where $|U|$ is the size of the training set and n is the number used to partition the confidence space. The fuzzy method transforms the confidence vector into a new vector, containing the values of the fuzzy membership function in $O(1)$ time. In the Euclidean case, we store the confidence vectors of all samples in the training set U ($O(|U|)$ space is required) and then the needed measures are computed on line in at most $O(|U|)$ time. So, this version of the fuzzy method is slower, but it is more accurate.

Second, let discuss a modification of the fuzzy algorithm. Let us consider three or more classes: x is a unknown pattern, ω_A is the class with highest score in the confidence vector, and classes satisfy the "overlapping" condition, i.e., $|\varphi_A(x)-\varphi_B(x)| < \tau$, $|\varphi_A(x) - \varphi_C(x)| < \tau$. In other words, the pattern is around the boundaries of three or more classes. Situations like this occur seldom, but still require some attention. A modification of the "fuzzy" method described can easily deal with situations like this. Using the confidence vector, we project space X of all patterns in a cube in $\mathbb{R}^{10}$, then we introduce a topology similar to the one(finite or Euclidean) described before and then use the following family of measures to define the fuzzy membership functions

$$\mu(\alpha_0, \alpha_1, \ldots, \alpha_9)(A) = \sum_{\{x \in V : \Phi(x) \in A\}} \sum_{i=0}^{9} \alpha_i f_i(x)$$

under the condition that $\alpha_i = 0$ if $|\varphi_A(x) - \varphi_B(x)| >= \tau$.

Another issue of importance is the way in which the "overlapping" area is determined. In this paper, we use one very simple and intuitive way to describe the "overlapping" area.

In the case of GSC, this was very effective (Fig. 2). In cases where other classifiers are used, this issue is more complex and requires more investigation. There some interesting problems have arisen: should more than one threshold be used to determine the overlapping area; what is the relation between the threshold and the size and density of the overlapping area; how does one choose the optimal neighborhood for each point in the fuzzy area and what topology corresponds to this set of neighborhoods; what is the relation between the misclassified patterns that have been classified correctly by our method and correctly classified patterns misclassified by our method. These and other open issues we plan to investigate in the future.

Finally, the method described is able to improve a recognition system while treating it as a black box. Further, the only resource required to enable this method is a large database of samples for retraining. More samples lead to the better results.

REFERENCES

[1] P. Arabie, L. J. Hubert, and G. De Doete, *Clustering and Classification*, P. Arabie, L. J. Hubert, and G. De Soete, Eds. River Edge, NJ: World Scientific, 1998.

[2] F. Aurenhammer, "Voronoi diagrams—Survey of a fundamental data structure," *ACM Computing Survey*, vol. 23, no. 3, pp. 345–405, Sept. 1991.

[3] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Norwell, MA: Kluwer, 1999.

[4] D. Bouchaffra, V. Govindaraju, and S. Srihari, "A methodology for determining probability of correctness of recognizer scores," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Santa Barbara, CA, June 1998, pp. 930–935.

[5] B. V. Dasarathy, *Nearest Neighbor (NN) Norms : nn Pattern Classification Techniques*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1991.

[6] G. Carapenter, S. Grossberg, N. Reynolds, and D. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimentional maps," *IEEE Tran. Neural Networks*, vol. 3, pp. 698–713, Sept. 1992.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., 1997.

[8] J. T. Favata and G. Srikantan, "A multiple feature/resolution approach to hand-printed digit and character recognition," *Int. J. Imaging Syst. Technol.*, vol. 7, pp. 304–311, 1996.

[9] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifer," *J. Mach. Learning*, vol. 29(2/3), pp. 131–163, Nov. 1997.

[10] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] J. Keller, M. Gray, and J. Givens, "A fuzzy K nearest neighbor algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 580–585, July/Aug. 1985.

[12] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision tree hibrid," in *Proc. 2nd Int. Conf. Knowledge Discovery Data Mining*, San Mateo, CA, Aug. 1996, pp. 114–119.

[13] L. Kuncheva and J. Bezdek, "An integrated framework for generalized nearest prototype classifier design," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 6, no. 5, pp. 437–457, 1999.

[14] S. Madhvanath, E. Kleinberg, and V. Govindaraju, "Empierical design of a multi-classifier threshold/control strategy for recognition of handwritten street names," *Int. J. Pattern Recognition Artificial Intell.*, vol. 11, no. 6, pp. 933–946, 1997.

[15] J. R. Munkres, *Topology—A First Course*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[16] M. Ramoni and P. Sebastiani, "Bayesian methods for intelligent data analysis," in *An Introduction to Intelligent Data Analysis*, M. Berthold and D. J. Hand, Eds. New York: Springer-Verlag, 1999.

[17] W. Rudin, *Real and Complex Analysis*. New York: McGraw-Hill, 1986.

Venu Govindaraju (M'97–SM'98) received the Ph.D. degree in computer science from the State University of New York at Buffalo in 1992.

He is currently the Associate Director of CEDAR and concurrently holds Associate Professorship in the Department of Computer Science and Engineering, State University of New York at Buffalo. He has been a coprincipal investigator on several federally sponsored and industry sponsored projects and is presently leading multiple projects on postal applications. He is the Associate Editor of the *Journal of Pattern Recognition*. He has coauthored more than 100 technical papers in various International journals and conferences and has one U.S. patent.

Dr. Govindaraju is the area Chair of the IEEE Systems, Man, and Cybernetics Technical Committee for pattern recognition.

Krassimir Ianakiev received the Masters (Hons.) degree from Sofia University, in 1989, and the Ph.D. degrees in math and computer science and engineering in 1998 in 2000, respectively .

He is currently a Research Scientist at CEDAR, State University of New York at Buffalo, Amherst, NY.