Optik 125 (2014) 1034–1037

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Sparse representation-based classification algorithm for optical Tibetan character recognition

Heming Huang [a,b], Feipeng Da [a,*]

[a] School of Automation, Southeast University, Nanjing, Jiangsu 210096, China
[b] School of Computer Science, Qinghai Normal University, Xining, Qinghai 810008, China

ARTICLE INFO

Article history:
Received 19 March 2013
Accepted 21 July 2013

Keywords:
Sparse representation
K-NN
Tibetan
Off-line
Character recognition

ABSTRACT

A sparse representation-based two-phase classification algorithm is proposed for off-line handwritten Tibetan character recognition. The first phase realizes coarse classification with the *K*-NN classifier by finding the *K* nearest neighbours of a test sample in the dictionary constructed by *K*-SVD with samples of all classes, and the classes of these neighbours are regarded as the candidate classes of the test sample. The second phase performs fine classification with the sparse representation classifier by sparsely representing the test sample with all elements of the dictionary constructed by *K*-SVD with samples of all candidate classes, and the test sample is finally classified into the candidate class with the maximal contribution in sparse representation. Experiments on the Tibetan off-line handwritten character database show that an optimal recognition rate of 97.17% has been reached and it is 2.12% higher than that of *K*-NN.

1. Introduction

Nowadays, there is a great demand for quickly inputting huge volumes of handwritten information data such as checks, mail, payment slips, and many other documents into the computer, and this kind of data has to be typed into the computer by human operators. Fortunately, such time-consuming and error-prone operations have been lightened with the invention of optical character recognition systems by recognizing handwritten characters automatically at high speeds.

Over the last several decades, a great amount of research work has been exploited for Latin, Chinese, and Arabic character recognition [1], and a rigorous theoretical foundation of character recognition methods has been laid especially to statistical methods [2,3], ANNs [4], SVMs [5], structure methods [6], and multiple classifier methods [7]. However, so far, the capabilities of the handwritten character recognition systems are still quite limited and the research on the recognition of handwritten minority scripts such as Tibetan is still open.

Sparse representation, which is originated from signal processing area, has been introduced in recent years to address pattern classification problems from a novel point of view. This method first represents a test sample with a sparse linear combination of the training samples from different classes, and then evaluates the representation contribution of each class on the test sample, and the test sample is finally classified into the class that has the greatest contribution. This method exhibits good performance in such pattern recognition problems as face recognition [8–10], speech recognition [11], digit recognition [12], and cancer biomarker classification [13].

In parse representation, a dictionary is always used and a dictionary in fact is a matrix that contains some training samples, one sample for each column. The most important step of sparse representation is to find the sparsest coefficients as a test sample is represented with all columns of a dictionary. As a matter of fact, this step is an iterative process of the sparse coding stage and the dictionary update stage; furthermore, each stage needs to be further resolved iteratively [14,15]. Therefore, the sparse representation-based classification method is generally time-consuming, especially to such classification problems as Chinese or Tibetan character recognition that have large category set and large training data set. There is no doubt that the decrease of the candidate classes will accelerate the recognition speed of sparse representation classification. Therefore, a *K*-NN classifier is introduced prior to sparse representation classification. When *K* is greater than one, the *K*-NN classification realizes coarse classification and obtain no more than *K* candidate classes for the test sample. The sparse representation classification is further implemented on these candidate classes, and the speed of the sparse representation classification is accelerated because of less candidate classes. Therefore, in this paper, a hybrid classifier that cascades *K*-NN and sparse

* Corresponding author at: School of Automation, Southeast University, Nanjing, Jiangsu 210096, China.
E-mail address: dafp@seu.edu.cn (F. Da).

0030-4026/$ – see front matter © 2013 Published by Elsevier GmbH.
http://dx.doi.org/10.1016/j.ijleo.2013.07.101

representation is proposed to classify the samples of handwritten Tibetan character.

The remainder of this paper is organized as follows. Section 2 introduces the proposed method in detail. Section 3 is dedicated to the experiments and Section 4 draws some conclusions.

2. Sparse representation-based two-phase classification method

The proposed method is a hybrid two-phase classifier that cascades K-NN and sparse representation. The first phase realizes coarse classification with K-NN and obtains some candidate classes while the second phase realizes accurate classification with sparse representation.

2.1. The K-NN-based coarse classification

Assume that there are N classes. For a test sample y, the first phase of the proposed method finds its K nearest neighbours from the dictionary D and regards the classes of these neighbours as the candidate classes. The dictionary D mentioned above is constructed in two steps. Firstly, construct a n-column dictionary D_i^n for class i (i = 1, 2, . . ., N) by the dictionary learning methods such as K-SVD [14]. Secondly, construct the dictionary D by concatenating the dictionaries $D_1^n, D_2^n, \ldots,$ and D_N^n horizontally, that is $D = [D_1^n, D_2^n, \ldots, D_N^n]$. Totally, there are $n * N$ training samples in dictionary D.

To a test sample y, find its K nearest neighbours $y_1, y_2, \ldots, y_K$ in the dictionary D and their class labels are denoted as $C_1, C_2, \ldots, C_M$, where $M \le K$. To perform an accurate classification in the second phase, we assume the genuine class of the test sample is one of the classes $C_1, C_2, \ldots,$ and C_M.

2.2. The sparse representation-based fine classification

The second phase of the proposed method realizes fine classification. This phase seeks to the sparsest representation of the determined dictionary D_C on the test sample y, and the test sample y is finally classified into the candidate class with the largest contribution.

In common with the construction of the dictionary D, the just mentioned dictionary D_C is also constructed in two steps. Firstly, construct an m-element dictionary $D_{C_j}^m$ for the jth candidate class C_j by the dictionary learning method K-SVD (j = 1, 2, . . ., M). Secondly, construct a dictionary D_C by concatenating all dictionaries $D_{C_1}^m, D_{C_2}^m, \ldots,$ and $D_{C_M}^m$ horizontally, that is $D_C = [D_{C_1}^m, D_{C_2}^m, \ldots, D_{C_M}^m]$. The sparsest representation of the dictionary D_C on the test sample y is the solution of the ℓ^0-minimization problem

$$X_C = arg\, \min_X ||X||_0 \quad \text{subject to} \quad ||D_C X - y||_2 \le \varepsilon. \tag{1}$$

This could be solved by executing the sparse coding stage and the dictionary update stage repeatedly until convergence. In sparse coding stage, the dictionary D_C is assumed to be fixed and the above optimization problem is considered as a search for sparse representation with coefficients summarized in the matrix X. The typical method for this stage is K-SVD [14]. The dictionary update stage is performed to search for a better dictionary. By fixing all columns except one in D_C, this process updates one column at a time and finds a new column and new values for the coefficients of D_C that best reduce the mean square error. The OMP, FOCUSS, and BP are well-known methods for this purpose [15].

Once the problem (1) is resolved, the test sample y is classified to the candidate class with the largest representation contribution. To this end, the coefficient vector X_C is divided into the sub-vectors $X_{C_1}, X_{C_2}, \ldots,$ and X_{C_M}, where X_{C_j} contains the corresponding coefficients of the dictionaries $D_{C_j}^m$ ($j = 1, 2, \ldots, M$). For each candidate class, the residual of the representation is computed with the equation

$$\gamma_j(y) = ||y - D_{C_j}^m X_{C_j}||. \tag{2}$$

A smaller residual indicates a greater contribution. Therefore, the test sample y is classified into the candidate class that produces the minimal residual. That is to say, the class label of the test sample y is decided by the minimization problem

$$\text{class}(y) = arg\, \min_j \gamma_j(y) \quad (j = 1, 2, \ldots, M). \tag{3}$$

2.3. The main steps of the proposed method

In summary, the main steps of the proposed method are as follows.

(1) Construct the dictionary D by horizontally concatenating $D_1^n, D_2^n, \ldots, D_N^n$, where D_i^n ($i = 1, 2, \ldots, N$) is a n-column dictionary of class i and it is constructed by K-SVD.
(2) To a test sample y, find its K nearest neighbours in D and regard the classes of these neighbours as the candidate classes.
(3) Construct the dictionary D_C by horizontally concatenating $D_{C_1}^m, D_{C_2}^m, \ldots,$ and $D_{C_M}^m$, where $D_{C_j}^m$ ($j = 1, 2, \ldots, M$) is a m-column dictionary of candidate class j and it is constructed by K-SVD.
(4) Solve the ℓ^0-minimization problem (1).
(5) Divide the coefficient vector X_C into sub-vectors $X_{C_1}, X_{C_2}, \ldots,$ and X_{C_M}, where X_{C_j} is the coefficient vector of the dictionary $D_{C_j}^m$ ($j = 1, 2, \ldots, M$).
(6) Compute the residual of each class with Eq. (2).
(7) Classify the test sample by solving the minimization problem (3).

3. Experiments

3.1. Database and feature extraction

The off-line Tibetan handwritten character database (THCDB) has collected handwritten samples of Tibetan characters. The sample number of each class ranges from 156 to 314 [16]. There are 30 consonants in Tibetan alphabet as shown in Fig. 1, and our experiments are implemented on the samples of these consonants.

Among all samples used in our experiments, 80% are used for training and the rest 20% are used for testing. Some essential preprocessing techniques such as sample image de-noising, size normalization, and slant correction have been implemented. Fig. 2

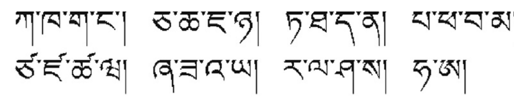

Fig. 1. The 30 consonants of Tibetan alphabet.

Fig. 2. Some samples of Tibetan consonants after essential preprocessing.