UNCONSTRAINED HANDWRITTEN CHARACTER RECOGNITION USING DIFFERENT CLASSIFICATION STRATEGIES

Alessandro L. Koerich
Department of Computer Science (PPGIA)
Pontifical Catholic University of Paraná (PUCPR), Curitiba, PR, Brazil

INTRODUCTION

This work focuses on the problem of unconstrained handwritten character recognition using different classification strategies.

Four multilayer perceptron neural network (MLP) classifiers were built and used into three different classification strategies: combination of a 26-class uppercase classifier and a 26-class lowercase classifier; a 26-metaclass classifier and a 52-class classifier.

Experimental results on NIST SD19 database show that better recognition performance is achieved by the metaclass classifier in which the uppercase and the lowercase representations of the characters are merged into a single class.

FEATURE EXTRACTION

We have developed several types of features such as surface, extrema, orientation, eccentricity, H/W ratio, and different zoning. We have carried out some exploratory experiments to determine which combination of features achieves the best recognition rates on the NIST SD19 database. The recognition rate was not the only criterion, we have also taken into account the dimension of the resulting feature vector where smaller is better. Such an empirical evaluation led us to build a 108-dimensional feature vector by combining 3 different types of features: projection excentricity, H/W ratio, and different zoning.

We have developed several types of features such as surface, extrema, orientation, eccentricity, H/W ratio, and different zoning. We have carried out some exploratory experiments to determine which combination of features achieves the best recognition rates on the NIST SD19 database. The recognition rate was not the only criterion, we have also taken into account the dimension of the resulting feature vector where smaller is better. Such an empirical evaluation led us to build a 108-dimensional feature vector by combining 3 different types of features: projection excentricity, H/W ratio, and different zoning. We have carried out some exploratory experiments to determine which combination of features achieves the best recognition rates on the NIST SD19 database. The recognition rate was not the only criterion, we have also taken into account the dimension of the resulting feature vector where smaller is better. Such an empirical evaluation led us to build a 108-dimensional feature vector by combining 3 different types of features: projection excentricity, H/W ratio, and different zoning.

We have developed several types of features such as surface, extrema, orientation, eccentricity, H/W ratio, and different zoning. We have carried out some exploratory experiments to determine which combination of features achieves the best recognition rates on the NIST SD19 database. The recognition rate was not the only criterion, we have also taken into account the dimension of the resulting feature vector where smaller is better. Such an empirical evaluation led us to build a 108-dimensional feature vector by combining 3 different types of features: projection excentricity, H/W ratio, and different zoning.

We have developed several types of features such as surface, extrema, orientation, eccentricity, H/W ratio, and different zoning. We have carried out some exploratory experiments to determine which combination of features achieves the best recognition rates on the NIST SD19 database. The recognition rate was not the only criterion, we have also taken into account the dimension of the resulting feature vector where smaller is better. Such an empirical evaluation led us to build a 108-dimensional feature vector by combining 3 different types of features: projection excentricity, H/W ratio, and different zoning.

CLASSIFIER DESIGN

We have designed a simple unconstrained character recognizer based on a multilayer perceptron (MLP) with one hidden layer. Many different classification strategies could be used to recognize unconstrained handwritten characters, however, in the scope of this work, we have considered the following:

- A 52-class classification problem: uppercase and lowercase representations of a single character are considered different classes (e.g. “A” and “a” are two distinct classes).

- A 26×26-class classification problem: uppercase and lowercase representations of a single character are considered different classes. Two networks with 26 outputs. Outputs are combined by several rules.

EXPERIMENTAL RESULTS

The recognition strategies were implemented and tested on the NIST Special Database 19 (SD19) which contains 614,255 binary alphanumeric characters. All experiment were conducted on a PC AMD Athlon 1.1GHz with 512MB of RAM and the average throughput is 4770 characters per second.

Training dataset (hsf0, hsf1, hsf2, hsf3): 1,440 samples per character class

Validation dataset (hsf7): 12092 uppercase characters + 11578 lowercase characters = 23670 characters

Test dataset (hsf4): 11941 uppercase characters + 12000 lowercase characters = 23941 characters.

The recognition rates on the NIST Special Database 19 (SD19) which contains 614,255 binary alphanumeric characters. All experiment were conducted on a PC AMD Athlon 1.1GHz with 512MB of RAM and the average throughput is 4770 characters per second.

Training dataset (hsf0, hsf1, hsf2, hsf3): 1,440 samples per character class

Validation dataset (hsf7): 12092 uppercase characters + 11578 lowercase characters = 23670 characters

Test dataset (hsf4): 11941 uppercase characters + 12000 lowercase characters = 23941 characters.

From the results shown in Table 2 it is clear that the metaclass classifier (NN26UpperLower) provides better recognition rates than the other classification strategies. The results achieved by the 52-class classifier are also very good even if more classes are involved in the network training. On the other hand the combination of two specialized classifiers did not have produced good results. It would be rash to conclude that the combination is not a good strategy. Maybe other combination rules might produce better results.

We have investigate why the recognition scheme based on metaclasses has performed better that the others. We observed that for some classes, the difference is very significant (e.g. over 10%) while for others, the difference is relatively small. For example, there is not an advantage in merging uppercase and lowercase “a’s”, on the other hand, merging uppercase and lowercase “c’s” seems to be very interesting because many confusions can be eliminated at the classification level. Such a conclusion agrees with differences in shapes. For example, uppercase and lowercase “c’s” have quite different shapes, so, merging both representation into a single metaclass will not bring any advantage.

Such observations suggest that neither a 26-metaclass nor a 52-class problem are the most appropriate approaches to deal with unconstrained handwritten characters. Possibly, a strategy in between both, that merges only the uppercase and lowercase characters that have similar shapes, will achieve better recognition results.

In spite of that, the recognition rates achieved in this work compare favorably with the results reported in other works, which vary from 69% to 83%. However, notice that a direct comparison is not fair, since the results were obtained on different test conditions and on different datasets.

CONCLUSION

Besides the good performance in terms of recognition rate, the recognition strategy based on the metaclass classifier is simple and requires short training and recognition times.