Pattern Recognition 47 (2014) 1002–1010

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A novel prototype generation technique for handwriting digit recognition

S. Impedovo, F.M. Mangini [*], D. Barbuzzi

Department of Computer Science, Bari University "Aldo Moro", Via Edoardo Orabona, 4-70125 Bari, Italy

ARTICLE INFO

Available online 3 May 2013

Keywords:
Handwriting digit recognition
Prototype generation
k-Nearest neighbor

ABSTRACT

The aim of this paper is to introduce a novel prototype generation technique for handwriting digit recognition. Prototype generation is approached as a two-stage process. The first stage uses an Adaptive Resonance Theory 1 (ART1) based algorithm to select an effective initial solution, while the second one executes a fine tuning designed to generate the best prototypes.

To this end, the second stage deals with an optimization problem, in which the objective function to be minimized is the cost function associated to the classification. A naive evolution strategy is used to generate the prototype set able to reduce classification time, without greatly affecting the accuracy. Moreover, as the ART1 based algorithm has incremental learning capability, the first stage is also useful for selecting the prototype set according to variations in handwriting style. The classification task is performed by the k-nearest neighbor classifier.

Experimental tests on the MNIST dataset demonstrated that our technique represents a good trade-off among accuracy, classification speed and robustness to handwriting style changes.

1. Introduction

Handwriting digit recognition has received remarkable attention in the field of character recognition. To meet industry demands, handwriting digit recognition systems must have good accuracy, acceptable classification times and robustness to variations in handwriting style. Currently, several approaches are able to reach competitive performance in terms of accuracy, including the ones based on multilayer neural networks [34], support vector machines (SVMs) [31] and nearest neighbor (NN) methods [5,23,29]. Neural networks require huge amounts of training data and time to learn effective models, but their feed-forward nature makes them very efficient during runtime. SVMs, using recent progresses in convex optimization theory to train classifiers, show a simpler training phase than neural networks, and in the test phase SVMs have a complexity which is only a fraction of a brute force k-nearest neighbor (k-NN) model as the number of support vectors generally is a small fraction of the training data. The main issue of these approaches is their low incremental learning capacity. As a matter of fact, conventional neural networks and SVMs must be retrained in order to learn new patterns. Furthermore, when new prototypes need to be learned, these systems generally forget the previous prototypes. Thus, the retraining process should involve all the new samples as well as the old ones in order to guarantee a relatively high level of recognition performance. It means that it should be necessary to combine new and old samples into a unique and large dataset and use it for retraining the classifier. Unfortunately, this is not efficient in terms of both time and space. Hence, in order to avoid retraining, a prototype based classification using the k-NN rule may be the best option for the classification method. In other words, the k-NN classifier combined with a suited prototype generation technique may be able to provide a trade-off among recognition accuracy, classification speed and robustness to handwriting style changes.

When the k-NN classifier is adopted, classifying an unknown input vector basically consists in finding the top k similar vectors in the given training set and identifying the predominant class among these k neighbors.

Therefore, the traditional k-NN rule requires the storage of the whole training set and performs classification based on the closest training samples in the feature space. So, there is a need for a small representative set of prototypes as k-NN algorithms have zero training time, but are usually expensive during runtime. In particular, for large data sets, the k-NN rule can lead to excessive amount of storage and large computation time in the classification stage. A way to mitigate these drawbacks is given by prototype optimization techniques [18,21,27]. They are aimed at achieving a representative training set with a lower size compared to the original training set and with a similar or even higher classification

[*] Corresponding author. Tel.: +39 080 5443398.
E-mail address: francescomaurizio.mangini@uniba.it (F.M. Mangini).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.016

accuracy for unknown input patterns. In the literature, two main categories of strategies can be identified: prototype selection and prototype generation. The first category strives to merge the samples from the training set into a small group of prototypes so that the performance of the k-NN rule is optimized. Examples of such techniques are the learning vector quantization algorithm [24,39], the k-means algorithm [15] and, more recently, for example the works of Garain [20] and Nanni and Lumini [30].

The second category attempts to reduce the initial training set and/or increase the generalization capability of the k-NN classifier. To this purpose, many editing and condensing algorithms have been proposed. Editing algorithms [9,19,41] remove those representatives that lead to the misclassification error. This can be done, for example, by removing "outlier" patterns or those patterns that are surrounded mainly by others from different classes. Condensing algorithms [1,17,22,42] try to build a small subset of patterns that is a part of the training set, leaving the nearest neighbor decision boundary substantially unchanged.

This paper presents a new prototype generation technique for improving handwriting digit recognition using the k-NN classifier. It is based on a two-stage process for finding the best prototypes to reduce the k-NN classification time, without greatly affecting the accuracy.

In summary the representative set, we are looking for, should be able to

1) drastically reduce the k-NN classification time since it depends just on the number of prototypes for each class (and, of course, this number is much smaller than the training data size);
2) allow the k-NN classification using only the prototypes that have been previously synthesized;
3) be incrementally adapted to changes in the writing styles by adding new prototypes or modifying the previous ones.

The experimental tests, that have been performed on the MNIST dataset using histograms of oriented gradients as image features and the Sokal and Michener dissimilarity as distance measure [36], demonstrate the effectiveness of the proposed solution compared to other strategies for building reduced prototype sets.

The remaining part of this paper is organized as follows. Section 2 presents the framework of the prototype based classification, focusing on our choice for feature extraction and the way we designed the distance measure. Section 3 describes the ART1 based algorithm to build an initial solution for our prototype generation approach. The naïve evolution strategy for synthesizing prototypes is illustrated in Section 4. The experimental tests and the results are discussed in Section 5. The conclusions are drawn in Section 6.

2. Prototype based classification

In the feature space representation, each sample consists of a feature vector v. Supposing a distance measure d (d is required to be nonnegative and to fulfill the reflexivity condition: $d(v, v)=0$, but it might be non-metric [43]), we call $v' \in \{v_1, \ldots, v_n\}$ a nearest neighbor to v if min $d(v, v_i)=d(v, v')$ where $i=1, \ldots, n$. The NN rule chooses to classify v into the class to which the nearest neighbor v' belongs: $v' \in c_n \rightarrow v \in c_n$.

For the k-NN rule, the predicted class of the unknown vector v is set equal to the most frequent true class among k nearest training samples.

Let us assume to have a representation set $R_S=\{P_1, P_2, \ldots, P_M\}$ as a collection of M prototypes and a distance measure d computed or derived from the entire training set. Now, the question that arises is whether the representation set R_S must be the complete training set or it may be a reduced set, and if so, how a small set should be extracted from the training set in order to guarantee a trade-off between recognition accuracy and classification speed. Of course, smaller representation sets are to be preferred because of a low complexity for both representation and evaluation of unknown samples.

Choosing discriminative features, as well as designing an effective distance measure [40], is a relevant task. If this task is properly accomplished and the reduced prototype set is representative, then the k-NN rule based on it is expected to perform well. Therefore, attention must be paid not only to build the representation set R_S, but also to feature extraction and distance measure definition.

2.1. Feature extraction

To increase discriminative power, the idea is to acquire information not only about digit local shape, but also about the spatial layout of shape. Local shape information is extracted by the distribution of gradient orientations within some image zones, while spatial layout description is obtained by applying to the image different grids at multiple resolutions. This leads to a scheme in which each resolution corresponds to a level (Fig. 1).

Moreover, in this paper, patterns are described through binary histograms of oriented gradients. They are considered very effective features for recognizing objects in images, because histograms of oriented gradients are tolerant to image changes in scale, rotation, translation and contrast [4,12,14,25]. They basically consist of calculating gradient values and constructing local histograms along certain orientations. Since each pixel of an image has an orientation and a magnitude depending on the local gradient, histograms are built by accumulating magnitude values over the same directions for all the pixels belonging to each zone of the image.

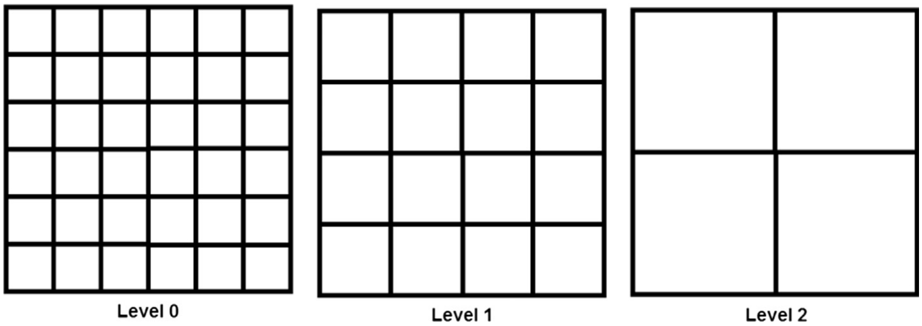

Fig. 1. Three level scheme for feature extraction.