Discovering Class-Specific Spatial Layouts for Scene Recognition

Chaoqun Weng, Hongxing Wang, Junsong Yuan, *Senior Member, IEEE*, and Xudong Jiang, *Senior Member, IEEE*

***Abstract*—Scene image is a spatial composition of objects and background contexts and finding discriminative spatial layouts is critical for scene recognition. In this letter, we propose an ℓ_1-regularized max-margin formulation to discover class-specific spatial layouts by jointly learning the image classifier and the class-specific spatial layouts for scene recognition. Unlike previous methods that classify images into different categories either without considering the spatial layouts explicitly or only using class-generic spatial layout, our proposed method can discover a sparse combination of class-specific spatial layouts for different scenes and boost the recognition performance. Experiments on scene-15, landuse-21, and MIT indoor-67 datasets validate the advantages of our proposed algorithm.**

***Index Terms*—Discovering class-specific spatial layouts, scene recognition.**

I. Introduction

DIFFERENT from texts and audios analysis, the rich spatial information in images has been proved to play a critical role in scene recognition and object detection [1]–[5]. Specifically for scene recognition, many previous work has demonstrated that the discriminative power is limited without considering encoding spatial information for local visual primitive features [6]–[12]. This is due to the fact that scene images are usually spatial compositions of foreground objects and background contexts with clear spatial layouts. Take the images in Fig. 1 for example, the "street" scene category often consists of "building" and "road" components, and the "coast" category often is composed of "sky," "coast," and "sea" components with clear spatial layouts. However, it remains challenging problem to leverage the spatial layout information in scene recognition applications to boost the recognition performance.

Many research work have focused on how to model spatial layout for scene and object recognition [8], [13]–[19]. One of the most intuitive ways to leverage the spatial layout information is to partition the image space into predefined grid cells and then compute the corresponding visual features for each grid cell and finally concatenate all of them to form a global image representation. For example, the spatial pyramid matching [6] method using hand-crafted features such as HOG [20] and SIFT [21] has illustrated the effectiveness of encoding the spatial pyramid information. It significantly improved the classification performance of the previous bag-of-visual-word method [22]. The SPP-Network given in [7] also proposed to use the spatial pyramid pooling for the fully connected layers to utilize the spatial layout information and boosted the accuracy of a variety of CNN networks in spite of their different architecture designs. In [8], Jiang *et al.* proposed a boosting method to select discriminative visual features under different spatial layouts for scene recognition. In [17], Yang *et al.* introduced a maxout layer in the CNN structure for spatial layout selection and achieved superior performance compared to previous CNN structure without explicitly using spatial layout information.

In spite of the great successes of previous work, there still exist many limitations. First, the class-generic predefined spatial layout is applied in many previous spatial pyramid pooling methods [6], [7], [23], [24], but is not necessarily an optimal choice for classification. This is due to the large interclass variations among different scene categories. As shown in Fig. 1, the "street" scene images have different spatial layout compared with the "coast" scene images and therefore the class-generic

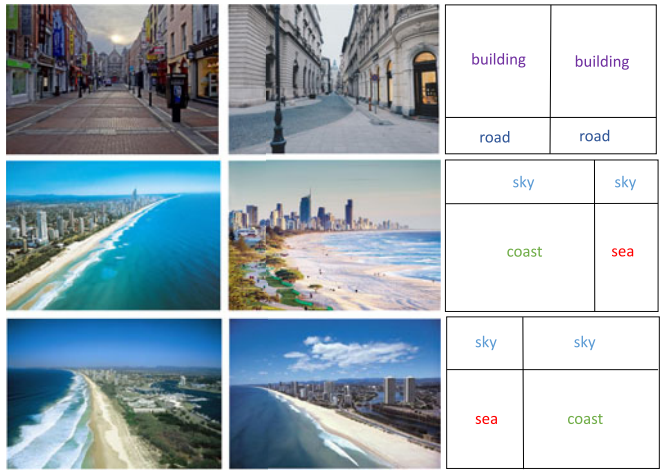

Fig. 1. Illustration of the spatial layouts for street and coast scene categories. Comparing the first row with the second and third rows, we can see that street and coast categories have different spatial layouts due to large interclass variations. From the second and third rows, we can see that within even the same category, images can also exhibit different spatial layouts due to large intraclass variations.

Manuscript received August 17, 2016; revised November 1, 2016; accepted December 5, 2016. Date of publication December 16, 2016; date of current version June 26, 2017. This work was supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114 and Tier 1 RG27/14, in part by the National Natural Science Foundation of China under Grant 61602069, and in part by Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2016jcyjA0468. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vasileios Mezaris.

C. Weng, J. Yuan, and X. Jiang are with Nanyang Technological University, Singapore 639798 (e-mail: weng0018@e.ntu.edu.sg; jsyuan@ntu.edu.sg; exdjiang@ntu.edu.sg).

H. Wang is with Nanyang Technological University, Singapore 639798, and also with Chongqing University, Chongqing Shi 400030, China (e-mail: ihxwang@cqu.edu.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2016.2641020

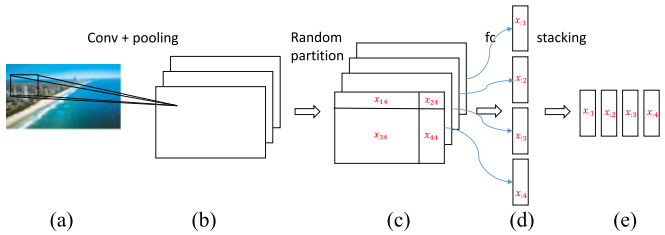

Fig. 2. Illustration of the feature extraction pipeline for the proposed method: (a) shows the input image; (b) shows the 2-D feature maps of the convolution and pooling layers of the convnet; (c) shows the multiple random partitions to form different spatial layouts; (d) shows the extracted features using the fc layer for different partitions; and (e) shows the stacked data matrix for the input image. From the figure we can see that we insert a random partitioning layer between the convolution/pooling layers and the fc layers of the convnet, and finally output a stacked data matrix as the data representation for each image.

spatial layout cannot work well for both scene categories. In such a case, class-specific spatial layout should be considered. Second, due to the large intraclass variations, scene images from even the same category could exhibit different spatial layouts. As a result, a unique spatial layout may not be optimal to capture all the variations. For example, the images in the second and third row of Fig. 1 are both from the "coast" category but they clearly have different spatial layouts.

From the above observations, this letter contributes to explicitly encoding the class-specific spatial layouts into the image classifier to boost the recognition performance. We first generate multiple random spatial layouts and then propose to jointly learn the class-specific spatial layouts and the image classifier by solving an ℓ_1-regularized max-margin optimization problem. The objective function can be optimized and converge to a local optima by our proposed alternating method. The introduced ℓ_1 regularized term induces sparsity of the discovered class-specific spatial layouts. As a result we are able to discover a sparse combination of class-specific spatial layouts and achieve superior performance compared to the method without explicitly considering the spatial layout information. Also thanks to the use of deep learning features [3] instead of traditional handcrafted features, our method achieves significant improvement over previous methods. Experiments on scene-15, landuse-21, and MIT indoor-67 datasets validate the advantages of our proposed algorithms.

II. Proposed Method

A. Feature Extraction

In this section, we will introduce the feature extraction pipeline for our proposed method. As shown in Fig. 2, first we obtain a set of two-dimensional (2-D) feature maps by forwarding the images through the convolution and pooling layers of the convnets, then we apply multiple random partitioning to the 2-D feature maps. After that, for each feature map we can obtain a concatenated fully connected feature vector, and finally we stack all the feature vectors to get the final matrix representation for each image.

Formally, we define a spatial pyramid of ℓ levels and each level i is randomly partitioned into nonoverlapping $2^i \times 2^i$ subregions. Instead of symmetrically dividing the image space into uniform cells as in [6], we randomly partition the image space into $2^i \times 2^i$ subregions of various sizes by using a uniform distribution. We repeat the random partition process independently for m times to obtain a set of predefined spatial layouts as candidates. As a result, we can obtain a set of subregions $R = \{R_{ij} | \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}\}$, where $n = \frac{1}{3}\left(4^{\ell+1} - 1\right)$ is the number of subregions for each spatial pyramid. For example, we can obtain $1 \times 1 + 2 \times 2 + 4 \times 4 = 21$ subregions for a spatial pyramid of three levels. Then, we input these subregions into the convnet [3] to get the corresponding feature vector $\mathbf{x}_{ij} \in \mathbb{R}^d$ for each subregion R_{ij}.

After that, we stack the feature vectors for different random partitions and define the data matrix $\mathbf{X} \in \mathbb{R}^{nd \times m}$ for each image in the following equation, where each column vector $\mathbf{x}_i \in \mathbb{R}^{nd}$ is the feature representation for each random partition and each $\mathbf{x}_{ij}$ is the ith subregion under the jth random partition

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] = \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{n1} & \cdots & \mathbf{x}_{nm} \end{bmatrix}. \quad (1)$$

Note that we explicitly encode the spatial layout information in the matrix representation for each image using multiple random partitions. The remaining problem is how to learn which spatial layouts are the optimal ones for each class-specific scene.

B. Discovering Class-Specific Spatial Layouts

In this section, we will introduce the proposed method to discover optimal spatial layout for each class given the data matrix extracted from Section II-A for each image.

For a particular scene category, due to the large intraclass variations, the optimal spatial layout may not be a unique one. We thus assume that the best spatial layout should be a mixture of spatial layouts, i.e., a linear combination of spatial layouts as follows:

$$\mathbf{z} = \sum_{i=1}^{m} v_i \mathbf{x}_i = \mathbf{X}\mathbf{v}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_m]^T \in \mathbb{R}^m$ is the coefficient weight for the m different random partitions and $\mathbf{z} \in \mathbb{R}^{nd}$ is the resulting feature representation for the image pyramid of in total n subregions.

Once we have obtained $\mathbf{z}$, we are interested in training a linear support vector machine (SVM) classifier as follows:

$$f(\mathbf{X}) = \mathbf{u}^T \mathbf{z} = \mathbf{u}^T \mathbf{X}\mathbf{v}$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_{\text{nd}}]^T \in \mathbb{R}^{\text{nd}}$ is the linear SVM weight vector of length nd. Note that we skip the bias terms to simplify the notations. It is also worth noting that the parameters number in the form of $f(\cdot)$ is $|\mathbf{u}| + |\mathbf{v}|$, compared to $|\mathbf{u}| * |\mathbf{v}|$ if we learn a linear SVM by concatenating all the columns of data matrix $\mathbf{X}$. The latter case is either prone to overfitting on the training dataset or infeasible due to the computational resource limitation, e.g., the lack of enough RAM or the need for much longer running time.

We follow the work on support vector machines [25] and define the empirical loss of classifier $f(\cdot)$ as the sum of the square hinge losses over a collection of T training images:

$$\sum_{t=1}^{T} \max(0, 1 - y_t f(\mathbf{X}_t))^2$$

where $\mathbf{X}_t$ is the data matrix as described in Section II-A for the tth image and $y_t \in \{1, -1\}$ is the corresponding label. Note that we use the square hinge loss instead of the hinge loss following the work in [25] to simplify the computation.

Since focusing solely on the empirical loss may result in overfitting the training set, we also add an ℓ_2 regularization on the classifier weight $\mathbf{u}$ and an ℓ_1 regularization on the partition coefficient $\mathbf{v}$, then the final objective function becomes as follows:

$$\arg\min_{\mathbf{u},\mathbf{v}} \frac{1}{2}\|\mathbf{u}\|^2 + \lambda\|\mathbf{v}\|_1 + C\sum_{t=1}^{T}\max(0, 1 - y_t f(\mathbf{X}_t))^2 \tag{2}$$

$$f(\mathbf{X}) = \mathbf{u}^T\mathbf{X}\mathbf{v}. \tag{3}$$

It is worth noting that we use the traditional ℓ_2-norm regularization for the weight $\mathbf{u}$ to learn the image classifier, and also the ℓ_1-norm regularization for the weight $\mathbf{v}$ to learn the combinations of random partitions. This is due to the consideration that the ℓ_2-norm regularization on the weight $\mathbf{u}$ follows the max-margin framework [26], while the ℓ_1-norm regularization on the coefficient vector $\mathbf{v}$ induces sparsity under certain conditions [27]. As a result, we can learn the class-specific weights $\mathbf{u}$ and $\mathbf{v}$ for different scene categories with different appearance features and different spatial layouts.

To optimize the loss function (2), we use an alternating method to iteratively optimize $\mathbf{u}$ and $\mathbf{v}$. When $\mathbf{v}$ is fixed, training $\mathbf{u}$ becomes an ℓ_2-regularized ℓ_2-loss SVM problem as shown in (4)

$$\arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u}\|^2 + C\sum_{t=1}^{T}\max(0, 1 - y_t f(\mathbf{X}_t))^2 \tag{4}$$

$$f(\mathbf{X}) = \mathbf{u}^T(\mathbf{X}\mathbf{v}). \tag{5}$$

Similarly once $\mathbf{u}$ is fixed, updating $\mathbf{v}$ also reduces to an ℓ_1-regularized ℓ_2-loss problem, as shown in (6)

$$\arg\min_{\mathbf{v}} \lambda\|\mathbf{v}\|_1 + C\sum_{i=t}^{T}\max(0, 1 - y_t f(\mathbf{X}_t))^2 \tag{6}$$

$$f(\mathbf{X}) = (\mathbf{u}^T\mathbf{X})\mathbf{v}. \tag{7}$$

Intuitively, optimizing $\mathbf{u}$ can be viewed as the conventional linear SVM classifier training process, given the feature vector $\mathbf{X}\mathbf{v}$ for each image, while on the other hand, updating $\mathbf{v}$ can be viewed as learning the sparse random partition coefficients, given the feature vector $\mathbf{u}^T\mathbf{X}$ for each image. To discover class-specific spatial layouts for multiclass dataset, we apply multiple one-versus–rest trainings. In Algorithm 1, we show the complete alternating optimization method.

Algorithm 1: Discovering class-specific spatial layouts

Input: Training samples $D = \{\mathbf{X}_1, \cdots, \mathbf{X}_t\}$, labels $Y = \{y_1, \cdots, y_t\}$, number of classes k.
Output: Weight matrix $\mathbf{U}$ for image classifiers and $\mathbf{V}$ for random partition coefficients.

```
for i ← 1 to k do
    Init v randomly
    while not converged do
        Obtain features Xv by Eq. 5
        Update u by solving Eq. 4
        Obtain features u^T X by Eq. 7
        Update v by solving Eq. 6
    U(:, i) = u; V(:, i) = v;
return U, V
```

III. Experiment

A. Scene-15 Dataset

The scene-15 dataset [6], [28] contains a variety of indoor and outdoor scenes. In the experiments, we use three-level spatial pyramid, i.e., the 1×1, 2×2, 4×4 structures, and we randomly partition the images by 30 times. To incorporate the spatial pyramid pooling [6], we also include the symmetrically partitioned spatial layout as one of the 30 partitions. Following the same settings in [6], we use 100 images per class for training and the rest for testing. Table I compares the results of our proposed method and other related methods.

TABLE I
Accuracy Results on the 15-Scene Dataset

Algorithm	Accuracy (%)
Linear SPM [29]	65.32
Kernel SPM [6]	81.40
Kernel codebook [30]	76.67
Sparse coding SPM [29]	80.28
Locality linear coding [31]	79.24
Geometric ℓ_p-norm pooling [32]	83.20
Data-driven LBP [9]	87.2
Boosting + ORSP [8]	83.9
Boosting + BRSP [8]	88.1
Convnet + SVM [2]	84.2
Convnet + random partition [17]	89.4
Resnet + SVM [3]	92.29
Resnet + weighted layout (Ours)	**94.47**

From the table we can see that our proposed method explicitly utilizes the class-specific spatial layout information and achieves better performance than the direct competitor that uses the same features, i.e., the Resnet + SVM method. And thanks to the use of Resnet-152 features [3] and the class-specific spatial layouts, our method achieves the best performance among the methods using both traditional hand-crafted features and deep learning features. It is also worth noting that our proposed method significantly outperforms the previous Boosting + ORSP/BRSP methods [8], which encoded different spatial layout information into different patterns and then applied feature selection method to find the discriminative patterns for classification. Our method also outperforms previous work [17] that selected the max response from a set of randomly generated spatial layouts. We show the confusion matrix in Fig. 3.

The random partition coefficients matrix $\mathbf{V}$ in absolute values for 15 classes is also shown in Fig. 4. It is interesting to see that the first row (the evenly partitioned spatial pyramid) contributes most to the coefficients. We can also see that the discovered random partition coefficients are sparse vectors, e.g., class $1-9, 11, 14, 15$. As for class 10,12,13, although the coefficient

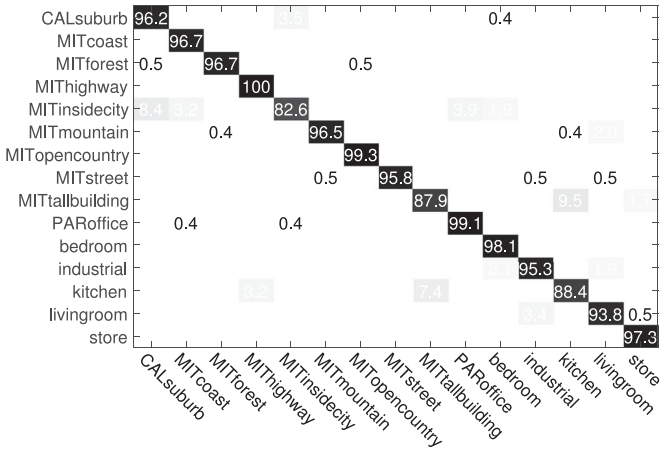

Fig. 3. Confusion matrix for scene-15 dataset.

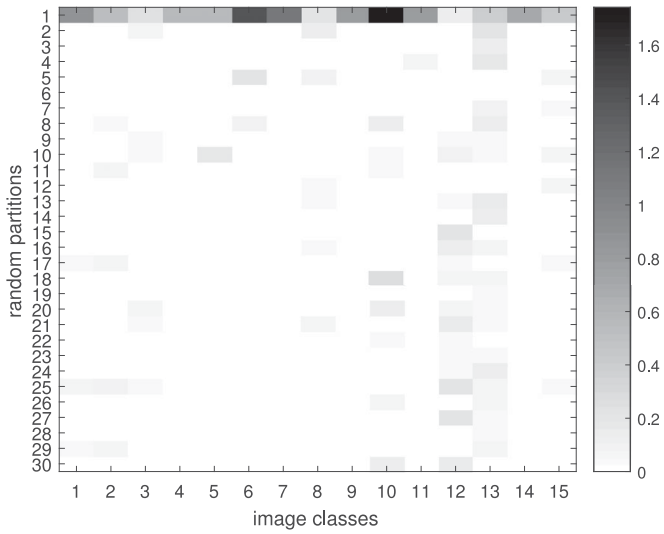

Fig. 4. Random partition coefficients for scene-15 dataset.

vectors are not so sparse, they are very different from other classes, which can lead to better classification performance. We also find that the discovered spatial layouts are discriminative if we compare one to the other, i.e., by using our proposed method we can select the class-specific spatial layouts which are discriminative for classification. The accuracy performance and the discovered random partition coefficients justify our proposed method of jointly learning the image classifier and the class-specific spatial layouts.

B. Landuse-21 Dataset

The landuse-21 dataset [33] consists of 21 classes of aerial orthoimagery from the United States Geological Survey National Map. For training/testing split, we follow the settings in [33] and use 80 images for each class for training and the rest 20 for testing. The spatial layout settings are the same as used in the scene-15 experiments. The results of our proposed method are shown in Table II. From the table we can see that our proposed method significantly outperforms previous hand-crafted features and also we can improve the baseline Resnet + SVM method.

TABLE II
ACCURACY RESULTS ON THE LANDUSE-21 DATASET

Algorithm	Accuracy (%)
BoVW [8]	71.9
Kernel SPM [6]	74.0
SPCK [33]	73.1
SPCK+[33]	76.1
SPCK++ [33]	79.24
Boosting + ORSP [8]	77.3
Boosting + BRSP [8]	75.5
Resnet + SVM [3]	98.10
Resnet + weighted layout (Ours)	**98.57**

TABLE III
ACCURACY RESULTS ON THE MIT INDOOR-67 DATASET

Algorithm	Accuracy (%)
D-Parts [35]	51.4
IFV [36]	60.8
MLrep [37]	64.0
Convnet + SVM [2]	58.4
Convnet + random partition [17]	62.0
Resnet + SVM [3]	78.88
Resnet + weighted layout (Ours)	**80.97**

C. MIT Indoor-67 Dataset

The MIT indoor-67 dataset [34] consists of 67 classes of indoor scene images. Due to the large number of classes and the large interclass variation, it is a more challenging dataset compared to the scene-15 and landuse-21 datasets. In the experiments, we use the standard training and testing split from [34], i.e., 80 images per class for training and 20 images per class for testing. The spatial layout settings are also the same as used in the scene-15 experiments. The results of our proposed method are shown in Table III. As can be seen from the table, our proposed method achieves the best performance among the methods using both hand-crafted features and deep convolution features. It is worth noting that our method can still improve the Resnet + SVM method by about 2% on such a big dataset with 67 classes of indoor scenes, which further justifies the advantages of our proposed method.

IV. CONCLUSION

Finding discriminative spatial layouts is critical to scene recognition. To discover class-specific spatial layouts, we first generate random spatial layouts and then learn weighted spatial layouts by an ℓ_1 regularized max-margin optimization problem for scene recognition. Our proposed joint learning of class-specific spatial layouts and image classifiers can improve the scene recognition performance compared to existing approaches that do not explicitly exploit the spatial layout information. Experiments on scene-15, landuse-21, and MIT indoor-67 datasets validate the advantages of our proposed algorithm.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 487–495.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[5] C. Weng and J. Yuan, "Efficient mining of optimal and/or patterns for visual recognition," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 626–635, 2015.

[6] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2169–2178.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.

[8] Y. Jiang, J. Yuan, and G. Yu, "Randomized spatial partition for scene recognition," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2012, pp. 730–743.

[9] J. Ren, X. Jiang, J. Yuan, and G. Wang, "Optimizing LBP structure for visual recognition using binary quadratic programming," *IEEE Signal Process. Lett.*, vol. 21, no. 11, pp. 1346–1350, Nov. 2014.

[10] Z. Zuo and G. Wang, "Learning discriminative hierarchical features for object recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1159–1163, Sep. 2014.

[11] B. Shuai, Z. Zuo, and G. Wang, "Quaddirectional 2D-recurrent neural networks for image labeling," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1990–1994, Nov. 2015.

[12] D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, "Supervised tensor learning," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2005, Art. no. 8.

[13] J. Yuan and Y. Wu, "Spatial random partition for common visual pattern discovery," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[14] M. S. Castelhano and A. Pollatsek, "Extrapolating spatial layout in scene representations," *Memory Cogn.*, vol. 38, no. 8, pp. 1018–1025, 2010.

[15] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1487–1494.

[16] C. Weng, H. Wang, and J. Yuan, "Learning weighted geometric pooling for image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2013, pp. 3805–3809.

[17] M. Yang, B. Li, H. Fan, and Y. Jiang, "Randomized spatial pooling in deep convolutional networks for scene recognition," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 402–406.

[18] Z. Yuan, H. Wang, L. Wang, T. Lu, S. Palaiahnakote, and C. L. Tan, "Modeling spatial layout for scene image understanding via a novel multiscale sum-product network," *Expert Syst. Appl.*, vol. 63, pp. 231–240, 2016.

[19] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3370–3377.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.

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

[22] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1470–1477.

[23] J. 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.

[24] Y. Xiao, J. Wu, and J. Yuan, "mCENTRIST: A multi-channel feature generation mechanism for scene categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 823–836, Feb. 2014.

[25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[27] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[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] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 186–190.

[30] J. van Gemert, J. M. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2008, pp. 1–7.

[31] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[32] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric LP-norm feature pooling for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2609–2704.

[33] Y. Yang and S. Newsam, "Spatial pyramid co-occurrence for image classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1465–1472.

[34] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.

[35] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3400–3407.

[36] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 923–930.

[37] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 494–502.