International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

Contour Detection-based Discovery of Mid-level Discriminative Patches for Scene Classification

Regular Paper

Jinfu Yang[1]*, Jizhao Zhang[1], Guanghui Wang[2] and Mingai Li[1]

1 Department of Control Science and Engineering, Beijing University of Technology, Beijing, P.R. China
2 Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, USA
*Corresponding author(s) E-mail: casiayang@gmail.com

Received 18 October 2015; Accepted 18 January 2016

DOI: 10.5772/62266

Abstract

Feature extraction and representation is a key step in scene classification. In this paper, a contour detection-based mid-level features learning method is proposed for scene classification. First, a sketch tokens-based contour detection scheme is proposed to initialize seed blocks for learning mid-level patches and the patches with more contour pixels are selected as seed blocks. The procedure is demonstrated to be helpful for scene classification. Next, the seed blocks are employed to train an exemplar SVM to discover other similar occurrences and an entropy-rank criterion is utilized to mine the discriminative patches. Finally, scene categories are identified by matching the discriminative patches and testing images. Extensive experiments on the MIT Indoor-67 dataset, the 15-scene dataset and the UIUC-sports dataset show that the proposed approach yields better performance than other state-of-the-art counterparts.

Keywords Mid-level Feature, Scene Classification, Sketch Tokens, Contour Detection

1. Introduction

Scene classification is an important and challenging task for robots to understand the content of images in computer vision, while feature extraction and representation is a fundamental step in scene classification. Human beings, when seeing a picture, can extract the context information between global features and local features, which is helpful for inferring the place they are looking at. For example, the place with lots of clothes is probably a wardrobe; while a computer is unlikely to be found in a kitchen. To make a computer understand an image efficiently like a human being, the ability to capture and analyse such context information should be embedded in a computer vision system.

Generally speaking, image representation can be categorized into three levels: low-level, mid-level and high-level. In low-level representation, great progress has been achieved with the invention of local invariant descriptors, such as SIFT [1], SURF [2] and ORB [3]. For instance, *Nister et al.* [4] proposed an extremely efficient image retrieval method based on SIFT features. It takes only around 0.2 seconds to extract features on a 640 × 480 frame and the database query takes only 25ms on a dataset with 50,000 images. However, low-level features based methods cannot provide sufficient semantic information for scene recognition, since they depend mainly on corner points.

In order to extract semantic information, high-level features based on object detection have been introduced to

image representation in recent years. In [5], *Dalal et al*. proposed a gradient descriptor, called histogram of oriented gradient (HOG), which has been proved to be an effective pedestrian detection descriptor. The HOG models of pedestrians were trained in a sliding window framework to represent the global shape features of pedestrians. *Felzenszwalb et al.* [6] extended the HOG model to a deformable part model (DPM) by adding object parts. It has become a state-of-the-art object detection framework for multiple classes. Such high-level representation has also been applied to scene classification. In [7], *Li et al*. presented an object bank (OB) method to encode the object appearance and spatial location information in images. Thus, an image can be represented as objects appearing in it. *Zhu et al*. [8] proposed an upstream model that modelled the interaction of objects and scene topics jointly. Since the number of trained object classes is small, an image could not be well represented using the model. *Pandey and Lazebnik* [9] applied the standard DPM model to scene classification rather than object detection. In their study, the DPM model can capture recurring visual elements and salient objects in different scenes. By integrating the standard global image features, the method obtained a good performance in terms of classification accuracy on the MIT Indoor-67 dataset. However, the issues of object part initialization and learning were not addressed in the article. Moreover, since the labels of the objects are necessary for all high-level feature based methods, the models have to be retrained for reorganization of a new object class.

Compared with the low-level and high-level representation of images, mid-level representation has attracted a lot of attention, since it is more flexible and powerful for visual recognition. Mid-level representation is more adaptive to appearance distributions in the real world than the low-level features; on the other hand, it does not require the semantic grounding of high-level entities. In [10], *Vogel and Schiele* proposed a model to detect a set of visual concepts locally over image regions; in this method, images were represented by the frequency of the detected local concepts. In [11], a method that discovered a set of discriminative image patches without any supervision was proposed by *Singh et al*. Some mid-level patch candidates were chosen by an SVM; these candidates were further refined according to purity and discriminativeness. In addition, *Singh et al.* [11] also proposed a notion of "doublet", which denotes a combination of two noticeable mid-level patches satisfied with a certain spatial relationship. The "doublet" helps to effectively represent the context of an image. The method obtained promising classification results on the MIT Indoor-67 dataset. *Carl et al.* [12] proposed a discriminative clustering approach to discover geographically representative image elements automatically from Google Street View imagery. They demonstrated that these elements were visually interpretable and perceptually geo-informative. In their work, a large number of randomly sampled candidate elements are initialized; those candidate elements with too many neighbours in the negative training dataset are then rejected; finally, clusters are gradually built by applying iterative discriminative learning to each candidate. *Mittelman et al.* [13] proposed a weakly supervised approach to learn mid-level features, where only class-level supervision is provided during training. They developed a novel extension of the restricted Boltzmann machine (RBM) by incorporating a Beta-Bernoulli process factor potential into hidden units. The method uses the class labels to promote category-dependent sharing of learned features, which tends to improve the generalization performance. *Juneja et al.* [14] proposed a simple, efficient and effective method to discover discriminative image patches (or parts) for scene classification. Starting from a single patch occurrence, or seed block, the additional patches are incrementally discovered by training SVMs to find the occurrences of the seed block. During learning patches, a superpixels-based segmentation method [15] was employed to initialize the locations of seed blocks. A seed block is initialized for each superpixel by centring a 64×64 pixels block at the centre of the superpixel whose area is in the range of 500 to 1,500 pixels. However, the initialization strategy may lose some significant mid-level features since the constraint is rigid.

In this paper, we propose a new mid-level patch learning method for scene recognition based on contour detection. First, in order to obtain sufficient candidate patches, a contour detection algorithm is applied to image preprocessing, which detects the occurrences of the contours in the images in a sliding window framework. The seed blocks are then initialized by selecting the image patches containing more contour pixels. Next, these seed blocks are used to build models of image regions to find more patch occurrences in the training data. After that, an entropy-rank evaluation criterion [14] is employed to select discriminative patches from all the candidates discovered above. Finally, the discriminative patches are utilized to categorize scenes by matching test images. The main contributions of this paper are as follows:

1. The proposed learning method with little supervision can automatically discover image patches with semantic information, which is helpful for improving scene recognition performance.
2. The proposed method of patch learning based on contour detection performs better than other counterparts.
3. The proposed method based on mid-level features obtains better performance than other methods in terms of recognition accuracy.

The rest of the paper is organized as follows: the contour detection-based learning method of discriminative patches for scene classification is presented in Section 2. In Section 3, a novel patches-based scene classification method is described. The experimental results and discussions are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Learning Discriminative Patches

Some discriminative objects usually present in specific scenes. For instance, computers typically appear in computer rooms; pots and pans usually stay in kitchens; books are generally the major feature in libraries. Thus, by seeing some discriminative objects in a scene, a person can determine what type of scene he/she is possibly or impossibly in. Compared with low-level features, mid-level features can provide more abstract semantic information, which is helpful in identifying the scenes. Mid-level features usually include some structures of visual fragments, such as corners of a room, a part of a potted plant, etc.

In this section, we introduce a patch learning algorithm to automatically discover the discriminative patches in the images. The learning approach consists of three procedures: initialization, learning and discovery. Since the initialization is a prerequisite for the latter two procedures, it is crucial to generate candidate image patches (i.e., seed blocks) to train the learning models. In this paper, a contour detection-based initialization approach is proposed to generate the seeds and the patches with more contour points are chosen to be seed blocks.

2.1 Contour detection

Inspired by [16], we introduce a contour detection algorithm based on sketch tokens in the initializing procedure of patch learning. Sketch tokens are mid-level representations of images that are learned from image patches containing hand-drawn contours. In the training step, patches with human-generated contour pixels are clustered to form sketch tokens by using a random forest classifier. In the contour detection step, when a patch is classified into a class of sketch tokens, the centre pixel of this patch is regarded as a contour pixel. In this paper, the patches with more contour points are regarded as seed blocks, which are then used to find discriminative patches.

2.1.1 Definition of sketch tokens

The goal of defining sketch tokens is to represent the variety of image local edge structures, such as straight lines, curves, corners, t-junctions and y-junctions. Human-generated image contours in [16] are adopted to discover and define the sketch tokens. The image dataset for learning sketch tokens was built by *David Martin et al.* [17]. In the dataset, there are a set of images I and a corresponding set of hand-drawn binary contour images S.

The sketch tokens are defined as the clusters of the patches sampled from the binary image set S, which satisfy the following two conditions: first, that all of the patches have a fixed size of 35 × 35 pixels; second, that these patches contain labelled contours at their centre pixels. The clustering is performed on the features of their corresponding patches in images I using a K -means algorithm. Fig. 1 shows some examples of sketch tokens. The details of feature extraction and classification are discussed below.

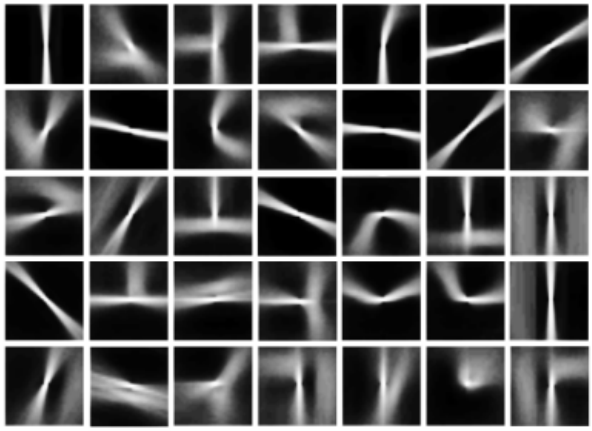


Figure 1. The average values of sketch token clusters

2.1.2 Feature extraction and classification

Given the notation of sketch tokens, their occurrences in every location of the input images can be detected. If a patch of the input images is considered to be a kind of sketch token, the centre pixel of the patch is selected as a contour pixel. As described in [18], the features are grouped into two types. One is where the features are directly indexed into multiple channels and the other is the self-similarity features. The channels of patches in images I consist of colour, gradient magnitude and oriented gradient. The colour channels are computed in the CIE-LUV colour space. The gradient magnitude channels are calculated using Gaussian blur with σ =0, 1.5 and 5 respectively. In the gradient magnitude channels with σ =0 and σ =1.5, the oriented gradients are quantified into four channels (voting into four orientation bins in 0 ~360) respectively. Therefore, there are three colour channels, three gradient magnitude channels and eight oriented gradient channels. The pixels in these channels are regarded as the first type of feature. The second type of feature is based on self-similarity [19]. The aim of this type of feature is to find out the portions of an image patch that contains texture boundaries. To compute this type of feature, a 35 × 35 patch is divided into 5 × 5 cells, where each cell is a region with 7 × 7 pixels. For the cells i and j in channel k, the self-similarity feature f_{ijk} is defined as

$$f_{ijk} = s_{jk} - s_{ik} \tag{1}$$

where s_{jk} is the sum of cell j in channel k and s_{ik} is the sum of cell i in channel k. Fig. 2 shows the distances in a colour channel. There are 25 L1 distances $\sum_k | f_{ijk} |$ in a colour channel between one cell with a yellow box and the other cells in the range of the original patch.

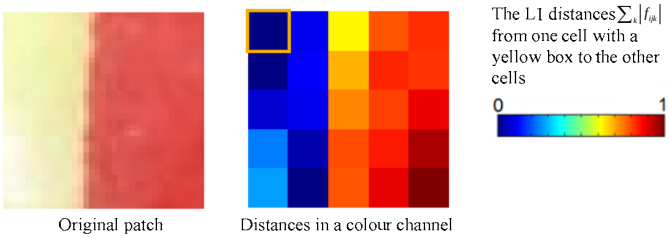


Figure 2. Illustration of distances in a colour channel. The left is the original patch and the right is the L1 distances $\sum_k |f_{ijk}|$ from one cell with a yellow box to the other cells.

Since there are multiple classes of sketch tokens, a random forest is used to categorize their types. Similar to [16], 150,000 contour patches (1,000 per token class) and 160,000 non-contour patches (800 per training image) are randomly sampled to train each tree. For each tree, the Gini impurity measure is employed to select a feature and decision boundary for each branch node from possible features. If there are m classes of sketch tokens and one non-contour class in the tree, the Gini impurity index for a feature is defined as

$$I = 1 - \sum_{j=0}^{m} p_j^2 \tag{2}$$

where p_j is the probability of a feature that belongs to the class j.

2.1.3 Contour detection

As described above, the features of all 35 × 35 patches in the images are classified using the random forest method. The outputs of these patches are the probabilities of sketch tokens' classes (including non-sketch token classes). The tokens label of a patch is assigned to the sketch token class that holds the highest probability value. Since all the centre pixels of sketch tokens are contour pixels, the centre pixel of a patch categorized to a sketch token is considered as a contour point.

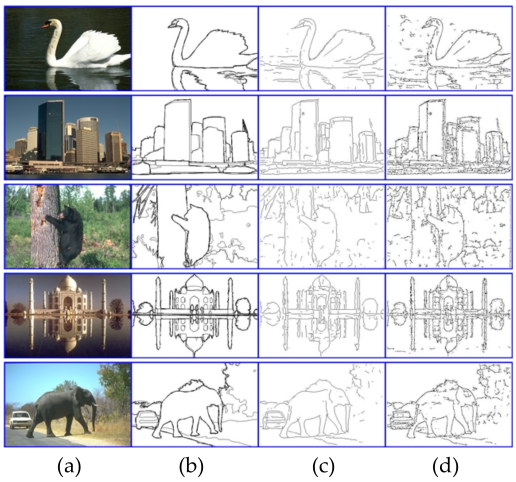


Figure 3. Example results of contours. The sketch tokens approach captures more details, such as the structure of the building on the second row. (a) Original (b) Ground truth (c) SCG [20] (d) Sketch tokens [16].

Let t_{ij} be the probability of patch x_i belonging to token j and t_{i0} be the probability of patch x_i belonging to the "no contour" class; the estimated probability of the patch centre containing a contour is

$$e_i = \sum_j t_{ij} = 1 - t_{i0} \tag{3}$$

If there are n trees in the random forest, the probability of the patch x_i belonging to the token j produced by the k-th decision tree is defined as

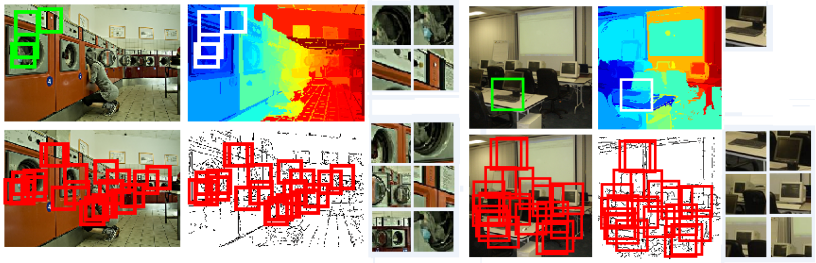
$$t_{ij} = \frac{1}{n} \sum_{k=1}^{n} p\left(C_{ij} \mid T_k\right) \tag{4}$$

In this way, all pixels in an image can be classified as contour points or non-contour points. In this paper, 25 trees are trained until every leaf node is pure or contains five examples or less. All of the features and parameters of the random forest are performed on the popular Berkeley segmentation dataset and benchmark (BSDS500) [17]. We define edge strength according to equation (3) for sketch tokens. Fig. 3 shows some qualitative comparison results of the contours. It is evident that the sketch tokens method captures more details, such as the structure of the building on the second row, than the SCG approach [20].

2.2 Learning discriminative patches

2.2.1 Initialization

In this paper, a pixel is considered as belonging to a contour when the probability e_i in equation (3) is larger than a threshold T (0.7 in our experiments). The seed blocks of the training images can be generated using the contour detection approach. First, for each image in the training dataset, we randomly sample S image patches with 64 × 64 pixels. The pixels of each patch are then categorized into contour pixels and non-contour pixels by utilizing the contour detection method discussed above. As a result, the top L (20 in our experiments) patches with the most contour pixels are chosen to be the seed blocks. The seed blocks initializing algorithm is shown in Algorithm 1, where the parameter *thres* stands for the threshold of sampling number, which is defined as 100 in this paper, and NUM stands for the total number of training images.



(a) launderette (b) computer room

Figure 4. Examples of initializing seed blocks. The top row is the results of [14]; the second row is the results of our method. In each figure (a or b), the middle column shows the corresponding superpixel image and contour pixels image respectively; the right column shows a subset of the seed blocks.

Algorithm 1. Seed blocks initialization

set the values of threshold T, sampling number S, sampling number threshold $thres$, the total number of training images NUM and seed number L;

for all training images NUM **do**

  while $S < thres$ **do**

    randomly sample a patch P_S with 64×64 pixels in each image;

    calculate the probability pr of each pixel in P_S using equation (3);

      if $pr > T$

        assign the pixel to contour;

        otherwise

        assign the pixel to non-contour;

    end if

    S = S+1;

  end while

  calculate the number of contour pixels in each patch P_S of each image;

  select top L patches with the most contour pixels as seed blocks for each image;

end for

get total $NUM \times L$ seed blocks

Fig. 4 shows some examples of initializing seed blocks. The top row in Fig. 4 is the results of [14]; the second row is the results of our method. From Fig. 4, we can see that the proposed approach obtains more seed blocks than [14]. More seed blocks are helpful to effectively discover occurrences of the seed blocks.

2.2.2 *Learning patches*

Given the seed blocks, other similar occurrences of each seed block can be discovered. An exemplar SVM [21] is applied to induce a further expansion process to make more

(a) a seed block (b) some expanded occurrences

Figure 5. The left is a seed instance. The right is some expanded example patches.

variable occurrences for each seed blocks. First, each seed block and a certain number of randomly sampled negative instances from the training images are represented as HOG features. The exemplar SVM is then trained by optimizing a convex objective to discover the occurrences of the seed block in the entire training dataset using a slide window strategy. The convex objective of the exemplar SVM can be represented as

$$\Omega_E(w,b) = \|w\|^2 + C_1 h\left(w^T x_E + b\right) + C_2 \sum_{x \in N_E} h\left(-w^T x - b\right) \quad (5)$$

where $h(x)=\max(0, 1-x)$ is the hinge loss function; x_E denotes the HOG template of a positive example; x denotes a HOG template of a negative example; N_E represents negative windows from other classes of scenes; and C_1 *and* C_2 are the loss penalty coefficients for positive and negative samples. Finally, all the obtained occurrences, including the seed blocks, are regarded as candidate discriminative patches for further mining in subsequent processing. Fig. 5 illustrates an example of a seed block and its expanded occurrences. In contrast to the method in [14],

where a standard SVM is trained iteratively to find enough occurrences, the proposed method, thanks to sufficient initialized seed blocks, avoids the time-consuming process of iterations.

2.2.3 Discovery

In this paper, a discriminative patch refers to one that appears frequently in many images of one class but seldom in the other classes. For example, a wheel is a discriminative patch in both bike class and car class. On the contrary, a patch appearing frequently in many classes may indicate that the patch is not discriminative, such as a featureless wall. In this section, a criterion, named entropy-rank (ER) curve [14], is employed to evaluate which patch is most discriminative. Specifically, each patch is evaluated in a sliding window framework on every validation image. For one patch, assuming that its occurrences in the validation dataset correspond to a set of vectors (z_i, y_i), z_i denotes the detection score of the i-th occurrence and y_i stands for the label of the i-th occurrence's scene class. All patches are sorted on their score z and the top r ranking patches are selected. The entropy $H(Y \mid r)$ is then defined as

$$H(Y \mid r) = -\sum_{y=1}^{N} p(y \mid r) \log_2 p(y \mid r) \tag{6}$$

where N is the number of scene classes and $p(y \mid r)$ is the probability of the top r patches (z_i, y_i) with label $y_i = y$.

Based on the definition, an entropy-rank curve is plotted, where the x-axis is the value r and the y-axis is the corresponding entropy $H(Y \mid r)$. According to the definition of entropy, the patches whose occurrences' distribution concentrates in fewer classes have lower entropy. Fig. 6 shows two entropy-rank curves, where one stands for discriminative patches and the other for non-discriminative patches. The discriminative performance is reflected by the area under curve (AUC). Since the AUC of the red curve is less than that of the blue one, the image patches of the red one have lower average entropy. As a result, the top discriminative patches can be selected from the candidate patches. The discriminative patches are helpful in scene classification since they contain more semantic information. Fig. 7 shows some examples of the discriminative patches and non-discriminative patches discovered from the corridor images.

3. Scene Classification Based on Discriminative Patches

Since the discriminative patches discovered above provide semantic information for scene recognition, we employ them to perform scene classification. First, a set number of image patches with 64 × 64 pixels are randomly sampled from each training image. The contour pixels of the sampled patches are detected based on sketch tokens. Therefore, the pixels of the sampled patches are classified

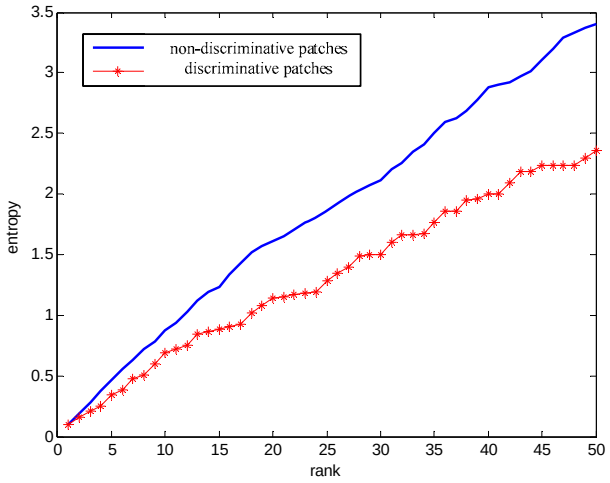


Figure 6. The red curve has low entropy at each value r, which indicates that the occurrences are mined from a few classes. The blue curve has more uniform entropy, which shows that the occurrences are discovered from many classes to make the patches less discriminative.

(a) Some examples of discriminative patches. It can be seen that they are the patches of corridor images.

(b) Some examples of non-discriminative patches. It is hard to tell which scene class they are most likely from.

Figure 7. Some examples of discriminative and non-discriminative patches

into contour pixels and non-contour pixels. The patches with the most contour pixels are chosen to be seed blocks. Each seed block and negative instances sampled randomly from the training images are then represented as HOG features to train an exemplar SVM by optimizing a convex objective. The occurrences of the seed block in the entire training dataset can be discovered by the trained exemplar SVM in a sliding window framework. All seed blocks and their corresponding occurrences are regarded as candidate discriminative patches. Next, the discriminative patches are further mined from candidate patches using a certain criterion. A patch with lower AUC will be considered to be discriminative. Finally, the top m discriminative patches of each category are utilized to recognize the scenes by matching with the testing images in a sliding window framework. Specifically, for a scene class c, the matching score of this class to the test image is defined as

$$score_c = \sum_{i=1}^{m} \left(\sum_{p_i=1}^{W-w_i+1} cos(HOG_{p_i}, hog_{p_i}) \right) \tag{7}$$

where p_i is the x-coordinate of the top-left pixel of a matched patch in the test image; W is the width of the test image, with w_i the width of the *i-th* discriminative patch; HOG_{P_i} denotes the HOG vector of the matched patch in the test image; and hog_{p_i} represents the HOG vector of the *i-th* discriminative patch. In equation (7), the cosine function is used to calculate similarity. Since every sampled patch has 64 × 64 pixels in our experiments, so w_i equals 64 and the sliding step is eight. Among all scene categories, the test image is classified into the one that gets the highest matching score. The proposed method based on the discriminative patches for scene classification is shown in Algorithm 2.

Algorithm 2. Discriminative patches-based scene classification

Step 1. Set the values of threshold T, sampling number S and seed number L; set the value of top discriminative patches m.

Step 2. Initialize seed blocks using *Algorithm 1* and obtain $NUM \times L$ seed blocks.

Step 3. Discover occurrences of the seed blocks.

for each seed block **do**

randomly sample a certain amount of negative instances (10 in our experiments);

represent the seed block and negative instances as HOG features;

train an exemplar SVM to discover the occurrences of the seed block in the entire training dataset;

end for

Step 4. Further mine the discriminative patches.

for all seed blocks and their occurrences **do**

calculate the entropy of each patch;

select m patches with the lowest AUC as discriminative patches of each class according to the ER curve;

end for

Step 5. Identify scene categories

for all testing images **do**

for m discriminative patches of each scene **do**

match with each testing image in a sliding window framework;

calculate the matching scores of each scene;

end for

Recognize scenes according to matching scores

end for

4. Experimental Evaluations

The proposed approach is evaluated using three popular datasets: MIT Indoor-67 dataset [22], 15-scene dataset [23] and UIUC-sports dataset [24].

4.1 MIT Indoor-67 dataset

The MIT Indoor-67 dataset [22], containing 67 indoor scene categories, is largely divided into shops, home, public spaces, leisure and work. Evaluation uses the protocol of [22], where each category has 64 training images, 16 validation images and 20 test images. Performance is reported in terms of average classification accuracy (Acc) and mean average precision (mAP) as described in [22]. The average classification accuracy is calculated as the mean over the diagonal values of classification confusion matrix. The advantage of this method is that it is less sensitive to unbalanced distributions of classes.

When categorizing scenes, we perform the experiments with different numbers of patches per category for matching. The results of the recognition rate are shown in Table 1; from which we can see that the highest accuracy is achieved when 50 patches per class are selected. Fig. 8 shows the corresponding curve of the recognition results. In the rest of this paper, we select 50 patches per category in our experiments.

Method	Number of parts selected per class			
	10	20	30	40
BoP [14]	42.34	44.81	44.96	46.00
Proposed	**43.28**	**45.30**	**45.90**	**47.69**
Method	Number of parts selected per class			
	50	60	70	80
BoP [14]	46.10	45.75	45.15	45.07
Proposed	**47.84**	**47.54**	**46.87**	**46.72**

Table 1. Results of classification accuracy with a different number of patches per category

The results of the proposed approach, compared with other state-of-the-art methods [7-9, 11, 14, 22, 25-27] over the MIT Indoor-67 dataset are shown in Table 2.

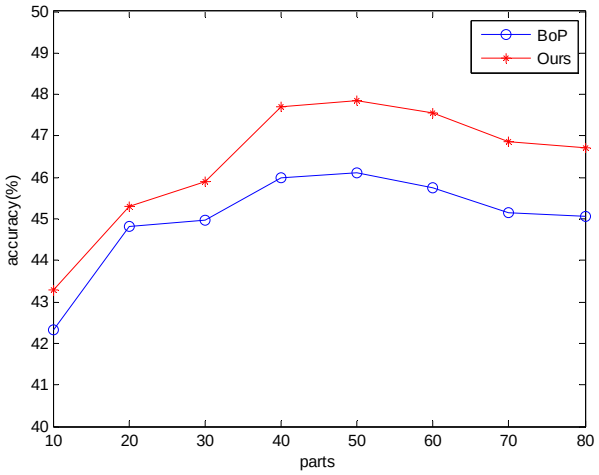


Figure 8. The accuracy curves at different numbers of patches per category

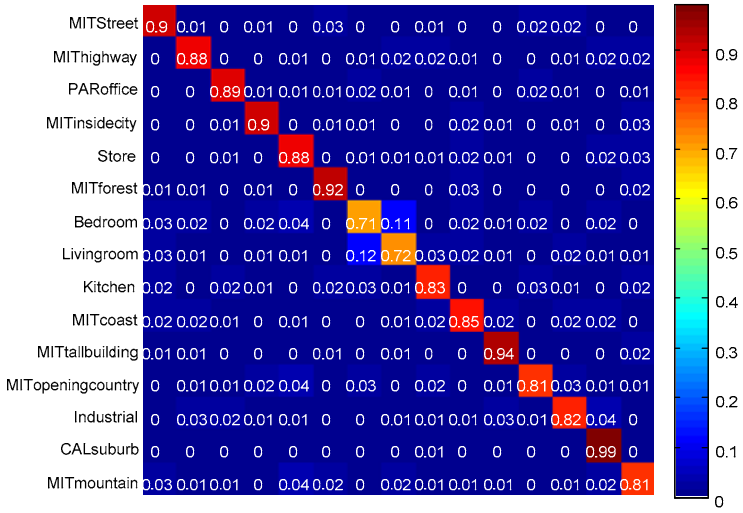


Figure 9. The confusion matrix of the results on the 15-scene dataset

As shown in Table 2, the proposed method outperforms other approaches in terms of Acc. In addition, the mAP (mean Average Precision) of the proposed method achieves 45.72%, which is better than that of [14].

Method	Acc (%)	mAP (%)
ROI+Gist [22]	26.05	-
MM-scene [8]	28.00	-
CENTRIST [25]	36.90	-
Object Bank [7]	37.60	-
DPM [9]	30.40	-
RBoW [26]	37.93	-
LPR [27]	44.84	-
Patches [11]	38.10	-
BoP [14]	46.10	43.55
Proposed	**47.84**	**45.72**

Table 2. The performances of scene classification over the MIT Indoor-67 dataset

4.2 15-scene dataset

The 15-scene dataset contains 4,485 images of 15 categories of indoor and outdoor scenes, such as bedroom, kitchen, coast, city, etc. We randomly chose 100 images per class for training and the rest for testing. The accuracy results are shown in Table 3 and the classification confusion matrix of the proposed method is shown in Fig. 9. It is evident from Table 3 that the proposed method outperforms other counterparts except for the Hybrid-Parts+GIST-color+SP [34] and the LScSPM [32]. In contrast to a HOG-based representation of the proposed approach, the method in [34] adopted multi-features composition, which leads to slightly better performance in terms of accuracy; while the LScSPM in [32] using low-level representation obtains the best performance; however, the extracted features have a lack of semantic information. From Fig. 9, we can see that the false rates of the bedroom and the living room are higher than other scenes. This is because the features in the images of the bedroom and the living room are too similar to identify their categories. Some confusing images of the bedroom and the living room are shown in Fig. 10.

Method	Acc (%)
GIST-color [28]	69.5
SP [23]	81.4
SP-pLSA [29]	83.7
CENTRIST [25]	83.9
HIK [30]	84.1
HG [31]	85.2
LScSPM [32]	89.8
Object Bank [7]	80.9
Classemes [33]	80.6
Hybrid-Parts+GIST-color+SP [34]	86.3
BoP [14]	84.7
Proposed	**85.7**

Table 3. Classification performances over the 15-scene dataset

4.3 UIUC-sports dataset

The UIUI-sports dataset contains 1,792 images of eight sports categories. We randomly selected 70 images for training and 60 images for testing in each category.

The performance comparison is shown in Table 4. The classification confusion matrix of the proposed method is shown in Fig. 11. As shown in Table 4, the proposed approach performs better than all other methods, except for the p.d.f (probability density function) method, the probable reason being that the p.d.f method uses an improved HOG features method for image representation.

4.4 Discussions

Mid-level representation is more adaptable to appearance distributions in real world than low-level features; howev-

er, learning mid-level features is a difficult task, since it usually involves many iterative operations. The proposed approach improves the performance by introducing a sketch tokens-based contour detection into the procedure of learning image patches. Since the contours contain more edge information, which is helpful for scene recognition, the proposed method can extract better mid-level features owing to the contour detection in initializing seed blocks. The experimental results on the MIT Indoor-67 dataset show that the proposed method obtains the best performance compared with other state-of-the-art methods. The experimental results based on the 15-scene dataset and the UIUC-sports dataset demonstrate that the proposed method performs better than most of the state-of-the-art approaches. While the proposed method works slightly worse than the LScSPM [30] and the Hybrid-parts+GIST-color+SP [29] on the 15-scene dataset, our method outperforms them over the UIUC-sports dataset.

Method	Acc (%)
GIST-color [28]	70.7
SP [23]	81.8
Graphical Model [24]	73.4
LDA [35]	66.0
CENTRIST [25]	78.3
HIK [30]	84.2
LScSPM [32]	85.3
Object Bank [7]	76.3
Classemes [33]	84.2
Hybrid-Parts+GIST-color+SP [34]	87.2
p.d.f [36]	90.4
BoP [14]	87.9
Proposed	**89.2**

Table 4. Classification performances on the UIUC-sports dataset

(a) bedroom

(b) living room

Figure 10. Confusing images of bedroom and living room

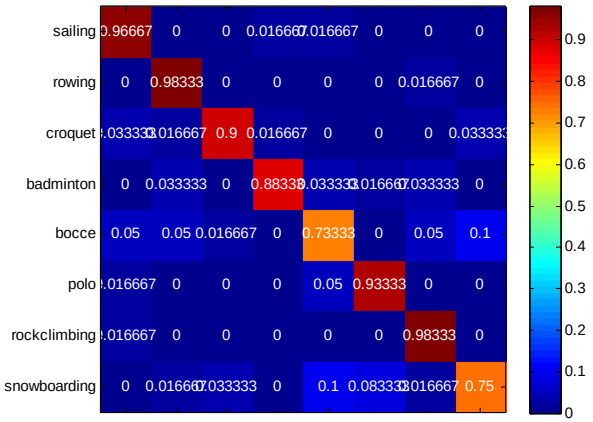


Figure 11. Confusion matrix of the results on the UIUC-sports dataset

5. Conclusions

We have proposed a contour detection-based mid-level feature learning method for scene classification. The proposed approach can automatically discover mid-level image patches with semantic information. First, since the initialization procedure is important for learning distinctive patches, we introduced a contour detection method based on sketch tokens to find seed blocks with more contour pixels, which results in improving the performance. The seed patches are then used to train exemplar SVMs to discover other similar occurrences under a sliding window framework. Finally, the discriminative patches are further mined by an entropy-rank criterion. Only the patches that appear frequently in one class but seldom in other classes are considered to be discriminative. Scene categories are identified according to the total response of matching scores between the distinctive patches and the testing images. Extensive experiments on the MIT Indoor-67 scene dataset, the 15-scene dataset and the UIUC-sports dataset demonstrate that the proposed method performs better than its counterparts. Although our approach has achieved promising results, further research will be carried out to improve the efficiency of feature extraction.

6. Acknowledgements

This work is partly supported by the National Natural Science Foundation of China under grant nos. 61201362, 61573351, 81471770 and 61273282; the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under grant no. CIT&TCD201404039; and the Scientific Research Project of Beijing Educational Committee under grant no. KM201410005005.

7. References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, Springer, 2006, pp. 404–417.

[3] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564–2571.

[4] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 2161–2168.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

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

[8] J. Zhu, L.-J. Li, L. Fei-Fei and E. P. Xing, "Large margin learning of upstream scene understanding models," in *Advances in Neural Information Processing Systems*, 2010, pp. 2586–2594.

[9] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1307–1314.

[10] J. Vogel and B. Schiele, "Semantic Modeling of Natural Scenes for Content-Based Image Retrieval," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 133–157, Apr. 2007.

[11] S. Singh, A. Gupta and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 73–86.

[12] C. Doersch, S. Singh, A. Gupta, J. Sivic and A. Efros, "What makes Paris look like Paris?", *ACM Trans. Graph.*, vol. 31, no. 4, 2012.

[13] R. Mittelman, H. Lee, B. Kuipers and S. Savarese, "Weakly Supervised Learning of Mid-Level Features with Beta-Bernoulli Process Restricted Boltzmann Machines," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 476–483.

[14] M. Juneja, A. Vedaldi, C. V. Jawahar and A. Zisserman, "Blocks That Shout: Distinctive Parts for Scene Classification," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 923–930.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[16] J. J. Lim, C. L. Zitnick and P. Dollar, "Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 3158–3165.

[17] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, vol. 2, pp. 416–423.

[18] P. Dollár, Z. Tu, P. Perona and S. Belongie, "Integral Channel Features," in *BMVC*, 2009, vol. 2, p. 5.

[19] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1–8.

[20] R. Xiaofeng and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Advances in neural information processing systems*, 2012, pp. 584–592.

[21] T. Malisiewicz, A. Gupta and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 89–96.

[22] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition, 2009. CVPR'09. IEEE Conference on*, 2009, pp. 413–420.

[23] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 2169–2178.

[24] L.-J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8.

[25] Jianxin Wu and J. M. Rehg, "CENTRIST: A Visual Descriptor for Scene Categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1489–1501, Aug. 2011.

[26] S. N. Parizi, J. G. Oberlin and P. F. Felzenszwalb, "Reconfigurable models for scene recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2775–2782.

[27] F. Sadeghi and M. F. Tappen, "Latent pyramidal regions for recognizing scenes," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 228–241.

[28] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[29] A. Bosch, A. Zisserman and X. Muoz, "Scene Classification Using a Hybrid Generative/ Discriminative Approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.

[30] J. Wu and J. M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 630–637.

[31] X. Zhou, N. Cui, Z. Li, F. Liang and T. S. Huang, "Hierarchical gaussianization for image classification," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1971–1977.

[32] S. Gao, I. W. Tsang, L.-T. Chia and P. Zhao, "Local features are not lonely–Laplacian sparse coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 3555–3561.

[33] L. Torresani, M. Szummer and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 776–789.

[34] Y. Zheng, Y.-G. Jiang and X. Xue, "Learning hybrid part filters for scene recognition," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 172–185.

[35] C. Wang, D. Blei and F.-F. Li, "Simultaneous image classification and annotation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1903–1910.

[36] T. Kobayashi, "BFO Meets HOG: Feature Extraction Based on Histograms of Oriented p.d.f. Gradients for Image Classification," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 747–754.