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/

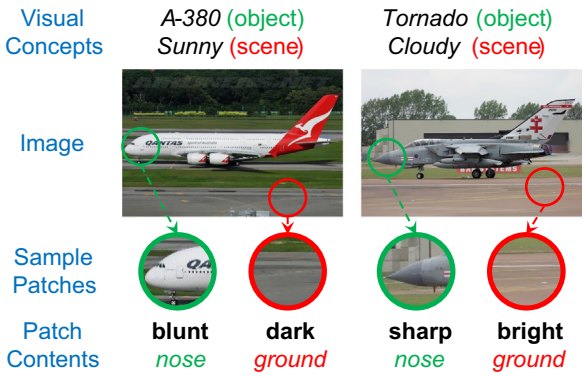


Fig. 1. Descriptive information, such as *shape* and *brightness*, helps to recognize visual concepts, such as the *weather* condition and the *model* of an *aircraft*. **Bold** and *italic* fonts indicate visual adjectives and nouns, respectively.

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.

The pooling stage is crucial in the BoF model, not only for it summarizes different numbers of local features into a vector of the same length, but also for its effect of canceling out the translation variance, allowing an object to appear on different positions of an image. A natural way of feature pooling is to calculate a global statistics based on all the quantized vectors. Max-pooling and average-pooling are probably the most widely adopted approaches. The max-pooling strategy calculates the maximal response on each codeword: $\mathbf{f}=\max_{1\le m\le M}\mathbf{w}_m$, while the average-pooling strategy calculates the average response: $\mathbf{f}=(1/M)\sum_{m=1}^{M}\mathbf{w}_m$. Here the notations $\max_m$ and $\sum_m$ denote dimension-wise maximization and summation, respectively. Researchers have discussed the choice of max-pooling versus average-pooling [22], showing that max-pooling gives more discriminative representation under soft quantization strategies, while average-pooling fits hard quantization better. Generalized Max Pooling (GMP) [23] discusses the relationship between max-pooling and Fisher vectors.

Feature normalization, or feature scaling, is a crucial data preprocessing stage aimed at avoiding attributes in greater numeric ranges dominating those in smaller numeric ranges, and controlling the similarity measure between feature vectors. One of the most popular feature normalization methods is the ℓ_p-norm normalization, in which we divide each feature vector by its length in the ℓ_p space: $\tilde{\mathbf{F}}=\mathbf{F}/\|\mathbf{F}\|_p$, so that all the vectors become ℓ_p-unit-length. The selection of the norm p might significantly impact the performance of generalized classifiers. For instance, it is demonstrated that in SVM, ℓ_2 normalized vectors have the minimal structural risk [24].

Representation vectors are finally fed into a generic classifier such as a Support Vector Machine (SVM). The detailed discussion of classifiers goes out of the goal of this paper.

2.2. Incorporating complementary information

It is well known that each type of feature has its speciality and limitation. For example, SIFT is good at describing *textual* features but less capable of capturing *color* [25] or *shape* [26] properties. To provide complementary description, researchers suggest to extract multiple types of local features, including Color-SIFT [25] and Local Color Statistics (LCS) [18] for *color* description, Shape Context (SC) [27] and Inner-Distance Shape Context (IDSC) [28] for *shape* formulation, etc.

Based on the various kinds of visual features, there are several research efforts aimed at combining multiple features into one image representation. Most of them could be categorized according to the *stage* on which features are fused. When two or more types of descriptors/features are extracted, it is possible to fuse them at the *descriptor-stage* (mixing both descriptors on the image plane) [29], at the *feature-stage* (concatenating encoded feature vectors into one) [26,30], or at the *classifier-stage* (fusing individual classification scores for final judgement) [31]. Besides extracting explicit features, visual clues could also be captured in an implicit manner. For example, Spatial Pyramid Matching (SPM) [32] encodes the position information of local patches by partitioning an image onto individual pooling regions and performing feature summarization in each bin. In [33], it is suggested to model spatial layouts in a more efficient way.

There are more previous works exploring a sophisticated way of visual feature fusion. In [34], the co-occurrence between common visual features (nouns) is considered, with the use of "prepositions" and "comparative adjectives" which help to express the relationship between objects. In [35], a top-down attention map is constructed category-specifically with color features, and then deployed to modulate shape features by weighting heavier on the images which are likely to contain an object instance. In [36], a feature fusion algorithm based on logistic regression is presented, which takes the advantage of the different cues on different features, without being tied to any one of them. In [37], orientational information is extracted on indoor scene images, and local features are pooled into the pre-defined orientational bins as high-level visual representation. In [38], an adaptive fusion strategy is designed to compute the weight on each component automatically. In all the above works, visual adjectives serve as complimentary information to assist visual recognition with nouns. Visual adjective information could also be applied to other high-level visual tasks, such as attribute learning [39,40] and context discovery and prediction [41].

In this work, we adopt a similar idea to [33], but allows more types of descriptive information to be encoded. There are merely complicated modules in the algorithm, the only thing we do is pre-classification fusion. A main attraction of our algorithm lies in that we are dealing with generic image settings, therefore our method could be applied to any type of image collections (e.g., object and scene recognition). Meanwhile, it does not require extra computational costs on the online classification stage, which could be friendly to real-world applications such as those run on mobile devices.

3. The proposed method

3.1. The framework

The goal of this work is to incorporate **visual adjectives**, i.e., descriptive information, to help image classification. We suggest to extract additional signatures at the descriptor-stage, since such an early fusion strategy increases the flexibility of the classification model [29].

Formally, let $\mathbf{d}_0$ be a noun (i.e., the SIFT descriptor), and $\{\mathbf{d}_1,\ldots,\mathbf{d}_L\}$ be a set of L adjectives (descriptive vectors). We aim at obtaining a vector $\mathbf{d}$ that contains all the above information. We follow a popular strategy used for fusing descriptors [25] and features [42], which directly concatenates individual descriptors into one and allows a simple weighting scheme: $\mathbf{d}=\left[\mathbf{d}_0^\top;w_1\cdot\mathbf{d}_1^\top;\ldots;w_L\cdot\mathbf{d}_L^\top\right]^\top$. In which, w_l controls the weight of the lth adjective, for $l=1,2,\ldots,L$. In this way, we treat the appended adjectives as new channels and maximally preserve their descriptive power.

The main observation is that the weight w_l could be computed using the ℓ_2-norm of the noun $\mathbf{d}_0$ and the corresponding adjective $\mathbf{d}_l$, i.e., $w_l\approx\|\mathbf{d}_0\|_2\cdot\|\mathbf{d}_l\|_2^{-1}$. An intuitive explanation is to guarantee that the ℓ_2-norm of each adjective is approximately the same as the noun (SIFT), i.e., $\|\mathbf{d}_0\|_2\approx\|w_l\cdot\mathbf{d}_l\|_2$. Since SIFT is originally designed to be cooperated with Euclidean (ℓ_2) distance [2], we are actually assuming that the contribution of the noun is the same as each of the appended adjectives.

The proposed framework is illustrated in Fig. 2. We shall emphasize that our model is capable of integrating an arbitrary number of visual adjectives. Appending these adjectives provides the possibility of incorporating various types of visual clues for image classification. To prevent redundancy, we adopt Principle Component Analysis (PCA) to reduce the descriptors into a fixed dimension. After PCA, related components from nouns and adjectives are projected onto the same dimension, thus later modules (codebook training, feature encoding, etc.) could be considered as a joint learning process of nouns and adjectives.

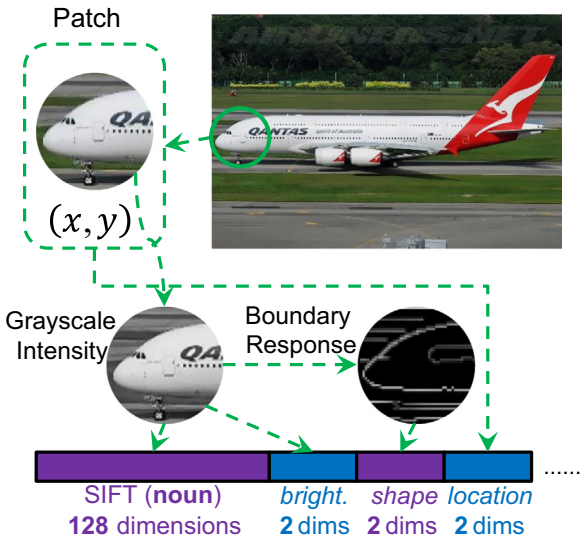


Fig. 2. Illustration of the proposed framework. *Brightness* (or *color*), *shape* and *location* adjectives are computed from different visual clues, i.e., grayscale intensity, boundary response and coordinates, providing complementary information to the original SIFT descriptor (noun).

3.2. Typical visual adjectives

In this part, we suggest several visual adjectives to provide complementary information to the SIFT descriptor.

- *Color* is the visual perceptual property corresponding in humans deriving from the spectrum of light. It degenerates to *brightness* (also known as *luminance*) after grayscale transformation. We borrow the idea from a simple color descriptor, LCS [18], which partitions a local patch into 4×4 grids and computes the mean and standard deviation on each channel and each grid. Such a 96-dimensional descriptor is too long to be an adjective, so we shorten it by computing the statistics merely on the whole patch, resulting in a 6-dimensional vector. We can similarly obtain a 2-dimensional *brightness* adjective on the grayscale intensity map.
- *Shape* is the form of an object or its external boundary, outline, or external surface, as opposed to other properties such as color, texture and material composition. To capture shape features, we compute boundary responses with Ultrametric Contour Map (UCM) [43], which is a closed contour extraction method based on a preliminary edge operator [44]. Similar to the extraction of *brightness* adjectives, we compute a 2-dimensional *shape* adjective for each patch by computing the mean and standard deviation on the boundary response (a grayscale intensity map).
- The importance of *location* for visual recognition is well known. We take the central coordinate of a patch as a 2-dimensional *location* adjective. This is very similar to [33] which also appends scale description. It is instructive to note that in many image retrieval systems [45,46], geometric information of local patches, such as location, scale and orientation, is also extracted to provide richer description. We verify that efficient embedding of geometric information into local features helps classification, just like what happens in image retrieval cases [47].

To normalize the adjectives, we first minus each number by the average of the corresponding dimension, and then divide each number by the maximum of the absolute value of that dimension. Finally, each number is multiplied by the factor L which is the ℓ_2-norm of the noun, i.e., the SIFT descriptor. In this way, the average ℓ_2-norm of each adjective is approximately the same as that of the corresponding SIFT descriptor.

We shall admit that many other options, including more types of adjectives and more ways of computing adjectives, could be taken. One might even extract multiple adjectives to describe one visual property in different aspects, e.g., computing both RGB and LAB *color* statistics or adding Shape Contexts (SC) [27] for *shape* description. It is also possible to append long adjectives to provide stronger descriptive information. Here we only present three straightforward examples to reveal the ability of our model (see the next section).

4. Experimental results

4.1. Datasets and settings

We evaluate the proposed algorithm on six popular image classification datasets. Among these datasets, three of them are designed for scene recognition, and the other three for fine-grained object recognition.

For scene recognition, we use the **Scene-15** dataset [32] (15 *general scenes*, 4485 images), the **LandUse-21** dataset [48] (21 *land-use scenes*, 100 images for each class), and the MIT **Indoor-67** dataset [49] (67 *indoor scenes*, 15,620 images). The numbers of randomly selected samples per category for training are 100, 80 and 80, respectively. For fine-grained object recognition, we use the **Aircraft-100** dataset [50] (100 *aircraft* models, 100 images for each model), the Oxford **Flower-102** dataset [51] (102 *flower* categories, 8189 images), and the Caltech-UCSD **Bird-200** dataset [52] (200 *bird* species, 11,788 samples). The numbers of randomly selected samples per category for training are about 66, 20 and 30, respectively.

We follow a recently published BoF model [42] for image representation. An image is rescaled, with its aspect ratios preserved, so that the larger axis is 300 pixels. When a bounding box is available, we only use the region within the box. We use the VLFeat [53] library to extract dense RootSIFT [54] descriptors. The spatial stride and window size of dense sampling are 6 and 12, respectively. On each patch, one or more visual adjectives might be extracted and appended, The dimension of the enhanced descriptor (no matter how long it is) is reduced to 64 using PCA. We then cluster the descriptors with a GMM of $K = 128$ components. The number of descriptors collected for clustering does not exceed 1 million. We use the improved Fisher vectors (IFV) [18] for feature encoding. We do not use a spatial pyramid in order to reveal the effect of *location* adjectives. We use LibLINEAR [55], a scalable SVM implementation for training and testing. The average accuracy over all the tested categories are calculated. We repeat the random selection 10 times and report the averaged accuracy.

4.2. Model and adjectives

We summarize the classification results on six datasets in Table 1. The standard deviations are often less than or close to 1% and we simply ignore them to save space. We enumerate every possible combination of *color* (or *brightness*), *shape* and *location* adjectives, except for the **Scene-15** dataset (a grayscale image set) on which *color* adjectives are not computed. We also test the "baseline" algorithm in which no adjectives are extracted.

One may observe that adjectives produce consistent accuracy gain on every single case. This signifies that we do obtain additional visual clues that help recognition. Sometimes, the improvement might be surprisingly large, implying the extra information is most useful. For example, it is well known that *color* is a crucial property to recognize a *flower*. Our experiments verify

Table 1
Classification accuracy (%) of different models. Among which, "**SIFT**+C" enhances SIFT with *color* adjectives, "**SIFT**+B" enhances SIFT with *brightness* adjectives, etc. Most often, combining multiple types of adjectives leads to higher classification accuracy. In all the cases, the feature vectors are of 16,384 dimensions, thus the computational costs are almost the same.

Algorithm	S-15	L-21	I-67	A-100	F-102	B-200
Baseline (**SIFT**)	80.03	89.10	43.12	54.41	53.17	27.44
SIFT+B	81.63	91.57	46.52	56.89	57.11	30.87
SIFT+C	–	90.98	51.09	57.36	69.60	36.81
SIFT+S	81.18	90.76	44.71	56.72	55.80	30.17
SIFT+B+S	81.80	91.43	47.39	58.31	59.51	33.25
SIFT+C+S	–	91.95	51.83	58.50	69.92	38.74
SIFT+L	82.31	90.95	47.49	58.46	57.16	32.03
SIFT+B+L	83.03	91.64	49.32	60.17	59.85	32.96
SIFT+C+L	–	92.26	53.46	60.61	71.04	38.71
SIFT+S+L	82.57	91.62	48.04	59.73	59.18	33.56
SIFT+B+S+L	**83.25**	92.14	50.12	61.69	62.00	34.61
SIFT+C+S+L	–	**92.88**	**53.83**	**61.91**	**71.48**	**40.10**

this point, since *color* adjectives produce more-than-15% boost over the baseline algorithm. As a compressed version of *color*, *brightness* adjectives also work well on these cases, although the gain is much lower than using *color* (+4% versus +15%). Therefore, we prefer *color* to *brightness* except for grayscale datasets (e.g., **Scene-15**).

Similar cases also appear when either *shape* or *location* adjectives are integrated. In distinguishing fine-grained concepts such as an *aircraft*, a *flower* or a *bird*, incorporating *shape* might help to discriminative subtle differences between two categories/species, such as `long`-versus-`short` *bird tails*, and `sharp`-versus-`blunt` *plane noses*. Since we do not use spatial pyramids, *location* adjectives (coordinates) could provide complementary information in this aspect. As observed in [33], this implicitly causes the descriptors assigned to different codewords, leading to a soft-assignment version of spatial pooling.

The proposed framework also allows multiple types of adjectives being combined at one time. As shown in Table 1, integrating multiple adjectives often produces higher accuracy than using a single one. For example, appending *shape* and *color* adjectives produces an averaged 1.02% accuracy gain over using *color* adjectives alone, suggesting that complementary information is provided by different types of adjectives. Due to the marginal effect, the gain is lower than appending *shape* adjectives directly on baseline (2.18%). In overall, the best performance is always achieved when all types of adjectives are incorporated. However, it is still a kind reminder to carefully select proper adjectives for each task to avoid introducing redundant information.

4.3. Computational costs

Additional computational costs are mainly required at the stage of extracting adjectives. We need to compute patch statistics for *color* (*brightness*) and *shape* adjectives, as well as the boundary responses using the Ultrametric Contour Map (UCM) [43]. The elapsed time on each image depends on the image size. For an image with 300 pixels on its longer axis, computing UCM takes around 15 s, SIFT takes about 0.5 s, and all the other computation only involves simple patch statistics which could be finished within 0.5 s. If the image has 600 pixels on the longer axis, the computation often takes 4× time compared to above. To accelerate, we perform approximation on the most time-consuming part, i.e., boundary detection, by first extracting UCM on the half-sized image (both width and height are reduced by a half) and then rescale the detected boundary image to the same size as the original image with bicubic interpolation. Thus, a 600× 600 image could also be processed within 20 s. All the time costs are evaluated on a single 3.0 GHz CPU. Considering such computations are performed only once, the time/memory overheads are reasonable and affordable.

Table 2
Comparison of classification accuracy (%) with visual adjectives and multiple features. "× 2" implies to use double-sized codebook, and "4C" means to extract 4×-length *color* adjectives, as illustrated in Section 4.4. The algorithms **SIFT**+C, **SIFT**+S and **SIFT**+L produce feature vectors of 16,384 dimensions which is the same as the baseline method. However, other algorithms produce either 32,768-dimensional (*color* and *shape* adjectives) or 65,536-dimensional (*location* adjectives) feature vectors which are much longer.

Algorithm	S-15	L-21	I-67	A-100	F-102	B-200
Baseline (**SIFT**)	80.03	89.10	43.12	54.41	53.17	27.44
SIFT+**LCS**	–	91.71	54.24	58.43	**77.03**	**38.55**
RGB-SIFT	–	90.52	51.05	57.90	71.59	31.05
OPP-SIFT	–	91.14	53.57	48.03	76.19	35.20
SIFT+C	–	90.98	51.09	57.36	69.60	36.81
(**SIFT**+C) × 2	–	**91.98**	53.67	58.75	75.69	37.39
(**SIFT**+4C) × 2	–	91.90	**54.61**	**58.91**	76.50	38.51
SIFT+**Edge**−**SIFT**	81.87	90.98	45.91	58.10	56.51	31.63
SIFT+S	81.18	90.76	44.71	56.72	55.80	30.17
(**SIFT**+S) × 2	**82.10**	**91.17**	**46.48**	**58.19**	**57.30**	**31.89**
SIFT with SPM ($R=4$)	**84.79**	91.33	48.82	**59.61**	59.13	33.39
SIFT+L	82.31	90.95	47.49	58.46	57.16	32.03
(SIFT+L) × 4	84.28	**91.62**	**49.19**	59.48	**59.71**	**33.61**

One might notice that all local descriptors are reduced by PCA into a fixed dimension (64 in our implementation). By which, we guarantee that representation vectors are of the same dimension on every single case. In other words, we significantly improve the classification accuracy without requiring more computational costs after PCA reduction.

4.4. Discussions

There exist many previous works aimed at integrating multiple features for image classification. We implement three of them, i.e., Local Color Statistics (LCS) [18] for color features, RGB-SIFT and Opponent-SIFT (OPP-SIFT) descriptors [25] for color image description, Edge-SIFT [29] for shape features, and Spatial Pyramid Matching (SPM) [32] for location. The comparison with the proposed algorithm is summarized in Table 2. The standard deviations are ignored since they are often quite small (0.5–1%).

- In [42], it is suggested to extract *color* information by computing LCS descriptors, encoding them similarly as SIFT, and concatenating two representation vectors into one (of doubled length). To make the comparison fair, we train a larger codebook with 2× codewords based on SIFT with *color* adjectives. By this we obtain competitive results on scene recognition. For *bird* and *flower* recognition, *color* is fundamentally important, therefore it is not enough to use merely 6 dimensions for *color* description. When we extract 24-dimensional *color* adjectives on a 2× 2 grid, comparable accuracy is achieved.
- In [29], *shape* properties are described by computing SIFT on boundary responses, i.e., Edge-SIFT. Our algorithm with *shape* adjectives outperforms the competitor using equal-length representation vectors.
- SPM [32] is one of the most popular algorithms for encoding spatial information. We use $R=4$ pooling regions, namely the whole image and three horizontal stripes, on the baseline model. We also train a codebook with 4× codewords to jointly cluster the descriptors with *location* adjectives. These two models produce comparable classification accuracy.

Table 3
Comparison of classification accuracy (%) with the state-of-the-art on scene recognition. The term "VA" indicates visual adjectives.

Algorithm	**S-15**	**L-21**	**I-67**
Lazebnik et al. [32]	81.4	–	–
Quattoni et al. [49]	–	–	26.1
Yang et al. [16]	80.4	–	–
Boureau et al. [56]	84.3	–	–
Xiao et al. [30]	88.1	–	–
Yang et al. [48]	–	81.19	–
Xie et al. [29]	83.77	–	46.38
Kobayashi et al. [21]	85.63	–	58.91
Wang et al. [57]	88.7	–	–
Xie et al. [37]	–	–	63.48
Kobayashi et al. [58]	–	92.6	63.4
Ours (IFV [18])	89.70 ± 0.58	90.50 ± 1.20	59.17 ± 0.76
Ours (IFV + VA)	**91.89** ± 0.53	**93.88** ± 0.94	**65.04** ± 0.43

Table 4
Comparison of classification accuracy (%) with the state-of-the-art (without part detection) on fine-grained object recognition. The term "VA" indicates visual adjectives.

Algorithm	**A-100**	**F-102**	**B-200**
Nilsback et al. [51]	–	72.8	–
Wah et al. [52]	–	–	10.7
Chai et al. [59]	–	85.2	–
Maji et al. [50]	48.69	–	–
Murray et al. [23]	–	84.6	33.3
Pu et al. [60]	–	–	44.2
Ours (IFV [18])	71.71 ± 0.62	68.93 ± 0.71	38.86 ± 0.62
Ours (IFV + VA)	**77.98** ± 0.48	**86.31** ± 0.51	**50.12** ± 0.47

The difference between our algorithm and previous works mainly lies in the different stages and ways of performing feature fusion. We do not aim at presenting a novel algorithm, but evaluating the performance of various visual clues, and providing an alternative and efficient feature fusion method to combine them with texture descriptors such as SIFT. The highlight of our method is that we do not require more computational times on the online classification stage, which is often important for real-world algorithms, e.g., those run on mobile devices with limited computational resources.

4.5. Comparison to the state-of-the-art

We further compare our model with the state-of-the-art algorithms on all these datasets. We extract stronger features by rescaling the images into 600 pixels on the longer axis and using 1024 GMM components. Scene recognition and fine-grained object recognition results are summarized in Tables 3 and 4, respectively. In both tables, the notation "IFV" refers to the algorithm in which SIFT descriptors are used directly and the Improved Fisher Vectors (IFV) [18] are adopted to encode them, whereas by the notation "IFV+VA" we mean to combine visual adjectives into the SIFT descriptors and then encode them with IFV. One can see that our result is competitive among the state-of-the-art performance.

In fine-grained datasets, it is verified that aligned parts provide informative clues for visual recognition [63,61,62,64]. For comparison, we inherit the detected parts on the **Bird-200** dataset and the same experimental settings from [61,62]. Since they have extracted *color* features explicitly, we only append *shape* and *location* adjectives for this task. Results are shown in Table 5. One can observe that our algorithm achieves better performance without increasing computational overheads. Moreover, the use of *location* adjectives overcomes the difficulty that integrates spatial pooling with relatively small parts (it is often difficult to use SPM in such cases [62]).

Table 5
Comparison of classification accuracy (%) on the **Bird-200** dataset with detected parts. The baseline algorithm uses color-SIFT descriptors, so we do not extract *color* adjectives.

Algorithm	Chai et al. [61]	Gavves et al. [62]
Baseline (CSIFT)	56.6	62.7
Our implementation	55.92	61.94
CSIFT+S	57.75	63.40
CSIFT+L	58.02	63.81
CSIFT+S+L	**59.18**	**65.03**

4.6. Generalization

In this final part, we briefly discuss the possibility of generalizing our idea to other research areas, e.g., assisting algorithms based on deep learning and applying on other computer vision problems.

To inspire deep learning algorithms such as Convolutional Neural Networks (CNNs), one might notice that CNNs directly learn visual representation from raw-pixel inputs. This strategy highly depends on the amount of training data. We propose to incorporate visual adjectives, which are often simple statistics (e.g., intensity average or patch location) and could be considered as an additional neuron on the same input layer. We believe that such extra clues could help to alleviate the heavy requirement of large-scale training data.

Many other computer vision applications are based on local features, such as large-scale image retrieval. We could transplant the global features involving visual contexts [41] onto NN-based retrieval algorithms, or embed the visual adjectives into a scalable data structure such as the inverted index [65], just like those efforts incorporating geometric information [47,46]. In the task of near-duplicate image retrieval, e.g., the **Oxford buildings** dataset [45], we believe the use of *color* and *shape* adjectives could help to generate more accurate feature matches, and the *location* adjectives might provide another efficient solution of geometric verification.

5. Conclusions

In this paper, we propose to extract **visual adjectives** to help BoF-based image classification. With a simple and efficient framework, various adjectives are incorporated to provide complementary information and powerful image representation. Experiments verify that our system achieves the state-of-the-art recognition performance on both scene recognition and fine-grained object recognition tasks.

The success of our algorithm verifies the importance of incorporating multiple features, as well as the joint learning strategy to capture underlying relationship between different features. In the future, we will try to transplant this idea to inspire deep learning methods such as neural networks. Moreover, we will also explore its use on other computer vision applications such as large-scale image retrieval.

Acknowledgments

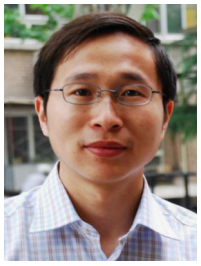
This work was supported by the National Basic Research Program (973 Program) of China (Grant nos. 2013CB329403 and 2012CB316301), the National Natural Science Foundation of China (Grant nos. 61332007, 61273023 and 61429201), and the Tsinghua University Initiative Scientific Research Program (Grant no. 20121088071). This work was also supported in part to Dr. Tian by ARO grant W911NF-15-1-0290 and W911NF-12-1-0057, and Faculty Research Awards by NEC Laboratories of America.

References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with Bags of Keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, 2004.
[2] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[3] X. Gu, Y. Zhang, D. Zhang, J. Li, Representative local features mining for large-scale near-duplicates retrieval, in: International Conference on Multimedia and Expo, 2014.
[4] T. Ye, D. Zhang, K. Gao, G. Jin, Y. Zhang, Q. Yuan, Salient region detection: integrate both global and local cues, in: International Conference on Multimedia and Expo, 2014.
[5] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. J. Comput. Vis. 60 (1) (2004) 63–86.
[6] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vis. Comput. 22 (10) (2004) 761–767.
[7] T. Tuytelaars, Dense interest points, in: Computer Vision and Pattern Recognition, 2010.
[8] A. Bosch, A. Zisserman, X. Munoz, scene classification via pLSA, in: European Conference on Computer Vision, 2006.
[9] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005.
[10] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.
[11] H. Bay, A. Ess, T. Tuytelaars, L.Van. Gool, Speeded up robust features (SURF), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.
[12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, BRIEF: computing a local binary descriptor very fast, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1281–1298.
[13] E. Tola, V. Lepetit, P. Fua, Daisy: an efficient dense descriptor applied to wide-baseline stereo, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 815–830.
[14] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: International Conference on Computer Vision, 2011.
[15] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems, 2007.
[16] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Computer Vision and Pattern Recognition, 2009.
[17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Computer Vision and Pattern Recognition, 2010.
[18] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, in: European Conference on Computer Vision, 2010.
[19] T. Jaakkola, D. Haussler, et al., Exploiting generative models in discriminative classifiers, in: Advances in Neural Information Processing Systems, 1999.
[20] X. Zhou, K. Yu, T. Zhang, T. Huang, Image classification using super-vector coding of local image descriptors, in: European Conference on Computer Vision, 2010.
[21] T. Kobayashi, BoF meets HOG: feature extraction based on histograms of oriented pdf gradients for image classification, in: Computer Vision and Pattern Recognition, 2013.
[22] Y. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in: International Conference on Machine Learning, 2010.
[23] N. Murray, F. Perronnin, Generalized max pooling, in: Computer Vision and Pattern Recognition, 2014.
[24] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, New York, 1995.
[25] K. Van De Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[26] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, in: Computer Vision and Pattern Recognition, 2007.
[27] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[28] H. Ling, D. Jacobs, Shape classification using the inner-distance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299.
[29] L. Xie, Q. Tian, M. Wang, B. Zhang, Spatial pooling of heterogeneous features for image classification, IEEE Trans. Image Process. 23 (5) (2014) 1994–2008.
[30] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, SUN database: large-scale scene recognition from abbey to zoo, in: Computer Vision and Pattern Recognition, 2010.
[31] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, C. Schmid, Transformation pursuit for image classification, in: Computer Vision and Pattern Recognition, 2014.
[32] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition, 2006.
[33] J. Sanchez, F. Perronnin, T. De Campos, Modeling the spatial layout of images beyond spatial pyramids, Pattern Recognit. Lett. 33 (16) (2012) 2216–2223.
[34] A. Gupta, L. Davis, Beyond nouns: exploiting prepositions and comparative adjectives for learning visual classifiers, in: European Conference on Computer Vision, 2008.
[35] F. Khan, J. Van de Weijer, M. Vanrell, Top-down color attention for object recognition, in: International Conference on Computer Vision, 2009.
[36] B. Fernando, E. Fromont, D. Muselet, M. Sebban, Discriminative feature fusion for image classification, in: Computer Vision and Pattern Recognition, 2012.
[37] L. Xie, J. Wang, B. Guo, B. Zhang, Q. Tian, Orientational pyramid matching for recognizing indoor scenes, in: Computer Vision and Pattern Recognition, 2014.
[38] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, Q. Tian, Query-adaptive late fusion for image search and person re-identification, in: Computer Vision and Pattern Recognition, 2015.
[39] O. Russakovsky, L. Fei-Fei, Attribute learning in large-scale datasets, in: Trends and Topics in Computer Vision, 2012.
[40] A. Lazaridou, G. Dinu, A. Liska, M. Baroni, From Visual Attributes to Adjectives through Decompositional Distributional Semantics, in: arXiv preprint, arXiv: arXiv:1501.02714, 2015.
[41] C. Doersch, A. Gupta, A. Efros, Unsupervised Visual Representation Learning by Context Prediction, in: arXiv preprint, arXiv: arXiv:1505.05192, 2015.
[42] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the Fisher vector: theory and practice, Int. J. Comput. Vis. 105 (3) (2013) 222–245.
[43] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, From contours to regions: an empirical evaluation, in: Computer Vision and Pattern Recognition, 2009.
[44] J. Canny, A computational approach to edge detection, Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[45] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Computer Vision and Pattern Recognition, 2007.
[46] W. Zhou, H. Li, Y. Lu, Q. Tian, Large scale image search with geometric coding, in: ACM Multimedia, 2011.
[47] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, 2008.
[48] Y. Yang, S. Newsam, Bag-of-Visual-Words and spatial extensions for land-use classification, in: International Conference on Advances in Geographic Information Systems, 2010.
[49] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Computer Vision and Pattern Recognition, 2009.
[50] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-Grained Visual Classification of Aircraft, in: arXiv preprint, arXiv: arXiv:1306.5151, 2013.
[51] M. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Indian Conference on Computer Vision, Graphics & Image Processing, 2008.
[52] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD birds-200-2011 Dataset, Technical Report: CNS-TR-2011-001.
[53] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, in: ACM Multimedia, 2010.
[54] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: Computer Vision and Pattern Recognition, 2012.
[55] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[56] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: Computer Vision and Pattern Recognition, 2010.
[57] P. Wang, J. Wang, G. Zeng, W. Xu, H. Zha, S. Li, Supervised Kernel descriptors for visual recognition, in: Computer Vision and Pattern Recognition, 2013.
[58] T. Kobayashi, Dirichlet-based histogram feature transform for image classification, in: Computer Vision and Pattern Recognition, 2014.
[59] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, A. Zisserman, TriCoS: A tri-level class-discriminative co-segmentation method for image classification, in: European Conference on Computer Vision, 2012.
[60] J. Pu, Y. Jiang, J. Wang, X. Xue, Which looks like which: exploring inter-class relationships in fine-grained visual categorization, in: European Conference on Computer Vision, 2014.
[61] Y. Chai, V. Lempitsky, A. Zisserman, Symbiotic segmentation and part localization for fine-grained categorization, in: International Conference on Computer Vision, 2013.
[62] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, T. Tuytelaars, Fine-grained categorization by alignments, in: International Conference on Computer Vision, 2013.
[63] T. Berg, P. Belhumeur, POOF: Part-based One-vs.-One Features for fine-grained categorization, face verification, and attribute estimation, in: Computer Vision and Pattern Recognition, 2013.

[64] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based R-CNNs for fine-grained category detection, in: European Conference on Computer Vision, 2014.
[65] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: International Conference on Computer Vision, 2003.

Lingxi Xie received the B.E. and Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2010 and 2015, respectively. He is currently a post-doctoral researcher at University of California, Los Angeles. He was a research intern in Microsoft Research Asia from August 2013 to January 2014, and from September 2014 to June 2015. He was a visiting researcher at the Department of Computer Science at the University of Texas at San Antonio (UTSA) from February to July 2014. His research interests include computer vision, multimedia information retrieval and machine learning.



Jingdong Wang received the B.Sc. and M.Sc. degrees in the Department of Automation from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in the Department of Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is currently a Lead Researcher at the Visual Computing Group, Microsoft Research. His areas of interest include computer vision, machine learning, and multimedia search. At present, he is mainly working on the Big Media project, including large-scale indexing and clustering, and Web image search and mining. He is an editorial board member of Multimedia Tools and Applications.

Bo Zhang received the B.E. degree from the Department of Automatic Control, Tsinghua University, China, in 1958. From 1980 to 1982, he visited University of Illinois at Urbana-Champaign, USA, as a scholar. He is currently a Professor in the Department of Computer Science and Technology at Tsinghua University. His research interests include artificial intelligence, machine learning, pattern recognition, knowledge engineering, intelligent robotics and intelligent control. He won the Prize of European Artificial Intelligence in 1984. He won the 1st-class of Science and Technology Progress Prize three times in 1987, 1993 and 1998, and the 3rd-class Prize two times in 1995 and 1999, respectively. He is a member of Chinese Academy of Sciences.

Qi Tian received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University in 1996 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign in 2002. He is currently a Professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA). He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008–2009. His research interests include multimedia information retrieval and computer vision.