Hierarchical Classifiers with Adaptive Computational Cost

C. Rodríguez, I. Soraluze, J. Muguerza, J.I. Martín, G. Álvarez
Computer Architecture and Technology Department,
Computer Science Faculty, UPV/EHU, Aptdo. 649, 20080 San Sebastián, Spain
E-mail: acrneura@si.ehu.es

Abstract

This paper analyses the application of hierarchical classifiers based on the k-NN rule to the automatic classification of handwritten digits. The discriminating capacity of a k-NN classifier increases as the size of the reference pattern set (RPS) increases. This supposes a problem for k-NN classifiers in real applications: the high computational cost required when the RPS is large. In order to accelerate the process of calculating the distance to each pattern of the RPS, some authors propose the use of condensing techniques (i.e. Hart's criterion). Our alternative proposal is based on hierarchical classifiers with rejection techniques that improve the computational cost of the classifier. When classifying a pattern in a superior level, the method takes into account the information got in the classification of the pattern in the previous levels of the hierarchy. We have used 270,000 digits (160,000 digits for training and 110,000 for the test) of the NIST Special Data Base 19 as experimental data set. The best non-hierarchical classifier achieves a recognition rate of 99.38%. The hierarchical classifier maintains the same recognition rate, but with 8 times lower computational cost than the cost of the best non-hierarchical classifier found in our experimentation. In relation to Hart's condensing technique, our approach achieves higher recognition rate (99.38% versus 99.26%) with lower computational cost (2,742 versus 6,297).

1 Introduction

Automatic recognition of handwritten characters or numerals is a typical field of pattern classification methods. Though a wide variety of techniques have already been proposed to recognise unconstrained handwritten characters, a lot of them use the *k*-NN rule (*k*-Nearest Neighbour) [1, 2] to select the patterns which take part in the classification of a new pattern. These patterns are selected from a previously learned set that is used as reference. In the *k*-NN rule, the class assigned to the pattern to be classified is determined depending on the classes of the *k* nearest patterns, and the selection criterion is the distance between the patterns in a particular metric space.

Our previous experience with typed digits [3,4] corroborates the good behaviour of these classification methods. In the experimentation carried out, features of global nature -zoning (with an 8x5 matrix)- and structural nature have been used. The robustness of the global feature with breaks, blurring and other deformations in the images of the digits, and the good behaviour of the *k*-NN classifier, has brought us to select them as starting point for our experimentation with handwritten digits.

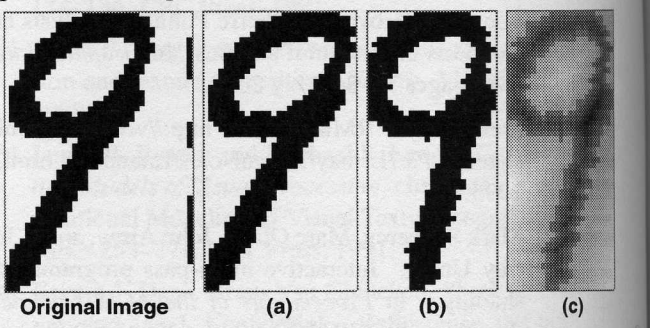


Figure 1. Preprocessing of the digit image.

In order to minimise the influence of the diversity of the structural forms of the handwritten digits, we have made some preprocesses that improves the classification process [5, 6]. In particular, the following transformations have been used [7]: (i) a filter for the digits images, with the aim of diminishing the negative influence of the segmentation phase in the recognition phase, and (ii) a normalisation in inclination and rotation of the digit image. In the first transformation, up to a 25% of the lateral columns of the image are eliminated when, at least, a whole white column is found in the image. In the top and bottom part of the digit the same process has been made, in this case with a limit of 10% of the rows (Figure 1a). The normalisation in inclination and rotation is based on the linear regression of the image of the digits (Figure 1b). Finally, we have applied the technique of contour

lines to the digit image for minimizing the influence of the digits thickness on the process of getting the feature vector. Taking as starting point the digits skeleton for black pixels and the digits contour for white pixels, a decreasing weight based on the distance of the pixel to the nearest starting point is associated (Figure 1c).

We have taken the Special Database 19 from NIST (NIST SD19) [8] as experimental base for the handwritten digits. This database contains about 380,000 handwritten digits. The authors indicate that a one by thousand is wrongly labelled. We have selected 160,000 digits for training and 110,000 different for the test. The number of patterns is the same for every class and these patterns

have been chosen randomly for each class. In order to study the evolution of the classifier according to the number of patterns learned we have defined five *Training Pattern Sets* (TPS), each one including the previous one, with 10,000, 20,000, 40,000, 80,000 and 160,000 patterns (labelled in the Figures with 10K, 20K, 40K, 80K and 160K). For the training process, a *1*-NN classifier has been used. A new pattern has been included in the reference set using the following criteria: (a) when the training pattern is misclassified, and (b) when, being correctly classified, the relative distance between the two nearest patterns of different classes is smaller than a certain percentage. The description of this algorithm is in Figure 2. In our classification space, there is no point in using a training percentage higher than 50%, since the recognition rates associated to the percentages 100% and 50% are very similar. The metric space used for the experimentation has been the Euclidean.

One of the aspects that makes difficult the use of classifiers as the *k*-NN in real applications is the high computational cost they require when the Reference Patterns Set (RPS) used is large. In previous works, we have presented an alternative based on hierarchical classifiers, which improves global classifier performance in computational cost maintaining the recognition rate. The computational cost is given as the cost of classifying each digit. Since the feature space and the metric used are the same for all the classifiers, the computational cost is calculated using the cardinal of the RPS (the number of operations required to classify a digit is directly proportional to this cardinal). Particularly, for individual classifiers the computational cost is the cardinal of the RPS. In hierarchical classifiers the computational cost is computed as the average of the computational cost required to classify all the digits. The computational cost to classify a digit is the sum of the cardinal of the RPS taking part in its classification. The best non-hierarchical classifier found in our experimentation achieves a recognition rate of 99.38% (with a TPS of 160K and a training percentage of 50%), with a computational cost of 21,912. In [9], we present hierarchical classifiers with two and three levels that maintain the recognition rate, but reduce the computational cost up to 4,051, which supposes a speed-up of 5.41.

In a hierarchical classifier, the last levels of the hierarchy are composed of more complex classifiers, with a bigger RPS size. This provides more information about the distribution of the patterns in the classification space to carry out a correct classification of patterns that could be misclassified in previous levels. The increase of the

$$
\begin{array}{l}
S = 0 \\
RPS_0 = \varnothing \\
TPS_0 = TPS \\
Repeat \\
\quad S = S + 1 \\
\quad RPS_S = RPS_{S-1} \\
\quad TPS_S = TPS_{S-1} \\
\quad \forall X \in TPS_S \\
\qquad given \quad Y \in RPS_s \; the\; 1-\text{NN}\; of\; X \\
\qquad\qquad Z \in RPS_s \; the\; 1-\text{NN}\; of\; X \; with\; class(X) \neq class(Z) \\
\qquad if\left[class(X) \neq class(Y)\right]\; or \\
\qquad \left[\left(class(X) = class(Y)\right)\; and \left(\frac{dist(X,Z) - dist(X,Y)}{dist(X,Z)} \times 100 < training\_percentage\right)\right] \\
\qquad then\; RPS_s = RPS_s \cup \{X\} \\
\qquad\qquad TPS_s = TPS_s - \{X\} \\
until\left(size(RPS_{s-1}) = size(RPS_s)\right) \\
where \\
\quad S = step\; of\; the\; training\; process \\
\quad RPS_s = Reference\; Pattern\; Set\; on\; the\; step\; S \\
\quad TPS_s = Training\; Pattern\; Set\; on\; the\; step\; S \\
\quad X, Y, Z\; are\; patterns\; of\; the\; TPS \\
\quad dist(v,w) = is\; the\; distance\; between\; the\; patterns\; v\; and\; w\; in\; the\; metric\; selected
\end{array}
$$

Figure 2. Training algorithm.

classifiers complexity along the hierarchy makes indispensable minimising the computational cost associated to each patterns classification. We propose the use of information provided by the hierarchy first levels to make a selective search in the different RPS - only among the patterns of the most probable classes - with the corresponding reduction of the computational cost.

The paper has the following structure. Section 2 summarises the previous experimentation using hierarchical classifiers. These classifiers reduce the computational cost in comparison with non-hierarchical classifiers. Section 3 presents the technique used to take advantage of the pattern classification information in the previous levels of the hierarchy when the pattern is classified in later levels. This scheme minimises the computational cost of the classifier maintaining the recognition rate. Finally, Section 4 is devoted to the conclusions and future work.

2 Hierarchical classifier

One of the factors with more influence on the efficiency of the neighbourhood-based classifiers is the size of the RPS. On the one hand, it is better to have as many patterns as possible to have available more information about the distribution of the classes in the classification space. On the other hand, the computational cost associated to the classification is bigger as the number of patterns used as reference grows, since to classify a new pattern is necessary to calculate the distance to each pattern of the RPS and to select the k nearest patterns. That is the reason why some techniques to accelerate this process are being studied [1]. Some methods try to reduce the size of RPS without losing classification power (condensing techniques) [10, 11, 12, 13, 14].

This Section summarises an alternative proposal based on hierarchical structures. The objective is to build classifiers minimising the computational cost but keeping the recognition rate. We propose the use of a hierarchy with the different classifiers we have got in an independent way with the different training sets and training percentages. In a hierarchical classifier with rejection techniques in the classifier levels, a pattern to be classified will use the next level of the hierarchy depending on the certainty its classification has in the present level. The certainty of a pattern classification in a level depends on the number of votes the class gets within the k nearest patterns (Voting k-NN). Three certainty percentages have been applied: 50%, 75% and 100% of the votes within the k nearest patterns. The *Distance-Weighted k-NN* (Weighted k-NN) [15] has been used in the last level of the hierarchical classifier.

We have defined an ordination criterion for the different classifiers. A classifier in a superior level determines the possible classifiers in the first levels: they can not use more training patterns than a classifier in a superior level and, in addition, the size of the RPS must be smaller. Therefore, when advancing in the hierarchy, the classifiers have bigger RPS. This supposes a higher classification capacity, but also more computational cost.

Although we have proved hierarchies with more than three levels, we have made an exhaustive study for hierarchies with two and three levels: 945 hierarchies of two levels and 52,920 of three levels [9]. Increasing the certainty percentage of the classifiers in the hierarchy, the behaviour of the recognition rate is more stable, but the reduction in computation is smaller. Nevertheless, the configurations of three levels reduce the computational cost considerably more than the two level ones.

Figure 3 shows a comparison between some reference cases: non-hierarchical classifiers trained with a percentage of 0%, *nonHier_0%*, (classifier with the lowest computational cost), and with a 50%, *nonHier_50%*, using the different training sets (20K, 40K, 80K and 160K), and the best hierarchical classifiers found with these percentages, *Hier_0%* and *Hier_50%*. As it can be observed, the recognition rate is similar for all training sets with the same learning percentage (0% or 50%). Nevertheless, the computational cost of hierarchical classifiers is much lower than the individual classifier cost. In this experimentation with handwritten digits, the best recognition rate reached by a non-hierarchical classifier is 99.38%, with a computational cost of 21,912. The best hierarchical classifier reaches the same recognition rate with a computational cost of 4,051. This classifier uses a classifier trained with a TPS of 160K and a training percentage of 50% in the last level of the hierarchy.

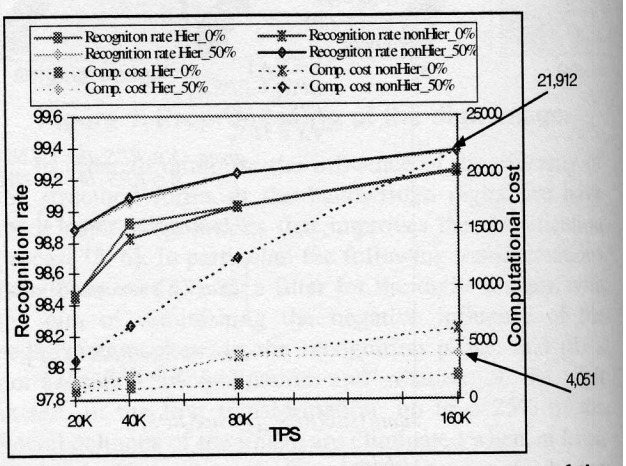


Figure 3. Computational cost improvement of the hierarchical classifiers comparing with non-hierarchical ones.

Taking into account this result, the following section presents a method to minimise the computational cost in the hierarchical classifiers. When a pattern is rejected in a

level, we propose the use of the information about the classification in that level to minimise the computational cost of the search in the next levels.

3 Adaptive computational cost in hierarchical classifiers

To carry out the classification process, a k-NN classifier looks for the k nearest patterns in its RPS. In our experimentation, we have worked with a maximum value of 23 for k. In a level of the hierarchy, an optimum k value is selected according to the specific classifier to make voting in this level. This optimum k value has been found among the values in the range {9..15} for all the proved classifiers. When a pattern is rejected in a level, next level of the hierarchy will try to classify it. This next classifier of the hierarchy will be more complex than the classifier of the previous level. This way, the classifier will have available more information about the distribution of the patterns in the classification space and it will be able to classify correctly more complex patterns.

So far, the pattern classification in a level does not use the information obtained in the previous level. However, this information can be used to search the k nearest patterns in the new RPS only among the patterns belonging to the classes more voted in the previous level. The computational cost is reduced, and possible interference among classes is eliminated. Anyway, the recognition rate should be the same or similar.

When a pattern is rejected in a level, we propose that the classification in the next level uses the information of the previous level classification in two different ways. On the one hand, taking into account only the n most voted classes in the previous level in order to search in the new RPS, and, on the other hand, combining the 23 nearest patterns obtained in the previous levels with the 23 nearest patterns corresponding to the new RPS. This way, the information of the previous level classification is reused to estimate the membership of the pattern to a class in a superior level of the hierarchy.

The value n is variable and depends on the number of different classes found in the 23 nearest patterns of the previous level. Experimentally, it has been proven that the correct class of the pattern is inside these n classes in a 99.97%. In our experimentation, the error due to the possibility of not having found the correct class of the pattern in this subset is minimum, 35 patterns of 110.000 test patterns. The Figure 4 shows the number of patterns whose class is not in the class subset of the k nearest patterns in function of the k value. These patterns would be irrecoverable errors for the classifier. Therefore, the election of the k value will depend on the error rate that the application can allow. The influence of small values of k will be bigger in the final error rate.

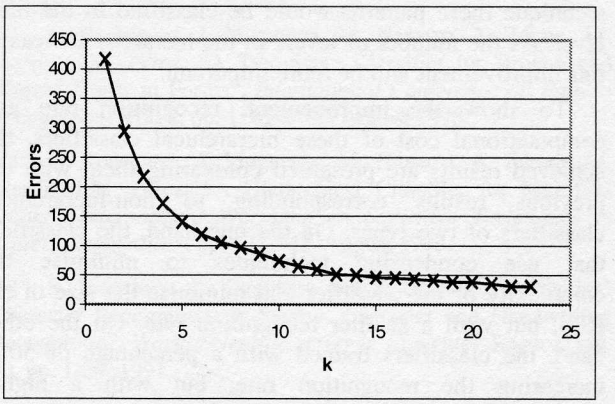


Figure 4. Distribution of the errors due to the selection of some candidate classes in function of *k*.

Table 1 shows the mean distribution of classes got in the classification of the test patterns for different hierarchical classifiers. It can be seen that in 107,720 patterns, in the 97.99% of the cases, the correct class of the pattern is the most voted class and in 86,352 patterns,

Order of the correct class	Number of different classes in the 23 nearest patterns										TOTAL
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	
1st	1,299	22,113	35,038	26,713	12,960	5,093	3,618	861	25	—	**107,720**
2nd	—	146	409	445	377	197	75	9	1	—	**1659**
3rd	—	—	67	81	103	65	27	9	1	—	**353**
4th	—	—	—	19	39	23	20	2	1	—	**104**
5th	—	—	—	—	17	12	9	1	1	—	**40**
6th	—	—	—	—	—	5	1	2	—	—	**8**
7th	—	—	—	—	—	—	3	1	—	1	**5**
8th	—	—	—	—	—	—	—	1	—	—	**1**
Not found	—	3	8	11	8	3	2	—	—	—	**35**
TOTAL	**1,299**	**22,262**	**35,522**	**27,269**	**13,504**	**5,398**	**3,755**	**886**	**29**	**1**	**109,925**

Table 1. Distribution of different classes and position of the correct class for the test set.

the 78.56%, only four different classes appear in the 23 nearest patterns. This provides an important reduction in the computational cost of the patterns classification in the next level because it is not necessary to examine more than those four classes in the whole new RPS.

We have also kept in mind the 23 nearest patterns obtained in the classification of the previous level ordered together with the 23 new patterns (obtained searching only among the patterns belonging to the classes previously selected as candidates) and, finally, the repeated patterns have been eliminated. The application of this technique reduces the computational cost because the patterns tend to be classified in former levels, and it does not reduce the recognition rate. In the case of three level hierarchies, on average, there are 600 patterns more classified in the second level, whereas without using this technique these patterns would be classified in the third level. As the number of levels in the hierarchy increases, this improvement will be more important.

To show the improvement, recognition rate and computational cost of these hierarchical classifiers, the achieved results are presented comparing them with the previous results corresponding to non-hierarchical classifiers of two types. On the one hand, the classifiers that use condensing techniques to minimise the complexity of the classifier and minimise the size of the RPS, but with a smaller recognition rate. On the other hand, the classifiers trained with a percentage of 50%, increasing the recognition rate, but with a higher computational cost. These two classifiers are the reference points, computational cost/recognition rate, in our experimentation. Among the different condensing techniques, the technique we have chosen to carry out the experimentation has been Hart's criterion [10]: it trains the classifier with a training percentage of 0%.

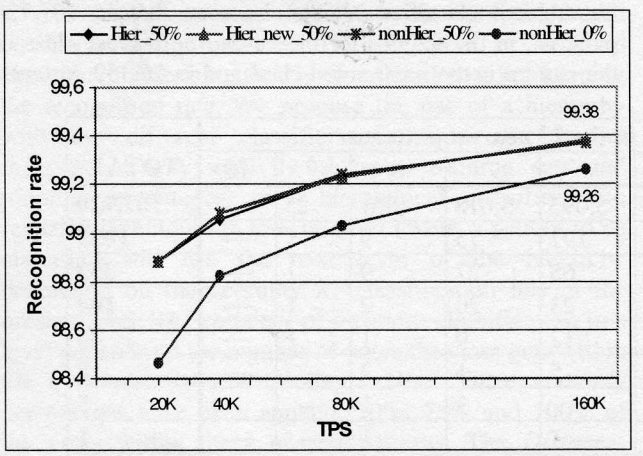
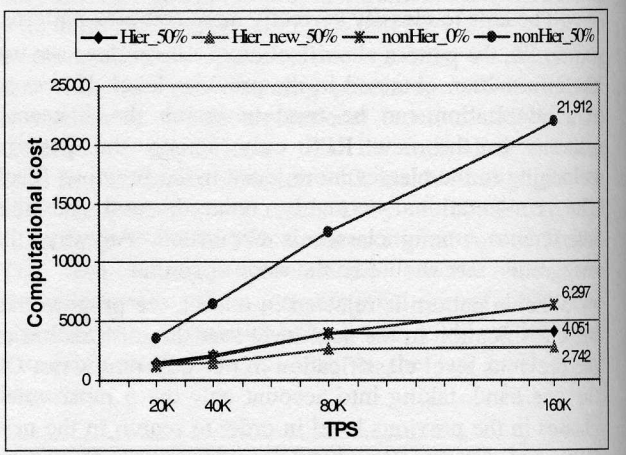
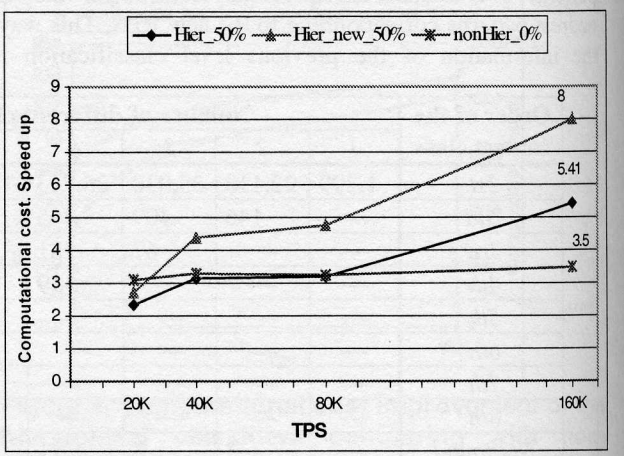


Figure 5. Recognition rate achieved with non-hierarchical and hierarchical classifiers.

The Figures 5 and 6 present the recognition rate and computational cost results corresponding to the experimentation carried out using hierarchical classifiers under these new conditions compared with the non-hierarchical classifiers referenced in the previous paragraph. The Figure 5 presents the recognition rate obtained for non-hierarchical classifiers (with training percentages of 0%, *nonHier_0%*, and 50%, *nonHier_50%*), hierarchical classifiers without using the information of the previous levels, *Hier_50%*, and hierarchical classifiers that use this information, *Hier_new_50%*. The hierarchical classifiers of this figure are four sets of classifiers of 3 levels finished in a classifier trained with a TPS of 20K/40K/80K/160K, all of them with a training percentage of 50%. As it can be seen, the recognition rate is the same for the classifiers that use a 50% of training percentage and superior to the recognition rate of the classifiers trained using the Hart's condensing technique (99.38% versus 99.26%). It can be seen that the use of these new approaches in the classification of the different levels in the hierarchy maintains the global recognition rate.



(a)



(b)

Figure 6. (a) Computation cost of the classifiers of Figure 5, and (b) speed-up achieved.

Figures 6a and 6b present the computational cost corresponding to the same classifiers presented in Figure 5. Figure 6b presents the speed-up achieved taking as reference the best non-hierarchical classifier, *nonHier_50%*. The improvement of the proposed methods is remarkable: the hierarchical classifier *Hier_new_50%* has a computational cost of 2,742, whereas the initial cost was 21,912 (classifier *nonHier_50%*), with 8 times lower computational cost. In regarding to the hierarchical classifier that does not take into account the information of the previous levels, *Hier_50%*, the achieved speed-up is 1.5 (it has a computational cost of 4,051). In all the cases, these hierarchical classifiers get a recognition rate of 99.38%, superior to the recognition rate obtained with a classifier trained with the Hart's approach, and with a smaller computational cost, 2,742 versus 6,297.

4 Conclusions

The use of hierarchical classifiers is an alternative to the classifiers that use condensing techniques to reduce the classification computational cost: they get a higher recognition rate with a smaller computational cost. Together with the use of hierarchical classifiers, when a pattern is rejected in a level, this work proposes that the classification in the next level uses the information of the previous level classification. On the one hand, we propose to take into account only the n most voted classes in the previous level in order to search in the new RPS. On the other hand, the proposed classifier combines the k nearest patterns obtained previously with the k nearest ones corresponding to the new RPS.

In the experimentation with handwritten digits, using the NIST Special Data Base 19, the best recognition rate reached by an individual classifier is 99.38%, with a computational cost of 21,912. The proposed hierarchical classifier reaches the same recognition rate with a computational cost of 2,742.

We are using more complex selection techniques that apply the symmetry concept to the selection of the k nearest patterns (k-NCN), reaching better results. We are also working on increasing the feature dimensionality (13x8) to improve the recognition rate for certain patterns, when a resolution of 8x5 is insufficient. Both aspects increase considerably the computation needed to classify a pattern. For this reason, it is still more necessary the use of hierarchical classifiers with more than three levels. In this case, the proposals here outlined will be to the advantage of a bigger reduction of the computational cost. The first experiments carried out have reached a recognition rate of 99.58%.

5 Acknowledgements

This work was supported in part by the University of the Basque Country (project number UPV139.226-G25/98), by the Spanish Government (CICYT project number TIC2000-0389), and by Gipuzkoako Foru Aldundia. We also want to thank the collaboration of the company ADHOC Synectic Systems, S.A.

6 References

[1] B. V. Dasarathy, *Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques*, Ed: IEEE Computer Society Press, 1991.

[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysi,*. Ed: Wiley, New York, 1973.

[3] C. Rodríguez, J. Muguerza, M. Navarro, A. Zárate, J.I. Martín, J.M. Pérez, "A Two-Stage Classifier for Broken and Blurred Digits in Forms", *International Conference on Pattern Recognition*, Vol. 2, pp. 1101-1105, August 1998.

[4] C. Rodríguez, J. Muguerza, M. Navarro, A. Zárate, J.I. Martín, J.M. Pérez, "A Hierarchical Classifier for Multifont Digits", *International Workshop on Statistical Techniques in Pattern Recognition*, pp. 937-943, August 1998.

[5] S. Impedovo, L. Ottaviano and S. Occhinegro, "Optical Character Recognition. A Survey. Character and Handwriting Recognition", *World Scientific series in Computer Science*, vol. 30, pp. 1-24, 1991.

[6] J. Skrzypek and J. Hoffman, "Visual Recognition of Script Characters and Neural Network Architectures", *Neural Networks: Advances and Applications*, Gelenbe Elsevier S.P. (North-Holland), 1991.

[7] C. Rodríguez, I. Soraluze, J. Muguerza, J.I. Martín, G. Álvarez. "Transformations and Neighbourhood Algorithms for the Classification of Handwritten Digits: an Experimental Study", *5th Iberoamerican Symposium on Pattern Recognition*, pp. 111-121, September 2000.

[8] P. J. Grother and G. T. Candela, *Comparison of Handprinted Digit Classifiers. NIST: Technical Report NISTIR 5209*, National Institute of Standards Technology, June 1993.

[9] C. Rodríguez, I. Soraluze, J. Muguerza, J.I. Martín, G. Álvarez, "An Alternative to the Application of Condensing Techniques in k-NN Classifiers", accepted for presentation on the SNRFAI-2001 Conference, 2001.

[10] P.E. Hart, "The Condensed Nearest Neighbor Rule", *IEEE Transactions on Information Theory*, vol. 14, pp. 515-516, 1968.

[11] I. Tomek, "Two Modifications of CNN", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6, pp. 769-772, 1976.

[12] G.T. Toussaint and R.S. Poulsen, "Some New Algorithms and Software Implementation Methods for Pattern Recognition Research", *Proceedigns of the 3rd International COMPSAC*, pp. 55-63, 1979.

[13] G.T. Toussaint, "The Relative Neighbourhood Graph of a Finite Planar Set", P*attern Recognition*, vol. 12, pp.261-268, 1980.

[14] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis", *Systematic Zoology*, vol. 18, pp.259-278, 1969.

[15] S. A. Dudani, "The Distance-Weighted *k*-Nearest-Neighbor Rule", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-6, pp. 325-327, April 1976.