Application of sparse coding with spatial pyramid matching for face expression classification

Dept. of Computer Science
The University of British Columbia

Abstract

In this work I evaluate the performance of image classification method based on spatial pyramid matching for sparse codes, using the JAFFE database of facial expressions. I show that the method is comparable to other methods, typically used for such datasets, and I also introduce some attempts that I made towards the improvement of the algorithm.

1 Introduction

Facial expression recognition problem in human-computer environment is both significant and challenging. There have been developed a number of facial recognition systems that make use of various facial databases. For instance, [2] used AR database to classify facial expressinos into three differen classes: smiling, angry and neutral. At the same time, some databases have much more rich emotion representation, such as JAFFE database. It contains seven main facial expressions: happy, sad, fearful, angry, disgusted, surprised and neutral. It has also been extensively used for facial expression classification tasks for a number of researchers [3], [4], which make it an attractive database for algorithm performance comparison.

Most facial expression recognition systems conduct two stages: feature extraction and expression classification. For feature extraction such methods as Gabor filter, *principal component analysis* (PCA) and *independent component analysis* (ICA) are frequently used. At the same time, *linear discriminant analysis* (LDA), *support vector machine* (SVM) and *hidden markov model* (HMM) are frequently used.

At the same time, *bag-of-features* (BoF) [6] and *spatial pyramid matching* (SPM) [5] are main components of state-of-the-art object and scene classification algorithms. In this work we study the performance of the algorithm, presented in [11], which is based on SPM and have been designed for object classification problems such as Caltech-101 and Caltech-256 [7]. In order to compare the performance of the algorithm to the other ones, we use the algorithms performance comparison results, presented in [1]. We make a coclusion that the algorithm is comparable to the algorithms, designed for the face recognition tasks.

2 Method description

The method description is organized as follows. In section 2.1 we briefly introduce Scale Invariant Feature Transform method [15], which will be used for image feature extraction. In section 2.2 we describe sparse coding method and explain how we use it to further process image features. Section 2.3 briefly describes spatial pyramid matching method and its application to to representation of feature vectors for this algorithm. Finally, in section 2.4 we depict the support vector machine classification algorithm, that we use for final classification of images.

2.1 Scale Invariant Feature Transform

In order to classify images we need to know how to extract good features from images and which features are good. It has been shown [14], that primate vision system uses complex features that are largerly shift, scale and illumination invariant. In the same manner, in computer vision tasks it is useful to be able to extract image features that posses the necessary invariance properties.

One of the best performances for local descriptors extration is demonstrated by *Scale Invariant Feature Transform* (SIFT) [15]. SIFT-based descriptors exhibit the highest performance evaluation on both textured and structured scenes, outperforming other local descriptors for slight image rotations and change of scale, significant amount of blur and illumination change [16].

For the purposes of image classification, we extract SIFT descriptors over regular grid with uniform spacing from each image in the database. Applying extraction over uniform grid rather than interest points may be beneficial for the purposes of scene recognition, as the low-contrast image regions may also play an important role in classification [13]. We will use the extracted features for Sparse Coding subsequently.

2.2 Sparse coding

Let $X = [x_1, ..., x_M]^T$ be a set of SIFT descriptors of all images in the database, $x_i \in \mathbb{R}^D$. The *Sparse Coding* (SC) method aims to solve the following problem:

$$\min_{U,V} \sum_{m=1}^{M} \|x_m - u_m V\|^2 + \lambda |u_m|, \qquad \|v_k\| \leq 1, \quad k = 1, 2, ..., K, \tag{1}$$

where L^2-norm constraint on v_k is necessary to avoid trivial solutions when u_m and V are respectively divided and multiplied by a large constant, making $|u_m|$ small. Here $V = [v_1, ..., v_K]^T$ is called a *codebook* and $U = [u_1, ..., u_M]^T$ is a collection of *sparse codes* of descriptors $x_1, ..., x_M$. We also assume that the size of the codebook K is known. Sparse Coding takes its name from the fact that L^1 regularizer ensures that the majority of elements of sparse codes u_m are zero, making matrix U sparse.

It is worth noting that SC is a more generalized version of *principal component analysis* (PCA); both methods represent the data as linear combinations of bases. However, the number of bases in PCA is less or equal to space dimension D, while codebook size K is usually greater than D.

SC has a training and a code (test) phase. During training phase the problem (1) is solved with respect to U and V; the codebook V is preserved afterwards. During the code phase the algorithm is provided with a set of descriptors $x_1, ...x_N$ of a new image. Since the method has previously learned codebook V, the problem (1) is solved with respect to U only.

In order to solve (1) with repsect to U and V we observe that the optimization problem in U with fixed V is convex and in V with fixed U is also convex, but not in both simultaneously. Thus, we can solve the problem iteratively by optimizing for U or V while fixing the other. The optimization with respect to U can be obtained by solving (1) in each u_m separately:

$$\min_{u_m} \|x_m - u_m V\|^2 + \lambda |u_m|. \tag{2}$$

This is a Lasso problem and we optimize it by a recently proposed *feature-sign search* algorithm [12]. Problem (1) with fixed U is essentially a least square problem with quadratic constraints:

$$\min_{V} \|X - UV\|_F^2, \qquad \|v_k\| \leq 1, \quad k = 1, 2, ..., K. \tag{3}$$

Problem (3) can be solved by Lagrange dual [12]. As a result of sparse coding for SIFT features $x_1, ..., x_M$ we obtain the codebook V, which is retained for later use.

2.3 Spatial pyramid matching

There is some psychophysical evidence that people recognize scenes as a whole, meanwhile overlooking the details of its components [9]. This suggests that it might be possible to develop image representations, that make use of the local features in order to reconstruct global semantics of the scene, without going through an intermediate step of segmenting the scene into its constituent parts.

Some advanced orderless methods such as *bag-of-features* (BoF), which follow this strategy have demonstrated high level of performance for image classification [8]. However, these methods do not preserve the information about spatial layout of features and thus cannot take advantage of spatial structure of a scene.

The *spatial pyramid matching* (SPM) method incorporates the BoF approach, while preserving some information about spatial features layout. In particular, SPM places a sequence of increasingly finer grids over the image and computes the histogram of local features for each grid cell. Adding these histograms together yields in a SPM feature vector, representating the image based on the set of its features and their spatial layout. Among the many extensions of BoF approach, SPM method has demostrated the best performance [5].

For the purposes of classification, for all images in the database we form SPM feature vectors based on sparse codes of their SIFT descriptors, obtained in previous section. The usual histogram for the number of features that fall in each cell we replace with max-pooling strategy for sparse codes of features that belong to each cell. Max-pooling for a number of vectors yields in a vector, formed elementwise as a maximum of the corresponding elements in vectors used for max-pooling. This strategy is shown to work better for this method [11]. As a result of the operations above we obtain a single feature vector for each image in the database that we will use for classification.

2.4 Support vector machine

Linear *support vector machine* (SVM) is a method for supervised classification, which aims to separate two classes in the feature space with a plane, most distant to points in both classes.

Suppose we are given a set of features $f_1, ..., f_N$, $f_i \in \mathbb{R}^n$ and a set of class-membership indicators $y_1, ..., y_N$, $y_i \in \{-1, 1\}$. SVM solves the following optimization problem:

$$\min \|w\|, \qquad y_i(w^T x_i + b) \geq 1, \tag{4}$$

where $w \in \mathbb{R}^N$ and $b \in \mathbb{R}$ determine the separating plane.

For the multiclass classification setting, consider training data $f = [f_1, ..., f_N]$, $y = [y_1, ..., y_N]$, where $f_i \in \mathbb{R}^n$, $y_i \in \{1, ..., L\}$ and L is the number of classes. Multi-class SVM aims to find L linear functions $\{w_c^T f | c \in \{1, ..., L\}\}$, such that for a test datum f, its class will be predicted by

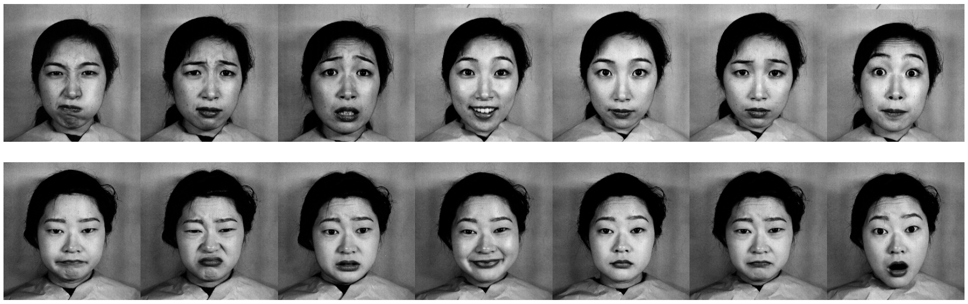
$$\max_{c \in \{1,...,L\}} w_c^T f$$

The method that we study takes one-against-all strategy and finds L classifiers by solving the following convex problem for each of them:

$$\min_{w_c} \left\{ \|w_c\|^2 + C \sum_{i=1}^{N} \left[\max(0, w_c^T f y_i^c - 1)\right]^2 \right\},$$

where $y_i^c = 1$ if $y_i = c$ and $y_i^c = -1$ otherwise. Here $\left[\max(0, w_c^T f y_i^c - 1)\right]^2$ is a quadratic hinge loss function, which is differentiable everywhere. This enables us to solve the problem with conventional gradient-based optimization methods; in this work L-BFGS is used.

The last step in the described algorithm is to train the SVM classifier on SPM feature vectors of a certain training set of images.



(a) Angry (b) Disgust (c) Fear (d) Happy (e) Neutral (f) Sad (g) Surprised

Figure 1: Samples of two subjects with seven different face expressions.

3 Experiments

3.1 Dataset

Since the purpose of the work was to evaluate the algorithm on the facial expression classification, it was important that recognition was not affected by other factors such as head position, rotation and significant change in illumination. Furthermore, high variety of facial expressions was desired, since it would allow me to test the algorithm on a problem with sufficient number of classes. The authors of the algorithm in their experiments used training sets of 15 and 30 images. For the similar experiments I needed the dataset that would contain at least 30 images in each class. In addition, in order to compare the performance of the algorithm to the that of the other algoritms, it was essential to have access to algorithm evaluation papers for the same dataset.

Based on the criterias above I selected the JAFFE dataset, consisting of 213 images of faces of 10 japanese femails. Each subject represented in the database has 2-4 photos of 7 different facial expressions. In total, there are 28-32 images in each of 7 classes of facial expressions (see Figure 1).

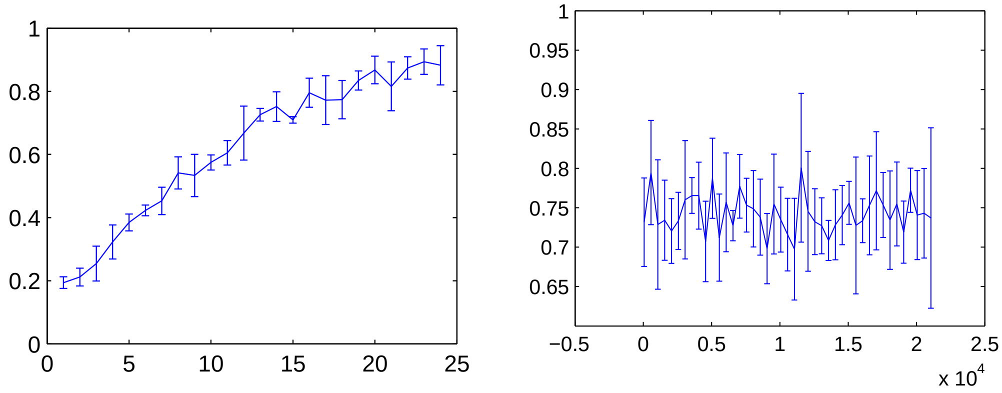
3.2 Results

In order to compare the described method to the other methods designed for facial epression classification, I use the results presented in [1]. Authors of [1] applied five different algorithms to JAFFE dataset and presented the results of classification rates for these methods. I applied the method, described in this paper to the JAFFE dataset; the comparison of the performances for the methods are represented in Table 1.

I also observed the influence of the training dataset on the classification rate, which is shown in Figure 2.

Table 1: Perormance comparison in JAFFE dataset

METHOD	RECOGNITION RATE
LDA + SVM	91.27%
2D-PCA + SVM	92.06%
ICA + SVM	93.35%
PCA + SVM	93.43%
SC+ SPM+ SVM	**93.51%**
2D-LDA + SVM	94.13%



(a) Recognition rate against training size

(b) Recognition rate against number of PCA bases

Figure 2: Recognition rate against different parameters of the method. Graph (a) represents the dependence of recognition rate on the training set. Recognition rate approaches 0.9 as we increase the training set to 25. Graph (b) shows that there is almost no correlation between number of PCA bases extracted from the final feature set and recognition rate.

3.3 Attempts for method improvement

SIFT descriptors represent image gradient histograms at the locations around keypoints they were extracted from. Since images in the same class have certain similar visual features, they must have a number of similar SIFT descriptors. In this light it is reasonable to assume that SIFT descriptors of images in a particular class form clusters in the feature space, concentrated around particularly distinctive features, which may representative for the class. For instance, in the more general classification setup if the class is ”Dogs”, such distinctive features can represent small image patches on eyes of dogs and therefore may be found in the images throughout the class, but are rarely seen in the images of the class ”Planes”. On the contrary, features that are far from centers of the clusters are not representative for the class, since they have less correlation with other features from the class. Furthremore, one could hope that removing non-distinctive features from the feature set may improve the classification.

In view of the aforesaid, I used SC for each of the sets of features of each class and removed all features that had ambiguities in their sparse codes. I conducted a number of experiments with different number of SC bases in the range from 10 to 100. However, the hypothesis that this would yield with the set of distinctive features was not corroborated. Final set of features did not correspond to similarities in images within classes. Apparently, the method did not work properly because feature set did not form the cluster structure with number of clusters between 10 and 100. Presumably the number of clusters is significantly higher and/or the clusters are too sprase to have influence on SC.

Another observation that I made was that the final data matrix had dimensions 213×21504, where each of 213 data points was represented by a vector of length 21504. These vectors were sparse, and the hope was that by applying PCA to the dataset prior to training the SVM I could eliminate the unnecessary dimensions and improve the algorithm. However, it appeared that the number of extracted PCA bases did not influence the classification rate (see Figure 2).

4 Conclusions and future work

In this work I evaluated performance of the method based on sparse coding with spatial pyramid matching on facial expressions classification task and compared the results to that of the methods, evaluated in [1]. I applied cross-validation on the final SVM classificator and showed that the performance of the method is comparable to that of the state-of-the-art methods in facial expression classification. I also made some attempts for improvement of the method. Although they did not work as I thought, it gave me better understanding of how the method works. The future work should be done on supervised codebook learning, as suggested in [11].

References

[1] F.Y. Shih, C. Chuang, and P.S.P. Wang. Performance Comparisons of Facial Expression Recognition in Jaffe Database. IJPRAI: 445-459, 2008.

[2] X.-W. Chen and T. Huang. Facial expression recognition: a clustering-based approach. Patt. Recog. Lett. 24: 12951302, 2003.

[3] S. Dubuisson, F. Davoine and M. Masson, A solution for facial expression representation and recognition, Sign. Process.: Imag. Commun. 17: 657673, 2002.

[4] I. Buciu, C. Kotropoulos and I. Pitas. ICA and Gabor representation for facial expression recognition, Proc. IEEE Int. Conf. Image Processing: 855858, 2003.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Proc. of CVPR06, 2006

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. Workshop on Statistical Learning in Computer Vision, ECCV: 122, 2004.

[7] G. Grifen, A. Holub, and P. Perona. Caltech-256 object category dataset. (7694), 2007.

[8] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual Categorization with Bags of Keypoints. ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[9] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV, 42(3):145175, 2001.

[10] W.E. Vinje and J.L. Gallant. Sparse coding and decorrelation in primary visual cortex during naturalvision. Science, 297(5456):12731276, 2000.

[11] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In NIPS, 2006.

[13] Fei-Fei Li, P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. in proc. CVPR, 2(15): 524-531, 2005.

[14] D. I. Perrett, M. W. Oram. Visual recognition based on temporal cortex cells: viewer-centred processing of pattern configuration. Zeitschrift fur Naturforschung Teil C Biochemie Biophysik Biologie Virologie, 53(7-8): 518-541, 1988.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2): 91-110, 2004.

[16] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10): 16151630, 2005.