Neurocomputing 182 (2016) 48–55

Contents lists available at ScienceDirect

Neurocomputing

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

Incorporating visual adjectives for image classification

Lingxi Xie [a], Jingdong Wang [b], Bo Zhang [a], Qi Tian [c,*]

[a] LITS, TNLIST, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[b] Microsoft Research, Beijing 100080, China
[c] Department of Computer Science, University of Texas at San Antonio, TX 78249, USA

ARTICLE INFO

Article history:
Received 2 August 2015
Received in revised form
21 October 2015
Accepted 1 December 2015
Communicated by Bin Fan
Available online 17 December 2015

Keywords:
Visual adjectives
Image classification
The Bag-of-Features model
Experiments

ABSTRACT

Image classification is a fundamental problem in computer vision which implies a wide range of real-world applications. Conventional approaches for image classification often involve image description and training/testing phases. The Bag-of-Features (BoF) model is one of the most popular algorithms for image description, in which local descriptors are extracted, quantized, and summarized into global image representation.

In the BoF model, all the visual descriptors are naturally treated as **nouns**, and plenty of useful contents are ignored. In this paper, we suggest to extract descriptive information, known as **adjectives**, to help visual recognition. We propose a simple framework to integrate various types of adjectives, i.e., *color* (or *brightness*), *shape* and *location*, for more powerful image representation. Experimental results on both scene recognition and fine-grained object recognition reveal that our approach achieves superior classification accuracy with reasonable computational overheads. It is also possible to generalize our model to many other multimedia applications such as large-scale image search.

1. Introduction

Image classification is a fundamental problem, which is closely related to a wide range of computer vision applications, including object recognition and detection, multimedia information retrieval, image tagging and recommendation, etc. Recent years have witnessed the emersion of fine-grained and large-scale image classification, introducing new challenges into this traditional research field.

The Bag-of-Features (BoF) model [1] is one of the most popular algorithms for image classification. It is a statistics-based model aimed at producing better image representation. Due to the limited descriptive power of raw pixels, handcrafted descriptors such as SIFT [2] are extracted. A visual vocabulary or codebook is then built to capture data distribution in the feature space. Descriptors are thereafter quantized on the codebook as compact signatures, and summarized as an image-level vector, which is the output of the BoF model. The high-dimensional representation vector could also be used for other multimedia applications, such as image retrieval [3] and object detection [4].

In the conventional BoF model, all the extracted descriptors are actually treated as **nouns**. By nouns we mean that they are concentrated on describing a specific aspect of an object, and no descriptive information is incorporated. We illustrate the shortcoming of this model in Fig. 1. When we are concerning about some fine-grained properties of an image, such as the *model* of an *aircraft*, or the *weather condition* of a *scene*, it is most often the subtle differences in local patches that reveal the answer. For example, an *A380* might be distinguished from a *Tornado* by the *shape* of the *plane nose*, and the *brightness* of the ground is the main evidence to judge if the *weather* is `sunny` or `cloudy`. Both *shape* and *brightness* are examples of **descriptive information** of a patch. Without such information, the BoF model might either ignore the subtle differences (e.g., quantizing `dark` and `bright` *grounds* into an identical word), or fail to capture the relationship between them (e.g., regarding `blunt` and `sharp` *plane noses* as two independent words). Both strategies might introduce considerable information loss and harm the discriminative power of image representation.

In this paper, we present a simple idea which integrates **visual adjectives** for image classification. Our main contribution is to design an efficient framework and suggest various types of adjectives, e.g., *color* (or *brightness*), *shape* and *location* signatures, to enhance local descriptors. By jointly training and testing with both concrete (noun) and descriptive (adjective) information, our approach achieves superior classification accuracy without requiring extra online computational overheads. It is also worth emphasizing that we do not aim at developing a novel approach, since all the adjectives extracted are pre-existed meanwhile we

* Corresponding author.
E-mail addresses: 198808xc@gmail.com (L. Xie), jingdw@microsoft.com (J. Wang), dcszb@mail.tsinghua.edu.cn (B. Zhang), qitian@cs.utsa.edu (Q. Tian).

http://dx.doi.org/10.1016/j.neucom.2015.12.008
0925-2312/

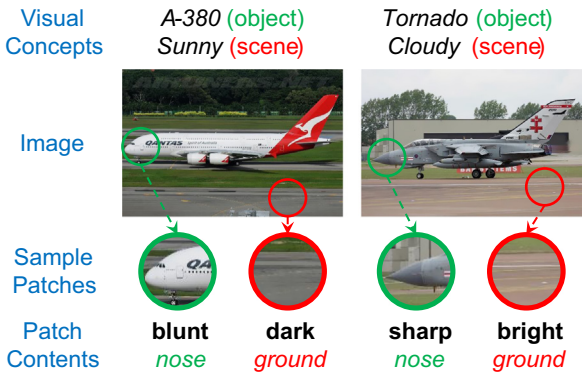

Fig. 1. Descriptive information, such as *shape* and *brightness*, helps to recognize visual concepts, such as the *weather* condition and the *model* of an *aircraft*. **Bold** and *italic* fonts indicate visual adjectives and nouns, respectively.

just perform feature fusion before the classification stage. What we want to deliver is an alternative efficient way of combining multiple sources of visual clues together.

The remainder of this paper is organized as follows. First, several related works are reviewed in Section 2. In Section 3, we illustrate our framework and introduce several descriptive adjectives for classification. After experimental results are shown in Section 4, we draw the conclusions in Section 5.

2. Related works

The related works to our research could be roughly partitioned into two parts, i.e., the conventional Bag-of-Features (BoF) Model for basic image description, and the approaches of incorporating complementary information into image representation.

2.1. The Bag-of-Features model

The Bag-of-Features (BoF) model is one of the most popular algorithms for image representation. It is composed of three major stages, i.e., descriptor extraction, feature encoding and feature summarization.

2.1.1. Descriptor extraction

The BoF model starts from extracting local descriptors. Due to the limited descriptive power of raw pixels, handcrafted descriptors are often extracted from small patches named interest points on an image.

For patch detection, gradient-based operators try to find local maxima which may correspond to well-defined interest points. Typical examples include Differential of Gaussian (DoG) [2], Hessian/Harris Affine [5], Maximally Stable Extremal Region (MSER) [6] operators and dense interest points [7]. Particularly, in image classification, it is also suggested to densely extract descriptors from a regular grid on the image [8].

For patch description, popular cases include Scale Invariant Feature Transform (SIFT) [2], and Histogram of Oriented Gradients (HOG) [9]. Other variants, such as Gradient Location and Orientation Histogram (GLOH) [10], Speeded Up Robust Features (SURF) [11], Binary Robust Independent Elementary Features (BRIEF) [12], DAISY descriptor [13] and Oriented FAST and Rotated BRIEF (ORB) [14], are also verified efficient and robust in image classification/retrieval tasks.

Either combination of patch detection or description algorithms yields a set $\mathcal{D}$ of local descriptors:

$$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\} \quad (1)$$

where $\mathbf{d}_m$ and $\mathbf{l}_m$ denote the D-dimensional description vector and the geometric location of the mth descriptor, respectively. M is the total number of dense descriptors. There might be more than one descriptor sets for an image in the cases of using multiple local descriptors.

2.1.2. Codebook training

After descriptor extraction and prior to feature encoding, a visual vocabulary (codebook) is trained to estimate the feature space distribution. The codebook is often computed with iterative algorithms such as K-Means or Gaussian Mixture Models (GMM).

K-Means is based on the kernel density model, which constructs K vectors with D dimensions:

$$\mathcal{B} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\} \quad (2)$$

The element $\mathbf{c}_k$, $k = 1, 2, \ldots, K$, is named a codeword, and each descriptor is then related to its nearest codeword(s) by Euclidean distance in the feature space.

On the other hand, the Gaussian Mixture Model (GMM) is trained to capture richer geometric contexts in the feature space. It describes the feature space with a mixture of K multi-variant Gaussian distributions:

$$\mathcal{M} = \{(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)\} \quad (3)$$

Parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ denote the prior, mean value and covariance of the kth Gaussian component, respectively, for $k = 1, 2, \ldots, K$.

Both K-Means and GMM could be solved iteratively with EM-based algorithms.

2.1.3. Feature encoding

Then, the feature encoding stage is aimed at quantizing each of the descriptors into a compact representation.

If the codebook is trained with K-Means clustering, i.e., composed of a set of codewords, then a descriptor could be encoded according to its distances to the codewords in the feature space. Hard quantization uses the nearest codeword to quantize a descriptor, resulting in a large quantization error. As an alternative solution, soft quantization allows a descriptor to be reconstructed by a small number of codewords. Sparse Coding [15] is a special case of soft quantization, which is verified very efficient in image classification [16,17]. After encoding, each descriptor $\mathbf{d}_m$ is represented as a K-dimensional, sparse feature vector $\mathbf{w}_m$, i.e., in which only one or few of the dimensions are non-zero.

If the codebook is trained with a GMM, i.e., geometric context information is preserved, richer discriminative features could be captured by computing the Fisher vectors [18]. It works by decomposing the Fisher Information Matrix towards maximal discrimination [19]. In this case, both the first-order and the second-order statistics are encoded, resulting in a much longer ($2DK$ dimensions) and denser (around 50% dimensions are non-zero) feature vector. Consequently, the time and memory costs are much more expensive than using K-Means based encoding. Similar ideas are also used in other high-dimensional features, such as Super Vector encoding [20] and Oriented SIFT/HOG encoding [21].

After the encoding stage, the set of local descriptors is transformed as a set of feature vectors:

$$\mathcal{W} = \{(\mathbf{w}_1, \mathbf{l}_1), (\mathbf{w}_2, \mathbf{l}_2), \ldots, (\mathbf{w}_M, \mathbf{l}_M)\} \quad (4)$$

In which, $\mathbf{d}_m$ in Eq. (1) is replaced by $\mathbf{w}_m$, for $m = 1, 2, \ldots, M$.

2.1.4. Feature summarization

As the final stage, quantized feature vectors are summarized into a compact image representation. For this respect, both feature pooling and normalization techniques are adopted.