Expert Systems With Applications 94 (2018) 250–264

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# A pooled Object Bank descriptor for image scene classification

Mujun Zang[a,*], Dunwei Wen[b], Tong Liu[a], Hailin Zou[a], Chanjuan Liu[a]

[a] School of Information and Electrical Engineering, Ludong University, Yantai, China
[b] School of Computing and Information Systems, Athabasca University, Alberta, Canada

ARTICLE INFO

*Article history:*
Received 12 January 2017
Revised 12 October 2017
Accepted 30 October 2017
Available online 31 October 2017

*Keywords:*
Image classification
Object Bank
Dimensionality reduction
Pooling
Image feature

ABSTRACT

Object Bank (OB) is a high-level image representation encoding semantic and spacial information, and has superior performance in scene classification tasks. However, the dimensionality of OB feature is high, which demands massive computation. Existing dimensionality reduction methods for OB are incapable of achieving both high classification accuracy and substantial dimensionality reduction simultaneously. In order to solve this problem, we propose a threshold value filter pooling method to avoid noise accumulation in histogram-pooling and represent more useful information than max-pooling. We also propose a Matthew effect normalization method to highlight the useful information, and thus boost the performance of OB-based image scene classification. Finally, we apply these two methods in a dimensionality reduction framework to simplify OB representation and construct more proper descriptors, and thus achieve both dimensionality reduction and classification accuracy increase. We evaluated our framework on three real-world datasets, namely, event dataset UIUC-Sports, natural scene dataset LabelMe, and mixture dataset 15-Scenes. The classification results demonstrate that our framework not only obtains accuracies similar to or higher than the original OB representation, but also reduces the dimensionality significantly. The computational complexity analysis shows that it can reduce the time complexity of classification. Therefore, our framework can improve OB-based image scene classification through both computational complexity reduction and accuracy increase.



## 1. Introduction

Image scene classification is an important problem in computer vision. The selection and setting of image representations often significantly affect the performance of the classification. An image representation first encodes the local patches according to low-level features (such as SIFT (Lowe, 1999), FilterBanks (Freeman & Adelson, 1991; Perona & Malik, 1990), GIST (Oliva & Torralba, 2001), HOG (Dalal & Triggs, 2005), GBPWHGO (Zhou, Zhou, & Hu, 2013), etc.), and then adopts pooling method (Lazebnik, Schmid, & Ponce, 2006; Shi, Wan, Wu, & Chen, 2017; Yang, Yu, Gong, & Huang, 2009), topic models (Bosch, Zisserman, & Muñoz, 2006; Fei-Fei & Perona, 2005; Niu, Hua, Gao, & Tian, 2012; Wang, Blei, & Li, 2009; Zang, Wen, Wang, Liu, & Song, 2015), Fishier Vector (Perronnin & Dance, 2007; Sánchez, Perronnin, Mensink, & Verbeek, 2013), or Attributes Learning (Su & Jurie, 2012; Wang, Yu, & Tao, 2013), to produce high-level features. Although these methods have some advantages, their low-level image representations are potentially not enough for high-level visual tasks, where features representing high-level semantic information seems to be more effective (Li, Su, Lim, & Fei-Fei, 2012; 2014), as demonstrated by Object Bank (OB), a recent high-level image representation that directly extracts high-level object appearance and spacial location information (Boureau, Le Roux, Bach, Ponce, & LeCun, 2011; Li, Su, Fei-Fei, & Xing, 2010; Li et al., 2012; 2014; Pandey & Lazebnik, 2011). It treats the objects as attributes for scene classification, each scene image is represented as a scale-invariant response map of a large number of pre-trained generic object detectors, and the feature is extracted from the responses in these detectors. In high-level visual recognition tasks, OB has achieved state-of-the-art performance (Li et al., 2010; Li et al., 2012; 2014), but its dimensionality is too high.

To reduce the dimensionality of OB, dimensionality reduction technologies can be applied. The feature dimensionality reduction technologies like PCA, LLE (Roweis & Saul, 2000), Laplacian (Belkin & Niyogi, 2001), and Isomap (Tenenbaum, De Silva, & Langford, 2000), have achieved great success in image classification. PCA method has been especially employed to reduction dimensionality of OB (Li, Su, Lim, & Fei-Fei, 2014). However, a definite size of dimensionality reduction is often accompanied by accuracy decrease. Pooling method can significantly reduce the dimensionality

* Corresponding author.
*E-mail addresses:* zangmj12@mails.jlu.edu.cn (M. Zang), dunweiw@athabascau.ca (D. Wen), tongliu13@mails.jlu.edu.cn (T. Liu), zhl_8655@sina.com (H. Zou), luckycj80@sina.com (C. Liu).

https://doi.org/10.1016/j.eswa.2017.10.057
0957-4174/


**Fig. 1.** Sample images of the *sailing* scene. For image scene classification task, the object "sailboat" is key for classification, but its number, position and size are less important.

by merging features of each image region and its neighborhoods, and has advantages in both dimensionality reduction and accuracy increase for image scene classification (Lazebnik et al., 2006; Yang et al., 2009). Although the generating process of OB has contained a max-pooling step, it should be noted that, if the pooling method is adopted to further reduce the dimension of OB, the classification accuracy will be decreased (The accuracy of dimensionality-reduced OB by directly pooling is about 2/3 of original OB. This experiment will be shown in Section 5.1).

In order to avoid a decrease of accuracy while reducing the dimensionality, the importance of object and other factors should be taken into account. We discuss those factors in two kinds of categories, namely, "key objects" dominated categories and "non-key objects" categories. In the first case, key objects play the decisive role in classification for some categories, while local features or scene environment features are auxiliary. For instance, observing an image containing the object "sailboat", one can almost certain that the image is corresponding to *sailing* category. The other relevant information, such as outdoor environment and the color and texture of each local area, may strengthen the result rather than determine its category. In the second case, the environment and local features may play a dominant role, while the object is just auxiliary. For instance, observing an image containing "car", "road" and "sky" objects is not quite enough to decide whether this image should be classified to *highway* or *insidecity*.

After knowing the significance of objects, it is important to find out what the primary factor is. On the one hand, for these key objects dominated categories, the most beneficial information is the possibilities of images containing the key object, rather than the number or position of the objects and so on. For example, an image of *sailing* category empirically contains at least one "sailboat" object, and this object usually has many descriptors whose response is very large. However, the other information related to "sailboat" object, such as the number, sizes, and positions of the sailboats, is unnecessary, just as shown in Fig. 1. On the other hand, for those images that the objects do not play the key role, we may not be able to get more useful information by making this unnecessary information clearer. As shown in Fig. 2, we cannot distinguish between *highway, street*, and *insidecity* by knowing that a "car" is contained in an image, or by knowing even more information related to "car" object such as its numbers, sizes, and positions in the image. That said, given that the image with "car" objects is more probable to be classified as *highway, street* or *insidecity* than *coast* or *bedroom*, the probabilities of these objects in an image are still important in this case.

From the above analysis, we can see that the most important information for image scene classification is the possibility of objects contained in an image. So, strengthening this information and weakening that of less important can be an effective strategy for OB dimensionality reduction. Under this observation, in order to reduce the dimensionality of OB, we propose two methods, namely, threshold value filter pooling and Matthew Effect normalization, in our new image scene classification framework. The former can avoid noise accumulation histogram-pooling and represent more useful information than max-pooling. The latter is a power normalization, which makes large feature components that have more useful information larger and smaller ones smaller. We will show that our framework can reduce dimensionality of OB to 1/252 (or 1/12 when keeping the spatial position).

The rest of the paper is organized as follows. We describe the existing Object Bank and its dimensionality reduction methods in Section 2. In Section 3, we introduce our methods and the implementation of our framework, which is followed by the datasets, process and results of our experiments in Section 4. In Section 5, we analyze factors of our methods. Finally we conclude the paper in Section 6.

## 2. Object filters and Object Bank descriptors

### 2.1. Object filters and responses

Object Bank is an object feature based on object filter response of Deformable Parts Model (DPM) (Felzenszwalb, McAllester, & Ramanan, 2008). It is necessary to introduce object filters and responses here before moving onto OB and our methods. Deformable parts consider models defined by a coarse root filter that


**Fig. 2.** Sample images. The existence of "car" object can distinguish *highway, street* and *insidecity* from *coast* and other indoor scenes. However, *highway, street* and *insidecity* cannot be distinguished only by the "car" objects, or by the number, positions and sizes of "car" objects.

covers the entire object, and higher resolution part filters that cover smaller parts of object.

Object filter is the feature vector of Histogram of Oriented Gradient (HOG) with size $w$ by $h$, defined by taking dot product of the weight vector and the feature in a $w \times h$ subwindow of a HOG pyramid. Let $H$ be a HOG pyramid and $p = (x, y, l)$ be the position $(x, y)$ at level $l$ of a filter, and $\phi(H, p, w, h)$ be the vector obtained by concatenating the HOG features in the $w \times h$ subwindow of $H$ with top-left corner at $p$. Then the response of a filter $F$ on this detection window can be written as $F \cdot \phi(H, p, w, h)$, which can also be written as $F \cdot \phi(H, p)$ when the dimensions are clear.

DPM for an object with $n$ parts is formally defined by a root filter $F_0$ and a set of part models $(P_1, \cdots, P_n)$, where $P_i = (F_i, v_i, s_i, a_i, b_i)$. Here $F_i$ is a filter for the $i$th part, $v_i$ is a two-dimensional vector specifying the center for a box of possible positions for part $i$ relative to the root position, $s_i$ gives the size of this box, while $a_i$ and $b_i$ are two dimensional vectors specifying coefficients of a quadratic function measuring a score for each possible placement of the $i$th part. $p_i = (x_i, y_i, l_i)$ indicates the position $(x_i, y_i)$ at level $l_i$ of the $i$th filter, then $F_i$ denote a root filter when $i = 0$, and $i$th part filter when $i > 0$. Response of object filters can be written as

$$R_{score} = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) + \sum_{i=1}^{n} [a_i \cdot (\tilde{x}_i, \tilde{y}_i) + b_i \cdot (\tilde{x}_i^2, \tilde{y}_i^2)] \quad (1)$$

where $(\tilde{x}_i, \tilde{y}_i) = ((x_i, y_i) - 2(x, y) + v_i)/s_i$ gives the location of the $i$th part relative to the root location. Both $\tilde{x}_i$ and $\tilde{y}_i$ should be between $-1$ and 1. The response also can be expressed in terms of dot product $\beta \cdot \psi(H, z)$ to learn object filters by latent SVM (Andrews, Tsochantaridis, & Hofmann, 2002), where $\beta$ and $\psi$ are shown in (2) and (3) respectively. The learning method of filters is beyond our concern in this paper.

$$\beta = (F_0, \cdots, F_n, a_1, b_1, \cdots, a_n, b_n) \quad (2)$$

$$\psi(H, z) = (\phi(H, p_0), \phi(H, p_1), \cdots, \phi(H, p_n), \tilde{x}_1, \tilde{y}_1, \tilde{x}_1^2, \tilde{y}_1^2, \cdots, \tilde{x}_n, \tilde{y}_n, \tilde{x}_n^2, \tilde{y}_n^2) \quad (3)$$

```
// Initialization
do train object filters
// Generate descriptors
for each image
  generate HOG features on 12 scales
  for each object
    for each grid
      for each level
        for each components
//Compute responses
          do R_score = Σ_{i=0}^{n} F_i · φ(H, p_i) + Σ_{i=1}^{n} [a_i · (x̃_i, ỹ_i) + b_i · (x̃_i^2, ỹ_i^2)]
//Pool responses to a descriptor
          do D_ob = max R_i
          end
//Pool descriptor of components to one
        D_level = max D_ob
        end
    end
  end
end
// OB representation is descriptors on each grid of each level of each object.
```

**Fig. 3.** Algorithm of dimensionality reduced OB.

### 2.2. Generating descriptors from responses

Object Bank (OB) descriptors are computed by using the response of "generalized object convolution". The responses are obtained by extracting the response value that is generated by applying a set of pre-trained object detectors at multiple scales. Then, spatial pyramid grids are generated according to Lazebnik et al. (2006), and responses at each scale are divided into these grids. Within each grid, pooling methods are employed to compute the object attribution of the grid.

OB employs a DPM proposed in Felzenszwalb, Girshick, McAllester, and Ramanan (2010), which builds two components of each filter for representing different viewpoints of each object at each level. So OB has 12 scales for each object, representing 6 levels and 2 components at each level. The process of generating OB is shown in Fig. 3. Firstly, HOG features are computed on each of the 12 scales, and then object convolution of each grid

in each scale for each object is generated, which is object filter's responses on all locations of the grids. Finally, pooling method is employed to merge responses at the same grid into a descriptor. In the improved OB, one of the three pooling methods can be adopted, which are max-pooling $D_{ob} = maxR_i$, average-pooling $D_{ob} = \frac{1}{N_R}\sum R_i$, and histogram-pooling $D_{ob} = \sum R_i$, where $R_i$ is the response of object filter and $N_R$ is the number of responses in each grid.

Let *No.Objects* be the number of object filters and *No.Grids* be the total number of grids of an image ($No.Grids = No.Scales \times No.Gridsperscale$ i.e. $12 \times (1^2 + 2^2 + 4^2) = 252$). An OB representation is constructed by concatenating responses of object filters across all grids. The overall dimensionality of OB representation is $N = No.Objects \times No.Grids$, which can easily grow into tens of thousands as the number of object filters increases.

### 2.3. Existing dimensionality reduction methods

When OB descriptor is employed as image feature, linear classifiers can be used to classify image scenes, as shown in Li et al. (2012; 2014). While linear classifiers have linear computational cost, dimensionality reduction is still an important task for further improving OB descriptor, because its very high dimensional feature causes a large amount of computation. For instance, in Li et al. (2010, 2012; 2014), 177 object filters are employed and a 44604-dimension feature is generated. Fortunately, OB is a robust representation with high dimensionality, and can be effectively compressed to a lower dimensionality representation (Li et al., 2014). A number of OB dimensionality reduction methods have been proposed, including PCA-OB, which reduces dimensionality of OB by employing PCA, and Viewpoints-pooling OB, which combines components belonging to different viewpoints.

As a well-known projection method, PCA is directly used for OB dimensionality reduction in PCA-OB (Li et al., 2014), which reduces the training time from 35 s to lower than 1 s for UIUC-Sports dataset, as well as the testing time from 5 ms to 0.01 ms per image. Although PCA-OB has a much higher accuracy than low-level features with the same or much higher dimensionality (Li et al., 2014), its performance obviously gets worse than original OB. For instance, on UIUC-Sports dataset, the accuracy decreased from 82.3% for the original OB to 75.4% for PCA-OB.

Another effective way to reduce dimensionality of OB is to merge components belonging to different viewpoints of same object at each level. As discussed in Section 2.2, OB has 2 components representing different viewpoints on each object. The Viewpoints-pooling OB adopts pooling to merge components after "pool responses to a descriptor" step (Fig. 3) of the original OB, and can reduce dimensionality to 1/2 of the original feature by merging 2 viewpoints. It can yield similar accuracy to the original feature. However, as the original dimensionality is too high, 1/2 dimensionality reduction is still insufficient for effectively decreasing the computational amount of classification.

Our framework aims to compress the descriptor for preserving useful information, rather than the existing dimensionality reduction methods (Belkin & Niyogi, 2001; Roweis & Saul, 2000; Tenenbaum et al., 2000) that aim to condense the feature. Because of this, our proposed framework can increase accuracies of dimensionality reduction methods when the dimensionality of descriptor is reduced to the same size. Different from the dimensionality reduction method in Li et al. (2014) and Li, Zhu, Su, Xing, and Fei-Fei (2013), our framework can merge scale and spatial information, and thus can reduce dimensionality to a much lower level.

## 3. Our pooled dimensionality reduction framework

### 3.1. Dimensionality reduction by merging levels and scales

OB representation has a pyramid structure, where a higher level represents more generalized information and a lower represents more detailed information. In the original OB, different responses in each component at the 1st level are merged by pooling method, and the descriptors are built at the 2nd level with a dimensionality of $No.Objects \times No.Scales \times No.Gridsperscale$. Viewpoints-pooling OB further merges descriptors on different viewpoints at each level by pooling method, and builds the descriptors at 3rd level with a dimensionality reduced to 1/2. The success of viewpoints-pooling OB suggests that it is reasonable to further pool descriptors and build an OB representation at the 4th or 5th level to obtain further reduced dimensionality. For instance, under the setting of the original OB, it can reduce the dimensionality to 1/12 when building the representation at the 4th level or to 1/252 when at the 5th one.

Building representation at a higher level also accords with the practice. Given an image, we can get the feature containing information on spatial, scale and potential *No.Objects*. As discussed in Section 1, the most beneficial information for classification is the probability that an image contains important objects. So a higher level representation has less useless information and more useful information. Based on this observation, we proposed a dimensionality reduction framework that could build representation at the 5th level and further simplify OB representation by merging descriptors at scales and levels through histogram-pooling. However, the resulting descriptor through direct pooling led to a reduced accuracy of 2/3 of the original OB in our test, so we propose two methods to improve its performance (Fig. 4).

### 3.2. Our methods

OB descriptor is a set of max responses of object detectors at each grid. That an object has little max response means that the grid has little probability of containing the object. These little responses usually have less useful information but the number of them is large. For example, we generated the OB descriptor of object "sailboat" and "bear" for an *sailing* category image as shown in Fig. 5. The object "sailboat" has a lot of large max responses as shown in Fig. 5(a) and the object "bear" has a lot of small max responses as shown in Fig. 5(b). When OB descriptor is directly adopted as feature, the objects whose descriptors have small responses will not obviously influence the classification, and thus can be seen as inessential objects.

#### 3.2.1. Threshold value filter pooling

We focus on dimensionality reduction by employing pooling method to merge the scales and grids of OB. One choice of pooling is histogram-pooling that accumulates all responses to represent pooled area. However, when histogram-pooling is employed to reduce the dimensionality of OB, these inessential objects will have fatal influence because the accumulation will combine small descriptors into large ones. The merged feature, however, may lose such important information that the image has a large probability of containing a key object, because we can not distinguish large descriptors generated before from those obtained by merging. Taking the image of Fig. 5 as an example, although there is no "bear" or something like a "bear" in this image, if we adopt histogram-pooling method to merge the scales of each grid (each column of Fig. 5(a) and (b)), it will cause the confusion that some responses of "bear" are larger than "sailboat".

Another choice is max-pooling method, which uses the maximum response to represent the pooled area. This method discards

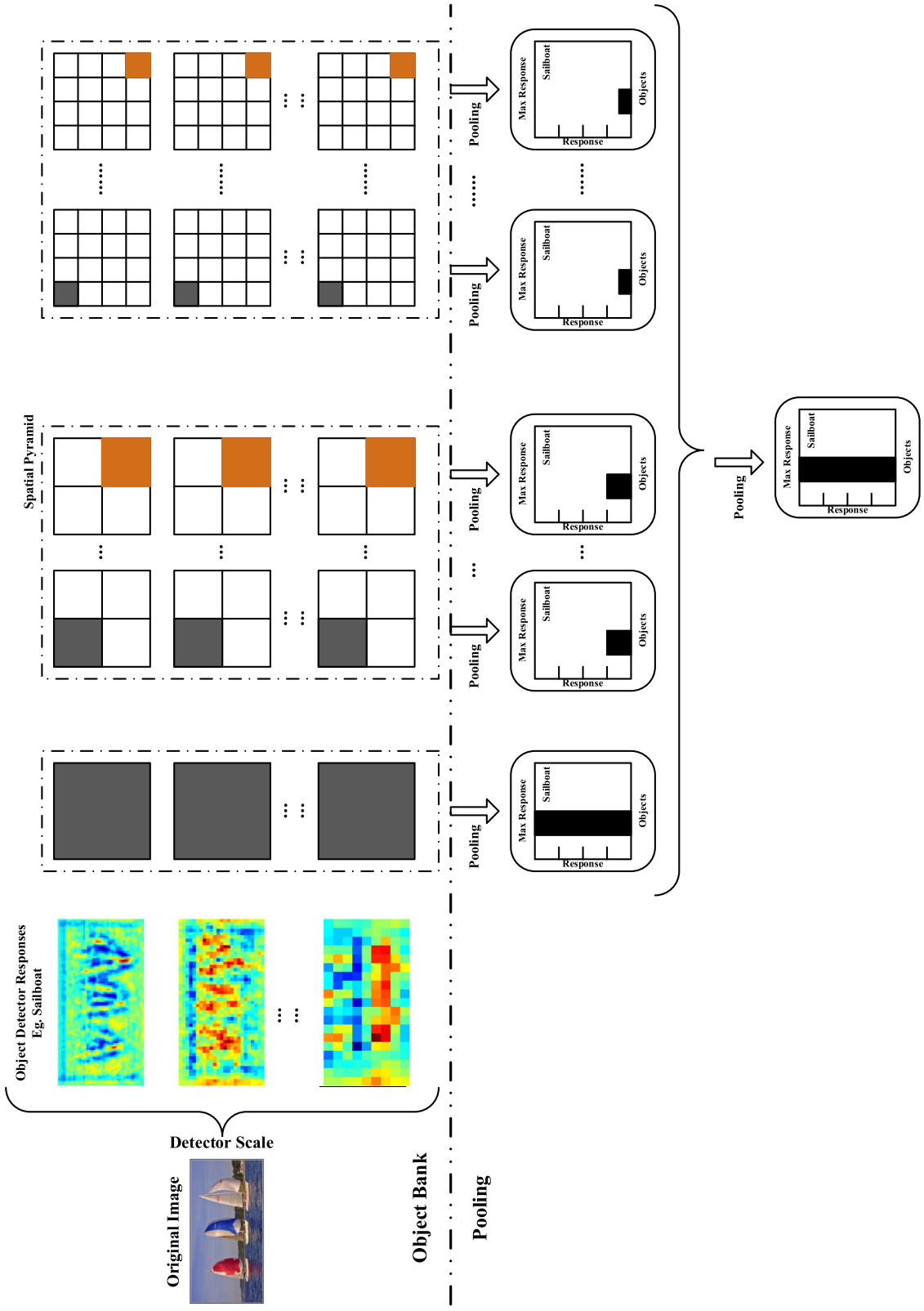


**Fig. 4.** The pooling reduction process of OB descriptor.

all information in the responses except for the largest one, thus avoiding inessential objects disturbance. However, some information from large responses, but not the largest, is also important. For example, as shown in Fig. 5(a), object "sailboat" has some large (but not the largest) responses in some area, these responses may also have important information, and discarding these large response can lead to a decrease of classification accuracy.

In OB merging, the histogram-pooling method can represent information of large response but it suffers from the problem of increasing the responses of low-responses-detectors. The

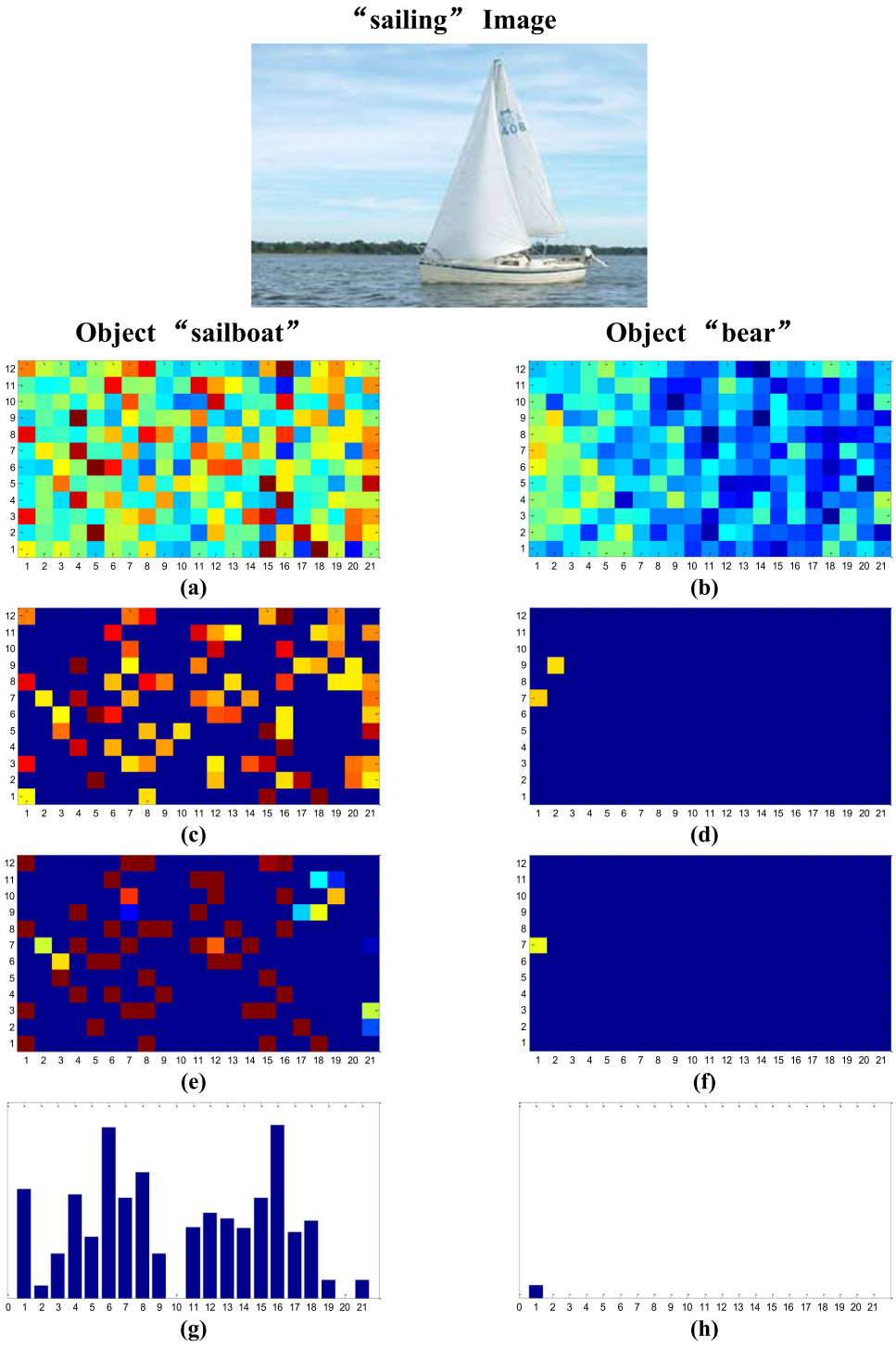


**Fig. 5.** An example to illustrate the role of our proposed methods. Hot color means large response, and cool color means small response. (a) and (b) are the OB descriptor of image for the object "sailboat" and "bear" respectively, where the rows represent the 12 detection scales and the columns represent the 3 spatial pyramid levels ($L = 0, 1, 2$). For example, the color of 2nd row 1st column represents the size of OB descriptors that corresponds to 2nd scale 1st area. Original OB is the vector of these descriptors. This representing is also adopted in (c)–(f). (c) and (d) are descriptors processed by our threshold value filter pooling. (e) and (f) are descriptors processed by our Matthew Effect normalization.

max-pooling method can avoid the problem of increasing low responses, but it discards much useful information. Thus we propose a threshold value filter pooling method that can find a compromise between histogram-pooling and max-pooling. The threshold value filter pooling can be defined as:

$$\gamma_i = \begin{cases} 1 & x_i \geq g_i \\ 0 & \text{else} \end{cases} \tag{4}$$

$$y = \sum_{i=1}^{n} \gamma_i x_i \tag{5}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]$ is $n$ descriptors in the pooled area, $y$ is output of the pooling, $\mathbf{g} = [g_1, g_2, \cdots, g_n]$ is given threshold value of $n$ descriptors. If threshold value $\mathbf{g}$ is small such that $\forall i$, $x_i > g_i$, the method will degenerate into histogram-pooling; else if $\mathbf{g}$ is

large such that $|\gamma| = 1$, the method will degenerate into max-pooling. We call the threshold $\mathbf{g}$ in the former case as $g_{histogram}$ and in the latter case as $g_{\max}$. Threshold $\mathbf{g}$ of our pooling is usually given a value between $g_{histogram}$ and $g_{\max}$, and thus histogram-pooling and max-pooling are two special cases of our method.

In fact, if we simply set the threshold value as middle value of descriptor, the advantage that small descriptors are removed will be obvious, as shown in Fig. 5(c) and (d). A more effective setting of threshold value will be introduced in Section 3.3, and the relationship between classification performance and threshold value will be introduced in Section 5.1.

#### 3.2.2. Matthew effect normalization

Power normalization (Perronnin, Sánchez, & Mensink, 2010) is one of the most popular normalization method for image descriptors. The mathematical expression of power normalization can be written as follows:

$$f(x_i) = sign(x_i)|x_i|^{\beta} \tag{6}$$

where the parameter $\beta$ is limited to [0, 1]. The normalization makes the large feature components smaller and the small ones larger and thus improves the classification performance of Fisher kernel (Perronnin & Dance, 2007; Sánchez et al., 2013).

Power normalization is motivated by an empirical observation on Fisher kernel based image classification (Perronnin et al., 2010): the classification accuracy of some cases may decrease because Fisher kernel descriptors are too sparse, which means that fewer descriptors are assigned with a significant value.

Most of existing classifiers such as SVM and LR have a common characteristic, i.e., larger components of feature are more decisive in classification. The characteristic usually becomes a disadvantage when involving sparse features, and can be overcome by normalization that adjusts values measured on different scales to a notional common scale. However, the empirical observation on OB is opposite: larger components of feature have more useful information, so the characteristic is an advantage for OB when descriptors are sparser. In order to enhance this advantage, we propose a Matthew Effect normalization, which generates results opposite to power normalization by setting parameter $\beta$ to a value larger than 1. So, the normalization is to make the large feature components larger, and make the small ones smaller.

As shown in Fig. 5(e) and (f), we adopt $\beta = 5$ to sparse the OB descriptor. The descriptors of interferential object "bear" have been observably reduced. At the same time, large descriptors are highlighted. Then we pool the descriptor according to its scale, as shown in Fig. 5(g) and (h). It is easy to find that at least one "sailboat" and no "bear" in the picture, so this picture can be obviously classified to *sailing* according to the pooled descriptor.

### 3.3. Implementation

In order to implement our framework, as shown in Fig. 6. Firstly, we sparsify OB through our threshold value filter and Matthew Effect normalization. Secondly, we use histogram-pooling method to merge the descriptor components of the same scale. Finally, the histogram-pooling method is utilized again to merge pooled descriptor components of same spatial position.

Before "generate the OB descriptor" step, a set of trained object detectors need to be selected and the responses of different regions in images need to be calculated. Li et al. have proposed an approach of training and selecting these object detectors (Li et al., 2010; Li et al., 2012). They randomly selected some objects from databases, and labeled images that contain every object. Then they utilized object detectors training method (Felzenszwalb et al., 2008; Felzenszwalb et al., 2010; Hoiem, Efros, & Hebert, 2005) to train each detectors. Finally, some valuable object detectors were

```
// Original OB as shown in figure3
  ... ...
// Our framework
  do normalization (7)
  do threshold value filtering (4)(5)
  do Matthew Effect normalization (6)
// Pooling
  do histogram-pooling D_grid = 1/No.level ∑ D_level
  end
// Pooling
do histogram-pooling D_object = 1/No.grid ∑ D_grid
end
```

**Fig. 6.** The algorithm of our framework.

selected from all trained detectors by cross-validation experiment. These selected object detectors consist of a detector group that are adopted for computing OB descriptor. In Li et al. (2010, 2012), 86 object detectors on LabelMe dataset and 177 object detectors on ImageNet dataset were obtained (Deng et al., 2009), finally 200 best object detectors were selected by cross-validation experiment. However, only 177 trained object detectors obtained on ImageNet dataset were published. On the one hand, the detector training is not the focus of this paper. On the other hand, manually labeled objects involve some subjectivity in detectors training. Besides, we hardly obtained the same detectors even though we repeated their job. Therefore, we employ the published 177 trained object detectors in our implementation, and remove cross-validation experiment step for selecting object detectors.

After the object detectors are obtained, we utilize the detectors by sliding grid in different scales to obtain the response of the whole image. Then we compute max response of each object on each local region. Our setting for scale and local region is same as Li et al. (2010, 2012), so detector responses will be obtained on 12 scales and 21 grids. Under this setting, the dimensionality of the OB descriptor that can be obtained is 44604.

Our framework, which contains our methods, is implemented on this OB descriptor. In the framework, the proposed methods are different from their individual forms formally but not substantially. Firstly, we normalize all of OB descriptors to (0, 1] by

$$\epsilon_i = \arctan(\theta_i) \cdot 2/\pi \tag{7}$$

where $\theta_i$ represents $i$th descriptor, $i \in [1, N]$. Secondly, given a threshold value vector $\mathbf{g}$, our threshold value filter method is performed to remove small response components of descriptors as follows,

$$t_i = \begin{cases} 1 & \epsilon_i \geq g_i \\ 0 & \text{else} \end{cases} \tag{8}$$

$$\mathbf{z} = \epsilon \cdot \mathbf{t}' \tag{9}$$

where $\mathbf{z}$ represents filtered descriptors. Then, Matthew Effect normalization is performed on filtered descriptors with $\beta > 1$ as follows,

$$\mathbf{D}_{level} = sign(\mathbf{z})|\mathbf{z}|^{\beta} \tag{10}$$

Finally, we adopt histogram-pooling twice to merge the scale and local region.

We define a default setting for threshold value and normalization. All experiments will adopt the following default setting, unless a special description is given otherwise:

Threshold value $g_i$ is set to the mean of $i$th OB descriptors component obtained in a subset of training set, $g_i = \frac{1}{M}\sum_{m=1}^{M} x_i^{(m)}$, where $M$ is the number of subset samples. We adopt $\beta = 5$ for

**Table 1**
Classification accuracy compared with baseline on UIUC-Sports dataset.

| Algorithms | Dimensionality | Avg.accuracy | Classifier |
|---|---|---|---|
| Org OB (Li et al., 2010) | 50400 | **76.3%** | **SVM-Linear** |
| OB-177 | 44604 | 72.9% | SVM-Linear |
| | | 72.7% | L2 regularized L2 loss SVM |
| | | 73.3% | L2 regularized L1 loss SVM |
| | | 73.7% | L1 regularized L2 loss SVM |
| | | **75.8%** | **L1 regularized LR** |
| | | 74.4% | L2 regularized LR |
| OB-177 PCA | 177 | **75.3%** | **L2 regularized L2 loss SVM** |
| OB-177 LLE | | 73.6% | L2 regularized L2 loss SVM |
| OB-177 Laplacian | | 71.6% | L1 regularized L2 loss SVM |
| OB-177 Isomap | | 65.1% | SVM-Linear |
| Our Framework | 177 | 75.2% | SVM-Linear |
| | | **77.1%** | **L2 regularized L2 loss SVM** |

Matthew Effect normalization to sparse the OB descriptor. The histogram-pooling can be represented as $y = \sum_i x_i$. In the first pooling, **x** is OB descriptor that processed by our methods, and $i$ is 12 scales. In the second pooling, **x** is the pooled results of the first pooling, and $i$ is the 21 local regions.

## 4. Experiments and results

We evaluated our framework on three diverse datasets: UIUC-Sports (Li & Fei-Fei, 2007), LabelMe (Oliva & Torralba, 2001), and 15-Scenes categories (Lazebnik et al., 2006). We compared our framework with the original OB descriptor (Li et al., 2010) and some of its deformation because they are the most relevant to our work. All experiments were repeated ten times with different randomly selected training and testing images. The final results were obtained as average classification accuracy.

Multi-class classification is done with a simple linear SVM or LR classifier, which is trained under the one-versus-all rule: a classifier is learned to separate each class from the rest, and a test image is assigned to the label of the classifier with the highest response.

More specifically, the three methods and their settings that were used in comparison with our framework are listed as follows:

- Original OB (Org OB) (Li et al., 2010): It adopts 177 object detectors obtained on ImageNet dataset (this part is same to ours), 86 object detectors obtained on LabelMe dataset, and 200 best object detectors selected by cross-validation experiment.
- OB with 177 object detectors (OB-177): It adopts 177 published object detectors obtained on ImageNet dataset. We used the same 177 detectors in our framework so this is the most important baseline.
- Object Bank2014 (state-of-the-art) (Li et al., 2014)
  - OB 2014 with 177 improved object detectors (OB2014-177): It adopts 177 object detectors obtained on ImageNet dataset, but these improved detectors have better performance than detectors in OB-177.
  - OB with 200 improved object detectors (OB2014): It adopts 1000 pre-selected object, and then selects the 200 best by cross validation experiment.

Existing OB dimensionality reduction method:

- Dimensionality reduction by using PCA (PCA-OB): It adopts the same detectors as OB2014-177 and uses PCA method to reduce dimensionality to below 150.
- Dimensionality reduction by combining different views (V-Pooling OB): It adopts the same detectors as OB2014-177 and incorporates multiple views of each object by using max-pooling and average-pooling method. This method can be seen as the image representation on 3rd level of OB structure.

Dimensionality of OB-177 descriptor is reduced from 44604 to 177 by feature dimensionality reduction methods. The only difference of this method from the proposed method is the dimensionality reduction algorithm:

- Dimensionality reduction by using PCA (PCA-OB-177): It uses a linear dimensionality reduction method to reduce dimensionality of OB-177.
- Dimensionality reduction by using LLE (LLE-OB-177):It uses LLE (Roweis & Saul, 2000) method to reduce dimensionality of OB-177.
- Dimensionality reduction by using Laplacian (Laplacian-OB-177):It uses Laplacian (Belkin & Niyogi, 2001) method to reduce dimensionality of OB-177.
- Dimensionality reduction by using Isomap (Isomap-OB-177):It uses Isomap (Tenenbaum et al., 2000) method to reduce dimensionality of OB-177.

### 4.1. UIUC-Sports Dataset

This dataset is composed of eight complex event classes provided by Li and Fei-Fei (2007). It contains 1579 color images with different sizes. There are 194 images in *rockclimbing*, 200 in *badminton*, 137 in *bocce*, 236 in *croquet*, 182 in *polo*, 250 in *rowing*, 190 in *sailing* and 190 in *snowboarding*. We followed the experimental setting in Li et al. (2010) by using 70 randomly drawn images from each class for training and 60 for testing. We obtained the best parameters through cross-validation for each classifier, and then adopted these best parameters for experiments.

**Table 2**
Classification accuracy compared with state-of-the-art OB on UIUC-Sports dataset.

| Algorithms | Dimensionality | Avg.accuracy | Classifier |
|---|---|---|---|
| OB2014 (Li et al., 2014) | 50400 | **82.3%** | **L2 regularized L2 loss SVM** |
| OB2014-177 | 44604 | **82%** | **LR** |
| PCA-OB | Below 150 | **75.4%** | **LR** |
| V-Pooling OB Avg | 25200 | **82.5%** | LR |
| V-Pooling OB Max | | 70.5% | |
| Our Framework | 177 | **77.1%** | **L2 regularized L2 loss SVM** |

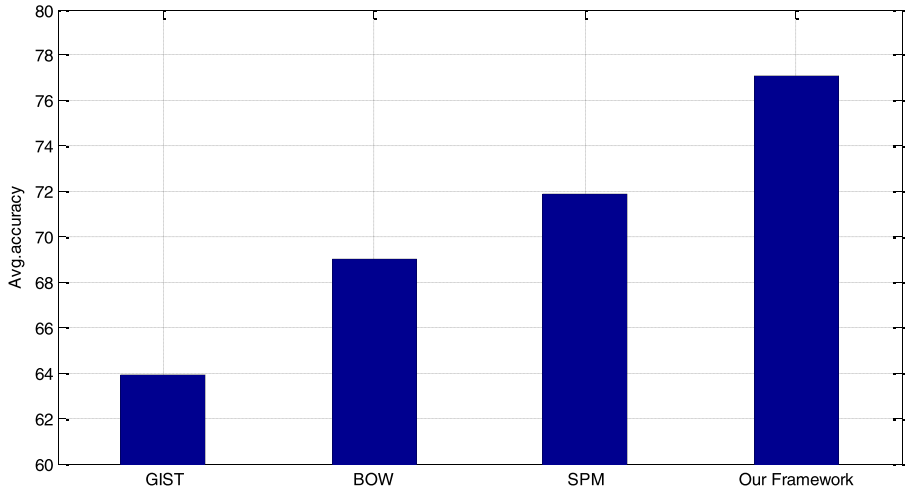


**Fig. 7.** Comparing our framework with popular methods based on low-level features.

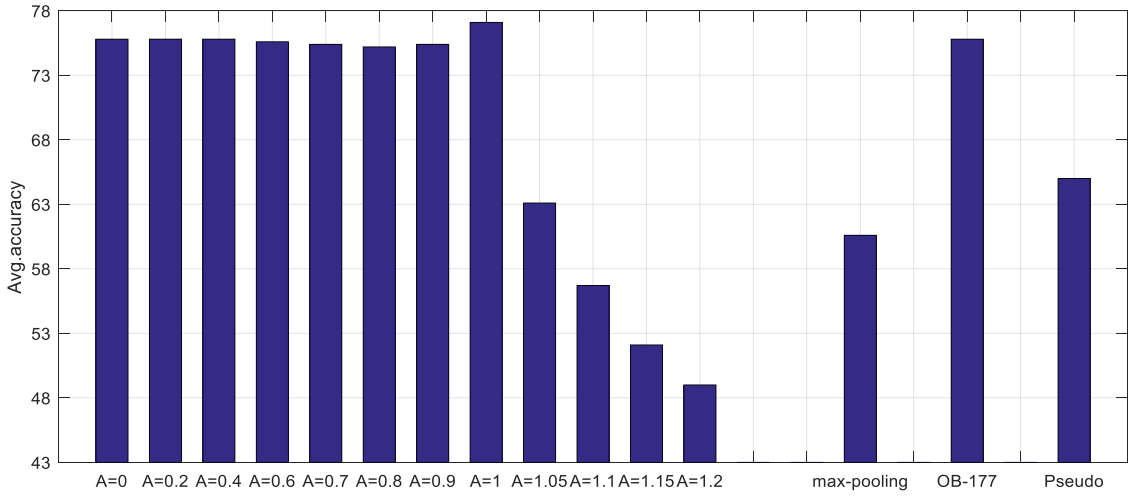


**Fig. 8.** The effect of our threshold value filter method. Larger *A* means that we can get sparser descriptor. OB-177 is the basic OB descriptor before pooling, and Pseudo is pooled OB descriptor without utilizing our methods.

Table 1 shows the results of our framework and baseline. While our framework achieves only a little superiority in classification performance, it reduces the dimensionality of feature from 44604 to 177, and thus can reduce the computational complexity significantly. If just comparing the accuracies, our proposed framework increases the accuracy about 1.8%–7%, compared with dimensionality reduction methods that reduce the dimensionality to the same size.

Table 2 shows the comparison between our framework and the latest Object Bank. We should point out that the latest Object Bank used a stronger condition (optimized object detector), so its basic accuracy is higher than that of ours. For example, descriptor generated from detector reported in Li et al. (2014) (OB2014-177) outperforms that in Li et al. (2010) (Org OB), even though these descriptors are generated by the same method.

Although we employ less object detectors in our implementation, our framework outperforms some dimensionality reduction methods based on OB2014 both in accuracy and computational complexity. If stronger object detectors are used, performance of our framework would be further improved in theory. (The latest object detectors have been not published yet, and the object detector technology is beyond the scope of this article, so we have not tested the framework on detectors of OB2014.)

We also compared our framework with popular methods based on low-level features, such as GIST (Oliva & Torralba, 2001), BOW (Csurka, Dance, Fan, Willamowski, & Bray, 2004), and SPM (Lazebnik et al., 2006), the results are shown in Fig. 7. Our method outperforms these methods more than 5% in classification accuracy. Some new methods (that learn high-level representation from low-level features) can obtain higher accuracy than the framework. For example, Gao, Chia, and Tsang (2011), Dixit, Rasiwasia, and Vasconcelos (2011), Gao, Tsang, and Chia (2010) and Bo, Ren, and Fox (2011) achieved classification accuracy of 79.37, 84.4, 84.92 and 85.7% respectively on the UIUC-Sports dataset. However, if we train the object detectors on the images of the training set, instead of datasets completely independent of the classification, the accuracy of the framework will increase to a similar level. Similar to Li et al. (2014), when training set is adopted to train detectors (this is named customized OB in their report), the accuracy increases from 75.8% to 84.54%.

### 4.2. LabelMe dataset

This is a dataset of eight natural scene classes provided by Oliva and Torralba (2001). It contains 2688 color images of the same size of $256 \times 256$. There are 360 images in *coast*, 328 in *forest*, 260 in *highway*, 308 in *insidecity*, 374 in *mountain*, 410 in *opencountry*, 292 in *street* and 356 in *tallbuilding*. 200 images were randomly drawn from each class scene, one half of which for training and the other for testing.

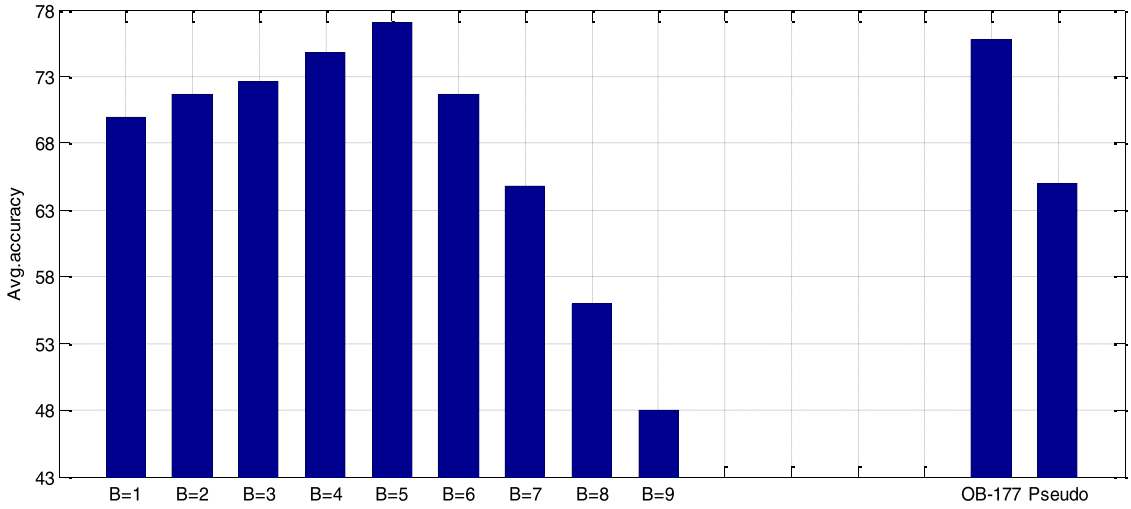


**Fig. 9.** The effect of our Matthew Effect method. Larger $B$ means more obvious Matthew Effect. OB-177 is the basic OB descriptor before pooling, and Pseudo is histogram pooled OB descriptor without utilizing our methods.

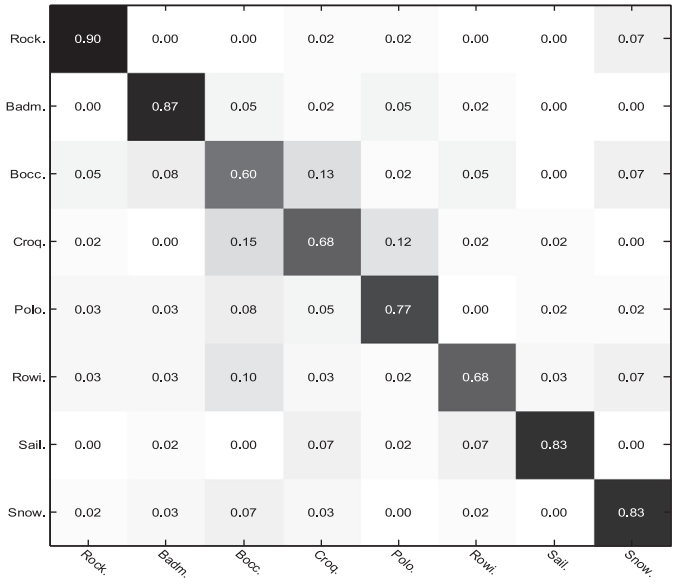


**Fig. 10.** Confusion table on UIUC-Sports dataset, average accuracy: 77.1%.

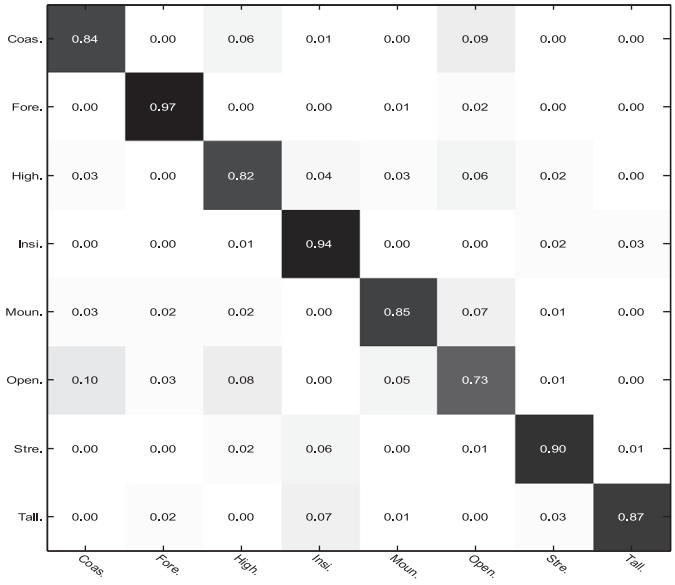


**Fig. 11.** Confusion table on LabelMe dataset, average accuracy: 86.5%.

Our framework achieved a similar accuracy level as the baseline. Moreover, our framework can significantly reduce the computational complexity because the feature dimensionality is much lower. What is more, the framework can increase the accuracy of dimensionally reduced feature from 78.7%–84.7% to 86.5% under the same size of dimensionality, which demonstrates the advantage of the proposed methods implemented in it (Table 3).

### 4.3. 15-Scenes dataset

The 15-Scenes dataset contains 15 scene classes and 4485 images provided by several researchers (Fei-Fei & Perona, 2005; Lazebnik et al., 2006; Oliva & Torralba, 2001), 8 of classes of which are the same as LabelMe dataset and the rest 7 classes consist of 216 images in *bedroom*, 241 in *suburb*, 210 in *kitchen*, 289 in *livingroom*, 215 in *office*, 315 in *store* and 311 in *industrial*. Each class has 200–400 images, and their average size is $300 \times 250$ pixels. Following the same experimental setting in Li et al. (2010), we used 100 images in each class for training and the rest for testing.

In this dataset, in view of accuracy, although the classification accuracy is also reduced, our framework outperforms the methods whose size of dimensionality are same to us (Table 4).

### 4.4. Analysis

On nature scene dataset, namely LabelMe, and event dataset, namely UIUC-sports, the descriptor whose dimensionality is reduced by our framework obtains higher or similar accuracy than that before dimensionality reduction, even if the dimensionality is reduced from 44604 to 177. These results suggest that our framework is capable of simplifying OB representation and yielding more proper descriptors to achieve both dimensionality reduction and classification accuracy increase. However, when tested on mixture of indoor and outdoor image scene dataset, namely 15-Scenes Dataset, the accuracy of our framework is a little lower than those that have large size of dimensionality. This is probably because we removed the spatial information, which is very important in the indoor scene images. If we preserve spatial information, our


**Fig. 12.** Samples of UIUC-Sports.

**Table 3**
Classification accuracy comparison on LabelMe dataset.

| Algorithms | Dimensionality | Avg.accuracy | Classifier |
|---|---|---|---|
| Org OB (Li et al., 2010) | 50400 | – | – |
| OB-177 | 44604 | 85.4% | SVM-Linear |
| | | 86.9% | L2 regularized L2 loss SVM |
| | | 87% | L2 regularized L1 loss SVM |
| | | 85.1% | L1 regularized L2 loss SVM |
| | | 87.1% | L1 regularized LR |
| | | **87.5%** | **L2 regularized LR** |
| OB-177 PCA | 177 | **84.7%** | **L2 regularized LR** |
| OB-177 LLE | | 84.6% | L2 regularized LR |
| OB-177 Laplacian | | 84.4% | L2 regularized LR |
| OB-177 Isomap | | 78.7% | L2 regularized L2 loss SVM |
| Our Framework | 177 | 83.8% | SVM-Linear |
| | | **86.5%** | **L2 regularized LR** |

**Table 4**
Classification accuracy comparison on 15-Scenes dataset.

| Algorithms | dimensionality | Avg.accuracy | Classifier |
|---|---|---|---|
| Org OB (Li et al., 2010) | 50400 | **80.9%** | **SVM-Linear** |
| SPM (Lazebnik et al., 2006) | 8400 | **81.4%** | **KSVM** |
| ScSPM (Yang et al., 2009) | 1681 | **80.3%** | **SVM-Linear** |
| GIST (Oliva & Torralba, 2001) | – | **73.1%** | **SVM-Linear** |
| BoW (Csurka et al., 2004) | – | **72.2%** | **SVM-Linear** |
| OB-177 | 44604 | **78.3%** | **SVM-Linear** |
| OB-177 PCA | 177 | **74.1%** | **SVM-Linear** |
| OB-177 LLE | | 71.3% | SVM-Linear |
| OB-177 Laplacian | | 72.5% | L2 regularized LR |
| OB-177 Isomap | | 67.8% | SVM-Linear |
| Our Framework | 177 | 74.9% | SVM-Linear |
| | 177 | 77% | L2 regularized L2 loss SVM |
| | 3717 | **81.5%** | **SVM-Linear** |

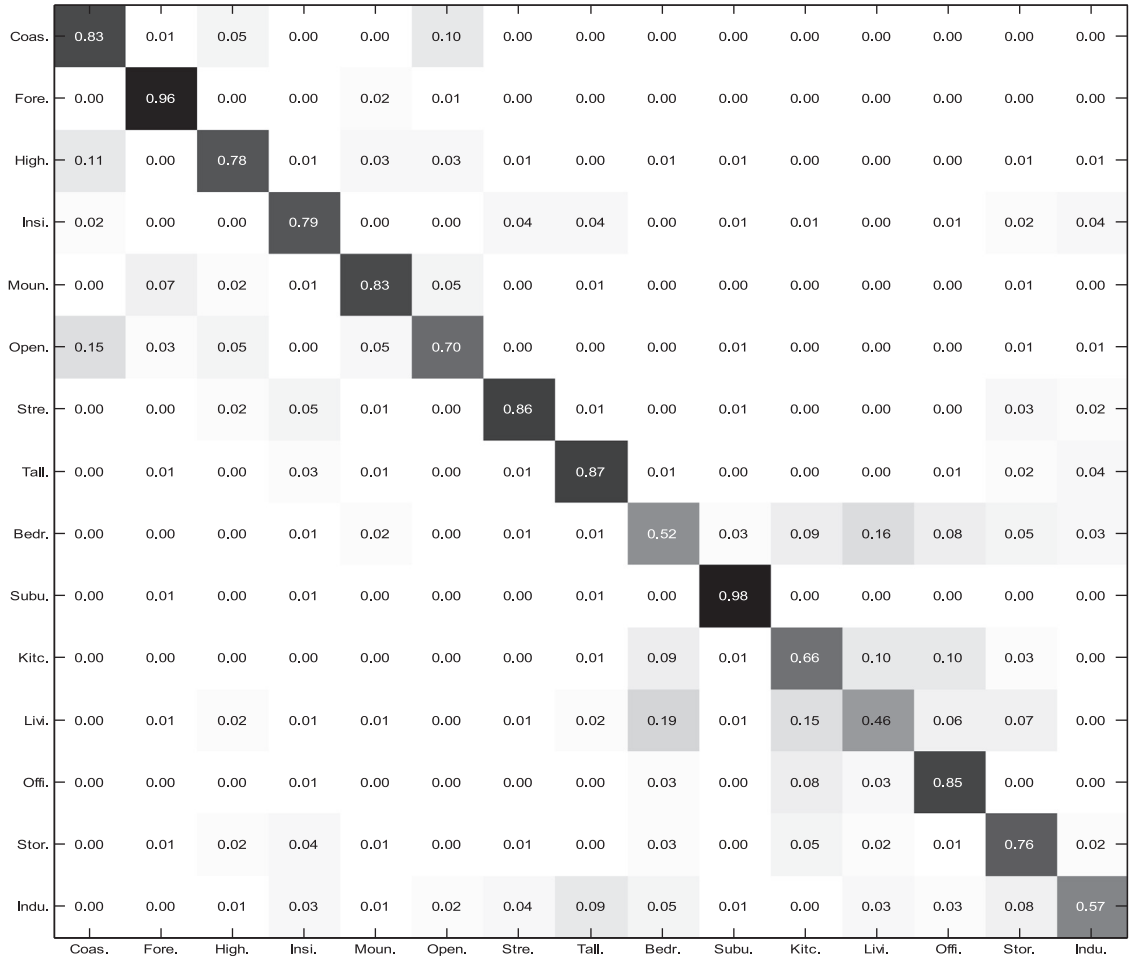


**Fig. 13.** Confusion table on 15-Scenes dataset, avg.accuracy: 77%.

framework can achieve the highest accuracy (81.5%) in this group of experiments. We will analyze this improvement in detail in the Section 5.3.

## 5. Further discussion

In our framework, pooling method has been used twice to reduce the dimensionality of OB descriptor, and two proposed methods have been implemented in it as well to improve the performance of pooled descriptor. In this section, we will further analyze the effectiveness of the proposed methods and discuss the computational complexity of our framework. We will also describe and analyze some disadvantages in classifying certain categories, and give an improved version of the framework for classifying these categories.

### 5.1. Role of our methods

We firstly discuss the effectiveness of the proposed methods. Just as discussed in Section 3.2, larger components of OB descriptor contain more useful information. The goal of our threshold value filter is to remove the smaller components, and thus makes the OB descriptor sparse. This method can avoid those small components being merged with larger ones in the pooling step. In our default setting, we used the mean of OB descriptor components as threshold value. Now we further analyze the effect of this method.

We multiply a coefficient $A$ on the threshold value, and rewrite it as $G_i = A \cdot \frac{1}{M}\sum_{m=1}^{M} x_i^{(m)}$. Apparently, larger $A$ means that we can get more sparse descriptor. When $A = 0$, our pooling method reduces to histogram-pooling; when $A = 1$, it is the same as adopting our default setting; and when $A$ is large enough that only one largest descriptor remains, our pooling method becomes max-pooling. We used the classification accuracy of L2-regularized L2-loss SVM that works on UIUC-Sports to evaluate the effect of the method.

The results are shown in Fig. 8. When our pooling reduces to histogram-pooling ($A = 0$), the accuracy of pooled OB is lower than OB without pooling (OB-177). Similarly, the accuracy of max-pooling is also lower than OB-177. With $A$ increasing from 0 to 1, the accuracy changes very small until $A = 1$, then overfitting appears when $A$ is larger than 1. Our threshold value filter method can achieve good performance when default setting is adopted, which means that the threshold value is equal to the mean of descriptor components on the training set. It suggests that the mean of detector responses has overwhelming influence, and those with responses larger than mean have much more useful information. So this method can improve not only the performance of OB detector, but also the other methods based on object detector technology.

The proposed Matthew Effect normalization has the similar effect of enhancing useful information. We adopt the concave function $f(x) = x^B$ to process the normalized descriptor, and thus large components that have more useful information are enhanced, and the smaller components that have less useful information are weakened. To analyze the effect of this method, we modified the index $B$ of concave function from 1 to 9, and adopted the

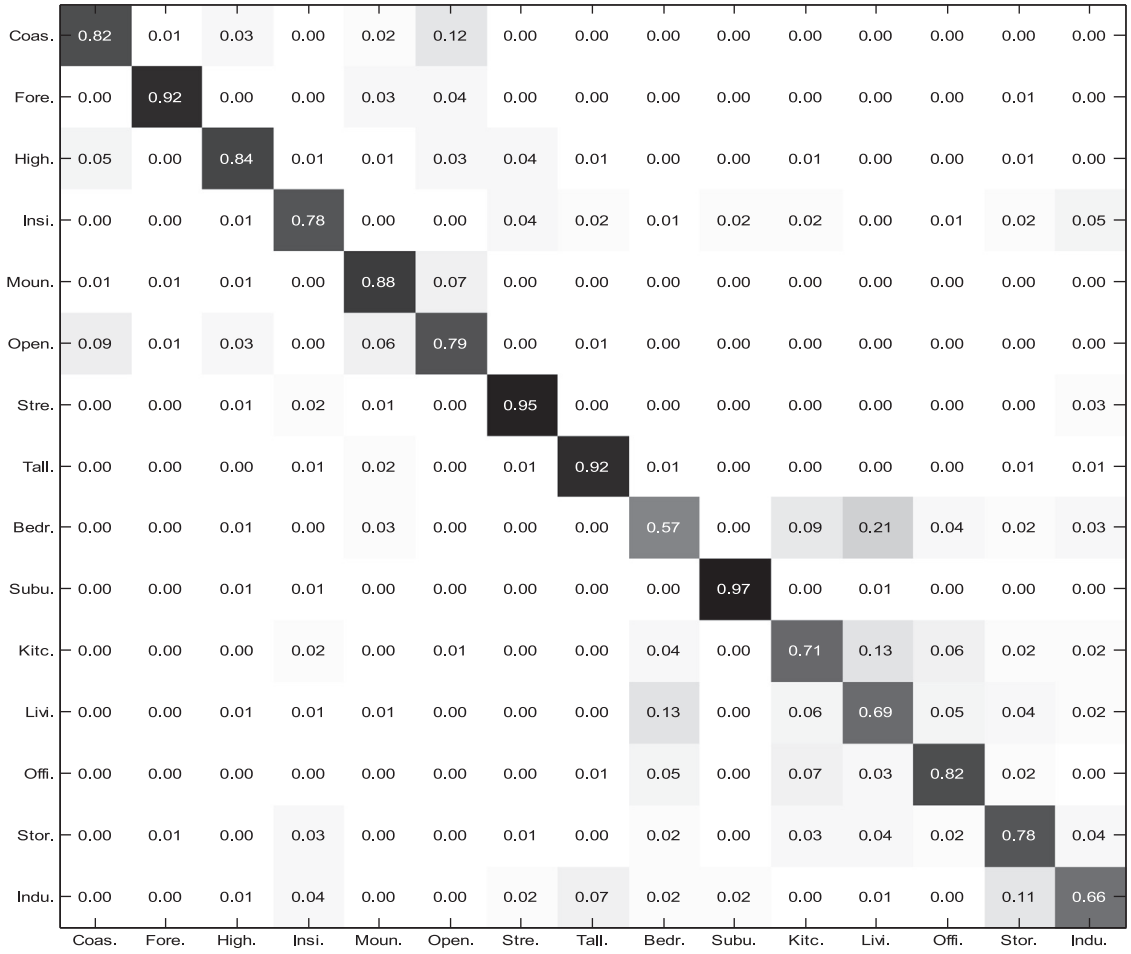


**Fig. 14.** Confusion table on 15-Scenes dataset, avg.accuracy: 81.5%.

classification accuracy of L2-regularized L2-loss SVM that works on UIUC-Sports to evaluate the effect. The Matthew effect is obliterated when $B = 1$, and a larger $B$ yields a stronger Matthew Effect.

The experiment results are shown in Fig. 9. When $B = 1$, i.e., no Matthew effect at all, the classification accuracy of pooled OB descriptor is lower than OB without pooling (OB-177), and it increases with the increase of $B$, until our default setting $B = 5$. At last, overfitting appears when $B > 5$.

We also tested the classification performance when both of our methods were obliterated (i.e., using directly histogram pooled OB descriptor, shown in Figs. 8 and 9 as Pseudo group). The accuracy of Pseudo group is not only lower than our method, but also lower than the baseline (OB-177). This suggests that our methods play the key role in improving the performance of pooled OB.

### 5.2. Computational complexity

Because the time complexity of the linear SVM and LR classifier is $O(n)$ for both training and testing, where $n$ is the number of samples, the overall computational complexity depends on the base computation, which is influenced by the feature dimensionality. Our pooled OB detector reduces the dimensionality by 1/252, so it can significantly reduce the computational complexity.

Our framework also brings in dimensionality reduction, including executing our methods and pooling, before classification. We should point out that our dimensionality reduction step has a very low computational complexity. The algorithm of our dimensionality reduction is shown in Fig. 6. The computation at the threshold value step is independent of the number $n$ of samples, and thus the pooling step has $O(n)$ time complexity. On the other hand, samples are loaded one by one from the dataset at the pooling step, so the space complexity is $O(1)$.

### 5.3. Spatial information

If object detectors are very exact such that they can identify every object, we can always find the key object to classify images. In this case, the classifier may have high accuracy without the scale and spatial information. Just as discussed in Li et al. (2014), if the object detectors are perfect, the classification accuracy will be 100% in theory. However, object detector technology is not perfect up to now, so we have to consider non-ideal situation.

In some sports or outdoor image classification tasks (eg. UIUC-Sports dataset and LabelMe dataset), most categories usually have key objects so that they are not obviously confused with other categories. However, the object detectors cannot accurately find every key object yet. Let's take the categories *bocce*, *croquest* and *polo* of UIUC-Sports dataset as examples. If the detectors can accurately identify "human" and different kinds of "ball", the accuracy of scene classification will be 100% in theory. However, the detectors can only identify if a "ball" or "human" is in the image, rather than detect the kind of "ball" or "human", so the obvious confusion between categories may appear, as shown in Fig. 10.

Figs. 10 and 11 show the confusion tables of the classification experiment results on UIUC-Sports and LabelMe dataset. In the confusion table, the rows represent the models for each scene category while the columns represent the ground truth categories of scenes.

In non-ideal situation, spatial information becomes important. For instance, in different categories of Fig. 12, "human" and "ball" are usually in different positions. Some non-key objects also become important factor, such as position and proportion of "sky" and "lawn". The most direct way to adapt this condition is to improve the performance of object detector technology, but this is not the focus of this paper.

Adding the spatial information in our feature is very easy. We just remove the step "**do** histogram-pooling $D_{object} = \frac{1}{No.grid}\sum D_{grid}$" in our algorithm (Fig. 6). The dimensionality of this feature is 21 times as our default implementation, in other words, 1/12 of original OB descriptor.

We adopted the same setting as in Section 4.3 to test the improved feature on 15-Scenes dataset. The result is shown in Figs. 13 and 14. Classification accuracy of the improved feature is about 4.5% higher than the default implementation, and the confusion between categories are lower. Compared with OB-177, the improved feature has advantage in both dimensionality reduction and accuracy increase; it can not only reduce the dimensionality to 1/12, but also increase the accuracy by more than 3%.

## 6. Conclusion

Object Bank has superior performance in scene classification, but its high dimensionality demands massive computation. The existing OB dimensionality reduction methods are incapable of simultaneously achieving both high classification accuracy and substantial dimensionality reduction.

We have proposed a framework that builds representation on important and useful information in images to simplify OB representation. In order to enhance useful information and remove useless information, and thus highlight important descriptors, we have proposed threshold value filter method and Matthew Effect normalization method, and then performed them in our framework. In the framework, OB descriptor is further pooled and thus its dimensionality is significantly reduced. We compared our framework with the exiting OB dimensionality reduction methods. The experimental results showed that the proposed feature is capable of both significant dimensionality reduction and accuracy increase.

Although our proposed framework has demonstrated accuracy increase and dimensionality reduction, some future research can be conducted to overcome its limitations and extend its applications. One limitation is that the parameters of the threshold value filter are empirically set. A study of the interaction between the parameters and recognition results may lead to new ways of boosting the classification performance of the proposed framework. Given that the threshold value filter pooling can avoid noise accumulation in histogram-pooling and represent more useful information than max-pooling, it is possible to extend it to feature dimensionality reduction tasks in other classification and expert systems. Moreover, being a opposite application of power normalization, the Matthew effect normalization is proposed mostly based on empirical observation. A systematical research on the empirical method may shed lights on how and when this opposite application works in general and for other systems. Finally, while our focus is on representing image scenes from object viewpoint, other information, for example, texture, color, background of images, are also important but have not been taken into account. A combination with such information may further help improve the classification performance.

## Acknowledgments

The authors would like to thank the authors in Li and Fei-Fei (2007), Oliva and Torralba (2001), Fei-Fei and Perona (2005) and Lazebnik et al. (2006) for making their experimental datasets available, and the authors in Li et al. (2010) and Felzenszwalb et al. (2008) for making their code available. This work was supported by the Doctoral Fund of Natural Science Foundation of Shandong Province under Grant no.ZR2016FB18, and Natural Science Foundation of China under Grant no.61702249.

## References

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support vector machines for multiple-instance learning. In *Advances in neural information processing systems* (pp. 561–568).

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips: 14* (pp. 585–591).

Bo, L., Ren, X., & Fox, D. (2011). Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in neural information processing systems* (pp. 2115–2123).

Bosch, A., Zisserman, A., & Muñoz, X. (2006). Scene classification via plsa. In *Computer vision–eccv 2006* (pp. 517–530). Springer.

Boureau, Y.-L., Le Roux, N., Bach, F., Ponce, J., & LeCun, Y. (2011). Ask the locals: multi-way local pooling for image recognition. In *Computer vision (iccv), 2011 ieee international conference on* (pp. 2651–2658). IEEE.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, eccv: 1* (pp. 1–2).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer vision and pattern recognition, 2005. cvpr 2005. ieee computer society conference on: 1* (pp. 886–893). IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 248–255). IEEE.

Dixit, M., Rasiwasia, N., & Vasconcelos, N. (2011). Adapted gaussian models for image classification. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 937–943). IEEE.

Fei-Fei, L., & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer vision and pattern recognition, 2005. cvpr 2005. ieee computer society conference on: 2* (pp. 524–531). IEEE.

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer vision and pattern recognition, 2008. cvpr 2008. ieee conference on* (pp. 1–8). IEEE.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32*(9), 1627–1645.

Freeman, W. T., & Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence, 13*(9), 891–906.

Gao, S., Chia, L.-T., & Tsang, I. W. (2011). Multi-layer group sparse coding í¬for concurrent image classification and annotation. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 2809–2816). IEEE.

Gao, S., Tsang, I. W.-H., & Chia, L.-T. (2010). Kernel sparse representation for image classification and face recognition. In *Computer vision–eccv 2010* (pp. 1–14). Springer.

Hoiem, D., Efros, A. A., & Hebert, M. (2005). Automatic photo pop-up. In *Acm transactions on graphics (tog): 24* (pp. 577–584). ACM.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 ieee computer society conference on: 2* (pp. 2169–2178). IEEE.

Li, L.-J., & Fei-Fei, L. (2007). What, where and who? classifying events by scene and object recognition. In *Computer vision, 2007. iccv 2007. ieee 11th international conference on* (pp. 1–8). IEEE.

Li, L.-J., Su, H., Fei-Fei, L., & Xing, E. P. (2010). Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in neural information processing systems* (pp. 1378–1386).

Li, L.-J., Su, H., Lim, Y., & Fei-Fei, L. (2012). Objects as attributes for scene classification. In *Trends and topics in computer vision* (pp. 57–69). Springer.

Li, L.-J., Su, H., Lim, Y., & Fei-Fei, L. (2014). Object bank: An object-level image representation for high-level visual recognition. *International Journal of Computer Vision, 107*(1), 20–39.

Li, L.-J., Zhu, J., Su, H., Xing, E. P., & Fei-Fei, L. (2013). Multi-level structured image coding on high-dimensional image representation. In *Computer vision–accv 2012* (pp. 147–161). Springer.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. the proceedings of the seventh ieee international conference on: 2* (pp. 1150–1157). Ieee.

Niu, Z., Hua, G., Gao, X., & Tian, Q. (2012). Context aware topic model for scene recognition. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (pp. 2743–2750). IEEE.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision, 42*(3), 145–175.

Pandey, M., & Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with deformable part-based models. In *Computer vision (iccv), 2011 ieee international conference on* (pp. 1307–1314). IEEE.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12*(7), 629–639.

Perronnin, F., & Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Computer vision and pattern recognition, 2007. cvpr'07. ieee conference on* (pp. 1–8). IEEE.

Perronnin, F., Sánchez, J., & Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Computer vision–eccv 2010* (pp. 143–156). Springer.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision, 105*(3), 222–245.

Shi, Y., Wan, Y., Wu, K., & Chen, X. (2017). Non-negativity and locality constrained laplacian sparse coding for image classification. *Expert Systems With Applications, 72*, 121–129.

Su, Y., & Jurie, F. (2012). Improving image classification using semantic attributes. *International Journal of Computer Vision, 100*(1), 59–77.

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science, 290*(5500), 2319–2323.

Wang, C., Blei, D., & Li, F.-F. (2009). Simultaneous image classification and annotation. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1903–1910). IEEE.

Wang, C., Yu, J., & Tao, D. (2013). High-level attributes modeling for indoor scenes classification. *Neurocomputing, 121*, 337–343.

Yang, J., Yu, K., Gong, Y., & Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1794–1801). IEEE.

Zang, M., Wen, D., Wang, K., Liu, T., & Song, W. (2015). A novel topic feature for image scene classification. *Neurocomputing, 148*, 467–476.

Zhou, L., Zhou, Z., & Hu, D. (2013). Scene classification using multi-resolution low-level feature combination. *Neurocomputing, 122*, 284–297.