

# Adaptive spatial pooling for image classification

Yinglu Liu<sup>a</sup>, Yan-Ming Zhang<sup>a</sup>, Xu-Yao Zhang<sup>a</sup>, Cheng-Lin Liu<sup>a,b,\*</sup>

<sup>a</sup> National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguang East Road, Beijing 100190, PR China

<sup>b</sup> CAS Center for Excellence in Brain Science and Intelligence Technology, 95 Zhongguang East Road, Beijing 100190, PR China

## ARTICLE INFO

### Article history:

Received 11 February 2015

Received in revised form

15 October 2015

Accepted 27 January 2016

### Keywords:

Weighted pooling

Spatial layout

Distribution matrix

Image classification

## ABSTRACT

In this paper, we propose an adaptive spatial pooling method for enhancing the discriminability of feature representation for image classification. The core idea is to adopt a spatial distribution matrix to define how the image patches are pooled together. By formulating the pooling distribution learning and classifier training jointly, our method can extract multiple spatial layouts of arbitrary shapes rather than regular rectangular regions. By proper mathematical transformation, the distributions can be learned via a boosting-like algorithm, which improves the efficiency of learning especially for large distribution matrices. Further, our method allows category-specific pooling operations to take advantage of the different spatial layouts of different categories. Experimental results on three benchmark datasets UIUC-Sports, 21-Land-Use and Scene 15 demonstrate the effectiveness of our method.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

Pooling is a crucial step in popular image classification methods, such as Bag-of-Visual-Words (BoVW [1,2]) and Convolution Neural Network (CNN [3,4]). It is used to aggregate a set of unordered local features into a vector representation. Based on this representation, discriminative classifiers (such as SVM [5,6], neural networks [7] and boosting [8,9]) can be trained for various classification tasks. Here we use  $\mathbf{v} = f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$  to define a pooling operation, where  $\mathbf{x}_i \in \mathbb{R}^d$  ( $i = 1, 2, \dots, m$ ) refer to local features within a spatial region  $\mathcal{R}$  and  $\mathbf{v}$  denotes the pooled vector. Here,  $d$  and  $m$  are the dimensionality and the number of local descriptors, respectively. There are two important factors for a pooling operation: One is the operator function  $f$  which defines the way to merge the local features, such as average pooling [10], max pooling [11,12] and  $l_p$ -norm pooling [13]. The other is the action region  $\mathcal{R}$  and it decides which local features will be selected for pooling. As we know, if  $\mathcal{R}$  covers the whole image as in BoVW methods, the pooled vector  $\mathbf{v}$  is invariant with the spatial shifts of  $\mathbf{x}_i$  because the spatial relationship is totally ignored within the action scope. This is helpful to tolerate spatial shifts, but it drops discriminative information about the spatial layout, which usually plays a very important role for image classification.

Several methods have been proposed to take advantage of the spatial layout of regions. One representative is the spatial pyramid matching representation (SPM [2]). Essentially, the SPM method

partitions images into uniform sub-regions at different levels of resolution, and then applies a pooling operator on these sub-regions separately. The final representation is obtained by concatenating the pooled features of different sub-regions. With the help of spatial information, SPM achieves significantly better performance compared with the BoVW model. However, the spatial partition patterns of SPM need to be predefined, and the number and the style of the spatial partition patterns are very limited, such as the  $1 \times 1$ ,  $2 \times 2$  and  $4 \times 4$  uniform grids. Some methods [14,15] improve the SPM by adopting abundant random action regions in the image, but these methods often suffer from exhaustive search from a large region pool and the action regions are still constrained to regular shapes. Other methods [16,17] are proposed to pool the features corresponding to foreground and background separately with the help of object detection. More details are discussed in Section 2.

In recent years, weighted pooling has been widely used in order to capture spatial layouts in images more flexibly. By giving each local feature a weight and then representing an image as the weighted sum of local features, the method can extract information from regions of arbitrary shapes (rather than rectangular regions), and define very flexible pooling operators (other than average pooling and max pooling). It is easy to see that both BoVW and SPM are special cases of weighted pooling, with globally uniform weights and rectangular region uniform weights, respectively. The design of weights for pooling is influential to the image classification performance. Harada et al. [18] select weights by maximizing the Partial Least Squares and Fisher criteria, while Huang et al. use multiple Gaussian distributions to depict the spatial structures [19]. Similar to our proposed method, the method of [20] (abbreviated as LSPR) optimizes weights along with the training of classifiers.

\* Corresponding author. Tel. +86 13811659042; fax: +86 10 8254 4594.

E-mail addresses: [yliu@nlpr.ia.ac.cn](mailto:yliu@nlpr.ia.ac.cn) (Y. Liu), [ymzhang@nlpr.ia.ac.cn](mailto:ymzhang@nlpr.ia.ac.cn) (Y.-M. Zhang), [xyz@nlpr.ia.ac.cn](mailto:xyz@nlpr.ia.ac.cn) (X.-Y. Zhang), [liucl@nlpr.ia.ac.cn](mailto:liucl@nlpr.ia.ac.cn) (C.-L. Liu).

However, the LSPR models the relationships between the weights and the classifiers using a multi-layer perceptron (MLP), which is computationally expensive because the weights corresponding to different structures need to be optimized simultaneously. The previous methods also have the limitation that the weights for pooling are shared by all categories. This leads to the under-utilization of discriminative spatial information because different categories usually have different spatial layouts.

In this paper, we propose an adaptive spatial pooling (ASP) method for image classification with the objective of overcoming the under-utilization of spatial information in previous methods. Our core idea is to adopt a spatial distribution matrix to define how the image patches are pooled together. It avoids the prior definition of how to partition images or how to design action regions, and learns a flexible pooling scheme on the whole image directly from the training data. We formulate the pooling distribution learning and classifier training into a unified framework and optimize the joint learning problem via a boosting-like algorithm. Compared with existing methods, our method has three advantages: (1) Since the pooling operator is parameterized as a matrix (each column denotes a distribution of patches), our method can extract various spatial layouts of flexible shapes embedded in images; (2) By proper mathematical transformation, our problem is efficiently solved via a boosting-like algorithm, especially for a distribution matrix of large size; (3) Category-specific pooling operator can be learned by discriminative training. This endows more discriminative power to our model. Fig. 1 shows some examples of the distributions learned by our method. It is obvious that the learned distributions reflect the spatial layout of images.

The rest of this paper is organized as follows: Section 2 reviews the related work about pooling; Section 3 introduces the proposed ASP method in detail; Section 4 presents our experimental results on several benchmark datasets. Finally, Section 5 gives concluding remarks.

## 2. Related work

Bag-of-Visual-Words (BoVW) and Convolutional Neural Network (CNN) are two popular image representation methods for image classification and object recognition. CNN learns image representations by performing convolution and pooling operation alternately on the whole image. It has achieved the state-of-the-art performance on many datasets, such as MNIST [21], NORB [22] and ImageNet [23]. However, it is computationally expensive and needs large training datasets to avoid over-fitting. On the contrary, The BoVW framework, based on hand-craft features, is much cheaper in computation and also achieves good performance on many real-world problems [24,25]. The proposed method is under the BoVW framework and can be combined with CNNs in the future, because the convolution outputs of CNNs can be taken as

local features for adaptive pooling. The BoVW framework involves several steps, each with many techniques proposed:

- *Local feature extraction*: In this stage, the patches of interest are located by either sparse sampling or dense sampling, and features are extracted from the sampled regions. In sparse sampling, patches are selected by interest point/region detectors, such as Harris detector [26,27], DoG [28] and MSER [29]. These methods are usually time-consuming and may miss some important regions, however. A simple and popular technique, called dense sampling, is to sample patches with a fixed step and patch size on the whole image. To extract features, we apply hand-craft feature descriptors, such as HoG [30], SIFT [28], SURF [31] and LBP [32], to each patch.
- *Codebook learning*: The codebook in computer vision is analogous to the vocabulary in natural language processing (NLP). It consists of some representative codewords from local features, which can be learned in either unsupervised or supervised manner. While k-means clustering [33] is most widely used for codebook learning, many advanced methods have been proposed for improving the discriminative ability of the codebook [34–37].
- *Encoding*: This step is to map the local features from the original feature space to a new space describing the weights of codewords. Accordingly, the dimensionality after encoding for each local patch equals the number of codewords. A simple coding scheme is the hard coding (HC [11]), which encodes each local feature with the most similar codeword. Unlike the HC, many advanced coding methods make full use of the codebook, such as the soft coding [38,39], sparse coding [12], local linear coding [40] and salient coding [41].
- *Pooling*: This is an operation to aggregate the codes of local patches into a vector representation of the image. Since our work improves the classification performance by proposing a new pooling method, we discuss the existing pooling methods in more details below.

The idea of feature pooling dates back to the research in 1960s [42]. Huber et al. discovered in the cat's visual cortex that the responses of high complex cells which receive signals from simple cells are insensitive to small spatial shift. This inspired the pooling operation widely used in vision recognition systems [43,44]. Many previous works aimed at finding a good pooling operator. The most popular operators are the average pooling and the max pooling. The average pooling [10] takes the average value of all local features  $\mathbf{x}$  within a region as the pooled feature  $\mathbf{v}$ , usually used along with hard coding. In max pooling [11], each dimension of  $\mathbf{v}$  is the maximum value of the corresponding dimension of set of  $\mathbf{x}$  in the region. Many advanced coding schemes, such as sparse coding [12] and localized soft coding [39], are combined with max pooling and have achieved considerable performance gain.

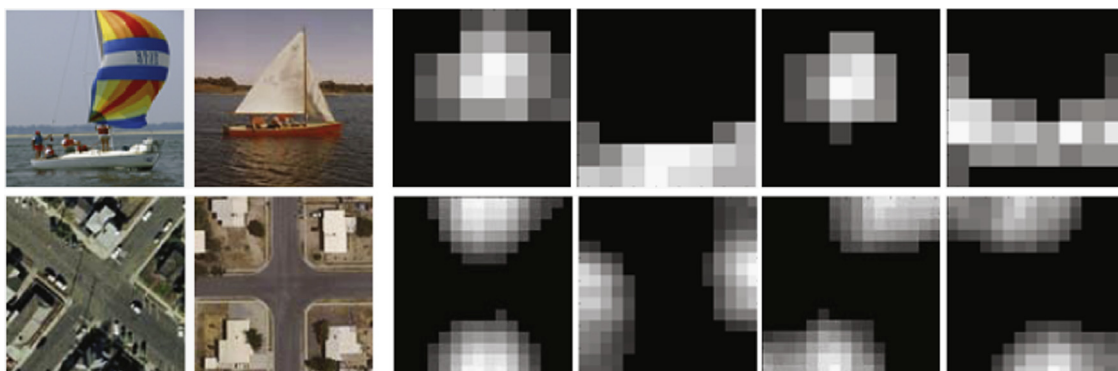


Fig. 1. Visualization of some learned distribution matrices. Each row stands for one specific category: two original images and four learned pooling distributions.

Some other works aimed at designing action regions for pooling. Sánchez et al. [17] and Russakovsky et al. [16] made use of object detection methods to pool foreground and background features separately to form the image-level representation. Zhu et al. [45] proposed to learn explicit and meaningful tangram templates for scene configurations through tree parsing. Based on the tangram model, Zhu et al. [46] constructed a hierarchical ROI dictionary (HRD) for spatial pooling. It utilized the compositionality among ROIs and employed partial least squares analysis to learn a compact and discriminative image representation. In [14], Jiang et al. proposed a method to select a set of partition patterns from a large number of randomized spatial partition patterns for each category. Similarly, Jia et al. [15] proposed a receptive field learning method for pooling image features. It first generates a high-dimensional representation on a over-complete set of receptive fields, and then uses a classifier with structural sparsity constraint to perform dimensionality reduction and classification.

The weighted pooling, with better flexibility, is getting increasing attention in recent years. Especially, by adopting weighted pooling on the whole image, the weights can effect on both the pooling operator and the action regions. Inspired by SPM which concatenates all the pooled features of image regions of different levels, Harada et al. [18] proposed a discriminative spatial pyramid representation (DSP) approach, which forms the image representation as the weighted sum of semi-local features over all the pyramid levels, and the weights are selected according to two discriminative criterions – the Partial Least Squares and the Fisher criterion (abbreviated as PlsSPR and FishSPR, respectively). In [13], Feng et al. proposed a weighted  $l_p$ -norm pooling method. Although this method utilizes weighted pooling operation as our method does, its weights are visual-word-specific while ours are category-specific. In [19], Huang et al. employed multiple Gaussian distributions to depict the global spatial layout of images, but the locations of Gaussian centers are uniformly sampled within the whole image. In [20], Malinowski and Fritz adopted a multi-layer perceptron, whose first hidden-layer parameters can be considered as the weights of smooth regions for pooling local features. Similar to our proposed method, the pooling operation of [20] is optimized along with the training of classifiers, but its sharing of weights among different categories may cause a loss of category-specific discriminative spatial information. Furthermore, it suffers from high computational complexity because the weights for different spatial layouts need to be optimized simultaneously, while our proposed method can learn weight distributions one after another via the boosting framework.

### 3. Adaptive spatial pooling method

Given an image  $I$ , we first extract local features by describing local patches with some descriptor (such as SIFT descriptor), and denote them as  $\mathbf{f}_i (i = 1, \dots, m)$ , where  $m$  is the number of patches of image  $I$ . Then a number of local features are selected at random from the training set to generate the codebook  $B$  by  $k$ -means clustering. Assuming that the codebook size is  $d$ , each image can be represented with a coding matrix  $\mathbf{X} \in \mathbb{R}^{d \times m}$ , in which the  $i$ -th column  $\mathbf{x}_i \in \mathbb{R}^d$  is the coding feature of the  $i$ -th patch.

#### 3.1. Model

To explore the discriminative spatial layout of images, we define the pooling operator by:

$$\mathbf{v} = \mathbf{X}\mathbf{p}, \quad (1)$$

where  $\mathbf{v}$  is the pooled vector,  $\mathbf{X}$  is the coding matrix described above and  $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_m]'$  refers to the distribution vector which is used to weight the local coding features.<sup>1</sup> Based on this representation, a bilinear classification function is defined as:  $f(\mathbf{X}) = \mathbf{w}'\mathbf{X}\mathbf{p} + b$ , where  $\mathbf{w} \in \mathbb{R}^d$  and  $b \in \mathbb{R}$  are the parameters of the classifier. As it is well known, applying traditional pooling operators (e.g. max pooling and average pooling) over the whole image involves the loss of spatial information. On the contrary, by using the operator defined in Eq. (1), discriminative spatial information is well preserved in the distribution vector  $\mathbf{p}$ .

Since there may be multiple spatial layouts for each category, one single distribution is not enough. Even with unitary spatial layout, images are usually composed of several distinct objects or regions, which should be pooled separately. Therefore, we define a distribution matrix  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T] \in \mathbb{R}^{m \times T}$  to account for various spatial layouts, where  $m$  is the number of patches,  $T$  is the number of distributions and each column in  $\mathbf{P}$  refers to a distribution of patches. The pooled feature is calculated by  $\mathbf{X}\mathbf{P} = [\mathbf{X}\mathbf{p}_1, \mathbf{X}\mathbf{p}_2, \dots, \mathbf{X}\mathbf{p}_T]$  and the classification function is defined by:

$$F_T(\mathbf{X}) = \text{tr}(\mathbf{W}'\mathbf{X}\mathbf{P}) + b = \sum_{t=1}^T (\mathbf{w}_t'\mathbf{X}\mathbf{p}_t + b_t), \quad (2)$$

where  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_T] \in \mathbb{R}^{d \times T}$  and  $b = \sum_t b_t \in \mathbb{R}$  are the parameters of the classifier.

To get uniform representations for images of different sizes, we perform a two-step preprocessing. First, each image is partitioned into the same number of blocks (e.g.  $8 \times 8$ ,  $16 \times 16$ ). Second, each block is represented by pooling the coding features of local patches within this block through max pooling. This operation not only unifies the number of patches among images of different sizes but also simplifies computation with a smaller  $m$  compared to the original number of patches.

#### 3.2. Objective function

Here we just consider the binary classification problem for simplicity. For the multi-category case, we transform it into multiple binary classification problems by the one-versus-all strategy and solve them separately.

To learn the classifier parameters  $\{\mathbf{W}, b\}$  and the spatial distribution matrix  $\mathbf{P}$ , we minimize the following objective function:

$$J(\mathbf{P}, \mathbf{W}, b) = \sum_i \exp(-y^{(i)}(\text{tr}(\mathbf{W}'\mathbf{X}^{(i)}\mathbf{P}) + b)) + C_1 \|\mathbf{W}\|_F^2 + C_2 \|\mathbf{P}\|_F^2 + C_3 \text{tr}(\mathbf{P}'\mathbf{L}\mathbf{P}), \quad (3)$$

where  $y^{(i)} \in \{+1, -1\}$  is the label of image  $i$  and  $\|\cdot\|_F$  refers to Frobenius norm. The first term in Eq. (3) measures the training error in exponential loss, while the last three terms are commonly used regularization constraints. Here,  $C_1$ ,  $C_2$ , and  $C_3$  are positive constants balancing the loss term against the regularization terms. In the last term,  $\mathbf{L} \in \mathbb{R}^{m \times m}$  is the Laplacian matrix of the adjacency graph of image patches. The graph, denoted by  $\mathbf{A} \in \mathbb{R}^{m \times m}$ , is constructed by  $\mathbf{A}_{ij} = 1$  if the patch  $j$  is a neighbor of the patch  $i$  in space and  $\mathbf{A}_{ij} = 0$  otherwise. Thus, the purpose of the last term is to smooth the distribution weights on nearby patches. To see that, we just recall:

$$\text{tr}(\mathbf{P}'\mathbf{L}\mathbf{P}) = \sum_{t=1}^T \mathbf{p}_t'\mathbf{L}\mathbf{p}_t = \frac{1}{2} \sum_{t=1}^T \sum_{i,j=1}^m \mathbf{A}_{ij} (\mathbf{p}_t(i) - \mathbf{p}_t(j))^2. \quad (4)$$

#### 3.3. Optimization via boosting-like algorithm

Due to the large number of parameters, directly optimizing  $\mathbf{W}$  and  $\mathbf{P}$  is time-consuming. A more efficient way is to treat

<sup>1</sup>  $\mathbf{p}$  is not a strict distribution because the sum of  $\mathbf{p}$  is not constrained to be 1.

$F_T(\mathbf{X})$  as the sum of  $T$  weak classifiers  $f_t(\mathbf{X}) = \mathbf{w}_t^T \mathbf{X} \mathbf{p}_t + b_t$  ( $t = 1, \dots, T$ ), and train them one after another via the boosting framework [9].

Thus, the objective function for the  $t$ -th iteration can be written as,

$$\begin{aligned} J_t(\mathbf{p}_t, \mathbf{w}_t, b_t) &= \sum_i \exp(-y^{(i)}(F_{t-1}(\mathbf{X}^{(i)}) + f_t(\mathbf{X}^{(i)}))) + C_1 \|\mathbf{w}_t\|_2^2 \\ &\quad + C_2 \|\mathbf{p}_t\|_2^2 + C_3 \mathbf{p}_t^T \mathbf{L} \mathbf{p}_t + M \\ &= \sum_i \alpha_t^{(i)} \exp(-y^{(i)} f_t(\mathbf{X}^{(i)})) + C_1 \|\mathbf{w}_t\|_2^2 \\ &\quad + C_2 \|\mathbf{p}_t\|_2^2 + C_3 \mathbf{p}_t^T \mathbf{L} \mathbf{p}_t + M, \end{aligned} \quad (5)$$

where  $\alpha_t^{(i)} = \exp(-y^{(i)} F_{t-1}(\mathbf{X}^{(i)}))$  is the weight of image  $i$  for the  $t$ -th iteration, and  $M$  is a constant. It is straightforward to show that  $\alpha_t^{(i)} \exp(-y^{(i)} \mathbf{w}_t^T \mathbf{X} \mathbf{p}_t + b_t)$ ,  $\|\mathbf{p}_t\|_2^2$  and  $\mathbf{p}_t^T \mathbf{L} \mathbf{p}_t$  are convex functions with respect to  $\mathbf{p}_t$ . Because the sum of convex functions is still convex, the objective function  $J_t$  is convex with respect to  $\mathbf{p}_t$ . Similar analysis can be applied to show that  $J_t$  is also convex with respect to  $\mathbf{w}_t$  and  $b_t$ . In consequence, we can use gradient descent to optimize  $\{\mathbf{w}_t, b_t\}$  and  $\mathbf{p}_t$  alternately, and the gradients with respect to  $\mathbf{w}, b$  and  $\mathbf{p}$  are written as follows:

$$\frac{\partial J_t}{\partial \mathbf{w}_t} = - \sum_i y^{(i)} \alpha_t^{(i)} \exp(-y^{(i)}(\mathbf{w}_t^T \mathbf{X}^{(i)} \mathbf{p}_t + b_t)) \mathbf{X}^{(i)} \mathbf{p}_t + 2C_1 \mathbf{w}_t, \quad (6)$$

$$\frac{\partial J_t}{\partial b_t} = - \sum_i y^{(i)} \alpha_t^{(i)} \exp(-y^{(i)}(\mathbf{w}_t^T \mathbf{X}^{(i)} \mathbf{p}_t + b_t)), \quad (7)$$

$$\frac{\partial J_t}{\partial \mathbf{p}_t} = - \sum_i y^{(i)} \alpha_t^{(i)} \exp(-y^{(i)}(\mathbf{w}_t^T \mathbf{X}^{(i)} \mathbf{p}_t + b_t)) \mathbf{X}^{(i)'} \mathbf{w}_t + 2C_2 \mathbf{p}_t + 2C_3 \mathbf{L} \mathbf{p}_t. \quad (8)$$

We summarize the ASP algorithm in Algorithm 1, and the learning procedure for each distribution and weak classifier is shown in Algorithm 2. Although the pooling operation is defined on the whole image, the patches whose corresponding distribution weights are zeros or very small can be viewed as being eliminated from the final decision function. Therefore, a sparse distribution learned from the data can be considered as an automatic region selection operator. However, in order to speed up the algorithm, we just adopt a simple operation to achieve sparsity, which is described in step 3 of Algorithm 2.

**Algorithm 1.** Adaptive spatial pooling via boosting-like algorithm.

**Given:**  $\{\mathbf{X}^{(i)}, y^{(i)}\}$ , the penalty parameters  $C_1, C_2, C_3$  and  $T$

Initialize:  $\alpha^{(i)} = \frac{1}{N}$ ,  $i = 1, 2, \dots, N$ ,  $t = 1$

**do:**

1. Optimize  $\mathbf{w}_t, b_t, \mathbf{p}_t$  using Algorithm 2.

2. Set  $f_t(\mathbf{X}) = \mathbf{w}_t^T \mathbf{X} \mathbf{p}_t + b_t$

3. Set  $\alpha^{(i)} \leftarrow \alpha^{(i)} \exp(-y^{(i)} f_t(\mathbf{X}^{(i)}))$  and normalize to

$$\sum_{i=1}^N \alpha^{(i)} = 1$$

4.  $t = t + 1$

**until**  $t = T$

**Output:** The classifier and distribution  $\{\mathbf{w}_t, b_t, \mathbf{p}_t, t = 1, 2, \dots, T\}$

**Algorithm 2.** Alternate optimization of  $\mathbf{w}$  and  $\mathbf{p}$ .

**Given:**  $\{\mathbf{X}^{(i)}, y^{(i)}\}$ , the penalty parameters  $C_1, C_2, C_3$ , the weights  $\alpha^{(i)}, i = 1, 2, \dots, N$ ,  $maxIter$  and  $\epsilon$

Initialize:  $\mathbf{w}$  and  $\mathbf{p}$  with random matrix,  $iter = 0$

**do:**

1. Fix  $\mathbf{p}$ , and optimize  $\mathbf{w}$  and  $b$  by:

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\partial J}{\partial \mathbf{w}} \text{ using Eq. (6)}$$

$$b \leftarrow b - \frac{\partial J}{\partial b} \text{ using Eq. (7)}$$

2. Fix  $\mathbf{w}$  and  $b$ , and optimize  $\mathbf{p}$  by

$$\mathbf{p} \leftarrow \mathbf{p} - \frac{\partial J}{\partial \mathbf{p}} \text{ using Eq. (8)}$$

3. Set  $\mathbf{p}_j = 0$  if  $\mathbf{p}_j < 0$  or  $|\mathbf{p}_j| < \text{mean}(|\mathbf{p}|)$

4.  $iter = iter + 1$

**until**  $|\delta J| \leq \epsilon$  or  $iter = maxIter$

**Output:** The classifier and distribution  $\mathbf{w}, b, \mathbf{p}$

### 3.3.1. Computational complexity

The main computational cost of Algorithm 2 comes from computing the gradients of  $J_t$  with respect to  $\mathbf{w}_t$  and  $\mathbf{p}_t$ . It is easy to see that computing  $\partial J_t / \partial \mathbf{w}_t$  needs  $O(dmN)$  operations, while computing  $\partial J_t / \partial \mathbf{p}_t$  needs  $O(dmN + m^2)$  operations, where  $N$  is the number of training images. Therefore, the overall computational complexity of Algorithm 2 is  $O((dmN + m^2)T')$ , where  $T'$  is the number of iterations. The computational complexity of Algorithm 1 is  $T$  times that of Algorithm 2, where  $T$  is the number of distributions.

### 3.3.2. Convergence analysis

In order to analyze the convergence of Algorithm 2, we denote the value of the objective function in the  $t$ -th iteration as  $J(\mathbf{p}^t, \mathbf{w}^t, b^t)$  (ref. Eq. (5)) while omitting the index of distributions.  $J(\mathbf{p}, \mathbf{w}, b)$  is convex with respect to classifier parameters  $\mathbf{w}$  and  $b$ . After updating it via Eqs. (6) and (7), we have

$$J(\mathbf{p}^t, \mathbf{w}^{t+1}, b^{t+1}) \leq J(\mathbf{p}^t, \mathbf{w}^t, b^t). \quad (9)$$

Next,  $\mathbf{w}$  and  $b$  are fixed, and  $\mathbf{p}$  is updated according to Eq. (8). Due to the convexity, we get

$$J(\mathbf{p}^{t+1}, \mathbf{w}^t, b^t) \leq J(\mathbf{p}^t, \mathbf{w}^t, b^t). \quad (10)$$

As mentioned above, the value of the objective function will decrease in each step of optimizing  $\mathbf{w}$  or  $\mathbf{p}$  by gradient descent. Considering that the value of the objective function is lower bounded by zero, we conclude that the alternate optimization in Algorithm 2 is convergent.

## 3.4. Classification

We introduce two approaches based on the proposed framework for image classification as follows.

### 3.4.1. ASP-original

By this approach, we directly use the learned  $\{\mathbf{W}^c, b^c\}$  and  $\mathbf{P}^c$  ( $c = 1, 2, \dots, C$ ) (ref. Eq. (3)) as the classifier and the distribution matrix of the category  $c$ , respectively. To classify a sample  $\mathbf{X}$ , we first calculate the score that  $\mathbf{X}$  belongs to each category by Eq. (11), and then classify it into the category with the highest score according to Eq. (12):

$$\text{Score}_c(\mathbf{X}) = \text{tr}(\mathbf{W}^c \mathbf{X} \mathbf{P}^c + b^c), \quad (11)$$

$$\text{Label}(\mathbf{X}) = \arg \max_c \text{Score}_c(\mathbf{X}). \quad (12)$$

### 3.4.2. ASP-enhanced

By this approach, the learned classifiers  $\{\mathbf{W}^c, b^c\}$  ( $c = 1, 2, \dots, C$ ) are ignored. New classifiers are trained under the enhanced image representations, which are generated by pooling local coding features with the learned distribution matrices  $\mathbf{P}^c$  ( $c = 1, 2, \dots, C$ ). This approach is dichotomized into two methods:

- **ASP-enhanced-Bi:** For each category, after generating the image representations with the corresponding category-specific distribution matrix, a binary classifier is trained by the one-versus-all strategy. The output of the learned classifier is taken as  $\text{Score}_c$  for classification (ref. Eq. (12)).

- *ASP-enhanced-Mul*: The ensemble strategy is adopted in this method. For each category,  $C$  binary classifiers are trained. Each classifier is learned with the image representations under one category-specific distribution matrix. Then the sum of the outputs of the  $C$  classifiers is taken as the final score for this category.

Because image representations and classifiers are learned separately, some advanced classifiers can be adopted in the ASP-enhanced method rather than linear classifiers. Our experimental results show that the performance with nonlinear classifiers is consistently better than that with linear classifiers.

## 4. Experiments

We compare our method with several baselines and related methods on three public datasets UIUC-Sports [47], 21-Land-Use [48] and Scene 15 [49].

### 4.1. Experimental setting

The setting for the three baselines and our method are as follows:

- *RDM*: Random Distribution Matrix. We use a random matrix as  $\mathbf{P}$ , in which each  $\mathbf{P}_{ij}$  is i.i.d. sampled from a uniform distribution on  $[0,1]$ . The number of distributions for RDM is set to 21 which is the same as that of the 3-level SPM representation.
- *BoVW*: Bags of Visual Words. This method takes the histogram as the image representation, which is equivalent to utilizing an all-one vector as  $\mathbf{P}$ .
- *SPM*: Spatial Pyramid Matching Representation. We adopt the most widely used 3-level ( $1 \times 1$ ,  $2 \times 2$ ,  $4 \times 4$ ) SPM model and obtain features both with the average pooling and the max pooling.
- *Proposed method*: Adaptive Spatial Pooling. For simplicity, we set the parameters  $C_1, C_2$  and  $C_3$  (cf. Eq. (3)) to the same value and select them from the set  $[0.01, 0.05, 0.1, 0.5, 1, 5]$  by cross validation. The number of distributions for our method is set to no more than 10. In Algorithm 2, we set the convergence parameter to  $\epsilon = 10^{-4}$  and the maximum number of iterations to  $maxIter = 200$ .

For all the above methods, we adopt the hard coding to encode local features and experiment on both the linear and the nonlinear versions. For the nonlinear SVM classification, we follow the strategy proposed in [50] as an approximation. First, an additive kernel is applied to map the features into a high-dimensionality space, and then a linear classifier is trained in the mapping space. In our experiments, the intersection kernel [51] is adopted and the resulting features are 7 times the length of the original ones.

**Table 1**  
Classification accuracy on the UIUC-Sports dataset.

Algorithm	Acc. (linear)	Acc. (nonlinear)
RDM	81.5	83.3
BoVW	76.1	82.7
SPM (average-pooling)	80.0	84.3
SPM (max-pooling)	79.5	85.5
ASP-original	$86.5 \pm 0.13$	–
ASP-enhanced-Bi	$85.9 \pm 0.06$	$87.6 \pm 0.05$
ASP-enhanced-Mul	$85.8 \pm 0.09$	$88.5 \pm 0.10$

### 4.2. Experimental results

We implement our proposed method and the baselines by ourselves, and also compare with the results of related work from the corresponding publications. In each experiment, we repeat the training procedure for 10 times with different random initialization for the parameter  $\mathbf{w}$  and  $\mathbf{p}$ , and report the average precision and variance.

#### 4.2.1. UIUC-Sports dataset

The UIUC-Sports dataset [47] contains 8 sports event categories: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). These high-resolution images are divided into easy and medium classes according to the human subjective judgement. In our experiments, each image is resized to less than  $300 \times 300$  by down-sampling. The SIFT descriptors with  $12 \times 12$  patch-size and 4 pixels step-size are first extracted, and then a subset of SIFT features from the training images are randomly selected to generate a codebook with 1000 words by k-means clustering. As in [47,14], we randomly select 70 images from each category for training and use the rest for testing. Except the results in Fig. 4 which show the influence of image partition patterns on the performance, the other results are obtained under the image partition pattern of  $8 \times 8$  blocks without overlapping with the parameters  $C_1, C_2$  and  $C_3$  set to 1.

We first compare the proposed method with three baselines with both linear and nonlinear classifiers. As Table 1 shows, our method (ASP) significantly outperforms the baselines. It is worth mentioning that our method, even with the linear kernel, exceeds the baselines with the nonlinear kernel. We also compare our method with several competing methods. As shown in Table 2, our method (ASP-enhanced-Mul) improves the accuracy of RSP and LSPR by about 9%. To the best of our knowledge, our performance of 88.5% outperforms all previously published results for a single type of SIFT descriptor.

Fig. 2(a) shows the dependency of the accuracy on the number of distributions  $T$ . We can see that, for all the five variations of the proposed method, the accuracy with  $T=2$  is significantly better than that with  $T=1$ , gaining improvements of 5.5%, 3.4%, 2.0%, 3.0% and 2.5%, respectively. For the ASP-original method, accuracy increases rapidly when  $T$  is small and saturates when  $T$  is larger than six. However, for the ASP-enhanced-Mul method, accuracy with  $T=2$  is very close to the best performance. This is because the classifier training for each category takes account of not only the self-distributions but also the distributions of other categories.

The codebook size determines the dimensionality of the coding feature, which affects both the accuracy and the computational complexity. Fig. 3 shows the influence of the codebook size on accuracy for the ASP-original method on the UIUC-Sports dataset. As it is shown, higher performance is achieved with larger codebooks when the codebook size is less than 1000. However, when the codebook is enlarged to 1200 codewords, the accuracy decreases due to overfitting. We also investigate the dependency of accuracy on image partition patterns, which are used to unify the number of patches among images of different sizes. As Fig. 4 shows,

**Table 2**  
Comparison with other related work on the UIUC-Sports dataset.

Algorithm	Acc.
Object-back [47]	76.3
RSP + optimal selection [14]	77.9
RSP + Boosting [14]	79.6
LSPR [20]	79.4
The proposed	<b>88.5</b>

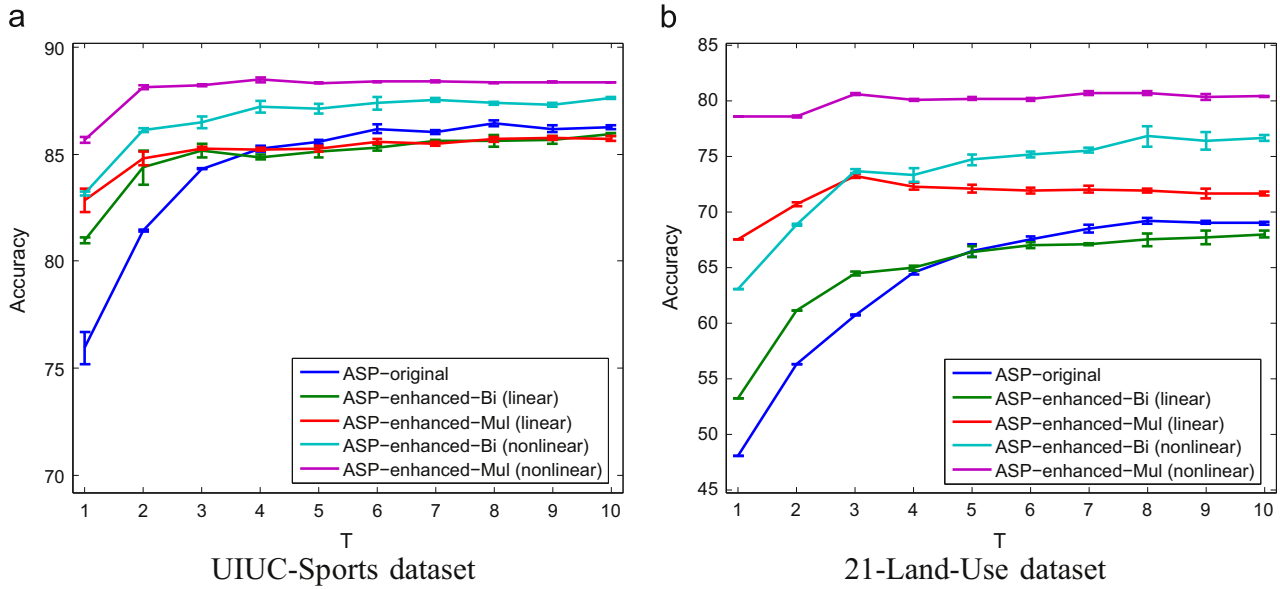


Fig. 2. Influence of  $T$  on accuracy.

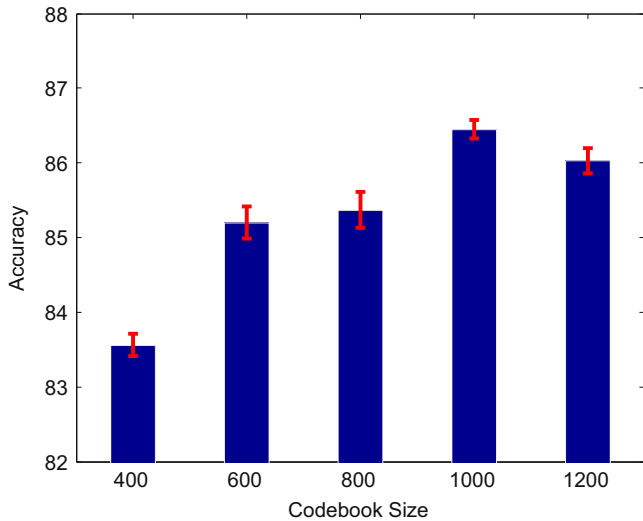


Fig. 3. Influence of the codebook size on accuracy for the ASP-original method on the UIUC-Sports dataset.

performance with image partition patterns  $8 \times 8$ ,  $12 \times 12$  and  $16 \times 16$  are similar, but are much better than the accuracy with the partition pattern  $4 \times 4$ . This is implied by the partition pattern  $4 \times 4$  which is too coarse to describe the spatial layout distinctly. However, with the same image partition pattern ( $4 \times 4$ ), our approach achieves 87.4% accuracy improving the 3-Level SPM method by 2%.

In order to intuitively describe the learned spatial distribution matrix  $\mathbf{P}$ , the first five learned distributions for each category are visualized by gray-scale maps shown in Fig. 5. We analyze the results from the following two aspects:

(1) *Number of distribution types.* For the categories whose spatial layouts are relatively simple, few distribution types will be learned. For example, most images of the category “snowboarding” are composed of two scenes: snow at the bottom and sky at the top. Thus, only two types of distributions are needed for pooling. In the last row of Fig. 5, the third and the fourth learned distributions are similar to the first two, which is consistent with our observation. In contrast, if the spatial layout is complex, more types of distributions will be learned, such as the first row for the category “badminton”.

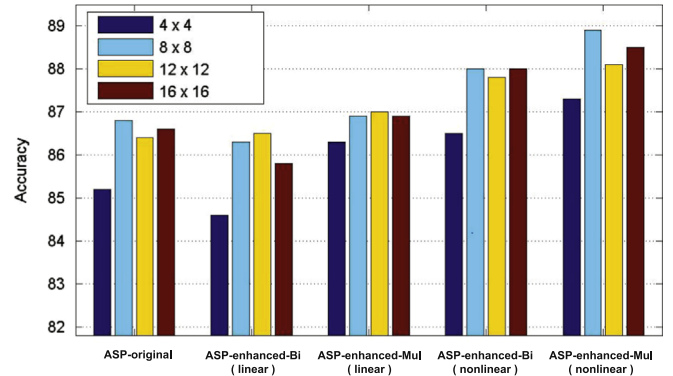


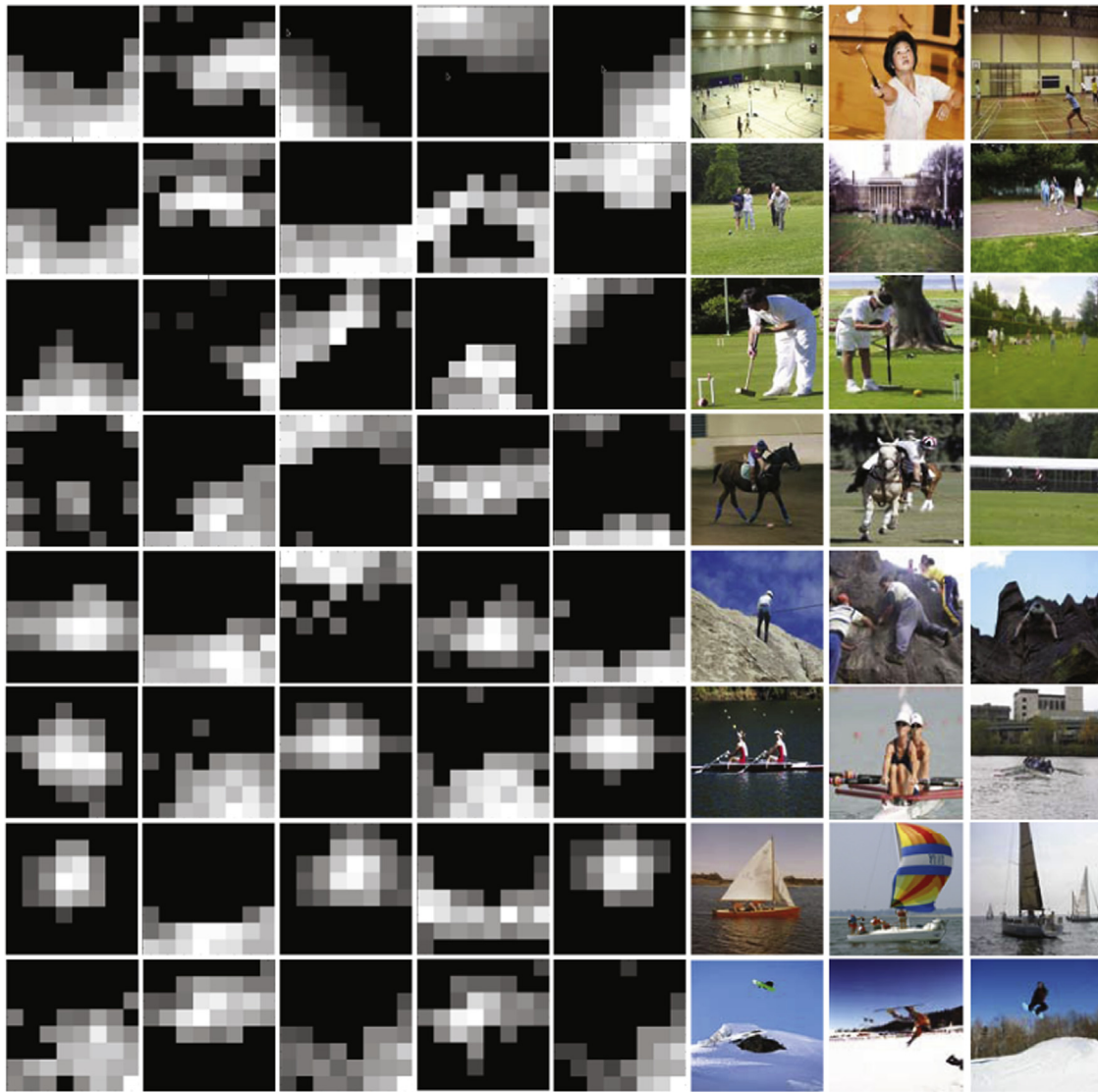
Fig. 4. Dependency of accuracy on image partition on the UIUC-Sport dataset. Four types of image partition patterns are considered and denoted by different colors. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

(2) *Discriminative ability.* For categories containing striking or salient objects, such as “rowing” and “sailing”,  $\mathbf{P}$  tends to extract foregrounds in the image, while for categories with complex spatial layouts or small objects, such as “badminton” and “golf”,  $\mathbf{P}$  tends to extract backgrounds or environments.

#### 4.2.2. 21-Land-Use dataset

The challenging 21-Land-Use dataset [48] is composed of 21 classes of aerial orthoimagery. There are 100 images for each of the following classes: agricultural, airplane, baseballdiamond, beach, buildings, chaparral, denseresidential, forest, freeway, golfcourse, harbor, intersection, mediumresidential, mobilehome, overpass, parkinglot, river, runway, sparseresidential, storagetanks, tennis, and tenniscourt. Each image ( $256 \times 256$  pixels) is manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around America. As in [14], we extract SIFT features with  $16 \times 16$  patch-size and 8 pixels step-size. The codebook size is set to 100. For each category, we randomly sample 80 images for training and use the rest for testing. In the preprocessing step, we merge patches into  $16 \times 16$  blocks and set  $C_1$ ,  $C_2$  and  $C_3$  to 0.5.

Table 3 gives the results for our proposed methods along with the baselines. As it is shown, the proposed methods perform consistently better than the baselines. In addition, different from



**Fig. 5.** Visualization of distribution matrices  $P$  on the UIUC-Sports dataset. Each row stands for one specific category: five learned pooling distributions and three original images.

**Table 3**  
Classification accuracy on the 21-Land-Use dataset.

Algorithm	Acc. (linear)	Acc. (nonlinear)
RDM	63.1	71.3
BoVW	58.6	68.3
SPM (average-pooling)	67.6	74.7
SPM (max-pooling)	62.9	68.8
ASP-original	$69.2 \pm 0.27$	–
ASP-enhanced-Bi	$68.0 \pm 0.28$	$76.8 \pm 0.91$
ASP-enhanced-Mul	$73.3 \pm 0.16$	$80.7 \pm 0.15$

the UIUC-Sports dataset, the ASP-enhanced-Mul method is significantly superior to the ASP-enhanced-Bi method. This is because the spatial layouts among categories are quite different in this dataset as shown in Fig. 5 and hence making use of distributions from other categories will bring an informative complement.

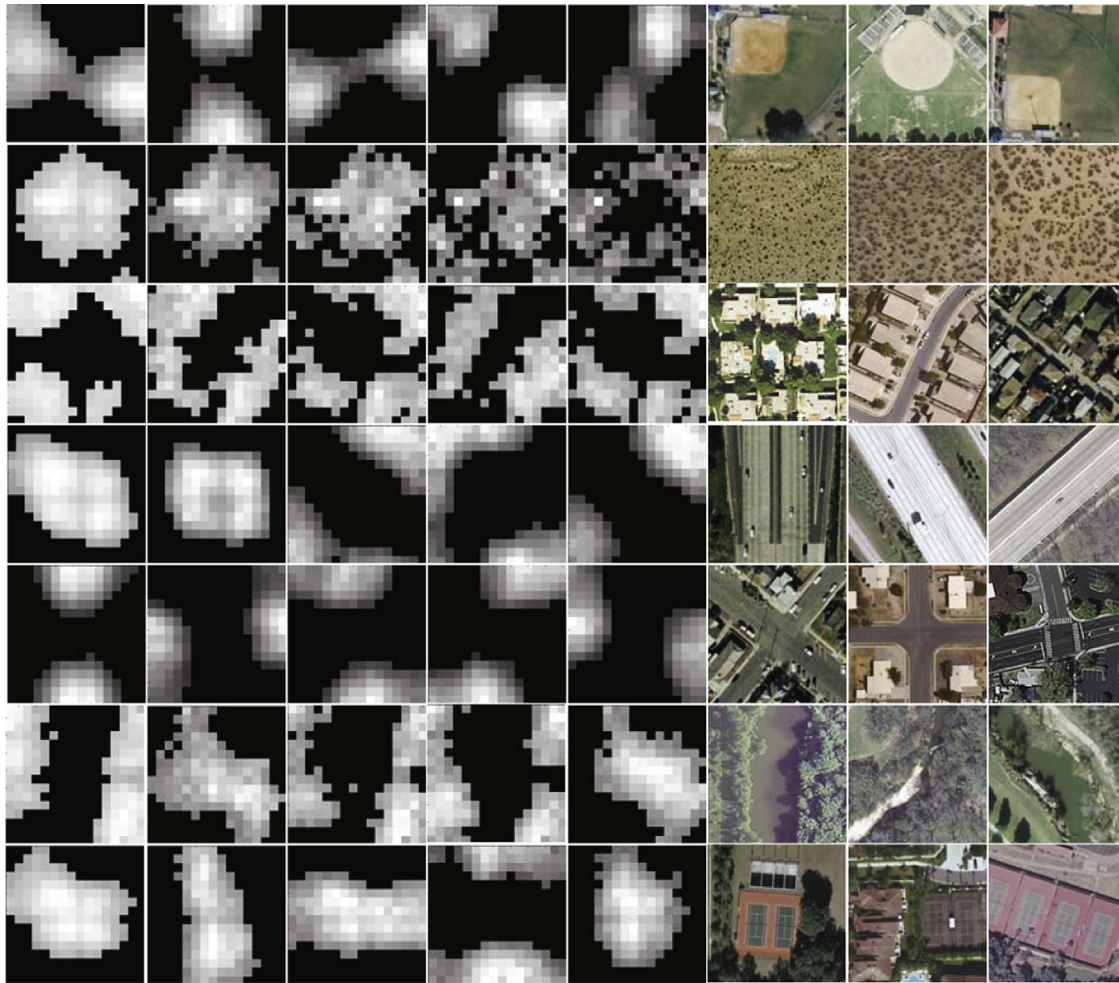
Table 4 lists the comparative results with several state-of-the-art methods. SPCK [48] is an algorithm proposed for overhead imagery, which captures both the absolute and the relative arrangement of words by making use of the co-occurrence relationship to represent images. SPCK+ and SPCK++ are both extended methods of SPCK. The proposed method (ASP-enhanced-Mul) achieves 80.7% accuracy,

**Table 4**  
Comparison with state-of-the-art methods on the 21-Land-Use dataset.

Algorithm	Acc.
SPCK [48]	73.1
SPCK+ [48]	76.1
SPCK++ [48]	77.3
RSP + optimal selection [14]	75.5
RSP + Boosting [14]	77.8
The proposed	<b>80.7</b>

which improves the accuracy of SPCK++ and RSP – Boosting by 3.4% and 2.9%, respectively.

Fig. 2 (b) shows the dependency of performance on the number of distributions  $T$ . We can see that the performance saturates when  $T \geq 5$  for ASP-enhanced-Bi and  $T \geq 3$  for ASP-enhanced-Mul. We randomly select 7 categories in this dataset and visualize the learned distribution matrix  $P$  with grey-scale maps in Fig. 6. It is interesting to observe that the learned distributions generally possess symmetric property on this dataset.



**Fig. 6.** Visualization of distribution matrices  $\mathbf{P}$  on the 21-Land-Use dataset. Each row stands for one specific category: five learned pooling distributions and three original images.

**Table 5**  
Classification accuracy on the Scene15 dataset.

Algorithm	Acc. (linear)	Acc. (nonlinear)
RDM	76.6	80.8
BoVW	74.7	78.9
SPM (average-pooling)	80.5	82.5
SPM (max-pooling)	75.5	77.1
ASP-original	83.1 $\pm$ 0.11	-
ASP-enhanced-Bi	80.9 $\pm$ 0.06	83.2 $\pm$ 0.01
ASP-enhanced-Mul	82.9 $\pm$ 0.03	84.7 $\pm$ 0.02

**Table 6**  
Comparison with other related methods on the Scene15 dataset.

Algorithm	Acc.
KSPM [2]	81.4
GLP [13]	83.2
MSP (hard coding ) [19]	78.7
MSP (super-vector coding ) [19]	84.3
RSP+optimal selection [14]	83.9
RSP+Boosting [14]	88.1
PlsSPR [18]	81.8
Tangram model [45]	81.6
HRD [46]	82.4
The proposed	<b>84.7</b>

#### 4.2.3. Scene 15 dataset

The Scene 15 dataset [49] is composed of 15 natural scene categories: bedroom, suburb, industrial, kitchen, livingroom, coast, forest, highway, insidicity, mountain, opencountry, street, tallbuilding, office and store. There are 4485 images in total, with the number of each category ranging from 216 to 400 images. We extract SIFT features for three patch sizes ( $12 \times 12$ ,  $18 \times 18$  and  $24 \times 24$ ) with a fixed step-size of 4 pixels, and the codebook size is set to 1024. For each category, we randomly select 100 images for training and use the rest for testing. We merge patches into  $16 \times 16$  blocks in the preprocessing phase with parameters  $C_1$ ,  $C_2$  and  $C_3$  set to 0.5.

The comparative results of our method with the baselines and other related work are presented in [Tables 5](#) and [6](#), respectively. The ASP-enhanced-Mul method achieves the accuracy of 84.7%, which is higher than KSPM [2], GLP [13], MSP [19], PlsSPR [18], Tangram Model [45] and HRD [46] but lower than RSP+Boosting [14]. However, RSP+Boosting adopts a large number of random patterns (100 partition patterns for each category) via boosting, while only three distributions are used in our method. Therefore, the dimensionality of RSP+Boosting is tens of times that of ours. Because of space limitation, we do not show the visualization of distribution matrix  $\mathbf{P}$ . Note that the most discriminative distributions learned by our method look like rectangles with long side in horizontal direction, which is consistent with the discovery in previous works that the  $3 \times 1$  spatial layout [52] is superior to the  $4 \times 4$  image partition adopted by SPM [2].

**Table 7**Average time for optimizing one distribution  $\mathbf{p}$  and classifier  $\{\mathbf{w}, b\}$  in Algorithm 2.

Dataset	UIUC-Sports	21-Land-Use	Scene15
Time (s)	3	8	23

**Table 8**

Running time for the proposed method.

Dataset		UIUC-Sports	21-Land-Use	Scene15
(a) Training time (s)				
ASP-original	$T=1$	12	19	379
	$T=5$	35	121	2498
	$T=10$	60	294	5375
ASP-enhanced-Bi	$T=1$	13	22	386
	$T=5$	37	124	2519
	$T=10$	63	298	5408
ASP-enhanced-Mul	$T=1$	13	22	398
	$T=5$	38	126	2559
	$T=10$	63	302	5479
(b) Testing time (ms)				
ASP-original	$T=1$	3	2	3
	$T=5$	3	2	6
	$T=10$	4	2	7
ASP-enhanced-Bi	$T=1$	4	2	3
	$T=5$	4	2	8
	$T=10$	4	2	13
ASP-enhanced-Mul	$T=1$	3	2	4
	$T=5$	4	2	10
	$T=10$	4	2	14

#### 4.3. Running time

All experiments are implemented in Matlab and performed on a CPU server with 2 Xeon E5-2690 2.9 GHz CPUs with 16 cores in total. First, the average time for optimizing one distribution  $\mathbf{p}$  and classifier parameters  $\{\mathbf{w}, b\}$  (ref. Algorithm 2) is shown in Table 7. We also report the training and testing time for the proposed method in Table 8, where the time for both ASP-enhanced-Bi and ASP-enhanced-Mul are counted under the linear vision. The value in Table 8(b) shows the average testing time per image, which is calculated by dividing the total testing time by the number of testing images. Note that the training time of the Scene15 dataset is much longer than that of the other two datasets. This is because parallel computing with 12 cores is adopted for the UIUC-Sports dataset and the 21-Land-Use dataset, while one single core is applied to the Scene15 dataset due to the memory limitation for Matlab parallel setting.

## 5. Conclusion

In this paper, we proposed a novel pooling operation with the aim of extracting discriminative spatial information from images. After characterizing this operation by a category-specific distribution matrix, we learned it along with the classifier under a unified optimization framework via boosting. With the help of the distribution matrix, our method can describe much more complex spatial layouts than the traditional image partitioning schemes, and thus extract more discriminative information to improve the classification performance. Experiments on three datasets demonstrated the superiority of our method. In future, we will extend the proposed method for the classification of large number of categories, which may suffer from severely imbalanced samples for the learning of category-specific distributions.

## Conflict of interest

None declared.

## Acknowledgment

This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316302, the National Natural Science Foundation of China under Grant 61203296, the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06040102), and the Xinjiang Uygur Autonomous Region Science and Technology Project (No. 201230122). The authors thank Xinwen Hou for helpful discussions.

## Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at <http://dx.doi.org/10.1016/j.patcog.2016.01.030>.

## References

- [1] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, In: European Conference on Computer Vision, 2004, pp. 1–22.
- [2] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, In: Computer Vision and Pattern Recognition, vol. 2, IEEE, New York, 2006, pp. 2169–2178.
- [3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324.
- [4] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, In: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
- [5] C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.* 20 (1995) 273–297.
- [6] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, *ACM Trans. Intell. Syst. Technol.* 2 (3) (2011) 27.
- [7] A. Jain, J. Mao, K. Mohiuddin, Artificial neural networks: a tutorial, *IEEE Comput.* 29 (3) (1996) 31–44.
- [8] R.E. Schapire, The boosting approach to machine learning: an overview, In: Workshop on Nonlinear Estimation and Classification, 2002.
- [9] C. Zhang, J. Liu, Q. Tian, Y. Han, H. Lu, S. Ma, A boosting, sparsity-constrained bilinear model for object recognition, *IEEE MultiMed.* 19 (2) (2012) 58–68.
- [10] Y.-L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, In: International Conference on Machine Learning, IEEE, Haifa, 2010, pp. 111–118.
- [11] Y. MarcAurelio Ranzato, L. Boureau, Y. LeCun, Sparse feature learning for deep belief networks, In: Advances in Neural Information Processing Systems, vol. 20, 2007, pp. 1185–1192.
- [12] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, In: Computer Vision and Pattern Recognition, IEEE, Miami, 2009, pp. 1794–1801.
- [13] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric lp-norm feature pooling for image classification, In: Computer Vision and Pattern Recognition, IEEE, Colorado Springs, 2011, pp. 2609–2704.
- [14] Y. Jiang, J. Yuan, G. Yu, Randomized spatial partition for scene recognition, In: European Conference on Computer Vision, Springer, Firenze, 2012, pp. 730–743.
- [15] Y. Jia, C. Huang, T. Darrell, Beyond spatial pyramids: receptive field learning for pooled image features, In: Computer Vision and Pattern Recognition, IEEE, Providence, 2012, pp. 3370–3377.
- [16] O. Russakovsky, Y. Lin, K. Yu, F.-F. Li, Object-centric spatial pooling for image classification, In: European Conference on Computer Vision, Springer, Firenze, 2012, pp. 1–15.
- [17] J. Sánchez, F. Perronnin, T. Campos, Modeling the spatial layout of images beyond spatial pyramids, *Pattern Recognit. Lett.* 33 (2012) 2216–2223.
- [18] T. Harada, Y. Ushiku, Y. Yamashita, Y. Kuniyoshi, Discriminative spatial pyramid, In: Computer Vision and Pattern Recognition, IEEE, Colorado Springs, 2011, pp. 1617–1624.
- [19] Y. Huang, Z. Wu, L. Wang, C. Song, Multiple spatial pooling for visual object recognition, *Neurocomputing* 129 (2014) 225–231.
- [20] M. Malinowski, M. Fritz, Learning smooth pooling regions for visual recognition, In: British Machine Vision Conference, 2013.
- [21] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, In: Computer Vision and Pattern Recognition, IEEE, Providence, 2012, pp. 3642–3649.

- [22] D.C. Ciresan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, In: International Joint Conference on Artificial Intelligence, ACM, Barcelona, 2011, pp. 1237–1242.
- [23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, *arXiv:1409.4842*, 2014.
- [24] <http://pascallin.ecs.soton.ac.uk/challenges/VOC/>.
- [25] <http://trdcvid.nist.gov/>.
- [26] C. Harris, M. Stephens, A combined corner and edge detector, In: Alvey Vision Conference, 1988, pp. 147–151.
- [27] T. Tuytelaars, L.V. Gool, Matching widely separated views based on affine invariant regions, *Int. J. Comput. Vis.* 59 (2004) 61–85.
- [28] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2) (2004) 91–110.
- [29] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, In: British Machine Vision Conference, BMVA Press, University of Cardiff, 2002, pp. 36.1–36.10.
- [30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, In: Computer Vision and Pattern Recognition, IEEE, San Diego, 2005, pp. 886–893.
- [31] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, *Comput. Vis. Image Underst.* 110 (3) (2008) 346–359.
- [32] T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, In: International Conference on Pattern Recognition, vol. 1, 1994, pp. 582–585.
- [33] S.P. Lloyd, Least squares quantization in PCM, *IEEE Trans. Inf. Theory* 28 (2) (1982) 129–137.
- [34] J. Mairal, F. Bach, Ponce, J., G. Sapiro, A. Zisserman, Supervised dictionary learning, In: Advances in Neural Information Processing Systems, 2008.
- [35] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, In: Computer Vision and Pattern Recognition, IEEE, San Francisco, 2010, pp. 3501–3508.
- [36] M. Yang, D. Dai, L. Shen, L.V. Gool, Latent dictionary learning for sparse representation based classification, In: Computer Vision and Pattern Recognition, IEEE, Columbus, 2014, pp. 4138–4145.
- [37] Q. Qiu, Z. Jiang, Sparse dictionary-based representation and recognition of action attributes, In: International Conference on Machine Learning, IEEE, Bellevue, Washington, 2011, pp. 704–714.
- [38] J.C. Gemert, J.-M. Geusebroek, C.J. Veenman, A.W. Smeulders, Kernel codebooks for scene categorization, In: European Conference on Computer Vision, Springer-Verlag, Marseille, 2008, pp. 696–709.
- [39] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, In: International Conference on Computer Vision, IEEE, Barcelona, 2011, pp. 2486–2493.
- [40] 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, IEEE, Colorado Springs, 2010, pp. 3360–3367.
- [41] Y. Huang, K. Huang, Y. Yu, T. Tan, Salient coding for image classification, In: Computer Vision and Pattern Recognition, IEEE, San Francisco, 2011, pp. 1036–1039.
- [42] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.* 160 (1) (1962) 106.
- [43] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, Y. LeCun, Ask the locals: multi-way local pooling for image recognition, In: International Conference on Computer Vision, IEEE, Barcelona, 2011, pp. 2651–2658.
- [44] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, In: Computer Vision and Pattern Recognition, IEEE, Colorado Springs, 2010, pp. 2559–2566.
- [45] J. Zhu, T. Wu, S.-C. Zhu, X. Yang, W. Zhang, Learning reconfigurable scene representation by tangram model, In: IEEE Winter Conference on Applications of Computer Vision, IEEE, Breckenridge, Colorado, 2012, pp. 449–456.
- [46] J. Zhu, W. Zou, X. Yang, R. Zhang, Q. Zhou, W. Zhang, Image classification by hierarchical spatial pooling with partial least squares analysis, In: British Machine Vision Conference, BMVA Press, University of Surrey, 2012, pp. 102.1–102.11.
- [47] L.-J. Li, F.-F. Li, What, where and who? Classifying events by scene and object recognition, In: International Conference on Computer Vision, IEEE, Rio de Janeiro, 2007, pp. 1–8.
- [48] Y. Yang, S. Newsam, Spatial pyramid co-occurrence for image classification, In: International Conference on Computer Vision, Barcelona, IEEE, 2011, pp. 1465–1472.
- [49] F.-F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, In: Computer Vision and Pattern Recognition, vol. 2, IEEE, San Diego, 2005, pp. 524–531.
- [50] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (3) (2012) 480–492.
- [51] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, In: International Conference on Computer Vision, vol. 2, IEEE, Beijing, 2005, pp. 1458–1465.
- [52] M. Marszalek, C. Schmid, H. Harzallah, v.d.W. Joost, Learning object representations for visual object class recognition, In: Visual Recognition Challenge Workshop, 2007.

**Yinglu Liu** received the B.S. degree in automation from Xiamen University, Xiamen, China, in 2009, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015. She is currently working at the China Samsung Telecom R&D Center. Her research interests include image detection and classification, scene parsing, and deep learning.

**Yan-Ming Zhang** received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2004, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently an Assistant Professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include machine learning and pattern recognition.

**Xu-Yao Zhang** received the B.S. degree in computational mathematics from Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. From July 2013, he has been an assistant professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include machine learning, pattern recognition, handwriting recognition, and deep learning.

**Cheng-Lin Liu** is a professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences, Beijing, China, and is now the director of the laboratory. He received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, the Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, in 1989, 1992 and 1995, respectively. He was a postdoctoral fellow at Korea Advanced Institute of Science and Technology (KAIST) and later at Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a research staff member and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. His research interests include pattern recognition, image processing, neural networks, machine learning, and the applications to character recognition and document analysis. He has contributed many effective methods to different aspects of handwritten document analysis. Some of his algorithms have been transferred to industrial applications including mail sorting, form processing and video text indexing. He has published over 200 technical papers at prestigious international journals and conferences. He is on the editorial board of journals *Pattern Recognition*, *Image and Vision Computing*, *International Journal on Document Analysis and Recognition* and *Cognitive Computation*. He is a Fellow of the IAPR and the IEEE.