Information Sciences 180 (2010) 4084–4098

Contents lists available at ScienceDirect

Information Sciences

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

A fast method for the implementation of common vector approach

Mehmet Koç *, Atalay Barkana, Ömer N. Gerek

Department of Electrical and Electronics Engineering, Anadolu University, Eskisehir, Turkiye

ARTICLE INFO

Article history:
Received 13 May 2009
Received in revised form 23 February 2010
Accepted 21 June 2010

Keywords:
Common vector approach
Fast classification algorithm
Classifier implementation
Face recognition

ABSTRACT

In this paper a novel computation method is proposed to perform the common vector approach (CVA) faster than its conventional implementation in pattern recognition. While conventional CVA calculations perform the classification with respect to the distance between vectors, the new method performs the classification using scalars. A theoretical proof of the equivalence of the proposed method is provided. Next, in order to verify the numerical equivalence of the proposed computation method to the conventional (vector-based) method, numerical experiments are conducted over three different face databases, namely the AR Database, extended Yale Face Database B, and FERET Database. Since the computational gain may depend on (i) the dimension of the feature vectors, (ii) the number of feature vectors used in training, and (iii) the number of classes, the effects of these items are clearly verified via these databases. Our theoretically equivalent (but faster) method provided no difference in the classification rates despite its improved classification speed as compared to the classical implementation of CVA. The new method is found to be about 2.1–3.0 times faster than the conventional CVA implementation for the AR face database, 1.9–3.3 times faster for the extended Yale Face Database B, and 1.9–3.1 times faster for the FERET Database.

1. Introduction

The recent demand on camera-based automated security systems has increased the importance of face recognition. Automatic face recognition suffers from real-time detection and recognition of faces in terms of speed and accuracy. Many methods have been proposed to improve the efficiency of automatic face recognition [1,10,15,16]. In face recognition, computation time is almost as important as the recognition accuracy for real-time applications [3]. In appearance-based methods, the whole face picture is taken as a $w \times h$ dimensional feature vector. For this reason, high-resolution pictures automatically become very high dimensional feature vectors. This degrades the computational efficiency of the system.

The common vector approach (CVA) is a subspace method used in pattern recognition problems such as speech recognition and face recognition [2,4,6,7,9,14] and is also used in feature selection [8]. High dimensional feature vectors also adversely affect the computational performance of CVA in training and testing stages.

In this paper, a new method that performs CVA in a shorter time than the conventional calculations is proposed. The method performs the classification with respect to scalar values while the conventional CVA calculations do it using vectors with a corresponding feature dimension. Consequently, it must be noted that the proposed method does not alter the recognition performance of the classical CVA. A brief review of the CVA is given in Section 2, where the classical implementation methods are explained. The proposed complexity reduction algorithm together with the necessary theoretical proofs is given

* Corresponding author. Address: Anadolu Üniversitesi, İki Eylül Kampüsü, Mühendislik–Mimarlik Fakültesi, Elektrik–Elektronik Mühendisliği Bölümü, PK: 26555, Eskisehir, Turkey. Tel.: +90 222 321 35 50x6465; fax: +90 222 323 95 01.
E-mail address: mkoc6@anadolu.edu.tr (M. Koç).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.06.027

in Section 3. Despite the theoretical proof regarding the equivalence of the proposed computation method and conventional implementations of CVA, one could expect numerical issues to arise and yield marginal differences in real-life experiments. In order to observe the actual speed of improvement figures and to check whether numerically the same recognition rates are achieved, three face databases were considered as in Section 4. The databases were selected in such a way that each one has a distinct characteristic regarding the (i) feature vector size, (ii) number of images per subject, and (iii) number of subjects. Overall, the numerical effect of the proposed speed improvement is tested over a wide variety of cases. It was observed that the proposed "faster" algorithm achieves exactly the same recognition rate as its classical implementation version while attaining classification speed improvements (up to 3.3 folds). Results are discussed in Section 5.

2. Common vector approach

In CVA, a feature vector is set to have two components: (1) a component that exhibits properties that are common to the class and (2) a residual component that has all the variations from the common properties, i.e., within-class variances. After subtracting the dissimilarities (second component) of each vector of a class, the invariant properties of the class (the first component) will remain. The vector that contains the invariant properties is called the common vector [6]. During the extraction of the common vector, there are two possible cases, depending on the vector size (n) and the number of samples (m) in each class. If $n < m$, the situation corresponds to the so-called sufficient case, where the observation samples are more than the feature size. The second situation is called the insufficient case ($n \geqslant m$). Despite the availability of CVA for either case, the derivation of the method was originally developed for the insufficient case. Following the same derivations, the sufficient case is omitted in this paper.

There are two conventional algorithms to evaluate the common vectors [7]. The first one uses the Gram–Schmidt orthogonalization process, and the second one uses the within-class covariance matrices. Their equivalence was proved in Ref. [7]. Since this paper is a contribution in the computation strategy, both of these evaluation methods are briefly described in the following subsections.

2.1. Determining common vectors using Gram–Schmidt orthogonalization process

Let the training set have C classes and let every class have n-dimensional m samples with $n \geqslant m$. Let a_i^j be the n-dimensional column vector which denotes the ith sample from the jth class. The *difference subspace* of jth class can be achieved as the span of the difference vectors:

$$B_1^j = \mathrm{span}\left\{a_2^j - a_1^j, \dots, a_m^j - a_1^j\right\}. \tag{1}$$

A vector set, U_1^j is obtained using these difference vectors that span B_1^j subspace. For this purpose Gram–Schmidt orthogonalization process is used to obtain the orthonormal vector set U_1^j:

$$U_1^j = \left\{u_1^j, u_2^j, \dots, u_{m-1}^j\right\}, \quad j = 1, \dots, C, \tag{2}$$

and this orthonormal vector set spans the same aforementioned *difference subspace* [5]:

$$B_1^j = \mathrm{span}\left(U_1^j\right). \tag{3}$$

If we select $V_1^j = \left\{v_m^j, \dots, v_n^j\right\}$ as the complementing orthonormal vector set where $\left(v_k^j\right)^T \cdot u_l^j = 0,\ k = m, \dots, n$ and $l = 1, \dots, m-1$, then an *indifference subspace* $\left(B_1^j\right)^\perp$ will be spanned by V_1^j, i.e.:

$$\left(B_1^j\right)^\perp = \mathrm{span}\left\{V_1^j\right\}. \tag{4}$$

Thus B_1^j and $\left(B_1^j\right)^\perp$ are the complementary subspaces, i.e., $B_1^j \cap \left(B_1^j\right)^\perp = \varnothing$ and

$$B_1^j \cup \left(B_1^j\right)^\perp = R^n. \tag{5}$$

The common vector of jth class can be calculated either by subtracting the projection of any feature vector into the difference subspace from itself

$$a_{com}^j = a_i^j - \left(\left(a_i^j\right)^T \cdot u_1^j\right)u_1^j - \cdots - \left(\left(a_i^j\right)^T \cdot u_{m-1}^j\right)u_{m-1}^j, \tag{6}$$

or by projecting any feature vector into the indifference subspace

$$a_{com}^j = \left(\left(a_i^j\right)^T \cdot v_m^j\right)v_m^j + \cdots + \left(\left(a_i^j\right)^T \cdot v_n^j\right)v_n^j. \tag{7}$$