IMPROVEMENT OF RECOGNITION ACCURACY USING 2-STAGE CLASSIFICATION

Iakaniev, Kr.; Govindaraju, V.

IMPROVEMENT OF RECOGNITION ACCURACY USING 2-STAGE CLASSIFCATION

KR. IANAKIEV and V. GOVINDARAJU
CEDAR, Department of Computer Science
UB Commons, Suite 202
Amherst, NY 14228, USA
E-mail: {ianakiev,govind} cedar.buffalo.edu

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 which uses a multiple classification scheme. The classification rules of each stage are 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. 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 GSC digit recognizer[7]. Our method successfully reduces 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.

1 Introduction

Pattern classification consists of assigning a class label to a set of unclassified patterns. Formally, given the pattern x (taken from the space of all unknown patterns X), a set of c class labels $(\omega_1 ... \omega_c)$ and a set of labeled prototypes, the classifier assigns a label to x such that there is a minimum probability of misclassification. 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 0 and 1. 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 Figure 1, x is "closest" to the center of the class ω_5 with a distance of s_5. The 2^{nd} and 3^{rd} 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.

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 153-165

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.

Figure 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

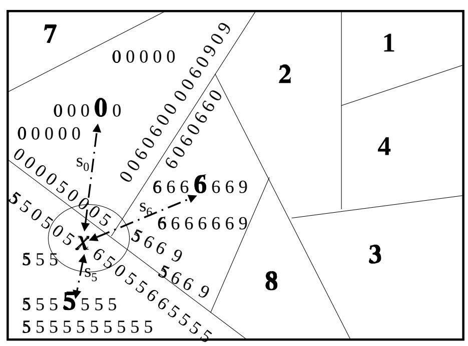

Figure 1: Feature space of a typical digit recognizer. Samples that are close to the center of a class are strong representatives of the class and are usually correctly classified. Samples that are away from the center (close to the boundaries of classes) are weaker representatives and are prone to misclassification. 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.

on top of the classifier in order to deal with the patterns that are more prone to misclassification. Using the KNN rule[4, 6] 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 Figure 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.

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[13] is another way to deal with the problem, especially in cases when a large database is available. This method is very attractive in view of recent empirical evaluations showing high accuracy[8,10].

2 Our Approach

We describe a method that is based on the following two ideas which appeal to common sense[3]:
(i) 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.
(ii) 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 look-up 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.

2.1 "Overlapping Area" OR "Fuzzy Area"

Assuming that s_i is the maximum possible score corresponding to the class ω_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 $|\phi_A(y) - \phi_B(y)| \leq \frac{s_A + s_B}{2}$, 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 Figure 2, the "overlapping" area can be narrowed even further.

2.2 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.

2.3 GSC

Although we have tested our method with other classifiers, we will discuss our experiments with the Gradient Structural Concavity (GSC) recognizer[7]. GSC uses 512 symbolic multi-resolutional features, which measure the image characteristics at local, intermediate, and large scales. The 192 **G**radient features measure edge curvature in a neighborhood of a pixel and provide information about stroke shape on a small scale. The 192 intermediate **S**tructural features measure short stroke types which span several pixels and give useful information about stroke trajectories. The 128 **C**oncavity features are used to detect stroke relationships at long distances which can span across the image. Features at all the 3 levels, **G, S,** and **C** are combined in a 512 binary feature vector and a weighted k-nearest neighbor algorithm is used for classification.

2.4 Methodology

Let us say the top two classes are ω_A and ω_B when presented with a test pattern x. A common practice [11] is to compute $|\ \phi_A(x) - \phi_B(x)\ |$. If $|\ \phi_A(x) - \phi_B(x)\ | \leq \tau$ (where τ is a pre-determined 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 $\mid \phi_A(x) - \phi_B(x) \mid \ggg \tau$. Clearly, the recognizer is confident about the identity of x as either ω_A or ω_B, depending on whichever is greater of the two. A sample x in the feature space falls in the "difficult" or the "overlapping" zone when $\mid \phi_A(x) - \phi_B(x) \mid \leq \tau$. Using the

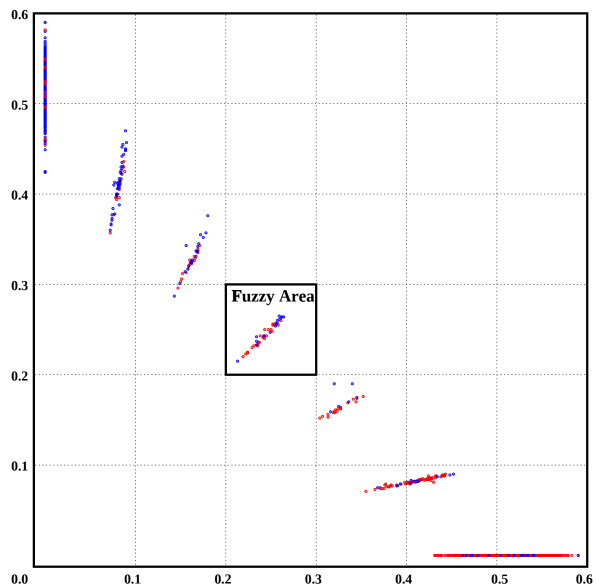

Figure 2: This is a confidence space of two classes '0' and '6'. The red dots represent the 0s and the blue dots represent the 6s. The GSC digit recognizer gives a confidence in the range of [0,0.6]. The truly "good" quality samples of '0' receive a score of 0.6 and the probability of misclassification is small as noted from the very few blue dots mingled with red dots 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.

confidence functions $\phi_A : X \rightarrow [0, 0.6]$ and $\phi_B : X \rightarrow [0, 0.6]$, we define a map $\Phi_{A,B} : X_{A,B} \rightarrow \mathbb{R}^{\not\approx}$, 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". Figure 2 shows the example of two classes ω_0 and ω_6 with τ is 0.1 (A $\equiv$ 0 & 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 Figure 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 towards 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 towards 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^o.

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^o. 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 Figure 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 7 clusters shown in Figure 2. One of the clusters is ambiguous about its class identity. This is the fuzzy area. The 6 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.

2.5 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.

2. *If* the quantity $\mid \phi_A(x) - \phi_B(x) \mid \ggg \tau$
   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.

2.6 Analysis

Figure 3 shows samples of images that will be confidently recognized as a '0' or '6' in the top choice by a traditional recognizer (such as the GSC recognizer[7]) and needs no adjustment to the classification. Figure 4 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. Figure 5 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 Figure 2. Figure 6 shows the new class boundary after re-classification based on the modified KNN for the samples in the fuzzy area (Figure 2). A straight classification based on the value $\phi_i(x)$

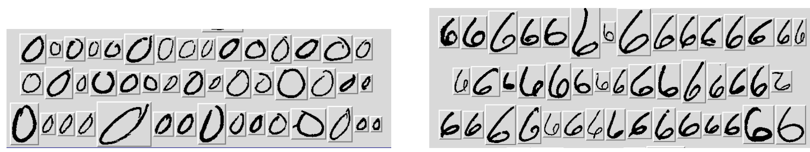

Figure 3: A sampling of "good" quality 0s and 6s. These are samples that receive $\phi_0 \approx 0.6$ and $\phi_6 \approx 0.0$ OR $\phi_0 \approx 0.0$ and $\phi_6 \approx 0.6$

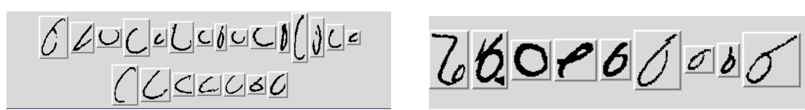

Figure 4: A sampling of "poor" quality 0s and 6s. These are samples that receive $\phi_0 \approx 0.0$ and $\phi_6 \approx 0.6$ OR $\phi_0 \approx 0.6$ and $\phi_6 \approx 0.0$

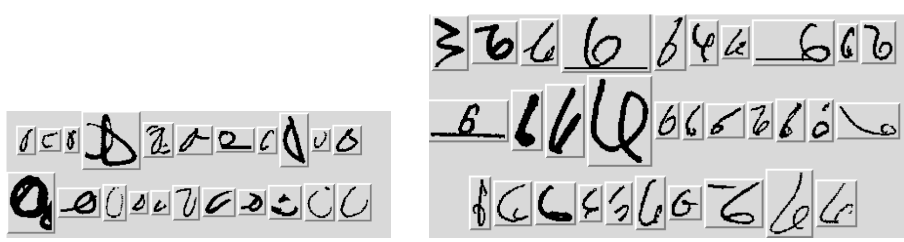

Figure 5: A sampling of "difficult" quality 0s and 6s. These are samples that receive $\phi_6 \approx 0.3$ and $\phi_0 \approx 0.3$ (Figure 2.)

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^o 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.

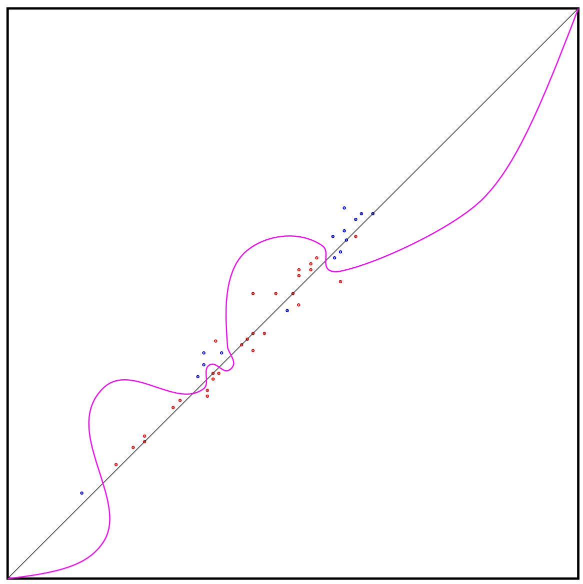

Figure 6: Re-defining the class boundary between 0s and 6s in the fuzzy area of Figure 2

3 KNN Modification

Let us start this section by repeating the assumption that the scores for each class, returned by the recognizer, are real numbers in interval $[0, 0.6]$. 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

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

Let x is an unknown pattern and ω_A and ω_B are the top two class cho ices according to the classifier. Let $W = [0, 0.6]^2$ be the squares in $\mathbb{R}^{\not\vDash}$ with vertices in points $(0,0)$, $(0,0.6)$, $(0.6,0.6)$ and $(0.6,0)$, i.e W is the projection of the set $X_{A,B}$ of all patterns, for which ω_A and ω_B are the top two classes choices according to the classifier, in $\mathbb{R}^{\not\vDash}$ by the map $\Phi_{A,B}$. Using lines parallel to axises, we devide W into n^2 disjoined parallelograms(squares). Let us enumerate these sets by $B_1, B_2, \ldots, B_M$, where $M = n^2$.

For every $y \in W$, there exists a unique neighborhood B_y among $B_1, B_2, \ldots, B_M$, containing y. Let us define the functions

$$\varphi_A(y) = |\{x \in V_A : \Phi_{A,B}(x) \in B_y\}|$$

and

$$\varphi_B(y) = |\{x \in V_B : \Phi_{A,B}(x) \in B_y\}|.$$

Let

$$\eta_A(y) = \begin{cases} \dfrac{\varphi_A(y)}{\varphi_A(y) + \varphi_B(y)} & \text{if } \varphi_A(y) + \varphi_B(y) \neq 0 \\ \\ 0.5 & \text{otherwise} \end{cases}$$

and

$$\eta_B(y) = \begin{cases} \dfrac{\varphi_B(y)}{\varphi_A(y) + \varphi_B(y)} & \text{if } \varphi_A(y) + \varphi_B(y) \neq 0 \\ \\ 0.5 & \text{otherwise} \end{cases} .$$

Now, we are in a position to define the new confidences corresponding to the different classes. Let τ be the threshold that determines the overlapping area. We can define the new confidences as follows:

$$\xi_A(x) = \begin{cases} \eta_A(\Phi_{A,B}(x)) & \text{if } x \text{ is in the overlapping area} \\ \phi_A(x) & \text{otherwise} \end{cases}$$

for class ω_A

$$\xi_j(x) = \begin{cases} \eta_B(\Phi_{A,B}(x)) & \text{if } x \text{ is in the overlapping area} \\ \phi_B(x) & \text{otherwise} \end{cases} .$$

for class ω_B

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

for all other classes. We say that the unknown pattern x belongs to the class ω_i, if $\xi_j(x) < \xi_i(x)$ for all $j \neq i$.

4 Experiments

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 1). To illustrate the significance of the improvement let us look at a simple application. Assuming we use GSC to recognize 5 digits in handwritten ZIP codes, taken from the US 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 has 50,000 images and the testing set has 30,398 images created using digit samples extracted from the US mailstream.

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 "overlapping 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.

One way to evaluate the performance of two recognizers is to compare the corresponding percentage correctly classified samples(Tables 1, 2). Actually,

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

Table 1: **Improvements to GSC recognizer using Fuzzy boundaries**

the best way to evaluate the performance of two classifiers is to compare the their graphs on a REJECT vs ERROR scale (Figure 7). Figure 7 shows the graph of GSC recognizer and the fuzzy method on a REJECT vs ERROR scale in order to compare their performances. Table 3, shows some selected points from the graphs of both recognizers (Figure 7) in order to compare the correct rates of classifiers given the reject rate. At the end, we have performed the similar experiment with other available recognizer and obtain similar results (Table 4).

5 Summary

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

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

Table 2: **Improvements to GSC recognizer using Fuzzy approach for any of the classes**

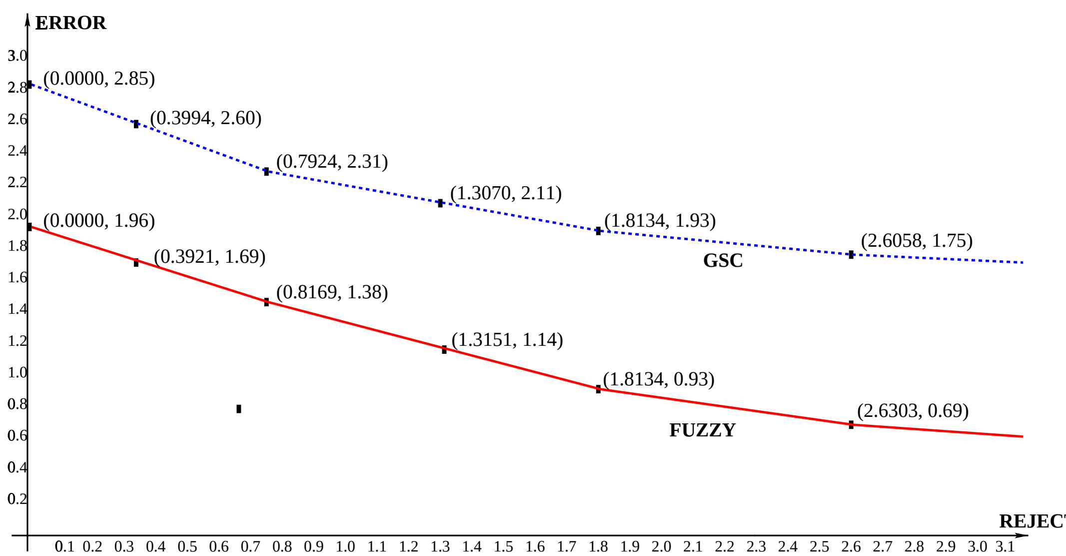

Figure 7: This figure shows how the performances of GSC recognizer and the proposed fuzzy method are compared on a REJECT vs ERROR scale. Based on the fact the graph of the first fuzzy method is upon the graph of GSC, we can conclude that the overall performance of our method is better the one of GSC recognizer

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 3: **This table shows some selected points from the graphs of both recognizers on a REJECT vs 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**

	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

Table 4: **Improvements to others recognizer using fuzzy method**

a large database of samples for retraining.

1. P. Arabie, L. J. Hubert, G. De Soete, editors, "Clustering and Classification", World Scientific, River Edge, New Jersey, 1998.
2. F. Aurenhammer, "Voronoi Diagrams-Survey of a Fundamental Data Structure", ACM Computing Servey, Vol. 23, No. 3, pp.345-405, September 1991.
3. D. Bouchaffra, V. Govindaraju, and S. Srihari, "A Methodology for determining probability of correctness of recognizer scores", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, June 1998.
4. Belur V. Dasarathy, "Nearest neighbor (NN) norms : nn pattern classification techniques", IEEE Computer Society Press, Los Alamitos, California, 1991.

5. G. Carapenter, S. Grossberg, N. Reynolds, D. Rosen, " Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimentional Maps", IEEE Tran. Neural Networks, vol. 3, No. 5. pp.698-713, 1992
6. R. O. Duda, P. E. Hart, D.G.Stork, "Pattern Classification", $2^n d$ edition, 1997.
7. J.T. Favata and G. Srikantan, "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition", Int. Journal of Imaging Systems and Technology, vol. 7, pp. 304-311, 1996.
8. N. Friedman, D. Geiger, M. Goldszmidt, "Bayesian Network Classifier", vol. 29(2/3), pp.131-163, November 1997.
9. A. K. Jain, R. C. Dubes, "Algorithms for Clustering Data", Prentice-Hall, Englewood Cliff, New Jersey, 1988.
10. R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: a decision tree hibrid", Morgan Kaufman, Proc. Sec. Int. Conf. Knowledge Discovery and Data Mining, San Mateo CA, 1996.
11. S. Madhvanath, E. Kleinberg, and V. Govindaraju, "Emperical Design of a Multi-classifier Thresholding/ Control Strategy for Recognition of Handwritten Street Names", Int. Journal of Pattern Recognition and Artificial Intelligence, 1997.
12. James R. Munkres, "Topology, a first course", Prentice Hall, 1975.
13. M. Ramoni, P. Sebastiani, "Bayesian Methods for Intelligent Data Analysis", in M. Berthold, D. J. Hand, editors, "An Introduction to Intelligent Data Analysis", Springer, New York, 1999
14. Walter Rudin, "Real and Complex Analysis", McGraw-Hill Companies, 1986.