Neurocomputing 144 (2014) 509–515

Contents lists available at ScienceDirect

Neurocomputing

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

Hierarchical feature coding for image classification

Jingyu Liu*, Yongzhen Huang, Liang Wang, Shu Wu

Institute of Automation, Chinese Academy of Sciences (CASIA), National Laboratory of Pattern Recognition (NLPR), Beijing 100190, China

ARTICLE INFO

Article history:
Received 28 January 2014
Received in revised form
4 April 2014
Accepted 22 April 2014
Communicated by Qingshan Liu
Available online 24 May 2014

Keywords:
Image classification
Hierarchical encoding
Higher level representation

ABSTRACT

Feature coding and pooling are two critical stages in the widely used Bag-of-Features (BOF) framework in image classification. After coding, each local feature formulates its representation by the visual codewords. However, the two-dimensional feature-code layout is transformed to a one-dimensional codeword representation after pooling. The property for each local feature is ignored and the whole representation is tightly coupled. To resolve this problem, we propose a hierarchical feature coding approach which regards each feature-code representation as a high level feature. Codeword learning, coding and pooling are also applied to these new features, and thus a high level representation of the image is obtained. Experiments on different datasets validate our analysis and demonstrate that the new representation is more discriminative than that in the previous BOF framework. Moreover, we show that various kinds of traditional feature coding algorithms can be easily embedded into our framework to achieve better performance.

1. Introduction

Image classification is a fundamental vision problem which is to classify images to the specified one or more categories. It has a wide range of applications in image retrieval [1–3], web analysis [4–6], etc. This is a very challenging task due to the variability of illumination, scales, rotation, viewpoints and occlusion. Inspired by the bag of words (BOW) model [7] in document analysis, the bag of features (BOF) model [8] has been demonstrated successful for image classification. In the BOF model, an image is modeled as an unordered composition of visual features which are encoded by a group of visual codewords. After that, features' responses on each codeword are pooled to one single value, and the image is finally described as a codebook histogram.

Coding and pooling are two critical procedures of the traditional BOF model. Many efforts have been dedicated to develop effective encoding and pooling algorithms. Though many algorithms have been proposed, the inherent characteristics of coding and pooling stay unchanged. Our proposed hierarchical framework is inspired by the essential drawbacks of coding and pooling, as can be summarized in the following two aspects:

1. The nature of coding is to partition the continuous feature space to discrete visual words. Different coding strategies are employed to assign each feature to its surrounding visual words. Inspired by Huang et al. [9], we interpret coding as a process of constructing connections. Features and visual words can be deemed as vertexes in the feature space. After coding, an undirected and weighted edge will bridge each local feature and their surrounding visual words. A more weighted edge characterizes an accurate approximation of features, whereas a less weighted edge indicates the ambiguity of visual words. Therefore, we believe such connections yield some valuable information, which yet, are not fully utilized in the traditional framework.
2. After coding, the traditional BOF framework will enter the next stage, pooling. The nature of pooling is to accumulate local features to a global appearance-based representation. For each local feature, the weighted connections with its surrounding visual words are obliterated in the process of pooling. Therefore the abundant and more subtle information of each local feature are abandoned in the process of pooling. Figs. 1 and 2 illustrate the phenomenon. Fig. 1 shows average pooling, where different appearances result in the same visual word histogram after pooling. As a result, two images from different categories might be wrongly classified into the same one. Fig. 2 shows max pooling, where only the largest response (0.5) is preserved. Though close enough, other values (0.49) are ignored.

Current studies on feature coding combined with feature pooling naturally result in the drawback of the traditional BOF framework. As analyzed above, the pooling operation ignores the connections of each local feature and their surrounding visual words. To address this, we deem the connections between features and visual words as a kind of "higher level" features (here,

* Corresponding author. Tel.: +86 1381 022 6465.
E-mail address: jyl_999@163.com (J. Liu).

http://dx.doi.org/10.1016/j.neucom.2014.04.022
0925-2312/

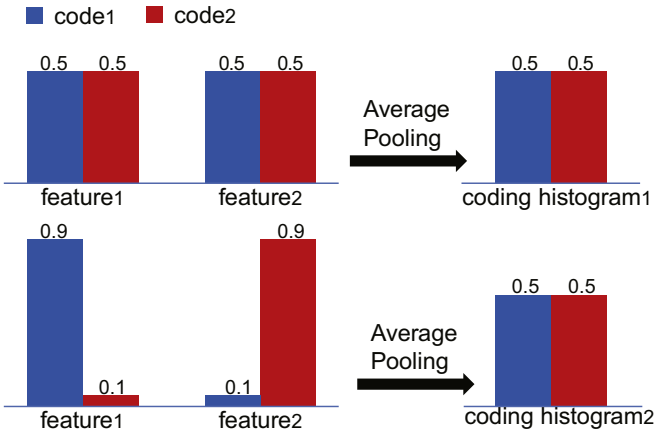

Fig. 1. Different feature appearances formulate the same visual word histogram after average pooling. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

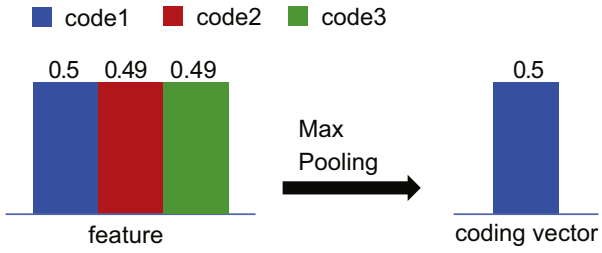

Fig. 2. Max pooling ignores other significant responses. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

"higher" is against the pixel level representation, e.g., SIFT [10] and HOG [11]). Based on this consideration, we propose a hierarchical BOF framework. In addition to the traditional pipeline, higher level features also generate the codebook and go through the stage of coding and pooling. In the end, a global histogram describing the frequency of connections between features and visual words are obtained.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 provides the details of various coding methods based on the hierarchical framework. Section 4 evaluates our framework on two different datasets and discusses why the two-layer framework improves the performance. Section 5 concludes the paper with discussion on future research.

2. Related work

In this section, we introduce related work of the BOF framework. A traditional BOF framework generally consists of the following stages:

(1) *Extract local features*: This step involves sampling local patches and describing them via classic feature descriptors. Local patches can be sampled in either a dense (with a fixed grid) or a sparse (with feature detectors) way. One of the typical feature descriptors is the scale-invariant feature transform (SIFT) descriptor [10]. It describes a local area by accumulating pixel gradients from each orientation weighted by their magnitude. In image classification, the general operation usually divides orientations into 8 bins in 16 sub-regions. Other typically used descriptors include local binary pattern (LBP) [12] and histogram of gradients (HOG) [11]. The inputs of this step are images, and the outputs are feature vectors.

(2) *Generate a codebook*: This step generates a codebook via learning from local features. For the computational efficiency, usually a subset of descriptors are randomly selected from all feature vectors obtained from the first step. The learning procedure is often implemented by unsupervised learning, e.g., K-means [13], or supervised learning [14]. Clustered centers are approximations of features and are often called codewords. In general, performance would be enhanced as the number of codewords becomes larger, since feature appearance spans over a large space and more codewords can present more sophisticated appearance of features. The inputs of this step are feature vectors and the output is the codebook consisting of codewords.

(3) *Encode features*: This step encodes local features to the codewords. Each feature will activate its nearest codewords measured in the feature space, and one or more codewords might obtain responses. Many encoding methods have emerged since it is not trivial to determine which codeword to activate as well as the weight with it. The input of this step is the codebook and the output is the coding vector. There are mainly five kinds of coding methods [15].

- Voting-based methods [8,16] apply a histogram to approximate the probability distribution of features. Each feature votes to its nearest one or multiple codewords, and the weight with the vote is obtained by hard quantization or soft quantization.
- Reconstruction-based methods [17–19] employ a subset of codewords to reconstruct a feature. Penalty is added to assure that few codewords are employed. So the optimization problem is formulated with certain constraints on the codewords, and the target is to minimize the reconstruction error. Sparse coding is widely used in reconstruction-based methods, wherein constraint terms are the main differences among various methods [20–26].
- Saliency-based coding [27] introduces the concept of codeword saliency, which is measured by relative proximity of the closest codeword compared with other codewords. Combining with MAX pooling, only the strongest response is preserved, indicating that the codeword can independently describe the feature without others.
- Local tangent-based coding [28] models features and codewords based on the manifold theory. It is assumed that codewords are located on the same smooth manifold constituted by all features. The encoding is formulated by using codewords to approximate the manifold. Lipschitz smooth function is applied to express the feature manifold.
- Fisher coding [29] is based on the Fisher kernel, which uses the gradient vector of its probability density function to describe a signal. IFK [30] employs Gaussian Mixture Model to estimate feature distributions. Each of the multiple Gaussian distributions reflects one pattern of features. Mean vector and covariance matrix are used to encode features.

(4) *Pool features*: This step is implemented via pooling votes obtained by each code. Typical pooling methods involve average pooling by averaging all the votes and MAX pooling by picking the most significant vote. One major drawback of pooling is that it ignores the spatial distribution in the process of the descriptor quantization. The problem can be partially resolved via spatial pyramid matching (SPM) [31] and multiple spatial pooling (MSP) [32]. SPM partitions an image into increasingly finer subregions and then employs pooling independently in them, which accords with the regular spatial structure of images from a particular category. An in-depth research on pooling can be found in [33].