Knowl Inf Syst (2017) 52:509–530
DOI 10.1007/s10115-016-1015-z

REGULAR PAPER

Learning extremely shared middle-level image representation for scene classification

Peng Tang[1] · Jin Zhang[1] · Xinggang Wang[1] · Bin Feng[1] · Fabio Roli[2] · Wenyu Liu[1]

Received: 23 May 2016 / Revised: 17 August 2016 / Accepted: 3 December 2016 /
Published online: 23 December 2016
© Springer-Verlag London 2016

Abstract Learning middle-level image representations is very important for the computer vision community, especially for scene classification tasks. Middle-level image representations currently available are not sparse enough to make training and testing times compatible with the increasing number of classes that users want to recognize. In this work, we propose a middle-level image representation based on the pattern that extremely shared among different classes to reduce both training and test time. The proposed learning algorithm first finds some class-specified patterns and then utilizes the lasso regularization to select the most discriminative patterns shared among different classes. The experimental results on some widely used scene classification benchmarks (15 Scenes, MIT-indoor 67, SUN 397) show that the fewest patterns are necessary to achieve very remarkable performance with reduced computation time.

Keywords Scene classification · Middle-level image representation · Extremely shared patterns

1 Introduction

Scene classification is a very challenging task in computer vision with applications in robot navigation, environment computing, Internet image classification, and so on. Typical solutions usually ignore the relevant correlations among the classes, and diversity inside the same class. For example, a scene image will contain more than one object that is always shared among the different scenes, which makes them visually similar. At the same time, even in the same scene, images are diverse and difficult to associate and recognize. Combined with the

✉ Xinggang Wang
xgwang@hust.edu.cn

[1] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

[2] Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy

difficulties, (1) the ever increasing size and categories of the data, and (2) requirements of many real time applications, all inspire us to dig a more compact and discriminative image representation and apply it for the scene classification problem.

One of the most popular image representation methods is the Bag-of-Features (BoF) [6]. The BoF model usually uses the unsupervised k-means [11] to learn a set of "visual words" as the codebook during training, then images can be represented by a histogram of these "visual words". But the BoF ignores some important information from the spatial layout of images, which is significant for scene classification. To deal with this problem, spatial pyramid matching (SPM) was introduced [19], and has greatly improved the performance of scene classification. However the histogram-based way of representation may lose much information of images, so some soft assignment methods have been proposed [25,47] to further boost scene classification performance.

But the unsupervised methods mentioned above to learn image representations ignore the information in image labels and have been proved not discriminative enough. To find the more discriminative patterns, many researchers are trying some weakly supervised methods. Like some part-based models [10,17,34,42,44] which find the objects (e.g., the human) or components of objects (e.g., the face or body of the human) in images as parts.[1] Each image can be represented by these parts, and usually in each certain class, images are represented by the parts confined in that class. These methods construct a collection of special parts for each image class as the learned patterns, which greatly improve the performance of unsupervised methods to classify scenes. Our method also makes a full use of the information in image labels during the procedure of pattern initializing and learning, to find the most discriminative ones that are shared with other classes.

Though many sophisticated methods have been proposed to learn the patterns for scene classification, they still have some limitations. Firstly, many previous methods use some handcrafted features like SIFT [28] and HOG [7] as local descriptors to represent patches in images. Though having achieved some satisfactory results, the codes generated by these methods are still very noisy to distinguish different patterns accurately. Recently, training a deep convolutional neural network (CNN) on large and diverse datasets like ImageNet [8] has achieved a breakthrough for image classification [18]. Meanwhile, training a deep CNN model on the large scene dataset also performs well for scene classification [53]. But using the deep CNN models trained on the ImageNet or scene dataset as the off-the-shelf feature extractor has been confirmed more feasible [9,14,31,37,41] than acting as classifier for scene dataset directly. In this work, we also verify that the features generated by deep CNN can be very efficient to represent and find the discriminative patterns.

Secondly, previous methods [10,17,34,42,44] require a large amount of patterns to construct the middle-level image representation, which results in unnecessary time and space costs in the procedure of learning patterns and representing images. For example, Singh et al. [42] and Juneja et al. [17] needs 210 and 50 patterns per class respectively, and even the most compact representation method MMDL [50] still requires 11 patterns per class. So they are unpractical for real applications because of the hardware and time demand. For instance, when dataset is very large and diverse like SUN 397 [51], which has 397 scene categories totally, their methods need tens even hundreds of patterns per class by estimation to represent the images, which is too time and space costly in the following procedure of patterns learning and image representation. Moreover, there is a trend for the increasingly large and complex scene datasets to emerge, like the large dataset ImageNet with thousands of object categories in total. So it is of great significance to propose a method to learn extremely shared middle-

[1] The "words", "parts" and "patterns" are interchangeable and this paper chooses "patterns" to represent them.

Fig. 1 The "chair" is shared among different classes, and the label of the image is marked on the *top left corner*

level image representations without the sacrifice of the discriminative ability, which needs less than five patterns per class. That would be a big contribution because current methods have tens even hundreds patterns per class and require too much training and testing time.

The limitations of previous methods originated that they only learn the patterns in each class separately, as they believe that the most discriminative patterns for a certain class only exist in that class. In some way it makes sense because if we can find some most special patterns for a class, it is more likely to distinguish that kind of images well. But in fact, in some classes we can never find patterns satisfying this requirement when the objects are too common or special but not exclusive. For example, the common but only objects (computers, desks, and chairs) in the scene of computer room are also distributed in other scenes like office and meeting room. Meanwhile, special objects in the scene of closet are not exclusive and shared with the clothing room. As shown in Fig. 1, the pattern "chair" is shared among many classes. All these facts inspire us to share some important patterns among different classes to represent images efficiently. Actually there already exist some research works in object detection which adopt the strategy of sharing parts [33,46]. The sharing strategy has the following advantages: (1) The number of self-adjustment parameters accompanies with that of the learned patterns, whose decline will reduce the chance of over-fitting; (2) it reduces the demand on the size of available training data; (3) it will reduce the computation time and storage space in the process of training and testing, which has great convenience when the dataset is very large. But for the task of scene classification, the problem of how to share patterns effectively and efficiently is still challenging. Because the target objects in scenes are more complicated and diverse. To solve this puzzle, we put forward a method to learn extreme compact image representation which utilizes the shared patterns among different classes.

Our work is related to the work by Parizi et al. [37], which implemented a jointly learned method to learn patterns, but they initially labeled a large pool of patterns in an unsupervised way, and then selected the most discriminative ones. As a consequence, it requires a lot of time in the process of labeling and selecting, and it is not compact and discriminative enough to represent the image. Based on their work, we propose a novel method to make several improvements. Initially, we adopted a weakly supervised strategy to learn a small number of patterns in each class, thus saving the computing time without the sacrifice of the discriminative ability of the patterns. Then, we introduced the lasso regularization [45] to select the most discriminative patterns and urge some of them shared among different classes, meanwhile maintaining some class-specialized ones. Results show that each class needs only four patterns on the average to achieve the remarkable performance, which are comparatively much smaller than referred by Parizi et al., and attest that it is a very compact way to represent the image through the extremely shared representation.

To summarize, the main contributions of our work are listed as follows.

– We demonstrate that there are many classes that have the same patterns and they are crucial for scene classification.

– We propose a novel method that firstly learns some class-specified patterns and then applies the lasso regularization to urge some patterns shared among different classes, which can generate a very compact and discriminative way for image representation. Experimental results show that our method takes the fewest patterns to achieve the excellent performance on three widely used scene benchmarks.

On the average, in each class, only four patterns are sufficient for the final image representation. It is a very time- and space-saving way to learn patterns and represent images for the recognition purpose. Moreover, the learned patterns are more discriminative after our training procedure, as the classification accuracy is boosted comparing with the classifier constructed by the initial patterns. In the experiments conducted on the three widely used scene benchmarks, our results using the fewest patterns are very competitive.

The rest of this paper is organized as follows. In Sect. 2, some related works are presented. In Sect. 3, the detailed procedure of image representation and learning shared patterns are unfolded. In Sect. 4, the experimental results and analysis are provided to illustrate the strength of our method. In Sect. 5, some discussions are given to further explain the proposed method. Section 6 concludes the paper.

2 Related work

Compared with low-level representations which can only capture some low-level image information like shape, color, and edge, middle-level image representations have the ability to find more semantic information like objects or components of objects, which are more discriminative to distinguish high-level semantic of images, e.g., the categories of image. Finding important patterns as middle-level image representation is a very popular way to recognize scene images. The traditional BoF model [6] utilizes some unsupervised clustering methods like k-means to learn patterns in images. But the patterns seem not discriminative enough, so many strongly supervised methods such as Attributes [13,36,38], Poselets [2], and Object Bank [20] have been proposed. Those methods can achieve better performance but they learn patterns using both image- and patch-level annotations, which are hard to obtain.

Recently, some weakly supervised strategies have been explored and proven very effective for image representation. They learn patterns under the supervision of image-level labels only, which are easier to obtain and it proved to be more discriminative to represent images than the unsupervised ones. Moreover, these methods are very robust because of avoiding the inaccuracy in labeling the patches. Thus the strategy to learn patterns via these weakly supervised methods is quite common. Singh et al. [42] employed linear SVM to distinguish patches in each class and then selected important ones as learned patterns on some criteria like purity and discriminativeness. Wang et al. [50] found patterns by introducing the multiple instance learning constraints to the dictionary learning. Some part-based models [10,17,34,44] were trying to find a set of class-specialized parts, i.e., the parts frequently appear in one class but rarely appear in other classes. But these methods learn patterns for each class separately, ignoring the fact that many discriminative patterns are appearing in more than one class. So patterns produced by their methods are not discriminative and compact enough. Our work can also be seen as a weakly supervised strategy, but we do not learn patterns for each class separately; instead, we share some important patterns among different classes to construct a compact way to represent the images.

Apart from unsupervised or weakly supervised ways, there are many other methods find patterns by the knowledge from experts [29,39]. Peraldi et al. [39] created patterns through

the Abox abduction to interpret multimedia like images. Neumann et al. [29] introduced description logics as a knowledge representation to interpret scenes. But these methods require knowledge from experts to develop some rules, which is hard to achieve. Our method can find some semantic patterns automatically, which is quite different from their methods.

With the powerful deep CNN model applied on image classification [18], we can also see the outstanding performance of CNN models trained on large diverse image dataset like ImageNet [8], when they are adopted as the local descriptor extractors to produce highly informative features for each patterns [9,14,23,24,31,41]. Liu et al. [24] chosen the activations from the fully connected 6th layer (fc6) of deep CNN model as the patch-level features, with the sparse coding based fisher vector to recognize images. Li et al. [23] also extracted the fc6 and learned patterns through pattern mining. Gong et al. [14] selected outputs from the fully connected seventh layer (fc7) to represent patches at multiple scales and then used VLAD to aggregate these outputs for scene classification and image retrieval. Dixit et al. [9] computed the activations from the fully connected eigth layer (fc8) and combined them with the semantic Fisher Vector for scene classification. All of these work selected deep features as local descriptors. Inspired by these experiences, we choose the activations from fc7 of the deep CNN model trained on ImageNet in four different scales to represent patches, which has proven to receive very satisfactory results for scene classification.

Actually the unsupervised methods [6,25,47] can also be regarded as methods to distribute patterns among different classes, but these patterns seem to be not discriminative enough to represent images, because the representation is the collection of these patterns which are shared among all different classes. Some works also learn shared parts for object detection [33, 46], and adopt the strategy of sharing patterns. Different from them, we faced with a more complicated puzzle in scene classification that the patterns are more diverse and complicated in one scene and more difficult to be shared. There are many other works trying to learn shared features. Some researchers learned shared features using the multitask learning [1,15,35,49]. Our method can also be regarded as the multitask learning because we learn patterns in each scene simultaneously and minimize the classification error during the learning procedure. But the difference is that we are trying to share the patterns among classes, which are more informative than the local features shared by their methods. Meanwhile, we introduce the image classifier during learning procedure aimed at sharing patterns among different scenes only. Actually we have demonstrated that retraining the final classifier is more effective than using the classifier during the process of learning patterns.

The most related works of our method were proposed by Lobel et al. [27] and Parizi et al. [37]. Lobel et al. [27] used hierarchical joint max-margin learning to learn the patterns and image classifier. But they required the weights of patterns to be positive and used some handcrafted descriptors, which may restrict the performance. Parizi et al. [37] also learned patterns and image classifier jointly. They selected the fc7 of deep CNN model trained on a large dataset for scene categorization [53] as local descriptors, and also learned negative patterns by allowing negative weights for them. Actually some patterns are also shared in their method. Our work follows the path of their work, but we make several improvements. Parizi et al. first initialize a large pool of patterns (tens of thousands of patterns from a relatively smaller scene dataset MIT-indoor 67 [40]) using an unsupervised learning method, and then select the most discriminative ones through jointly learning procedure. Compared with our extremely shared strategy, this initialization strategy is too costly and may not discriminative enough when applied in the large scene dataset like SUN 397 [51]. We are working in a different way, that is, we first employ a weakly supervised strategy to initialize a small number of class-specified patterns, and then apply the lasso regularization [45] to select the discriminative patterns and share some ones among different classes, meanwhile maintain

the class-specialized ones. Moreover, they jointly learn image representation and classifier to just improve the discriminativeness of patterns, while in our method, we introduce the image classifier to force some patterns are shared among different classes, and find some most discriminative patterns at the same time. Though the image-level classifier is constructed during training, we still retrain it after the patterns are learned to improve the performance. So we are with quite different objectives. Experimental results have proven that our method to construct a more compact and discriminative representation than theirs.

3 Learning method

In this section, we propose our novel solution in detail to learn extremely shared patterns, which can generate a very compact and discriminative middle-level image representation.

3.1 Pattern definition and image representation

Suppose $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ are images. For each image X_i, we can densely extract a set of local descriptors $F(X_i) = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{im_i}\}$ from different location and scales (i.e., different patch size), where $\mathbf{x}_{ij} = f(X_i, z_{ij}) \in \mathbb{R}^{d\times 1}$ is a d-dimensional feature vector of a patch at location z_{ij} in X_i, and m_i is the number of patches in image X_i. In this paper, patterns are defined as a cluster of linear filters $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$, $\mathbf{w}_k \in \mathbb{R}^{d\times 1}$, which can discover some most semantic patches, e.g., objects. The response of pattern k at location z_{ij} in X_i can be simply computed using the dot product

$$s_{ijk} = s\left(X_i, \mathbf{w}_k, z_{ij}\right) = \mathbf{w}_k^T \mathbf{x}_{ij}. \tag{1}$$

Similar to many previous work [27,37,42,44,50], each patch $\mathbf{x}_{ij}$ can be represented by the concatenation of pattern responses $[s_{ij1}, s_{ij2}, \ldots, s_{ijK}]^T \in \mathbb{R}^{K\times 1}$. In practice, an image is divided into some grids, i.e., some subregions using SPM [19]. A common way to aggregate the pattern responses in each grid $l \in \{1, 2, \ldots, L\}$ is max-pooling. That is, the representation of grid l in image X_i is $\mathbf{s}_i^l = [s_{i1}^l, s_{i2}^l, \ldots, s_{iK}^l]^T \in \mathbb{R}^{K\times 1}$, where $s_{ik}^l = max_{z_{i,j}\in l}\, s_{ijk}$ is the maximum response of pattern k in grid l. Then image X_i can be described as the concatenation of each grid representation $\mathbf{s}_i = [{\mathbf{s}_i^1}^T, {\mathbf{s}_i^2}^T, \ldots, {\mathbf{s}_i^L}^T]^T \in \mathbb{R}^{KL\times 1}$.

3.2 Formulation of shared compact deep discriminative patterns

Inputs to our learning system are n training images $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ and their labels $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_n\}$, $Y_i \in \{1, 2, \ldots, M\}$, where M is the number of image classes. Our objective is to learn the pattern filters $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$ and the final image-level classifier $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$, $u_m \in \mathbb{R}^{KL\times 1}$. After encoding the image X_i to $\mathbf{s}_i$, we can classify it via $Y_i = arg\, max_{m\in 1,2,\ldots,M}\, \mathbf{u}_m^T \mathbf{s}_i$.

Some work tries to jointly learn the pattern filters $\mathbf{W}$ and image-level classifier $\mathbf{U}$ [27,37], but as referred in Sect. 2, there are some limitations in their methods, which make their image representation not compact and discriminative enough, and not practical when applied on large dataset. To improve their work, we learn shared patterns with the lasso regularization [45] as in Eq. (2). Experimental results show that image representation constructed by these patterns is extreme compact and discriminative for final classification task.

$$\min_{\mathbf{W},\mathbf{U}} \lambda_u \sum_{i=1}^{M} \|\mathbf{u}_i\|_1 + \frac{\lambda_w}{2} \sum_{i=1}^{K} \|\mathbf{w}_i\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} L_{X_i}, \tag{2}$$

where $\mathbf{W}$, $\mathbf{U}$, $\mathbf{s}_i$, Y_i, M, K, and n are just as the definitions above. $\lambda_u \in \mathbb{R}$ and $\lambda_w \in \mathbb{R}$ are the regularization coefficients for $\mathbf{U}$ and $\mathbf{W}$ respectively.

The objective function Eq. (2) can be interpreted as follows. The second term is to avoid over-fitting. L_{X_i} in the third term is the image-level multiclass hinge loss by Crammer and Singer [5] as in Eq. (3), which minimizes the classification error of image X_i and generates discriminative patterns.

$$L_{X_i} = max\left(0, \left[1 - \left(\mathbf{u}_{Y_i}^T \mathbf{s}_i - \mathbf{u}_Y^T, \mathbf{s}_i\right)\right]\right) \tag{3}$$

where $Y = arg\,max_{j \in 1,2,\ldots,M, j \neq Y_i} \mathbf{u}_j^T \mathbf{s}_i$.

Note that Lobel et al. [27] and Parizi et al. [37] choose L2 regularization $\sum_{i=1}^{M} \|\mathbf{u}_i\|_2^2$ for image level classifier but the convex lasso regularization is chosen here. As we all know, the L2 regularization mainly focuses on avoiding the over-fitting, but the lasso which is a good approximation to the L0 regularization $\|\mathbf{u}_i\|_0$, can yield a sparse solution for matrix U, and avoid over-fitting in the meantime. Due to the fact that, in each image class, there are only a few important patterns, while the majority of them is redundant, the lasso can outperform the L2 regularization, as has been proven by NG [30].

The lasso regularization has the function of selecting the discriminative but shared patterns. Suppose $\mathbf{u}^j$ be the jth row of $\mathbf{U}$, i.e., $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M] = [\mathbf{u}^{1^T}, \mathbf{u}^{2^T}, \ldots, \mathbf{u}^{KL^T}]^T$, and the ith column jth row element in $\mathbf{U}$ be u_i^j. It is obvious that $\sum_{i=1}^{M} \|\mathbf{u}_i\|_1 = \sum_{j=1}^{KL} \|\mathbf{u}^j\|_1$, so this regularization is symmetric toward the two dimensions of $\mathbf{U}$. The term $\|\mathbf{u}_i\|_1$ encourages the sparsity of each column, which according with the phenomenon that only a few patterns are sufficient to represent the class i, so we adopt $\|u_i\|_1$ to select the most discriminative ones for class i. Meanwhile, the $\|\mathbf{u}^j\|_1$ ensures the sparsity of each row, which encourages the patterns shared among different classes, and control its number. There are some discriminative patterns shared among different classes, as we referred before, so we choose $\|\mathbf{u}^j\|_1$ to select these shared patterns, which can contribute to generate the extremely shared image representation. Furthermore, this constraint also makes sure that patterns shared among only a few classes to ensure the information amount. So the lasso rather than L2 regularization is more suitable to the traits mentioned above to select the most important patterns for each class.

Note that in a certain class, the Eq. (2) can not only learn the positive patterns but also negative patterns as the counter-evidence for the classification. For example, one would not expect to find a tree in the scene such as classroom and closet. We can observe that this goal is achieved by control the sign of u_i^j. That is, when $u_i^j > 0$, the pattern j should occur in the image class i; otherwise, when $u_i^j < 0$, the pattern j should not occur in the image class i, and when $u_i^j = 0$, the pattern j is trivial for class i.

3.3 Optimization method

The objective function Eq. (2) is non-convex so it is hard to optimize it directly. Inspired by the fact that when $\mathbf{W}$ is fixed, it descends to a typical convex SVM problem and can be solved directly (see Sect. 3.3.1); when $\mathbf{U}$ is fixed, it can be solved by some sophisticated methods (see Sect. 3.3.2). Therefore we choose the block coordinate descent [32] to learn

parameters through alternating between optimizing $\mathbf{U}$ when $\mathbf{W}$ is fixed, and optimizing $\mathbf{W}$ when $\mathbf{U}$ is fixed, as shown in Algorithm 1. When $\mathbf{W}$ is fixed, the Eq. (2) can be rewritten as the Eq. (4). When $\mathbf{U}$ is fixed, the Eq. (2) can be rewritten as Eq. (5). More details of the optimizing process can be seen in the following subsections.

$$L_U = \min_{\mathbf{U}} \lambda_u \sum_{i=1}^{M} \|\mathbf{u}_i\|_1 + \frac{1}{n} \sum_{i=1}^{n} max\left(0, [1 - \left(\mathbf{u}_{Y_i}^T \mathbf{s}_i - \mathbf{u}_Y^T \mathbf{s}_i\right)]\right), \tag{4}$$

$$L_W = \min_{\mathbf{W}} \frac{\lambda_w}{2} \sum_{i=1}^{K} \|\mathbf{w}_i\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} max\left(0, [1 - \left(\mathbf{u}_{Y_i}^T \mathbf{s}_i - \mathbf{u}_Y^T \mathbf{s}_i\right)]\right), \tag{5}$$

Algorithm 1 Learning Extremely Shared Middle-level Image Representation

Input: $\mathcal{X}, \mathcal{Y}, \lambda_u, \lambda_w$
Output: $\mathbf{W}$
1: Initialize pattern filters $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$, $\mathbf{w}_k \in \mathbb{R}^{d \times 1}$
2: **while** not convergence **do**
3: Fixing $\mathbf{W}$, optimizing $\mathbf{U} = \underset{\mathbf{U}}{arg\,min}\, L_U$ in Eq. (4)
4: Fixing $\mathbf{U}$, optimizing $\mathbf{W} = \underset{\mathbf{W}}{arg\,min}\, L_W$ in Eq. (5)
5: **end while**

3.3.1 Optimizing $\mathbf{U}$

The Eq. (4) is convex and differentiable to $\mathbf{U}$, so the optimizing scheme in the third line of Algorithm 1 can be the simple stochastic gradient descent (SGD). The partial derivative of Eq. (4) can be computed as

$$\frac{\partial L_U}{\partial \mathbf{u}_m} = \lambda_u \, sign(\mathbf{u}_m) + \frac{1}{n} \sum_{i=1}^{n} \frac{\partial L_{X_i}}{\partial \mathbf{u}_m}, \tag{6}$$

where the $sign(x)$ is the sign function which equals to 1 when $x > 0$ and -1 when $x < 0$. The $\partial L_{X_i} / \partial \mathbf{u}_m$ in the second term can be derived as follows,

$$\frac{\partial L_{X_i}}{\partial \mathbf{u}_m} = \begin{cases} -\mathbf{1}[\mathbf{u}_{Y_i}^T \mathbf{s}_i - \mathbf{u}_Y^T \mathbf{s}_i < 1]\mathbf{s}_i & \text{if } m = Y_i \\ \mathbf{1}[\mathbf{u}_{Y_i}^T \mathbf{s}_i - \mathbf{u}_Y^T \mathbf{s}_i < 1]\mathbf{s}_i & \text{if } m = Y \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathbf{1}[a < b]$ indicates whether $a < b$ is true and equals to 1 when a is smaller than b and 0 otherwise. Then given the learning rate η, the $\mathbf{u}_m$ can be optimized using $\mathbf{u}_{m_{t+1}} \leftarrow \mathbf{u}_{m_t} - \eta \partial L_{X_i} / \partial \mathbf{u}_{m_t}$.

3.3.2 Optimizing $\mathbf{W}$

To make the learning procedure of $\mathbf{W}$ easier to understand, we suppose there is no SPM i.e. $L = 1$, so we can rewrite $\mathbf{s}_i$ as $[s_{i1}, s_{i2}, \ldots, s_{iK}]$. Let $s_{ik} = s(X_i, \mathbf{w}_k, z_i^k)$ be the kth element of $\mathbf{s}_i$, where $z_i^k = \underset{z_{ij}}{arg\,max}\, \mathbf{w}_k^T \mathbf{x}_{ij}$ is the latent variable indicating the strongest response

location of pattern k. For Eq. (5), s_{ij} is convex in $\mathbf{W}$ as it is the maximum value of a linear function. But if $u^k_{Y_i} - u^k_Y > 0$, the $1 - (\mathbf{u}^T_{Y_i}\mathbf{s}_i - \mathbf{u}^T_Y\mathbf{s}_i)$ will be non-convex to $\mathbf{w}_k$. So the fourth line in Algorithm 1 refers to the non-convex optimization. Here we choose CCCP algorithm [52] to learn the $\mathbf{W}$.

As we can see, when $u^k_{Y_i} - u^k_Y < 0$, the L_{X_i} will be convex to $\mathbf{w}_k$, so we can optimize $\mathbf{w}_k$ directly. Meanwhile, when $u^k_{Y_i} - u^k_Y > 0$, if the latent variable z^k_i is given, the $1-(\mathbf{u}^T_{Y_i}\mathbf{s}_i - \mathbf{u}^T_Y\mathbf{s}_i)$ is just a linear function to $\mathbf{w}_k$, so L_{X_i} is also convex to $\mathbf{w}_k$. Inspired by this fact, we can alternately update latent variable z^k_i if $u^k_{Y_i} - u^k_Y > 0$ and learn the $\mathbf{W}$, as shown in Algorithm 2.

After z^k_i is given, learning $\mathbf{W}$ becomes a convex optimizing problem. As Eq. (5) is differentiable to $\mathbf{W}$, we can also choose SGD to learn $\mathbf{W}$. The partial derivative of Eq. (5) can be written as follows,

$$\frac{\partial L_W}{\partial \mathbf{w}_k} = \lambda_w \mathbf{w}_k + \frac{1}{n}\sum_{i=1}^{n}\frac{\partial L_{X_i}}{\partial \mathbf{w}_k}, \tag{8}$$

and the second term is derived as

$$\frac{\partial L_{X_{\mathrm{IM}_i}}}{\partial \mathbf{w}_k} = \mathbf{1}\left[\mathbf{u}^T_{Y_i}\mathbf{s}_i - \mathbf{u}^T_Y\mathbf{s}_i < 1\right]\left(u^k_Y - u^k_{Y_i}\right)\mathbf{x}^k_i, \tag{9}$$

where $\mathbf{x}^k_i = f(X_i, z^k_i)$ is the feature vector at location z^k_i in image X_i. As in Sect. 3.3.1, given the learning rate η, the $\mathbf{w}_k$ can be optimized using $\mathbf{w}_{k_{t+1}} \leftarrow \mathbf{w}_{k_t} - \eta\partial L_W/\partial \mathbf{w}_{k_t}$. The detailed pseudo-code of updating $\mathbf{W}$ is shown in Algorithm 2, where $\mathbf{W}^{old}$ is the pattern filters got from the later iteration, and T is the number of SGD iterations.

Algorithm 2 Updating $\mathbf{W}$ using CCCP and SGD

Input: $\mathcal{X}, \mathcal{Y}, \lambda_w, \mathbf{W}^{old}, T, \eta$
Output: $\mathbf{W}$
1: **while** not convergence **do**
2: Set $\mathbf{W}_1 = \mathbf{W}^{old}$
3: **for** $iter = 1$ **TO** T **do**
4: **if** $u^k_{Y_i} - u^k_Y > 0$ **then**
5: $z^k_i = \underset{z_{ij}}{arg\,max}\ \mathbf{w}^{old\,T}_k \mathbf{x}_{ij}$
6: **else**
7: $z^k_i = \underset{z_{ij}}{arg\,max}\ \mathbf{w}^T_{k_t}\mathbf{x}_{ij}$
8: **end if**
9: $\mathbf{w}_{k_{t+1}} \leftarrow (1-\eta\lambda_w)\mathbf{w}_{k_t} - \frac{\eta}{n}\sum_{i=1}^{n}\mathbf{1}[\mathbf{u}^T_{Y_i}\mathbf{s}_i - \mathbf{u}^T_Y\mathbf{s}_i < 1](u^k_Y - u^k_{Y_i})\mathbf{x}^k_i$
10: **end for**
11: $\mathbf{W}^{old} = \mathbf{W} = \mathbf{W}_{T+1}$
12: **end while**

Notice that there is only a little difference in the extension from no SPM to multilevel SPM, so we do not provide more details about the SPM form of learning $\mathbf{W}$. We should also note that even we get $\mathbf{U}$ via minimizing the classification error, we do not choose it as the final image classifier; instead, we retrain a multiclass linear SVM. The only purpose to introduce the $\mathbf{U}$ is to learn the discriminative and shared patterns to construct our extremely shared image representations. In the Sect. 4, we will show that the performance of the retrained SVM classifier is better than that constructed by learned $\mathbf{U}$ directly.

3.3.3 Pattern filters initialization

As described in the first line of Algorithm 1, we should first initialize the pattern filters $\mathbf{W}$. Here we choose an effective way to achieve this goal. Firstly, we use k-means on images in each class separately to assign every patch a cluster label. Then, a multiclass SVM by Crammer and Singer [5] can be trained using the cluster label as the supervised information. Here the SGD is also used for training the SVM due to its effective and efficient.

As we can see, this initialization strategy can be regarded as the weakly supervised BoF and can find special patterns in each class through the multiclass SVM.

4 Experiments

To evaluate our proposed method, we conduct some experiments on three well-known scene classification datasets, including 15 Scenes [19], MIT-indoor 67 [40] and SUN 397 [51]. Images in these datasets are collected from Google and Flickr. Some class-specific keywords are chosen to retrieve images on the search engine, and the collected images are measured by the Amazon's Mechanical Turk. Now all these three datasets are public available.[2] As the 15 scenes are quite a simple dataset, it is just a tentative experiment, then some detailed experiments are conducted on the MIT-indoor 67, and finally we do the experiment on a large scene dataset SUN 397.

4.1 Experimental setup

In the beginning of our experiments, we densely sample patches in four different scales from each image, with scale size $\{72\times72, 96\times96, 120\times120, 144\times144\}$ by every 32 pixels. Deep CNN features are chosen as the local descriptors. We select activations from the seventh layer (fc7) of the model trained by Chatfield et al. [3], with the Caffe library [16]. To make the training process faster and more tractable, PCA is introduced to reduce the dimension of local descriptors from the original 4096 to 256, along with the pipeline of whitening and L2-norm.

Apart from the local descriptors, other important experimental settings also deserve to be mentioned. As described in Sect. 3.1, the max-pooling with three level SPM (1×1, 2×2, 4×4) is adopted for image representation. The number of learned patterns is chosen by tested on MIT-indoor 67 dataset, and other parameters are chosen by cross-validation. In the SGD procedure, we set the initial learning rate equaling to 1 and after that divide the learning rate by 10 every T iterations. This operation is repeated five times so the final learning rate declined to 0.00001. In the initializing step, k-means is implemented by the VLFeat [48] library. Generating the final image classifier through SVM is implemented by the LibLinear [12]. All of the programs are written in MATLAB and running on an Inter(R) Xeon(R) i7-E5 2670 CPU (2.60GHz) 64GB RAM PC.[3]

4.2 15 Scenes

The 15-scene dataset includes 4485 scene images and 15 different classes in total. There are about 200–400 images in each class, with average size of 300 × 250. Following the

[2] 15 Scenes: http://www-cvr.ai.uiuc.edu/ponce_grp/data/scene_categories/. MIT-indoor 67: http://web.mit.edu/torralba/www/indoor.html. SUN 397: http://vision.princeton.edu/projects/2010/SUN/.

[3] The implementation code and trained models are available at https://github.com/hust-tp/ESMIR.

Table 1 Classification accuracy for different methods on 15 scenes

Methods	Accuracy (%)
Handcrafted features	
EMFS [43]	85.70
D-Parts [44]	86.0 ± 0.80
MMDL [50]	86.70 ± 0.40
Deep CNN features	
ImageNet fc7 + Linear SVM [53]	84.23 ± 0.37
Places fc7 + Linear SVM [53]	90.19 ± 0.34
Hybrid fc7 + Linear SVM [53]	91.59 ± 0.48
Ours (60 patterns)	$\mathbf{92.80 \pm 0.37}$

The best results are highlighted in bold

standard setup [19], we randomly select 100 images per class for training and the rests for testing. This operation is performed ten times and the average classification accuracy is reported.

For 15 scenes, the λ_w, λ_u are set to 0.1 and 5×10^{-6}, respectively, and 60 patterns (i.e., average four patterns per class) are learned in all. Results are reported in Table 1. As we can see, our method can achieve the mean accuracy of 92.80%, which is much higher than other methods with handcrafted features [43,44,50] or the deep CNN features [53], the latter of which even utilizes Hybrid deep CNN features trained by the combination of ImageNet and Places datasets. Notice that our method needs only 60 patterns, which is comparatively quite a small number.

4.3 MIT-indoor 67

The MIT-indoor 67 is a very popular scene classification benchmark. It contains 67 categories of indoor scenes and 15,620 images totally. Following the standard setup [40], the standard partition that separates 80 images for training and 20 images for testing per category is adopted in our experiment. T, the number of SGD iterations is set to 10,000, and the λ_w, λ_u are set to 0.1 and 5×10^{-7}, respectively, which are all chosen from cross-validation. The classification accuracy and detailed analyses is reported.

4.3.1 Impact of the number of patterns

The first set of experiments is designed to evaluate the impact of pattern numbers. As the red line in Fig. 2 shows, it takes only 268 patterns (i.e., average 4 patterns per class) to achieve 74.85% accuracy. However, when double the number of patterns (536 patterns), there is only 0.15% improvement (74.85–75.00%), which is not surprising because our formulation is trying to share some important patterns among different image classes. Many chosen patterns may be redundant when the number is increased. It demonstrates that simply adding patterns seems unhelpful to improve the performance, so only 268 patterns are necessary. To the best of our knowledge, the fewest patterns are required to achieve the excellent performance in the classification accuracy. 268 is a tiny number compared with that in other methods, such as many part-based models [10,17,44] using thousands even tens of thousands of patterns in total. So we can learn very compact representation after sharing some patterns.

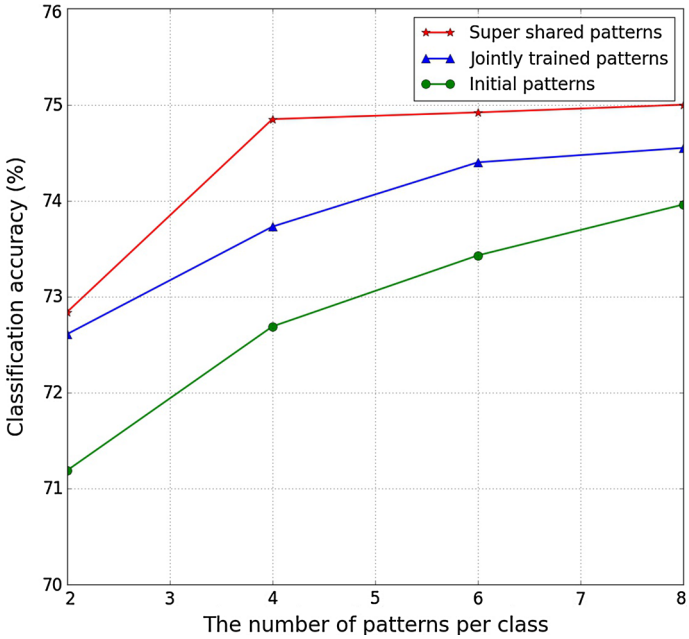


Fig. 2 Classification accuracy on MIT-indoor 67 over different number of patterns

4.3.2 Impact of the learning procedure

Then we compare the performance between the initial pattern filters and the final filters after the learning procedure. The green line in Fig. 2 is the classification accuracy with different pattern numbers. Actually our initializing strategy can be regarded as a weakly supervised BoF except that we replace the histogram-based method in BoF [6] with max-pooling based method. We can observe that the simple method to initialize patterns also has content performance, and after the learning procedure, we can get +2.16% improvement in performance (72.69–74.85%) using 268 patterns. So our method can greatly boost the discrimination level through selecting the discriminative patterns.

4.3.3 Comparing against jointly training

As referred in the last paragraph of Sect. 3, we can also utilize the learned **U** as the final image classifier, actually it can be regarded as the strategy of jointly training the pattern filters and constructing the image-level classifier. But we simply using learned **U** to sharing the patterns and retrain the classifier. To illustrate the advantage of our choice, we also compare our method with the jointly training method, as shown in the blue line in Fig. 2. It is obvious that our method can achieve higher accuracy than jointly training methods no matter whether the pattern number is 268 or 536. So we can learn more compact and discriminative representation after the learning procedure.

4.3.4 Comparison with other methods

To demonstrate the advancement of our method, some comparison experiments are also performed. We first compare our method with another shared representations method [37],

Table 2 Classification accuracy comparing with another shared representations method on MIT-indoor 67

Methods	Number of patterns	Accuracy (%)
Parizi et al. [37]	372	73.30
Parizi et al. [37]	13,400	77.10
Ours	268	74.85

Table 3 Classification accuracy for different methods on MIT-indoor 67

Methods	Accuracy (%)
Handcrafted features	
DPM + Gist-color + SP [34]	43.10
BoP [17]	46.10
miSVM [21]	46.40
EMFS [43]	48.20
Patches + GIST + SP + DPM [42]	49.40
MMDL [50]	50.15
D-Parts [44]	51.40
BoP + IFV [17]	63.10
LASC [22]	63.40
Doersch et al. [10]	64.03
Doersch et al. [10] + IFV	66.87
Deep CNN features	
ImageNet fc7 + TF-IDF (268 words)	49.85
ImageNet fc7 + Linear SVM [53]	56.79
Places fc7 + Linear SVM [53]	68.24
Hybrid fc7 + Linear SVM [53]	70.80
ImageNet fc6 + SC [24]	68.20
CL-45C [26]	68.80
OverFeat + SVM [41]	69.00
ImageNet fc7 + VLAD [14]	68.88
MDPM [23] (3350 patterns)	70.46
ImageNet fc8 + FV [9]	72.86
Parizi et al. [37] (372 parts)	73.30
Parizi et al. [37] (13,400 parts)	77.10
ImageNet fc8 + FV + Places fc7 [9]	**79.00**
Ours (268 patterns)	74.85

The best results are highlighted in bold

as shown in Table 2. We can observe that the method by Parizi et al. [37] outperforms our methods when they use 13,400 parts, but note that we only need 268 patterns, which are much less than theirs, and they make use of the Place CNN model trained on a large dataset which is particularly collected for scene classification [53]. Our method achieves higher accuracy than Parizi et al. [37] when they use 372 patterns (+1.55%). At the same time less patterns are necessary in our method (about 70% patterns of their method), which can demonstrate our method has the ability to learn more compact and discriminative representations.

More results of other methods are shown in Table 3. Firstly, our method outperforms the ones with handcrafted features to a large extent. Meanwhile, we also achieve better performance than the ones using ImageNet deep CNN features, like activations from the

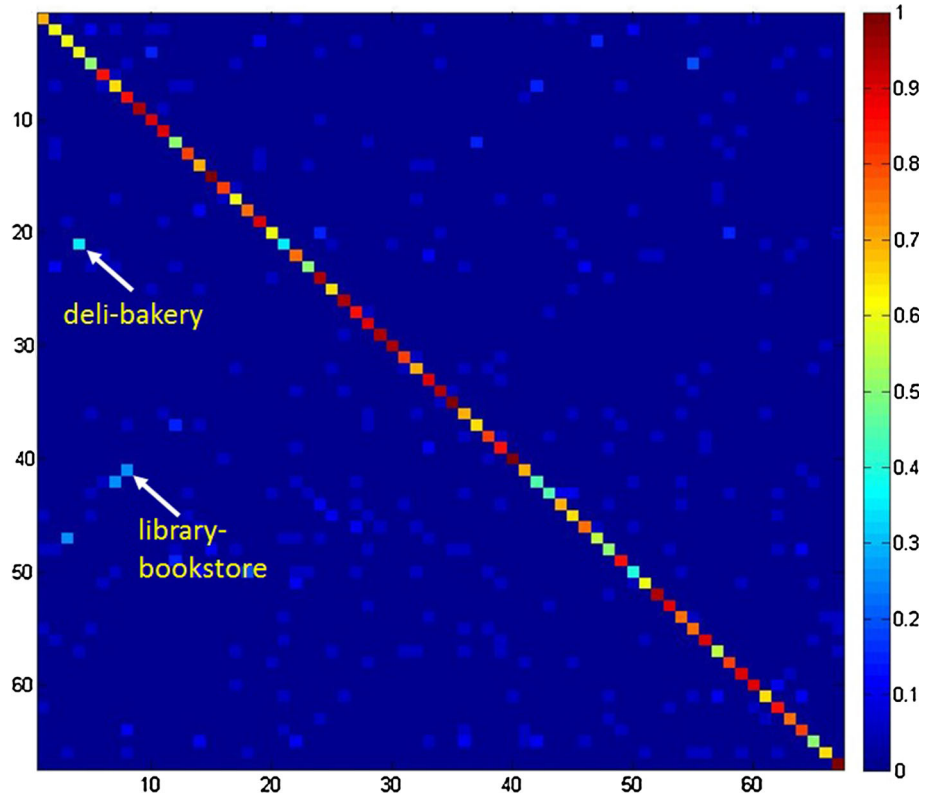


Fig. 3 The confusion matrix on MIT-indoor 67

ImageNet CNN convolutional layer with cross-convolutional-layer pooling [26], ImageNet fc6 with sparse coding [24], ImageNet fc7 with VLAD and Fisher Vector [14], ImageNet fc7 with pattern mining [23], and ImageNet fc8 with sematic Fisher Vector [9]. The result via concatenating semantic Fisher Vector with ImageNet fc8 and output from Place CNN fc7 is the best on MIT-indoor 67 [9]. But it also adopts the Place CNN features. We also compare our method with the Term Frequency–Inverse Document Frequency (TF-IDF) method, using the same number of patterns as our methods. It is obvious our method outperforms the TF-IDF method by a large margin (74.85 vs. 49.85). These results show that our method achieves very competitive results comparing with others.

Figure 3 is the confusion matrix of our method. As we can see, the incorrect classification made in our method (like differentiating bakery and deli in Fig. 4) are even unavoidable for human eyes.

4.3.5 Time costing for training and testing

The efficiency of our method can be demonstrated that, after the deep CNN features have been extracted, it only takes about 8 min to initialize four patterns in each class and 3 h for all of the learning procedure. The consuming time has positive linear correlation with the pattern number. In addition, if combined with the strategy in [4], which can accelerate the speed of extracting patches feature, our method could shorten the time costing further.

Fig. 4 Some pictures in bakery and deli

Table 4 Classification accuracy for different methods on SUN 397

Methods	Accuracy (%)
Handcrafted features	
Xiao et al. [51]	38.00
EMFS [43]	40.70
LASC [22]	45.30 ± 0.40
Deep CNN features	
ImageNet fc7 + Linear SVM [53]	42.61 ± 0.16
Places fc7 + Linear SVM [53]	54.32 ± 0.14
Hybrid fc7 + Linear SVM [53]	53.89 ± 0.21
ImageNet fc7 + VLAD [14]	51.98
ImageNet fc7 + FV [14]	53.00 ± 0.40
ImageNet fc8 + FV [9]	54.40 ± 0.30
ImageNet fc8 + FV + Places fc7 [9]	$\mathbf{61.72 \pm 0.13}$
Ours (1588 patterns)	56.57 ± 0.24

The best results are highlighted in bold

Moreover, for both training and testing, the procedure of generating the image representation only contains steps of calculating the dot product and taking the maximal value. It takes only 0.1 s to encode one image, which is so small that can even be neglected.

4.4 SUN 397

The SUN 397 is a very large dataset for scene classification. There are 397 different image classes in the dataset, including outdoor and indoor scenes, like alley, apartment building, hospital room, throne room, and so on. It totally contains more than 100 K images, and each class has at least 100 images. We follow the fixed ten different partition of training and testing

set by Xiao et al. [51], i.e., each class has 50 images for training and 50 for testing. The T is set to 20,000 and λ_w, λ_u are set to 0.1 and 5×10^{-8} respectively according to the cross-validation. According to the Sect. 4.3.1, we learn average 4 patterns per class (totally 1588 patterns).

The classification results are shown in Table 4. We can arrive at the same conclusion that our method can generate very compact image representation and achieve the best performance only using ImageNet deep CNN features. Moreover, our method is also capable of dealing with large datasets and performs well on these dataset.

5 Discussion

The results in Sect. 4 have shown many inspiring phenomenon. (1) Comparing with other pattern learning methods, our extremely shared strategy successfully produces very compact

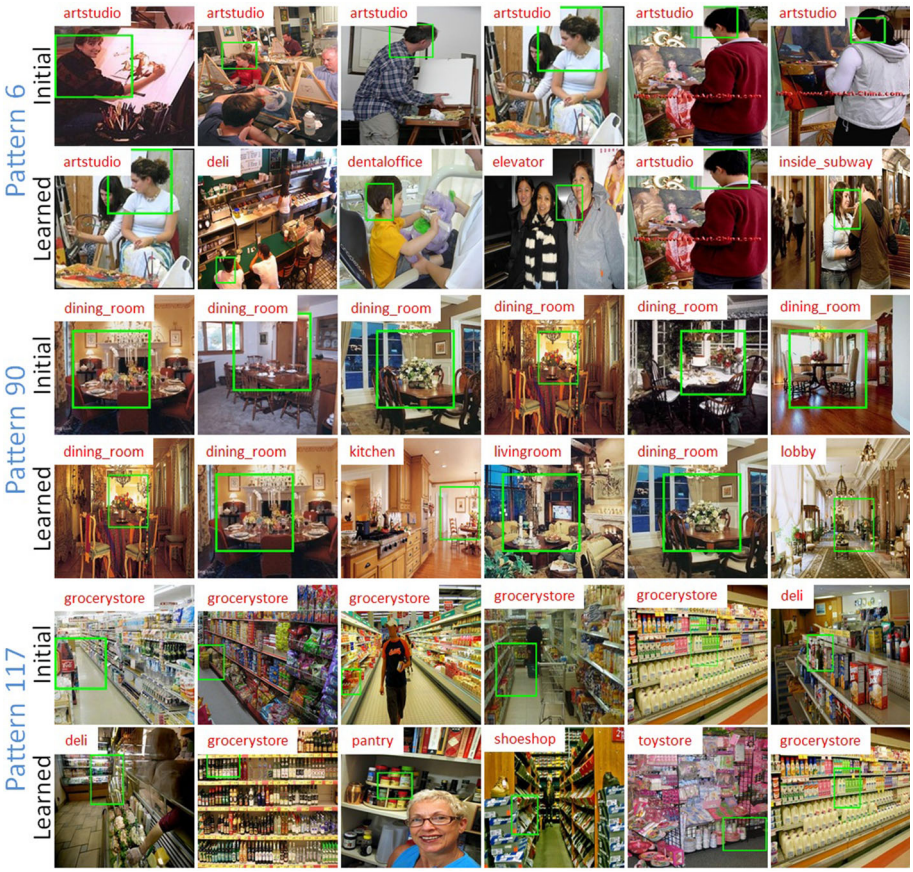


Fig. 5 The top detections in some patterns on the full MIT-indoor 67 training images. *Each row* represents patches in a certain pattern and the label of the image is marked on the *top left corner*. The "initial" means patterns generated by the initial step in Sect. 3.3.3, and the "learned" means patterns learned by our method. As we can observe, top detections in "initial" often belong to the images in one or two classes, and in "learned", they are shared among diverse images. For example, the pattern 117, with the semantic meaning of goods, initially concentrates on the grocerystore scene (*the fifth row*). After training, they are shared among the deli, grocerystore, pantry, shoeshop, and toystore scenes (*the sixth row*)

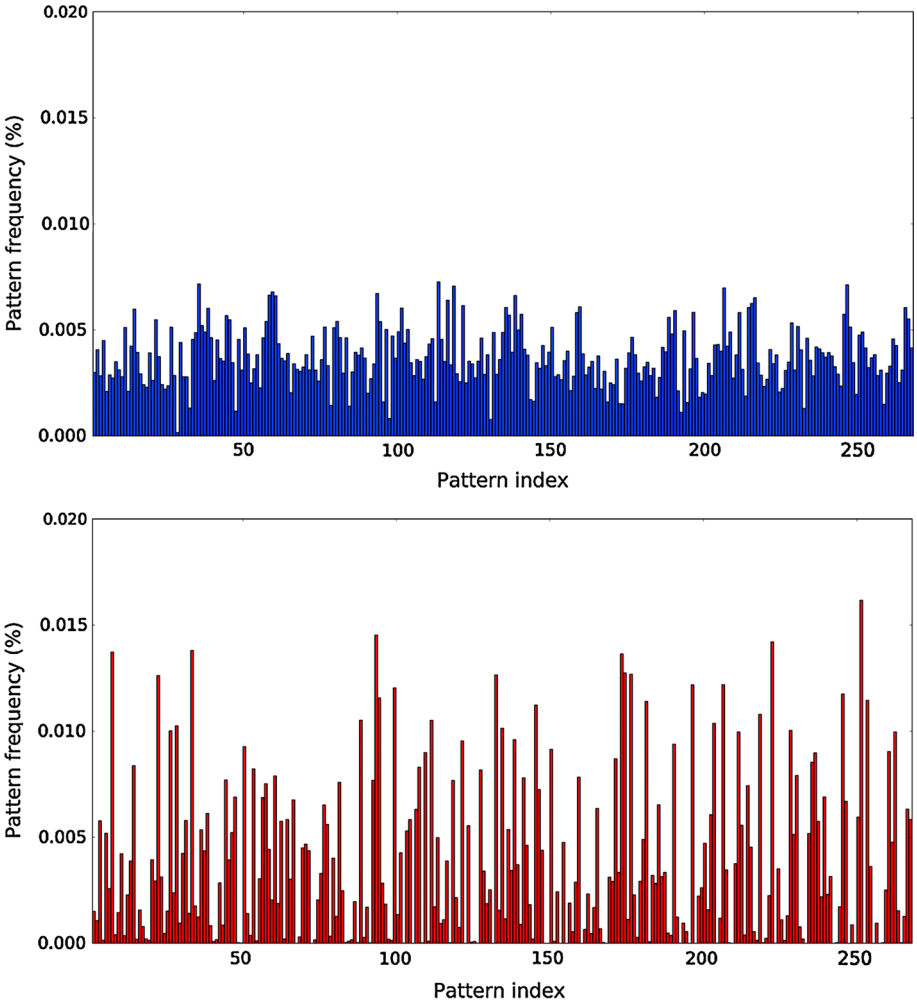


Fig. 6 The pattern frequencies before (*top*) and after (*bottom*) our learning strategy on MIT-indoor 67. The details of pattern frequencies are described in Sect. 5

representations, that is, only four patterns on the average are necessary for each class, which is the fewest patterns been used to our best knowledge. (2) The fewest pattern number does not harm our performance of scene classification. Our method outperforms other methods using ImageNet deep features, and is comparable with the methods using deep CNN models trained on large scene dataset. (3) Our method also outperforms the sophisticated Fisher Vector based methods, which achieve many the state-of-the-art performance in many applications. (4) Due to the very compact representation, our method is capable for dealing with large scene dataset like SUN 397 [51], which has not been tested by other pattern learning methods. All of them confirm that our method can generate very compact but discriminative representation for images.

Figure 5 shows the patches with the highest scores in some semantic patterns on the MIT-indoor 67 dataset. Previously, the most important patches are disturbed in the same

class. However, after the learning procedure of the extremely shared strategy, the important patches are dispersed among different classes, and meanwhile contain some semantic information, which demonstrated that our method can share the discriminative patterns to generate extremely compact image representation. For example, the pattern 117 (the fifth and sixth rows in Fig. 5) is concentrated in the scene grocerystore before training. After our learning procedure, it is shared among the deli, grocerystore, pantry, shoeshop and toystore scenes. At the same time, the learned patterns also convey much semantic information like the head (pattern 6), the tables (pattern 90), and the goods (pattern 117).

The pattern frequencies before and after our learning strategy on MIT-indoor 67 is shown in Fig. 6, where the frequencies are computed via dividing the occurrence number of each pattern by the number of patches. A pattern is occurred when the maximum response of a patch is corresponding to that pattern. We can observe that before learning, all pattern frequencies are located in the vicinity of a similar value, while after learning, some most discriminative patterns will be selected with greater frequencies, and what is more, the overall frequency distribution tends to be sparse. According to the pattern frequencies, we can compute the statistical significance of our results. Suppose the null hypothesis is that the pattern frequencies will not change after learning, and then the alternative hypothesis will be the pattern frequencies will change after learning. As a patch belonging to or not belonging to a pattern is similar to coin tossing with "head" or "tail," we can employ the Bernoulli distribution to compute the p value. According to these hypotheses, the p value infinitely close to 0, i.e., the probability of the null hypothesis being true is almost 0. We can also make a null hypothesis that the classification accuracy will not be improved by our learning strategy, and then the alternative hypothesis will be our learning strategy can improve the classification accuracy. Note that on MIT-indoor 67, as shown in Fig. 2, the accuracies before and after learning are 0.7269 and 0.7485, respectively. We can also use the Bernoulli distribution here, then the p value will be less than 0.1%, i.e., the probability of the null hypothesis being true is less than 0.1%. So we can reject these entire two null hypotheses. These statistical significance analyses show that our learning strategy can select some most discriminative patterns and, meanwhile, improve the results.

6 Conclusions

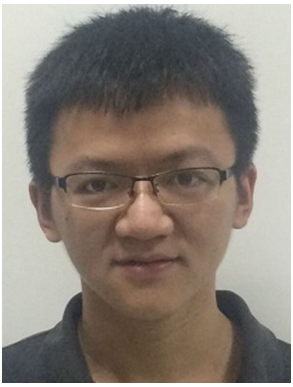
In this work, we propose a novel method to learn extremely shared middle-level image representation. The lasso regularization is adopted to enforce the pattern selection and sharing. Our extremely shared method can learn several discriminative patterns for different scene classes simultaneously and force them to be shared among different image classes. After the patterns are learned, we concatenate the scores of patterns, then use max-pooling to aggregate these scores into the final extremely compact image representation. Our method can achieve very remarkable scene classification performance. Only four patterns per class in average are required to represent images, and the performance of the learned patterns is very remarkable on the considered scene datasets. The code for reproducing the results is publicly available. For future work, we would like to explore more powerful methods to find extremely shared patterns which can generate more compact and discriminative way for image representation.

Acknowledgements We thank anonymous reviewers for their very useful comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant 61572207 and Grant 61503145, and the CAST Young Talent Supporting Program.

References

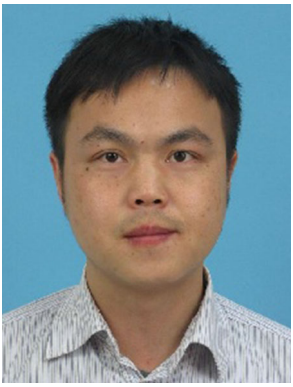
1. Argyriou A, Evgeniou T, Pontil M (2006) Multi-task feature learning. In: Proceedings of neural information processing systems, pp 41–48
2. Bourdev L, Malik J (2009) Poselets: body part detectors trained using 3d human pose annotations. In: Proceedings of international conference on computer vision, pp 1365–1372
3. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014) Return of the devil in the details: delving deep into convolutional nets. In: Proceedings of the British machine vision conference
4. Cimpoi M, Maji S, Vedaldi A (2015) Deep filter banks for texture recognition and segmentation. In: Proceedings of computer vision and pattern recognition, pp 3828–3836
5. Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2:265–292
6. Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Proceedings of workshop on statistical learning in computer vision, European conference on computer vision, pp 1–22
7. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of computer vision and pattern recognition, pp 886–893
8. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of computer vision and pattern recognition, pp 248–255
9. Dixit M, Chen S, Gao D, Rasiwasia N, Vasconcelos N (2015) Scene classification with semantic fisher vectors. In: Proceedings of computer vision and pattern recognition, pp 2974–2983
10. Doersch C, Gupta A, Efros AA (2013) Mid-level visual element discovery as discriminative mode seeking. In: Proceedings of neural information processing systems, pp 494–502
11. Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, New York
12. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
13. Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: Proceedings of computer vision and pattern recognition, pp 1778–1785
14. Gong Y, Wang L, Guo R, Lazebnik S (2014) Multi-scale orderless pooling of deep convolutional activation features. In: Proceedings of European conference on computer vision, pp 392–407
15. Hwang SJ, Sha F, Grauman K (2011) Sharing features between objects and their attributes. In: Proceedings of computer vision and pattern recognition, pp 1761–1768
16. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM international conference on multimedia, pp 675–678
17. Juneja M, Vedaldi A, Jawahar CV, Zisserman A (2013) Blocks that shout: Distinctive parts for scene classification. In: Proceedings of computer vision and pattern recognition, pp 923–930
18. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of neural information processing systems, pp 1097–1105
19. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of computer vision and pattern recognition, pp 2169–2178
20. Li LJ, Su H, Fei-Fei L, Xing EP (2010) Object bank: a high-level image representation for scene classification & semantic feature sparsification. In: Proceedings of neural information processing systems, pp 1378–1386
21. Li Q, Wu J, Tu Z (2013) Harvesting mid-level visual concepts from large-scale internet images. In: Proceedings of computer vision and pattern recognition, pp 851–858
22. Li P, Lu X, Wang Q (2015a) From dictionary of visual words to subspaces: locality-constrained affine subspace coding. In: Proceedings of computer vision and pattern recognition, pp 2348–2357
23. Li Y, Liu L, Shen C, van den Hengel A (2015b) Mid-level deep pattern mining. In: Proceedings of computer vision and pattern recognition, pp 971–980
24. Liu L, Wang L, Liu X (2011) In defense of soft-assignment coding. In: Proceedings of international conference on computer vision, pp 2486–2493
25. Liu L, Shen C, Wang L, van den Hengel A, Wang C (2014) Encoding high dimensional local features by sparse coding based fisher vectors. In: Proceedings of neural information processing systems, pp 1143–1151
26. Liu L, Shen C, van den Hengel A (2015) The treasure beneath convolutional layers: cross-convolutional-layer pooling for image classification. In: Proceedings of computer vision and pattern recognition, pp 4749–4757
27. Lobel H, Vidal R, Soto A (2013) Hierarchical joint max-margin learning of mid and top level representations for visual recognition. In: Proceedings of international conference on computer vision, pp 1697–1704

28. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
29. Neumann B, Möller R (2008) On scene interpretation with description logics. Image Vis Comput 26(1):82–101
30. NG AY (2004) Feature selection, L1 vs. L2 regularization, and rotational invariance. In: Proceedings of international conference on machine learning
31. Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of computer vision and pattern recognition, pp 1717–1724
32. Ortega JM, Rheinboldt WC (1970) Iterative solution of nonlinear equations in several variables. Academic Press, New York
33. Ott P, Everingham M (2011) Shared parts for deformable part-based models. In: Proceedings of computer vision and pattern recognition, pp 1513–1520
34. Pandey M, Lazebnik S (2011) Scene recognition and weakly supervised object localization with deformable part-based models. In: Proceedings of international conference on computer vision, pp 1307–1314
35. Parameswaran S, Weinberger KQ (2010) Large margin multi-task metric learning. In: Proceedings of neural information processing systems, pp. 1867–1875
36. Parikh D, Grauman K (2011) Relative attributes. In: Proceedings of international conference on computer vision, pp 503–510
37. Parizi SN, Vedaldi A, Zisserman A, Felzenszwalb P (2015) Automatic discovery and optimization of parts for image classification. In: Proceedings of international conference on learning representations
38. Pechyony D, Vapnik V (2010) On the theory of learning with privileged information. In: Proceedings of neural information processing systems, pp 1894–1902
39. Peraldi SE, Kaya A, Melzer S, Möller R, Wessel M (2007) Multimedia interpretation as abduction. In: Proceedings of the dl-2007: international workshop on description logics
40. Quattoni A, Torralba A (2009) Recognizing indoor scenes. In: Proceedings of computer vision and pattern recognition, pp 413–420
41. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of computer vision and pattern recognition workshop, pp 512–519
42. Singh S, Gupta A, Efros A (2012) Unsupervised discovery of mid-level discriminative patches. In: Proceedings of European conference on computer vision, pp 73–86
43. Song X, Jiang S, Herranz L (2015) Joint multi-feature spatial context for scene recognition in the semantic manifold. In: Proceedings of computer vision and pattern recognition, pp 1312–1320
44. Sun J, Ponce J (2013) Learning discriminative part detectors for image classification and cosegmentation. In: Proceedings of international conference on computer vision, pp 3400–3407
45. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Methodol) 58(1):267–288
46. Torralba A, Murphy KP, Freeman WT (2007) Sharing visual features for multiclass and multiview object detection. IEEE Trans Pattern Anal Mach Intell 29(5):854–869
47. VanGemert J, Veenman C, Smeulders A, Geusebroek J (2010) Visual word ambiguity. IEEE Trans Pattern Anal Mach Intell 32(7):1271–1283
48. Vedaldi A, Fulkerson B (2010) Vlfeat: an open and portable library of computer vision algorithms. In: Proceedings of Multimedia, pp 1469–1472
49. Wang G, Forsyth DA (2009) Joint learning of visual attributes, object classes and visual saliency. In: Proceedings of international conference on computer vision, pp 537–544
50. Wang X, Wang B, Bai X, Liu W, Tu Z (2013) Max-margin multiple-instance dictionary learning. In: Proceedings of the international conference on machine learning, pp 846–854
51. Xiao J, Hays J, Ehinger KA, Oliva A, Torralba A (2010) Sun database: large-scale scene recognition from abbey to zoo. In: Proceedings of computer vision and pattern recognition, pp 3485–3492
52. Yuille AL, Rangarajan A (2003) The concave–convex procedure. Neural Comput 15(4):915–936
53. Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: Proceedings of neural information processing systems, pp 487–495

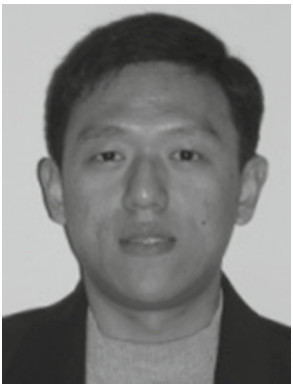


Peng Tang is a Ph.D. student in the School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST). He received his B.S. degree from HUST in 2010. He is a reviewer of neurocomputing. His research interests include computer vision and machine learning. In particular, he focuses on mid-level representation for image understanding.

Jin Zhang is currently a master student at the Electrical and Computer Engineering Department in Carnegie Mellon University, and received the B.S. Degree in Huazhong University of Science and Technology in 2016. Her research interest in computer vision focus on scene classification and object detection.



Xinggang Wang is an assistant professor of School of Electronics Information and Communications of Huazhong University of Science and Technology (HUST). He received his B.S. degree in communication and information system and Ph.D. degree in computer vision both from HUST. From May 2010 to July 2011, he was with the Department of Computer and Information Science, Temple University, Philadelphia, PA., as a visiting scholar. From February 2013 to September 2013, he was with the University of California, Los Angeles, as a visiting graduate researcher. He is a reviewer of IEEE Transaction on Cybernetics, pattern recognition, computer vision and image understanding, n, CVPR, ICCV and ECCV etc. His research interests include computer vision and machine learning.



Bin Feng received the B.S. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2001 and 2006, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include computer vision and intelligent video analysis.

Fabio Roli received his M.S. degree, with honours, and Ph.D. degree in Electronic Engineering from the University of Genoa, Italy. He was adjunct professor at the University of Trento, Italy, in 1993 and 1994. In 1995, he joined the Dept. of Electrical and Electronic Engineering of the University of Cagliari, Italy, where he is now professor of computer engineering and Director of the research lab on pattern recognition and applications (http://pralab.diee.unica.it). Dr. Rolis research activity is focused on the design of pattern recognition systems and their applications to biometric personal identification, multimedia text categorization, and computer security. He is a member of the governing boards of the International Association for Pattern Recognition and of the IEEE Systems, Man and Cybernetics Society. He is Fellow of the IEEE, and Fellow of the International Association for Pattern Recognition.

Wenyu Liu received the B.S. degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees, both in Electronics and Information Engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the School of Electronic Information and Communications, HUST. His current research areas include computer vision, multimedia, and machine learning. He is a senior member of IEEE.