IDIAP RESEARCH REPORT

CURSIVE CHARACTER CHALLENGE: A NEW DATABASE FOR MACHINE LEARNING AND PATTERN RECOGNITION

Francesco Camastra * Marco Spinetti **
Alessandro Vinciarelli ***

IDIAP–RR 05-79

DECEMBER 2005

PUBLISHED IN
To be presented at International Conference on Pattern Recognition 2006

* University of Naples Parthenope - francesco.camastra@uniparthenope.it
** Elbahome - Via del Renaio 516, 57034 Marina di Campo (Italy) - spinetti@elbahome.com
*** IDIAP Research Institute - vincia@idiap.ch

IDIAP Research Report 05-79

Cursive Character Challenge: a New Database for Machine Learning and Pattern Recognition

Francesco Camastra Marco Spinetti Alessandro Vinciarelli

December 2005

published in
To be presented at International Conference on Pattern Recognition 2006

Abstract. Cursive character recognition is a challenging task due to high variability and intrinsic ambiguity of cursive letters. This paper presents *C-Cube* (Cursive Character Challenge), a new public-domain cursive character database. *C-Cube* contains 57293 cursive characters manually extracted from cursive handwritten words, including both upper and lower case versions of each letter. The database can be downoloaded from the web and it provides predefined experimental protocols in order to compare rigorously the results obtained by different researchers.

1 Introduction

The assessment of the results obtained in domains like machine learning or pattern recognition depends critically on the availability of databases that can be shared by different groups. This paper presents a new database of cursive handwritten characters that can be used as an effective benchmark not only in cursive handwriting recognition (see below for more details), but also for the development of classification and clustering algorithms. The database contains around 60000 characters that have been manually labeled by three human assessors in order to reduce as much as possible the number of erroneously labeled items.

The new database is called *Cursive Character Challenge* (C-Cube) and presents three important advantages with respect to other datasets. The first is that the collection is explicitly split into training and test set. In this way, different groups can work in the same experimental conditions and the results obtained by different researchers can be compared rigorously. The second is that the dataset contains not only the character bitmaps in a format easy to read and manipulate, but also the feature vectors extracted from them using the technique described in [3]. This is important because it allows one to distinguish between the effect of the algorithms from the effect of the feature extraction process. In other words, different researchers can use exactly the same feature extraction process and can attribute the performance differences to the only algorithms rather than to the combination of feature extraction processes and algorithms. The third is that the results obtained so far over the data using state of the art methods leaves the space for significant improvement. While in the case of other benchmarks (e.g. digit databases) the performances are higher than 99% and any improvement is unlikely to be significant, in the case of C-Cube the best performance obtained so far is around 90.0% and a breakthrough improvement can still be obtained.

Moreover, the recognition of cursive handwritten characters is of interest in cursive handwriting recognition where one of the main approaches is based on a segmentation and recognition strategy [2]. Since no method is available to achieve a perfect segmentation, the word is first oversegmented, i.e. fragmented into primitives that are characters or parts of them, to ensure that all appropriate letter boundaries have been dissected. To find the optimal segmentation, a set of segmentation hypotheses is tested by merging neighboring primitives and invoking a classifier to score the combination. Finally, the word with the optimal score is generally found by applying Dynamic Programming techniques. A crucial module in the segmentation-based approach is a cursive character recognizer for scoring individual characters. In the last years some results [7][5][13] [15] on cursive character recognition have been presented, but it is difficult to compare them because they are obtained over different datasets that are often proprietary. The introduction of C-Cube can be a good solution for such a problem.

The rest of this paper is organized as follows: in Section 2 the database is presented; in Section 3 some experimental results on the database are reported; in Section 4 some conclusions are drawn.

2 The character database

C-Cube[1] can be dowloaded from `http://ccc.idiap.ch` and it contains cursive characters extracted from handwritten words after they have been desloped, deslanted and segmented [11]. The bitmaps of the characters are represented by means of ASCII characters '1' and '0' corresponding to black and white pixels respectively. For each character, the following informations are available: label, width, height, distance between upper extreme of the character and word baseline, distance between lower extreme of the character and word baseline, distance between upperline and baseline of the word from which the character is extracted.

The words from which the characters are extracted have been collected in several postal plants in the United States. Depending on the specific plant, the resolution is 212 dpi (corresponding samples are in the subsets named *blt*, *org*, *fls* and *rar*) or 300 dpi (corresponding letters are in the subset named *cdr*). The whole database contains 57293 samples and the letter distribution, shown in Figure 1, reflects the prior distribution of the postal plants where the data were collected. For this reason, some letters are very frequent while others are almost absent.

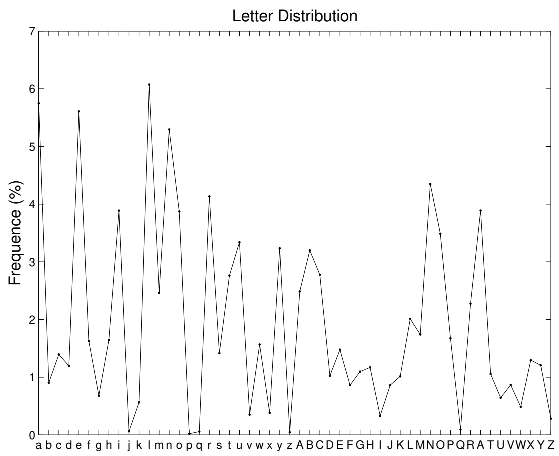

Figure 1: Letter distribution in the database.

The database has been split with a random process into training and test set containing respectively 38160 and 19133 characters and the lists corresponding to both sets are available on the database site. This defines an experimental protocol that must be respected in order to allow a rigorous comparison between results obtained by different groups.

[1]The database was collected in the project no. 62413, funded by Italian Ministry of University and of Scientific and Technological research, "Dispositivi con reti neurali e sensori per riconoscimento voce e teleoperazioni varie".

3 Experiments and Results

In this section some experiments performed on C-Cube are described. We applied to the database a feature extraction technique generating local and global features [3]. After the feature extraction process each character of the database was represented by a 34-dimensional feature vector. The character classification is achieved by using *Support Vector Machines* (*SVM*) [4] and *Neural Gas* (NG) [10]. NG allows one to obtain a suitable representation of classes, while SVMs perform the character recognition. The letters are present in C-Cube in both upper and lower case version. In some cases, the two versions are different and must be considered as separate classes. In some other cases, the two versions are similar and can be joined in a single class. NG is used to measure the overlapping in the feature space of the vectors corresponding to the two versions of each character. When the overlapping is high enough, upper and lower case versions of the letter are joined in a single class. Subsections 3.1 and 3.2 show how the optimal class representation was found and the recognition experiments respectively.

3.1 Optimal number of classes finding

Clustering verifies whether vectors corresponding to the upper and lower case versions of the same letter are distributed in neighboring regions of the feature space or not. The more the two versions of the letter are similar in shape, the more their vectors are overlapping (e.g. like *o* and *O*) and can be joined in a single class. On the other hand, when the two versions of a character are very different (e.g. g and G), it is better to consider them as separate classes. Clustering was performed by means of NG. We trained different NG maps, selecting the one with minimal *empirical quantization error*. The map neurons were labelled with a kNN technique, namely each node was labelled with the classes of the k closest feature vectors. Then the neurons were divided into 26 subsets collecting all the nodes showing at least one version of each letter α among the k classes in the label. For each subset, the percentage η_α of nodes having upper and lower case versions of the letter α in the label was calculated. The results are reported, for every subset, in Figure 2. The percentage can be interpreted as an index of the overlapping of the classes of the uppercase and lowercase versions of the letter. This information can be used to represent the data with a number of different classes ranging from 26 (uppercase and lowercase always joined in a single class) to 52 (uppercase and lowercase always in separate classes). For example a class number equal to 46 means that, for the six letters showing the highest values of η (i.e. *c*, *x*, *o*, *w*, *y*, *z*) uppercase and lowercase versions are joined in a single class.

3.2 Recognition experiments

The percentage η was used to look for the optimal number of classes. The letters showing the highest values of η were represented by a single class containing both upper and lower case versions. We trained SVMs with different number of classes. Since SVM is a binary classifier and the number of classes K was larger than 2 we have adopted *one-versus-rest*

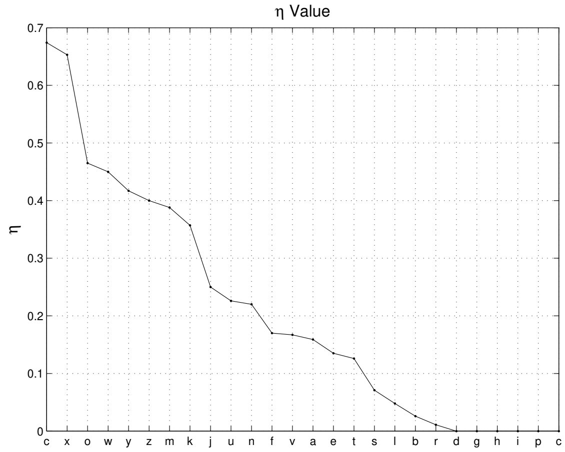

Figure 2: Value of η for each letter.

(*o-v-r*) method [12]. The method learns one classifier for each of the K classes against all the other classes choosing the classifier with maximal score. In each SVM trial we used the gaussian kernel and the variance σ and the regularization constant C were selected by means of *crossvalidation* [14]. In Figure 3 the confusion matrix of the SVM classifier is shown.

class number	performance
52	89.20%
38	90.05%
26	89.61%

Table 1: SVM Recognition rates on the Test Set, in absence of rejection, for some class numbers.

In Table 1, for different class numbers, the performances on the test set, measured in terms of recognition rate in absence of rejection, are reported. The performance is shown to be improved by decreasing the number of classes when this is higher than an optimal value, in this case 38. A further reduction of the number of classes results in a lower accuracy. The η parameter is then reliable in estimating the optimal number of classes. We compared SVM against *Learning Vector Quantization* (*LVQ*) [8] and *Multi-Layer-Perceptron* (*MLP*) [1]. SVM and LVQ trials were performed using respectively *SVMLight* [6] and *LVQ-pak* [9] software packages. In LVQ trials the learning sequence LVQ1+LVQ2+LVQ3 was adopted and the number of codevectors and learning rates were setup using crossvalidation. In MLP trials learning rates and the number of hidden neurons were setup by crossvalidation. LVQ, MLP and SVM recognition rates, in absence of rejection, are reported in Table 2. As shown in Figure 4 (upper plot), SVM recognizes better than LVQ, 25 letters on 26. The cumulative probability functions of the correct classification for LVQ and SVM are reported in Figure 4 (lower plot). The probabilities of classification of a character correctly in the top, top two and top three positions for LVQ are respectively 84.52%, 93.30% and 95.76%; whereas for

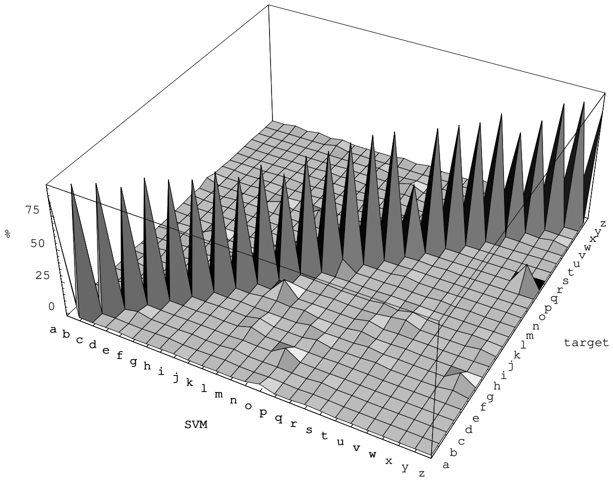

Figure 3: Confusion Matrix of the SVM.

model	class number	performance
SVM	38	90.05%
LVQ	39	84.52%
MLP	26	71.42%

Table 2: SVM, LVQ, MLP recognition rates on the Test Set, in absence of rejection.

SVM are 90.05%, 95.73% and 97.31 and for MLP they are 71.42%, 82.56% and 88.60%.

4 Conclusion

In this paper we have presented *C-Cube* a public-domain cursive character database. C-Cube is formed by 57293 cursive characters extracted from handwritten words. The letters are present in C-Cube in both upper and lower case version. We have also reported some results obtained on C-Cube, using different classifiers (e.g. a combination of SVM and Neural Gas, a combination of LVQ and Neural Gas, a Multi-Layer-Perceptron). The recognition rate, using the combination of SVM and Neural Gas, is among the highest presented in the literature for cursive character recognition.

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Cambridge University Press, Cambridge, UK, 1995.

[2] R.M. Bozinovic and S.N. Srihari. Off-line cursive script word recognition. *IEEE Trans. on Patt. Anal. and Mach. Int.*, 11(1):69–83, January 1989.

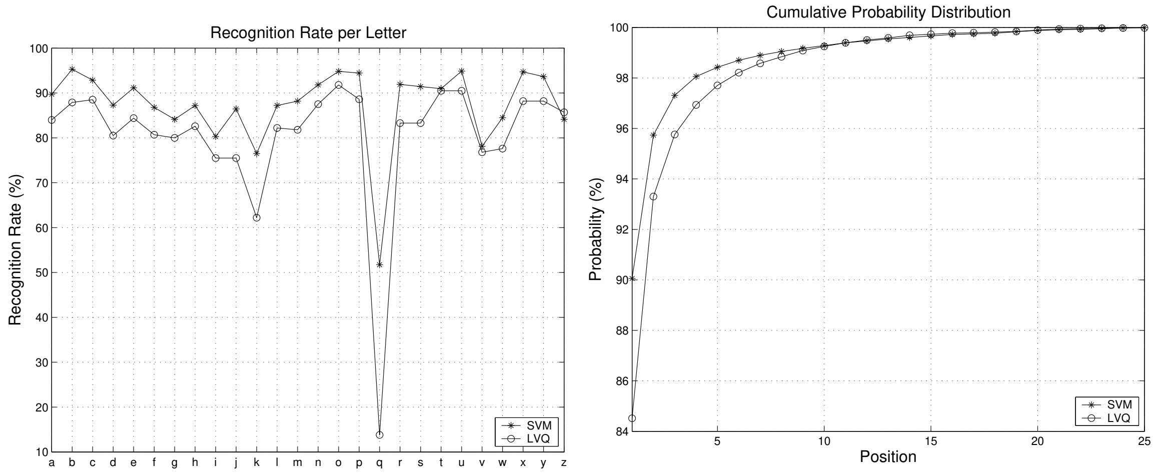

Figure 4: Recognition rate per letter (upper plot and correct position cumulative distribution (lower plot).

[3] F. Camastra and A. Vinciarelli. Cursive character recognition by Learning Vector Quantization. *Patt. Rec. Lett.*, 22(6):625–629, May 2001.

[4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[5] P.D. Gader, M. Mohamed, and J.-H. Chiang. Handwritten word recognition with character and inter-character neural networks. *IEEE Trans. on Syst., Man and Cyb.-Part B: Cybernetics*, 27:158–164, January 1997.

[6] T. Joachim. Making large-scale svm learning practical. In *Advances in Kernel Methods-Support Vector Learning*, pages 169–184. MIT Press, 1999.

[7] F. Kimura, N. Kayahara, Y. Miyake, and M. Sridhar. Machine and human recognition of segmented characters from handwritten words. In *Proc. of 4th Intl. Conf. on Document Analysis and Recognition*, pages 866–869. IEEE Press, 1997.

[8] T. Kohonen. Learning vector quantization. In *The Handbook of Brain Theory and Neural Networks*, pages 537–540. MIT Press, 1995.

[9] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola. Lvq-pak: The learning vector quantization program package. Technical Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[10] T. Martinetz, S. Berkovich, and K. Schulten. "Neural Gas" network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[11] G. Nicchiotti and C. Scagliola. Generalised projections: a tool for cursive character normalization. In *Proc. of 5th Intl. Conf. on Document Analysis and Recognition*, pages 729–732. IEEE Press, 1999.

[12] B. Scholkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge (USA), 2002.

[13] S. Singh and M. Hewitt. Cursive digit and character recognition on cedar database. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 569–572. IEEE Press, 2000.

[14] M. Stone. Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Stat. Soc.*, 36(1):111–147, 1974.

[15] B. Verma, M. Blumenstein, and M. Ghosh. A novel approach for structural feature extraction: contour vs. direction. *Patt. Rec. Lett.*, 25(9):975–988, 2004.