LEARNING DISCRIMINATIVE AND SHAREABLE PATCHES FOR SCENE CLASSIFICATION

Shoucheng Ni[⋆] *Qieshi Zhang*[†] *Sei-ichiro Kamata*[†] *Chongyang Zhang*[⋆]

[⋆]Institue of Image Communication and Network Engineering, Shanghai Jiao Tong University
[†]Graduate School of Information, Production and Systems, Waseda University

ABSTRACT

This paper addresses the problem of scene classification and proposes learning discriminative and shareable patches (LDSP) method. The main idea of learning discriminative and shareable patches is to discover patches that exhibit both large between-class dissimilarity (discriminative) and large within-class similarity (shareable). A novel and efficient re-clustering, based on co-occurrence relationship of first-step clustering, is proposed and conducted to further enhance the visual similarity of patches within each cluster. In order to establish appropriate criteria for selecting desired patches, a condensed representation of image features called feature epitome is introduced. In the classification, a patch feature involving pre-trained convolutional neural network model is investigated. The experimental result outperforms existing single-feature methods on MIT 67 scene benchmark in term of mean Accuracy Precision.

Index Terms— Learning discriminative and shareable patches, scene classification, deep-learned patch feature

1. INTRODUCTION

Scene classification, which aims at determining the scene category that one image is taken in, is one of the main tasks of computer vision. It can be applied in areas such as scene classification apparatus of video, autonomous robotics, digital libraries and so on.

Considering the complexity and diversity of scene images, several frameworks and methodologies have been proposed. Discovering distinct features for scene categories is the most essential motivation in previous works. [1, 2] focus on creating local features to capture distinct characteristics of each category. [3, 4, 5] attempt to construct mid-level features based on patches, such as patch filter banks or patch dictionary. Recently, [6, 7] have proven the effectiveness of applying deep learning methods in scene classification.

Meanwhile, some other different methods based on image parts or patches have also been proposed: [8] addresses the importance of learning image regions with higher occurrence possibility of discriminative parts; however, these parts are manually acquired. [9, 10, 11] aim at proposing an automatic mode for discovering distinct image parts. These distinct parts or patches are supposed to reveal the semantics of each scene category; however, previous works fail to emphasize the property of desired patches in seeking process.

The proposed learning discriminative and shareable patches method, referred to as LDSP, mainly solves two problems in previous works: (i) how to discover patches with higher visual similarity: besides certain clustering applied in previous works, proposed co-occurrence re-clustering will further improve the shareable property of patches; (ii) how to establish criteria for selecting desired patches: compared with Entropy-Rank in [10], representative power presented gives a straightforward constraint. Learned patches are denoted as discriminative and shareable patches (DSPs): being shareable implies patches sharing common patterns occur frequently in images from the very category, while being discriminative suggests that these patches occur rarely in images from other categories.

Our scene classification framework is based on LDSP. There are three main advantages of the classification framework involving DSPs: (i) **semantic diversity elimination**: DSPs are capable of capturing semantic essence of each scene category; (ii) **experimental time reduction**: compared with training images that are directly utilized, DSPs are of both smaller size and quantity; (iii) **universal application potentiality**: for two different databases that contain scene categories with the same or similar semantics, it is possible to share or exchange the DSPs for these categories.

The remaining of the paper is organized as follows. Section 2 gives detail explanations on our proposed LDSP method. The experimental results are presented and discussed in section 3. Finally, conclusions are drawn in section 4.

2. LEARNING DISCRIMINATIVE AND SHAREABLE PATCHES

The proposed LDSP method is capable of automatically discovering discriminative and shareable patches of both high between-class dissimilarity and within-class similarity. As shown in Fig. 1, LDSP method consists of three successive steps: K-means of Segmentation Component Centroids (KSCC), Hierarchical Clustering and Co-occurrence Re-clustering (HCCR), Discriminative and Shareable Patch Selection (DSPS). After the KSCC step, initial sets of patches are supposed to be sampled. The HCCR step conducts two-step clustering to seek several groups of visually similar patches for each scene category. The final DSPS step deals with the problem how to evaluate the representative power of each cluster of patches. The detailed procedure is illustrated in the following section.

2.1. K-means of Segmentation Component Centroids

Initial patches are supposed to be sampled from training images in the first step. In [9], randomly sampling, which means patches are completely randomly generated, is applied; however, these patches may not be semantic enough to be qualified representatives of original images. In our work, a sampling approach taking segmentation results into account is going to be applied.

To be specific, firstly original image is segmented by the efficient graph-based image segmentation [12]. Next, K-means is applied to cluster centroids of segmented connected components (or regions)

This work was partly funded by NSFC (No.61571297, No.61527804, No.61420106008) and China National Key Technology R&D Program (No. 2012BAH07B01).

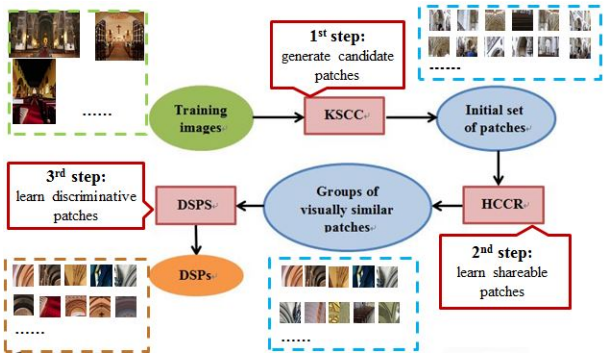

Fig. 1. Flow chart of proposed LDSP method. The first step is applied to generate initial candidate patches. The second and third step follow the criteria of being shareable and discriminative respectively.

into several clusters so that the centers of patches to be sampled can be generated. For one image *I*, N_p patches are to be sampled; at the same time, there are N_c connected components after segmentation: (i) If $N_p > N_c$, K-means is applied to cluster N_c component centroids (represented as the coordinates in image space) into N_p clusters. The centroids of newly-generated N_p clusters are treated as the centers of N_p sampled patches; (ii) If $N_p \leq N_c$, N_c component centroids are taken directly as the centers of sampled patches and the remaining $(N_p - N_c)$ centers are generated by random sampling.

Considering the complexity of scene images, the first situation is more likely to occur. Fig. 2 gives the example how our proposed sampling strategy performs.

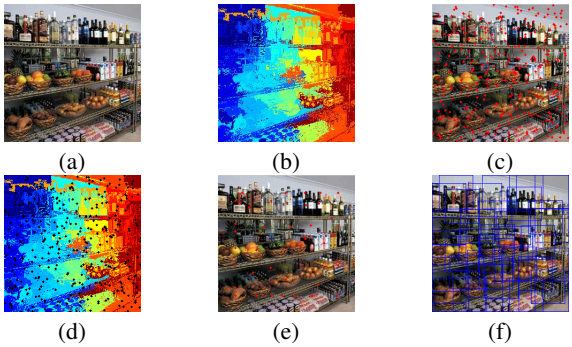

Fig. 2. Example of KSCC. (a) Original image, (b) Segmented image in which connected regions sharing the same color is a connected segmentation component, (c) Centroids of segmentation components shown in the original image, (d) Centroids of segmentation component shown in the segmented image, (e) Patch centers generated by clustering all centroids in (c), (f) Patches sampled by KSCC.

2.2. Hierarchical Clustering and Co-occurrence Re-clustering

2.2.1. Hierarchical Clustering and Cross Validation

Patch similarity measurement is the crucial component to seek visually similar patches. Histogram of Oriented Gradient (HOG) [13] is used as alternative option for similarity measurement in our work.

The properties of hierarchical clustering make it the most suitable for seeking visually similar patches. Firstly, for visually similar patches, the only prior knowledge is features of these patches are similar. Secondly, in sampling step, there may be some highly-overlapped patches or similar patches sampled from repeated pattern from one single image. However, similar patches from single image should be discouraged as shareable patches. Thus, the complete distance $dist_{max}(C_i, C_j)$, ensuring patches with smaller Euclidean distance between features to be clustered, is applied in hierarchical clustering; meanwhile, clustering patches sampled from the same image I is prevented by setting corresponding distance to infinity:

$$dist_{max}(C_i, C_j) = \max_{\forall p_i \in C_i, \forall p_j \in C_j} dist(p_i, p_j),$$
$$dist(p_i, p_j) = \begin{cases} \infty & p_i, p_j \in I \\ dist\big(HOG(p_i), HOG(p_j)\big) & otherwise \end{cases}, \quad (1)$$

where C_i is one cluster, p_i is one patch in cluster C_i, $p_i, p_j \in I$ indicates two patches are sampled from the same image and $dist\big(HOG(p_i), HOG(p_j)\big)$ is the Euclidian distance between HOG features of patches p_i and p_j.

Hierarchical clustering and cross validation, as described in Alg. 1, are combined in the first-step clustering:

Algorithm 1 Hierarchical clustering and cross validation

Input: S: Initial patch set for one category generated by KSCC; N: Patches sampled from non-scene images; N_P: Number of patches to be detected with highest SVM scores;
Output: C: Clusters of visually similar patches;
1: $S \rightarrow S_1, S_2$ (S_1, S_2: equal sized disjoint sets);
2: $C_{i,j} \leftarrow hierarchical\_cluster\{S_i\}$ $(1 \leq i \leq 2, 1 \leq j \leq M)$;
3: **while** not converge **do**
4: **for** $1 \leq j \leq M$ **do**
5: $SVM_{1,j} \leftarrow svm\_train\{C_{1,j}, N\}$;
6: $C_{1,j} \leftarrow svm\_detect\{SVM_{1,j}, S_2, N_P\}$;
7: **end for**
8: $swap\{S_1, S_2\}, swap\{C_1, C_2\}$;
9: **end while**
10: **return** C;

2.2.2. Co-occurrence Re-clustering

After observing the clusters created by hierarchical clustering and cross validation, we find that although the similarity among all the patches in the same cluster is low, there are some pairs of patches that will occur in the same cluster for more than once; at the same time, these two patches show a high degree of similarity. It inspires us to use the times of two patches existing in the same cluster as the measurement of similarity, which we denote as co-occurrence.

Co-occurrence re-clustering is established based on the clustering results of previous step, which to some extent implies desired relationship between elements within the same cluster. Thus, co-occurrence re-clustering will preserve the initial clustering information and lead to a better re-clustering consequence.

The following part is the detailed procedure of co-occurrence re-clustering: after the first-step clustering, there are N_c created clusters and N_e elements in these clusters. The co-occurrence matrix $M_{CO} \in \mathbb{R}^{N_e \times N_e}$ is constructed according to the following illustration: $M_{CO}(i,j) = \begin{cases} Co(e_i, e_j) & i \neq j \\ 0 & i = j \end{cases}$ and $Co(e_i, e_j)$ denotes how many times the *i*-th and the *j*-th element occur in the same cluster. Considering M_{CO} is a symmetric matrix, we sort the elements in the upper triangular matrix of M_{CO}, denoted as M'_{CO}, in descend order: $M'_{CO}\big(O_1(1), O_1(2)\big) \geq$

$M'_{CO}\big(O_2(1), O_2(2)\big) \geq \cdots \geq M'_{CO}\big(O_M(1), O_M(2)\big) \geq co_{thr} \geq M'_{CO}\big(O_{M+1}(1), O_{M+1}(2)\big) \geq \cdots \geq M'_{CO}\big(O_N(1), O_N(2)\big)$, where $N = \big[(N_e - 1) \times N_e\big]/2$, $\big(O_j(1), O_j(2)\big)$ is the index of j-th largest element in M'_{CO} and co_{thr} is a co-occurrence threshold to ensure two elements that have a higher co-occurrence value to be clustered. Co-occurrence re-clustering is conducted from O_1 to O_M and only N_P elements will remain in each of finally-clustered clusters. The selection is based on solving the following optimization problem in Eq. (2) and detailed procedure is shown in Alg. 2:

$$\underset{\forall e_1, \cdots, e_{N_P} \in C'_k}{\arg max} \sum_{l=1}^{N_P - 1} \sum_{m=l+1}^{N_P} Co(e_l, e_m). \tag{2}$$

Algorithm 2 Co-occurrence re-clustering

Input: M'_{CO}: Upper triangular matrix of M_{CO}; M: Number of element pairs with co-occurrence higher than co_{thr};
Output: C': Re-clustered clusters;

Initialize: the number of existing re-clustered clusters $N'_c = 0$, existing re-clustered clusters $C' = \emptyset$;
for $1 \leq j \leq M$ **do**
  if $(e_{O_j(1)} \notin C') \wedge (e_{O_j(2)} \notin C')$ **then**
    $C'_{N'_c+1} \leftarrow \{e_{O_j(1)} \cup e_{O_j(2)}\}$;
  end if
  if $(e_{O_j(1)} \in C'_{k_1}) \wedge (e_{O_j(2)} \in C'_{k_2}) \big(k_1, k_2 \in [1, N'_c]\big)$ **then**
    $C'_{k_1} \leftarrow \{C'_{k_1} \cup C'_{k_2}\}, C'_{k_2} = \emptyset$;
  end if
  if $(e_{O_j(m)} \notin C') \wedge (e_{O_j(3-m)} \in C'_k)\ \big(m \in \{1, 2\}\big)$ **then**
    $C'_k \leftarrow \{C'_k \cup e_{O_j(m)}\}$;
  end if
  Update N'_c, C';
end for
for $1 \leq k \leq N'_c$ **do**
  Solve Eq. (2) and $C'_k \leftarrow \cup_{i=1}^{N_p} e_i$;
end for
return C';

2.3. Discriminative and Shareable Patch Selection

2.3.1. HOG matrix and feature epitome

As we conclude from the property of DSPs, the representative power is highly associated with the issue of occurrence relationship between one patch and one image. The strategy frequently used for measuring the relationship is sliding window method: a bounding window is moving throughout the whole image to generate the response map, which is very time-consuming. Thus, feature epitome, a condensed summarization of patch features, is proposed to calculate the occurrence relationship more efficiently.

To generate the feature epitome for one patch, the first step is to construct the HOG matrix, which is inspired by how HOG feature is constructed: each pixel in the cell calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it and the votes are accumulated into 9 orientation bins over the cell [13]. The 9-dimensional vector can be further reshaped into a matrix $HOG\_M_c \in \mathbb{R}^{3\times3}$. As suggested in [13], 2×2 cells are grouped into a block and matrix HOG_M_b for 4 cells are put together according to the spatial layout and a HOG block $HOG\_M_B = \begin{bmatrix} HOG\_M_{c\_1,1} & HOG\_M_{c\_1,2} \\ HOG\_M_{c\_2,1} & HOG\_M_{c\_2,2} \end{bmatrix} \in \mathbb{R}^{6\times6}$ is generated and normalized finally. For the whole image, final HOG matrix HOG_M is constructed by grouping HOG_M_B of extracted overlapped blocks together as illustrated in Fig. 3.

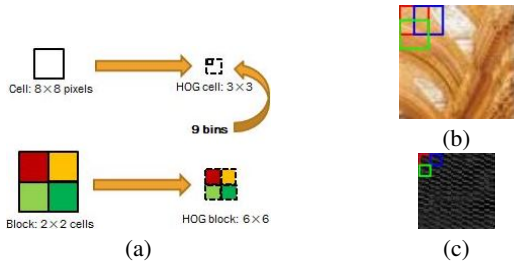

Fig. 3. Construction of HOG matrix. (a) Construction of HOG cell and HOG block, (b) Input patch (64×64), (c) HOG matrix (42×42).

The feature epitome is inspired by epitome [14], which is a condensed version containing the essence of the textural and shape properties of the original image. Given one image I, assume that N_p patches are sampled from the image and HOG matrix of these N_p patches are given by $H_k \in \mathbb{R}^{m\times n} (k \in [1, N_p])$, the feature epitome $e = (\mu, \sigma)$ $(\mu, \sigma \in \mathbb{R}^{M\times N}, M > m, N > n)$ is calculated by means of Gaussian Mixture Model (GMM): in each iteration, the generative model uses a hidden mapping Γ_k that maps HOG matrix H_k to the coordinates E_k in e. In Expectation step, mapping Γ_k is updated, while μ, σ are updated in Maximization step.

2.3.2. Occurrence quantization

The occurrence probability concerning one patch *P* and one image *I* is indicated by the HOG matrix $H \in \mathbb{R}^{m\times n}$ of patch *P* and the feature epitome $e = (\mu, \sigma)$ of image *I*, as illustrated in Eq. (3):

$$\begin{aligned} & p(H|\mu, \sigma) \\ &= \max_{\substack{\forall r\in[1,M-m+1] \\ \forall c\in[1,N-n+1]}} \prod_{i=r}^{r+m-1} \prod_{j=c}^{c+n-1} \mathcal{N}\Big(H(i,j); \mu(i,j), \sigma(i,j)\Big) \\ &= \max_{\substack{\forall r\in[1,M-m+1] \\ \forall c\in[1,N-n+1]}} \prod_{i=r}^{r+m-1} \prod_{j=c}^{c+n-1} \frac{1}{\sqrt{2\pi}\sigma(i,j)} e^{-\frac{[H(i,j)-\mu(i,j)]^2}{2\sigma(i,j)^2}}. \end{aligned} \tag{3}$$

The probability $p(H|\mu, \sigma)$ is the maximum response of the product of Gaussian distribution probability of all pixels over all possible patch locations in the epitome space. In information theory, the minus logarithm of probability is denoted as self-information:

$$\begin{aligned} SI(P|I) = \min_{\substack{\forall r\in[1,M-m+1] \\ \forall c\in[1,N-n+1]}} \sum_{i=r}^{r+m-1} \sum_{j=c}^{c+n-1} \Big\{ & \log[\sqrt{2\pi}\sigma(i,j)] \\ & + \frac{[H(i,j) - \mu(i,j)]^2}{2\sigma(i,j)^2} \Big\}. \end{aligned} \tag{4}$$

Smaller self-information $SI(P|I)$ is an indication that majority of the information existing in the patch *P* can be acquired from the image *I*. In other words, there is a higher probability that the similar pattern of patch *P* exists in the image *I*.

The power of one cluster of patches to represent one category is defined as $P_{re}(C, I_{val_j}) = \sum_{i=1}^{Num(j)} \min_{\forall P\in C} SI(P \mid I_{val\_j,i})$, where $I_{val\_j,i}$ is the *i*-th validation image for the *j*-th category and $Num(j)$ is the number of validation images for the *j*-th category.

Assume $C_{n,m}$ is the m-th cluster for the n-th category, the representative power concerning whole validation images is calculated by

$$P(C_{n,m}, I_{val}) = \sum_{j=1}^{N} \sum_{i=1}^{Num(j)} \min_{\forall P \in C_{n,m}} SI(P|I_{val\_j,i}) f(n,j), \tag{5}$$

where N is the number of scene categories and function $f(n,j) = \begin{cases} 1 & n = j \\ -1 & n \neq j \end{cases}$ indicates whether DSPs and validation images belong to the same category. For each cluster in each category, the representative power $P(C, I_{val})$ is calculated and the clusters with relatively larger $P(C, I_{val})$ will be selected. Fig. 4 shows examples of clusters of DSPs for MIT 67 scene database[15] generated by LDSP method.

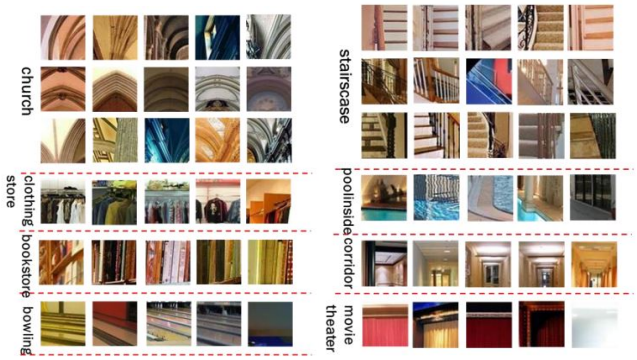

Fig. 4. Examples of discriminative and shareable patches. Five patches in one row belong to the same cluster.

3. EXPERIMENTAL RESULTS

In the experiment, MIT 67 scene database that contains 67 scene categories is tested to evaluate proposed classification framework. For each category, there are about 80 training images and 20 testing images. Grayscale [16] training images are discarded and 15 images are selected from remaining training ones as validation images.

Five kinds of novel feature encoding (FE) methods are investigated in the experiment: (i) Vector Quantization (VQ); (ii) Locality-constrained Linear Coding (LLC); (iii) Kernel-codebook encoding (KCB); (iv) Fisher Vectors (FV); (v) Vector of Locally Aggregated Descriptors (VLAD). The implementation of the feature encoding methods provided by [17] is applied in the experiment.

3.1. Scene Classification with SIFT Feature

SIFT features are extracted from discriminative and shareable patches and validation images. In the experiment, the number of clusters of DSPs used ranges from 4 to 8 and classification performance is evaluated by mean Accuracy Precision (mAP). The experimental results with SIFT features are shown in Table 1.

3.2. Scene Classification with Deep-learned Patch Feature

In our work, we attempt to introduce a deep-learned patch feature (DLPF) into the well-known bag-of-words model to classify scene images. The main idea of DLPF is to take advantages of the convolutional neural network (CNN) models trained from the ImageNet [18] database. The output of one layer that is close to the terminal layer in the CNN is considered as the feature of the input patch.

In the experiment, patches of size 32×32, with sampling step size 16, are sampled to generate the deep-learned patch features and the pre-trained CNN model named imagenet-vgg-verydeep-16 [19] is applied. The experimental results are shown in Table 2.

Finally, we compare our classification result with some other single-feature method in Table 3 and our best classification result outperforms the results of other single-feature methods on MIT 67 scene database. The slight improvement between SIFT-based and DLPF-based methods may result from the fact that much fewer DLPFs are extracted in codebook generation compared with SIFTs.

Table 1. Classification results using varying number of clusters of discriminative and shareable patches with SIFT feature encoding. The best classification performance with SIFT feature is 59.38%.

FE \ Range	4	5	6	7	8
VQ	49.27%	49.97%	**50%**	49.8%	49.61%
LLC	**48.89%**	48.48%	48.42%	48.48%	48.89%
KCB	49.77%	50.29%	**51.23%**	49.92%	51.09%
IFV	59.11%	58.88%	**59.38%**	58.88%	57.65%
VLAD	56.98%	57.39%	**58.4%**	56.45%	56.09%

Table 2. Classification results using 6 of clusters of discriminative and shareable patches with deep-learned patch feature encoding. The number in parentheses indicates the output layer of CNN.

FE \ Feature	DLPF	SIFT
VQ (35)	51.82%	50%
IFV (37)	59.73%	59.38%
VLAD (37)	57.15%	58.4%

Table 3. Comparisons of mean Accuracy Precision (mAP) with other proposed single-feature methods on MIT 67 scene database.

Methods	mAP(%)
Patches[9]	38.10
LPR[20]	44.84
RICA[21]	47.89
Part Detector[4]	51.40
DSFL[3]	52.24
DeCAF[7]	58.52
LDSP+SIFT	**59.38**
LDSP+DLPF	**59.73**

4. CONCLUSIONS

We propose learning discriminative and shareable patches method to classify scene images. The LDSP method is capable of automatically discovering discriminative and shareable patches of both high between-class dissimilarity and within-class similarity compared with previous works. Meanwhile, the deep-learned patch feature is introduced into the bag-of-words model. The experimental results show that proposed DLPF can improve the mAP compared with applying SIFT feature. The best classification result also outperforms other single-feature methods on MIT 67 scene database.

5. REFERENCES

[1] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2006, vol. II, pp. 2169–2178.

[2] R. Margolin, L. Zelnik-Manor, and A. Tal, "OTC: A novel local descriptor for scene classification," in *Proc. of European Conf. on Computer Vision*, 2014, vol. VII, pp. 377–391.

[3] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang, "Learning discriminative and shareable features for scene classification," in *Proc. of European Conf. on Computer Vision*, 2014, vol. I, pp. 552–568.

[4] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Proc. of IEEE Int'l Conf. on Computer Vision*, 2013, pp. 3400–3407.

[5] G. Papandreou, L.C. Chen, and A.L. Yuille, "Modeling image patches with a generic dictionary of mini-epitomes," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 2059–2066.

[6] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014, pp. 512–519.

[7] J. DDonahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. of Int'l Conf. on Machine Learning*, 2013, pp. 647–655.

[8] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 3726–3733.

[9] S. Singhand, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. of European Conf. on Computer Vision*, 2012, pp. 73–86.

[10] M. Juneja, A. Vedaldi, C.V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 923–930.

[11] C. Doersch, A. Gupta, and A.A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. of Advances in Neural Information Processing Systems*, 2013, pp. 494–502.

[12] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886–893.

[14] N. Jojic, B.J. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *Proc. of IEEE Int'l Conf. on Computer Vision*, 2003, pp. 34–41.

[15] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 413–420.

[16] Q. Zhang and S. Kamata, "A novel color space based on rgb color barycenter," in *Proc. of IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 2016.

[17] K. Chatfield, V.S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. of British Machine Vision Conference*, 2011, vol. II, p. 8.

[18] "ImageNet," `http://www.image-net.org/`.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[20] F. Sadeghi and M.F. Tappen, "Latent pyramidal regions for recognizing scenes," in *Proc. of European Conf. on Computer Vision*, 2012, vol. V, pp. 228–241.

[21] V.Q. Le, A. Karpenko, J. Ngiam, and A.Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Proc. of Advances in Neural Information Processing Systems*, 2011, pp. 1017–1025.