Isolated Character Recognition by using Support Vector Machines and an Optimal Feature Vectors

Hermenegildo Cisneros, Ricardo Barrón and Benjamín Cruz
Centro de Investigación en Computación del IPN
Av. Juan de Dios Batís esquina Miguel Otón de Mendizábal
Colonia Nueva Industrial Vallejo
México, D. F. 07738, MÉXICO
E-mail: rbarron@cic.ipn.mx

(Paper received on September 06, 2006, accepted on September 27, 2006)

Abstract. In this work, we present a combination of techniques for the recognition of isolated hand-written characters. We describe a way to obtain the feature vector that allows us to describe an isolated character. Flusser invariant moments and other geometric measures are combined as object descriptors. We implement an off-line system. We use so-called Support Vector Machines [SVM] as the main classifiers.

1. Introduction

An object recognition system usually incorporate three main modules (Figure 1):

1. Data acquisition and preprocessing module.
2. Feature extraction module.
3. Decision (classification) module.

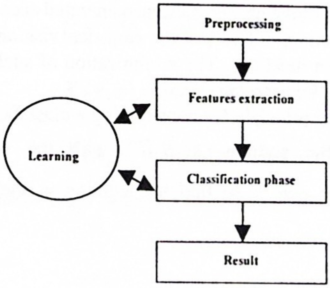

Figure 1. The three modules of an object recognition system.

After an image of an object (in this case the an image of the objects to recognize) is acquired, it is analyzed to get a compact description of the objects in the image. This compact representation model must fulfill certain desirable properties:

© H. Sossa and R. Barrón (Eds.)
Special Issue in Neural Networks and Associative Memories
Research in Computing Science 21, 2006, pp. 227-235

a) Unicity. Each object must have a unique representation.
b) Invariance. The description should experiment no changes under a group of image transformations: translations, rotations, changes of category, reflections, and the like.
c) Sensitivity. Capacity to discriminate equal objects.

These kind of descriptions are the so-called external representation schemes, which make use of the contour of the objects and its features, such that chain codes, Fourier descriptions, polygonal approaches. There are also the so called schemes of internal representation, that describe the region occupied by the object in a binary image, such area, perimeter, compactness factor, the moments, an so on. Some of these schemes can be found in [LP].

In this work we show how a careful selection of descriptors can help to recognize isolated written characters. In this work we have selected the well-known Flusser invariant moments, the elongation of the object, compactness factor, eccentricity and size. Flusser invariants have been chosen due to their capacity to recognize projected objects.

For classification we have used the multi-class support vector machines because their great capacity of classification, and because they are a safe alternative in regards to the most common methods: Bayesian, KNN, back propagation, among others.

2. Support vector machines

Most of pattern recognition methods are based on finding a classification function that minimizes the *empirical risk*, the error measure of the given patterns. The theory developed by Vapnik and Chervonenkis provides upper bounds of this *structural risk*, the error of misclassification of the unseen patterns generated according to an unknown but fixed probability distribution, in terms of the empirical risk and the VC-dimension of the family of classification functions. The minimization of such bounds follows the principle of *Structural Risk Minimization* [SRM].

Following the mentioned principle for constructing a classification function f(x) = sing (w·x+b) based on the patterns $x_i \in R^n$ with the corresponding labels $y_i \in \{-1,1\}, i = 1,\ldots,l$ the goal is to find the solution w and b of the following optimization problem:

$$\text{Minimize} \quad \frac{1}{2}\|w\| + C\sum_{i=1}^{l} \xi_i \tag{1}$$

$$\text{Subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \textit{with} \quad \xi_i \geq 0.$$

Here the quantity $\frac{1}{2}\|w\|$ the *margin*, is related to the VC dimension and is maximized. The parameter C can be regarded as a regularization parameter. To save some numeric and implementation problems, the problem (1) is transformed using the technique of Lagrange multipliers into the following dual problem.

$$\begin{aligned} Max \quad & W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i\cdot x_j \\ s.t. \quad & \sum_{i=1}^{l} y_i\alpha_i=0, \quad 0\le\alpha_i\le C, \quad i=1,\dots,l \end{aligned} \tag{2}$$

An important characteristic of SVMs is that f can be expressed in terms of the training vectors x_i for which the solution α_i of (2) are positive, called the *support vectors* (*SVs*) [SV], by setting w and b as follows:

$$w=\sum_{\alpha_i\in SV}^{l} y_i\alpha_i(x_i\cdot x),\ b=\frac{1}{2}(w\cdot x_+ + x_-) \tag{3}$$

where x_+ and $x\_$ are two SVs which belong to the positive and negative class, respectively. The SVs are interpreted as the relevant patterns in the classification process. Frequently, the number of SVs is small with respect at the number of training data; this gives a compact representation and efficient implementation of the classifier. Other important characteristic of SVM is that non-linearity can be introduced by replacing the inner product in (2) by a *kernel*

$$K(x_i\cdot x_j)=\phi(x_i)\cdot\phi(x_j),$$

where ϕ maps the patterns into a high (possibly infinity) dimensional inner product space. Some of the best known (and used) kernel functions are:

$$K(x,z)=\exp(-\gamma\ x-z\ ) \tag{4}$$

$$K(x,z)=\exp(\gamma((x\cdot z)+1))^d \tag{5}$$

$$K(x,z)=\tanh(\gamma(x\cdot z)-\theta) \tag{6}$$

which are know as *radial basis functions* (*RBFs*), *polynomial kernels* and *hyperbolic kernels* respectively.

2.1. Multi-class support vector machines

2.1.1. One-against-all (oaa) SVM

Consider an n-class problem. For oaa [OAA] SVM, we determine n direct decision functions that separate one class from the remaining classes. Let the ith decision function, with the maximum margin that separates class i from the remaining classes, be

$$D_i(x) = w_i^T g(x) + b_i \tag{7}$$

where w_i is the l-dimensional vector, $g(x)$ is the mapping function that maps x into the l-dimensional feature space, and b_i is the bias term.

The hyper plane $D_i(x) = 0$ forms the optimal separating hyperplane, and if the classification problem is separable, the training data belonging to class i satisfy $D_i(x) \geq 1$ and those belonging to the remaining classes satisfy $D_i(x) \leq -1$. Especially, support vectors satisfy $y_i D_i(x) = 1$. If the problem is not separable, unbounded support vectors satisfy $y_i D_i(x) = 1$ and bounded support vector satisfy $y_i D_i(x) \leq 1$. Remaining training data satisfies $y_i D_i(x) \geq 1$.

In classification, if for the input vector x

$$D_i(x) > 0 \tag{8}$$

is satisfied for one i, them x is classified into class i. This is due to only the sign of the decision function is used. The decision is discrete.

If Eq. 8 is satisfied for plural is or if there is no i that satisfies Eq. 8, x is unclassifiable.

3. Implementation

3.1. Pre-processing

The pre-processing [LP] phase begins with the capture of the image to process; it is in BMP or JPG format. Figure 2 shows an example of an image with 5 isolated characters.

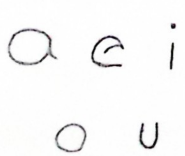

Figure 2. Original image

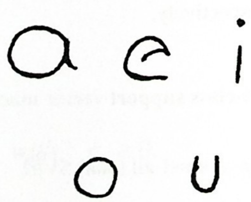

Figure 3. Threshold image.

Next a process to convert the image to grey levels is made, here the next phases are applied:

Thresholding. In this phase the image is taken from grey levels and a threshold is applied to it, so that the new image in two is obtained: 0 (black for the objects) and 1 (white for the background) (Figure 3).

Edge detection [DIPE]. Edges of each character are detected to find the limits among them (Canny's detector was used). Figure 4 shows the edges detected from figure 3.

Figure 4. Detection of edges

Figure 5. Dilatation of characters

Dilation [DIPD]. The edges of each character are dilated. This because in some cases its lines are not completely closed or there are some isolated dots nearly at the main lines. By using dilatation these dots are joined at the lines. More formally, the dilatation of A by B note $A \oplus B$ and is defined by means of:

$A \oplus B = \{c \in E^n \mid c = a + b \textit{ for some } a \in A \ y \ b \in B\}$. As the addition is commutative, the dilatation also is it $A \oplus B = B \oplus A$.

It is possible to demonstrate that the following definitions of the dilatation are equivalent:

$$A \oplus B = \left\{x \mid \left(\hat{B}\right)x \cap A \neq 0\right\} \cup b \in BA_b$$

Actually the A and B sets are not symmetrical. The first element of the dilatation, A, is associated with the image being processed, the second one is named as structural element. The result of applying B on A produces the dilatation of A, $A \oplus B$ (Figure 5).

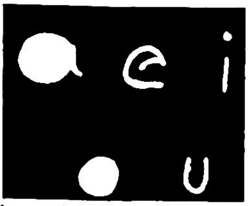

Figure 6. Filling of characters

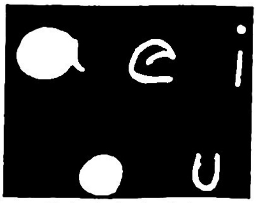

Figure 7. Labeled of characters.

Filling. A filling of pixels is made at the characters, to give them greater form. Figure 6 shows the filled characters of Figure 5.

Labeling [LP]. Labelling of characters is made to introduce them in the classifier, by using the labelled of connected components. Later we get the features vector of each object. Figure 7 shows the labelled characters from Figure 6.

3.2. Distinction of characters

Before continuing with the following stage of the system, some aspects are mentioned that we considered can help us to make a greater distinction of the characters. For example, we can observe that some high characters as the capital letters and some small letters exist (b, d, h, g, f, j, k, l, q, p, t...), as well as we observed low characters and in this category we found to the remaining small letters.

Also the ambiguous characters exist that are those in which the version in capital letter is morphologically equal to the small letter (K/k, P/p, Y/y, I/i I/l), and other that are not ambiguous but their height are very similar (P/f, h/k, j/g) among others.

In this sense, our system makes the classification on the basis of some of the aspects mentioned before. For example, if the character to recognize is greater to certain height, then it belongs to the category of high characters, otherwise to the class of low characters.

3.3. Feature vector

Since we have mentioned early, to get the feature vector of the objects we have used a combination of techniques. This with the idea to find a set of optimal features that allows us to recognize isolated handwritten characters.

In this work we show a selection of descriptors using the Flusser invariant moments, due to the capacity that they have to recognize projected objects (we considered that they can be very useful since many handwritten characters have this characteristic of projection) in addition to the elongation of the object, compactness factor, eccentricity and size.

Flusser invariant moments

Affine invariant moments [IMF] are derived by means of the theory of algebraic invariants, are invariant under general affine transformations.

$$u = a_0 + a_1x + a_2y$$
$$v = b_0 + b_1x + b_2y \tag{9}$$

The affine transformation (1) can be discomposed in six parameters of transformation:

$$\begin{array}{llllll} u = x + a & u = x & u = \omega \cdot x & u = \delta \cdot x & u = x + t \cdot x & u = x \\ v = y & v = y + \beta & v = \omega \cdot y & v = y & v = y & v = t' \cdot x + y. \end{array}$$

Any function F of moments that are invariant under these six transformations will be invariant under the general affine transformation. Next, we show the parameters used for the obtaining of the feature vector:

$$I_1 = (u_{20}*u_{02} - u^2_{11})/u^4_{00}$$

$$I_2 = (u^2_{30}*u^2_{03} - 6*u_{30}*u_{21}*u_{12}*u_{03} + 4*u_{30}*u^3_{12} + 4*u^3_{21}*u_{03} - 3*u^2_{21}*u^2_{12})/u^{10}_{00}$$

$$I_3 = (u_{20}*(u_{21}*u_{03} - u^2_{12}) - u_{11}*(u_{30}*u_{03} - u_{21}*u_{12}) + u_{02}*(u_{30}*u_{12} - u^2_{21}))/u^7_{00}$$

Compactness factor

$$F = \frac{4*\pi*A}{P^2}, \quad \pi = 3.1416.$$

P = perimeter.
A = Area.

Elongation

$$E = \frac{P^2}{A}.$$

Size

$$Size = \frac{P}{A}.$$

Eccentricity

$$e = \sqrt{1 - \frac{b^2}{a^2}}.$$

b: smaller semi axis
a: greater semi axis.

4. Experimental results

The tests were made with the vowels and the numbers of 0-9.The letters were written by a person. For the case of the vowels, 50 letters were used for each class, that is, a total of 250 samples. For each class we selected in random way 50% of data for training and 50% for the tests.

For the case of the numbers, 50 digits were used by each class, that is, a total of 500 samples. As it happened with the vowels, we selected at random 50% of data for training and 50% for the tests.

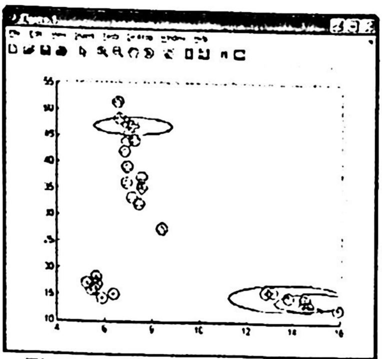

Figure 8. Training vowels with oaa SVM.

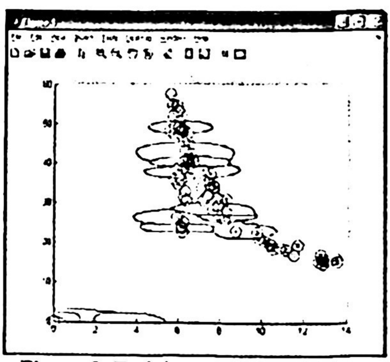

Figure 9. Training numbers with oaa SVM.

Our system uses oaa (one against all) as multi-class SVM classifier. We use the RBF kernel for all classification nodes with the implementation of optimized minimum sequence [SMO] to train and a constant of 10. The programming was made in Matlab using a Toolbox of Support Vector Machines [Stprtool].

To compare the results we trained a neural network with the RPROP algorithm (with standard parameters values) and the Euclidean classifier.

In Figures 8 and 9 we respectively show the training of data of the vowels and the digits with multi-class oaa SVM.

The following tables summary the results obtained for each classifier.

Table 1. Vowel recognition.

Vowels	SVM	ANN	Euclidian
a	100%	100%	90%
e	100%	100%	90%
i	100%	100%	90%
o	100%	100%	80%
u	100%	100%	80%
Total	**100%**	**100%**	**86%**

Table 2. Number recognition.

Numbers	SVM	ANN	Euclidian
0	100%	100%	90%
1	100%	100%	90%
2	90%	90%	90%
3	70%	70%	60%
4	90%	90%	80%
5	60%	50%	50%
6	100%	100%	90%
7	100%	100%	80%
8	100%	100%	90%
9	100%	100%	90%
Total	90.1%	90%	81%

5. Conclusions

In this work we have shown that for isolated character recognition support vector machines are an efficient way of classification.

We have combined the three first Flusser moments, the compactness factor, eccentricity, size and elongation of the object as features to describe the objects. In this case we have observed that the percentage of recognition is considerable high for the five vowels and the numbers.

We have noticed that the percentage of recognition is not sufficient, because the goal is to increase it until one hundred percent, reason why still lack to find a better set of features vector that get this purpose.

References

[SVM] Support vector machines for pattern classification / Shigeo Abe, Page 83.

[SRM] V. Vapnik. Statistical Learning Theory. John Wiley & Sons, New York, 1998.

[IMF] Pattern recognition, 1993, vol. 26, pag. 167 – 174.

[LP] Handbook of character recognition and document image analysis. Pag. 35, 41-42.

[DIPE] Digital Image Processing. Rafael C. Gonzales, Pag. 416.

[DIPD] Digital Image Processing. Rafael C. Gonzales. Pag 518.

[ER] Recognition of On-line Handwritten Mathematical Formulas in the E-Chalk System. Ernesto Tapia and Raúl Rojas, Free University of Berlin Institute of computer Science Takurstr 9, D-14195 (ICDAR 2003).

[LR] Support Vector Machines for Handwritten Numerical String Recognition. Luiz S. Oliveira and Robert Sabourin, Pontificia Universidade Católica de Paraná, Curitiba, Brazil, Ecole de Technologie Supérieure – Montreal, Canada

[Stprtool] Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University Technick´a 2, 166 27 Prague 6, Czech Republic

[OAA] Support vector machines for pattern classification, Shigeo Abe, page. 84.

[SV] Support vector machines for pattern classification, Shigeo Abe, page. 18.

[SMO] Support vector machines for pattern classification, Shigeo Abe, page. 166