SCENE IMAGE CLASSIFICATION USING REDUCED VIRTUAL FEATURE REPRESENTATION IN SPARSE FRAMEWORK

Krishan Sharma *Shikha Gupta* *Dileep A.D* *Renu Rameshan*

School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi

ABSTRACT

In this paper, we address the task of scene image classification in sparse framework. Recent scene image datasets consist of thousands of different size images with size of the order of 10^6 pixels. Motivated by the fact that every image has a different size, we propose a dynamic kernel[1] which works over set of feature maps obtained for an image from last convolutional pooling layer of a pre-trained CNN. The size of feature maps depends on the input image size leading to the requirement of a dynamic kernel to compute similarity score between feature maps of different images. The kernel matrix obtained by using a dynamic kernel is large in size owing to the large number of training examples. To handle this we propose to use the concept of reduced virtual features (RVFs) obtained by diagonalizing the kernel matrix. RVF is a fixed length representation of a scene image irrespective of its true size. Classification is done in sparse framework by applying block sparsity constraint over sparse coefficients using dictionary built from RVFs. The proposed approach tested over standard datasets like Vogel-Schiele, MIT-8, MIT-67 and SUN-397 yields good results.

Index Terms— Convolution neural networks, block sparse representation, dynamic kernel, set of feature maps, scene image, reduced virtual features.

1. INTRODUCTION

Scene image classification is one of the active areas of research for pattern recognition community since a decade [1–5]. A scene may be indoor or outdoor and can consist of several entities (concepts) like mountain, chair, tree, car etc. These entities can be found common across different scene classes (like a car may be present in both garage and parking class) which results in high intra-class variability and less inter-class variability [6, 7]. The goal of this work is to classify a test image to its corresponding scene class. To accomplish this task, convolutional neural networks (CNNs) are used to extract features from the images. From [5, 8] one can infer that convolutional layers in CNN are responsible for generating the discriminative features by preserving spatial information. However CNN needs a fixed size input image (e.g., 227×227) which is obtained by reducing or enlarging the original image. This results in loss of several concepts information [5] from image present at very small scale and hence leads to poor discriminative features. So a better solution is to feed images to CNN in their true size. However, this solution is not feasible as fully connected layers of CNN architecture require a fixed length input. To overcome this problem, we feed images to CNN in their true size by considering the CNN architecture only upto the last convolutional pooling layer. At the output of last convolutional pooling layer, an input image gets represented as a set of feature maps $\mathcal{X}_m$, where

$$\mathcal{X}_m = \{\mathbf{x}_{m1}, ..., \mathbf{x}_{mi}, ..., \mathbf{x}_{mf}\}, \quad (1)$$

where $\mathbf{x}_{mi} \in \mathbb{R}^{m_p \times m_q}$. The number of feature maps in $\mathcal{X}_m$ is f, where f is the number of filters in last convolutional layer. The entire training data is represented by the set $\mathcal{X} = \{\mathcal{X}_1, ..., \mathcal{X}_m, ..., \mathcal{X}_N\}$ with N being the number of training examples. Here, the feature map size $(m_p \times m_q)$ depends on the true size of corresponding image. To measure the similarity between these varying size feature map sets ($\mathcal{X}_m$ and $\mathcal{X}_n$), a dynamic kernel known as deep spatial pyramid matching kernel (DSPMK) is proposed which uses L spatial pyramid levels to compute the similarity score between two sets of feature maps.

Since recent scene datasets contain thousand of images across large scene categories, the kernel matrix generated using DSPMK is of $N \times N$ size and has high memory complexity of the order $O(N^2)$, where N denotes the total training examples. Our objective is to generate a reduced set of features without compromising the classification accuracy. To achieve this, we introduce the concept of reduced virtual features (RVFs). These RVFs are d dimensional feature vectors $(d << N)$ which are obtained by diagonalizing the kernel matrix. Classification is done in sparse framework by applying block sparsity constraint [9] over coefficients obtained in training RVFs basis. The core contributions of this work are as follows:

- Obtaining deep varying length discriminative sets of feature maps from last convolutional pooling layer by passing the input images to CNN in their true size.
- A novel DSPMK is proposed to find the similarity score between these varying length deep feature maps sets.

[1] A dynamic kernel measure similarity between varying length features.

- Generating d-dimensional RVFs representation irrespective of image true size by diagonalizing the kernel matrix.

- Classifying the test image in sparse framework by applying the block sparsity constraint over obtained coefficients.

The paper is organized as follows: Section 2 gives brief description about the proposed DSPMK with details given in Algorithm 1. The proposed RVFs framework and block sparse representation classifier are explained in Section 3 and 4, respectively. Experimental analysis is given in Section 5 followed by conclusion in Section 6.

2. THE PROPOSED DYNAMIC KERNEL

Inspired by the work of [5, 10], we propose a deep spatial pyramid matching kernel (DSPMK) for scene classification task. DSPMK measures the similarity score between two sets of feature maps $(\mathcal{X}_m, \mathcal{X}_n)$ having same or different size. A known fact for the scene images is that the spatial arrangement of entities present in image remain unchanged to some extent for all images of same class irrespective of size of entity. DSPMK makes use of this fact to find similarity score between corresponding spatially divided blocks of generated feature maps. When an RGB image $I_m \in \mathbb{R}^{m'_p \times m'_q \times 3}$ is given as input to CNN in its original size, set of feature maps $\mathcal{X}_m$ (equation 1) is generated from last convolutional pooling layer of CNN. This set can be compactly represented as an element in $\mathbb{R}^{m_p \times m_q \times f}$. DSPMK operates over set of features maps $\mathcal{X}_m$ at several pyramid levels. At each pyramid level, feature maps are divided into fixed number of spatial blocks and a feature vector is generated by sum pooling over blocks. Level-wise similarity score between these feature vectors is obtained by using histogram intersection function. Final matching score K_{DSPMK} is obtained by weighted combination of level-wise similarity score as given in Algorithm 1. K_{DSPMK} is used to generate the $N \times N$ kernel matrix $\mathbf{K}_{train}$.

3. REDUCED VIRTUAL FEATURES GENERATION

The kernel matrix, $\mathbf{K}_{train}$ generated using Algorithm 1 gives the measure of similarity between two images in high dimensional space $\mathcal{F}$. The main difficulties while working with the kernel matrix are: (1) the size of kernel matrix is dependent on the total number of training examples, which restricts its use to small datasets in order to avoid increase in memory and run time complexity, (2) only similarity between the examples gets reflected in $\mathbf{K}_{train}$, i.e. $\mathbf{K}_{train}$ is discriminative rather than descriptive and does not define an example in terms of its own features value as defined in input space. To generate a descriptive fixed length representation of scene image we propose to use the concept of virtual samples [11]. Here an approximation to the original sample, in the space F is obtained as explained below. From the virtual sample we derive the RVF.

Algorithm 1 Deep spatial pyramid matching kernel $K_{DSPMK}(\mathcal{X}_m, \mathcal{X}_n)$

Inputs:

(i) Feature maps set $\mathcal{X}_m$ and $\mathcal{X}_n$, where
$\mathcal{X}_m = \{\mathbf{x}_{m1}, ..., \mathbf{x}_{mi}, ..., \mathbf{x}_{mf}\}$; where $\mathbf{x}_{mi} \in \mathbb{R}^{m_p \times m_q}$
$\mathcal{X}_n = \{\mathbf{x}_{m1}, ..., \mathbf{x}_{ni}, ..., \mathbf{x}_{nf}\}$; where $\mathbf{x}_{ni} \in \mathbb{R}^{n_p \times n_q}$

(ii) L: number of pyramid levels.

Procedure:

for l=0 **to** $L-1$ **do**

Divide each feature map of $\mathcal{X}_m$ into 2^{2l} blocks.
$\mathcal{X}_m^l = \{\mathbf{x}^l_{m1(1)} ... \mathbf{x}^l_{m1(2^{2l})}, ..., \mathbf{x}^l_{mi(1)} ... \mathbf{x}^l_{mi(2^{2l})}, ..., \mathbf{x}^l_{mf(1)} ... \mathbf{x}^l_{mf(2^{2l})}\}$.

Apply sum pooling over each block such that
$x^l_{mi(j)} = \sum_u \sum_v \mathbf{x}^l_{mi(j)}(u,v)$
$\mathbf{X}^l_m = [x^l_{m1(1)} ... x^l_{m1(2^{2l})}, ..., x^l_{mi(1)} ... x^l_{mi(2^{2l})}, ..., x^l_{mf(1)} ... x^l_{mf(2^{2l})}] \in \mathbb{R}^{(2^{2l} \times f) \times 1}$.

ℓ_1- normalize the generated feature vector $\mathbf{X}^l_m$
$\hat{\mathbf{X}}^l_m = [\hat{x}^l_{m1(1)} ... \hat{x}^l_{m1(2^{2l})}, ..., \hat{x}^l_{mi(1)} ... \hat{x}^l_{mi(2^{2l})}, ..., \hat{x}^l_{mf(1)} ... \hat{x}^l_{mf(2^{2l})}] \in \mathbb{R}^{(2^{2l} \times f) \times 1}$.

Compute intermediate matching score using histogram intersection function
$\Gamma_l = \sum_{j=1}^{f} \sum_{k=1}^{2^{2l}} min(\hat{x}^l_{mj(k)}, \hat{x}^l_{nj(k)})$.

end for

Compute final matching score between $\mathcal{X}_m$ and $\mathcal{X}_n$
$K_{DPSMK} = \sum_{l=0}^{L-2} \frac{1}{2^{(L-l-1)}} (\Gamma_l - \Gamma_{l+1}) + \Gamma_{L-1}$.

Outputs:

(i) $K_{DSPMK}(\mathcal{X}_m, \mathcal{X}_n)$.

Since kernel matrix satisfies Mercer's conditions [12], each of its entry is an inner product, $\mathbf{K}_{train}(m,n) = \phi(\mathcal{X}_m)^\top \phi(\mathcal{X}_n)$, where $\phi(\mathcal{X}_m)$ is $\mathcal{X}_m$ mapped to higher dimensional feature space $\mathcal{F}$. This can be generalized to the full kernel matrix as: $\mathbf{K}_{train} = \Phi(\mathcal{X})^\top \Phi(\mathcal{X})$, where $\Phi(\mathcal{X})$ is the representation of complete training data in $\mathcal{F}$. Since kernel matrix is symmetric and positive semidefinite, it can be diagonalized as:

$$\mathbf{K}_{train} = \mathcal{U} \Sigma_N \mathcal{U}^\top, \tag{2}$$

Where $\Sigma_N \in \mathbb{R}^{N \times N}$ is a diagonal matrix with diagonal entries as the singular values arranged as $\sigma_1 > \sigma_2 > ... > \sigma_N$. $\mathcal{U}$ contains orthogonal eigenvectors arranged according to these singular values. Kernel matrix would also have the form $\mathbf{K}_{train} = (\hat{\boldsymbol{\psi}}^N_{train})^\top (\hat{\boldsymbol{\psi}}^N_{train}) = \Phi(\mathcal{X})^\top \Phi(\mathcal{X}) = \mathcal{U} \Sigma_N \mathcal{U}^\top$. Here $\hat{\boldsymbol{\psi}}^N_{train} \in \mathbb{R}^{N \times N}$ is the N-dimensional virtual feature

Algorithm 2 The proposed framework

Inputs:

(i) Training scene image database $\mathcal{D} = \{I_j, \mathbf{t}_j\}_{j=1}^{N}$, where $\mathbf{t}_j \in 1, 2, ..., c$ scene classes.

(ii) Test scene image example I_{test}.

(iii) Pre-trained CNN model.

1: **Procedure:**

2: Extract the varying length feature maps sets $\{\mathcal{X}_j\}_{j=1}^{N}$ and $\mathcal{X}_{test}$ of $\{I_j\}_{j=1}^{N}$ and I_{test} respectively from last convolutional pooling layer of pre-trained CNN.

3: Compute $\mathbf{K}_{train}$ and $\mathbf{k}(.,\mathcal{X}_{test})$ using kernel function $\mathbf{k}(.,.)$ from Algorithm 1.

4: Apply SVD decomposition over $\mathbf{K}_{train}$, $\mathbf{K}_{train} = \mathcal{U}\Sigma_N\mathcal{U}^\top$.

5: Generate the reduced virtual features of dimension d $(d << N)$

$$\hat{\boldsymbol{\psi}}^d_{train} = {\Sigma_d}^{-\frac{1}{2}}\mathcal{U}^\top\mathbf{K}_{train},$$
$$\hat{\mathbf{y}}^d_{test} = {\Sigma_d}^{-\frac{1}{2}}\mathcal{U}^\top\mathbf{k}(.,\mathcal{X}_{test}),$$

where $\Sigma_d = \Sigma_N(1:d, 1:N)$.

6: Solve equation (6) to obtain sparse coefficient $\hat{\boldsymbol{\alpha}}$.

7: Obtain label using equation (7) by minimizing residual error.

$$label(\hat{\mathbf{y}}^d_{test}) = \underset{i=1,2,...,c}{\arg\min}\ ||\hat{\mathbf{y}}^d_{test} - \hat{\boldsymbol{\psi}}^d_{train}\boldsymbol{\xi}_i||_2^2.$$

Outputs:

(i) $label(I_{test})$.

representation of complete training data and can be written as:

$$\hat{\boldsymbol{\psi}}^N_{train} = {\Sigma_N}^{-\frac{1}{2}}\mathcal{U}^\top\mathbf{K}_{train}, \tag{3}$$

It may be noted that the inverse in equation (3) is only for the non-zeros entries along the diagonal. RVFs representation is obtained by selecting d topmost eigenvalues.

$$\hat{\boldsymbol{\psi}}^d_{train} = {\Sigma_d}^{-\frac{1}{2}}\mathcal{U}^\top\mathbf{K}_{train}, \tag{4}$$

where, $\Sigma_d \in \mathbb{R}^{d\times N}$ is the first d rows of Σ_N and $\hat{\boldsymbol{\psi}}^d_{train} \in \mathbb{R}^{d\times N}$ is the training data matrix of d-dimensional RVFs.

4. BLOCK SPARSE REPRESENTATION BASED CLASSIFIER

In sparse representation [17, 18], a signal is expressed as a combination of few signals/atoms from a dictionary. We consider the dictionary formed from RVFs training data ($\hat{\boldsymbol{\psi}}^d_{train} \in \mathbb{R}^{d\times N}$) of all c scene classes. Sparse representation for the test feature $\hat{\mathbf{y}}^d_{test}$ can be obtained by solving:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\arg\min} \quad \lambda||\boldsymbol{\alpha}||_1 + ||\hat{\mathbf{y}}^d_{test} - \hat{\boldsymbol{\psi}}^d_{train}\boldsymbol{\alpha}||_2^2 \tag{5}$$

where $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_1 ... \hat{\boldsymbol{\alpha}}_i ... \hat{\boldsymbol{\alpha}}_c] \in \mathbb{R}^N$ is the sparse representation of $\hat{\mathbf{y}}^d_{test}$ and $\hat{\boldsymbol{\alpha}}_i$ denotes i^{th} class coefficient vector. However, the major drawback of equation (5) is that it does not consider the multiple low dimensional subspace structure for class specific training data. This is overcome by adding block sparsity constraint [2] in equation (5).

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\arg\min} \quad \lambda\sum_{j=1}^{m}||\boldsymbol{\alpha}[j]||_q + ||\hat{\mathbf{y}}^d_{test} - \hat{\boldsymbol{\psi}}^d_{train}\boldsymbol{\alpha}||_2^2 \tag{6}$$

In equation (6), first term denotes the block sparsity constraint using ℓ_q norm with λ as trade-off parameter. Theoretical guarantees given in [9] prove that for $q \geq 1$, optimization problem (6) is convex and can be solved using any convex optimization tool. A test signal $\hat{\mathbf{y}}^d_{test}$ is classified to that class which minimizes the representation residual ($||\hat{\mathbf{y}}^d_{test} - \hat{\boldsymbol{\psi}}^d_{train}\boldsymbol{\xi}_i||_2^2$), where $\boldsymbol{\xi}_i$ define a characteristic function which picks up the coefficients corresponding to i^{th} class.

$$\boldsymbol{\xi}_i = \begin{cases} \hat{\boldsymbol{\alpha}}[j], & \forall j \in i^{th}\text{class} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Pseudo code for proposed classification method is given in Algorithm 2.

5. EXPERIMENTAL ANALYSIS

In this section, the effectiveness of RVFs in sparse framework with block sparsity constraint for scene classification task is studied. VGGNet-16 architecture [13] already trained on Places205 [15], Places365 [16] and ImageNet [14] is considered. Pre-trained VGGNet-16 architecture is used as it is upto last convolutional pooling layer. Set of feature maps for a given scene image is obtained from last convolutional pooling layer. Number of feature maps for VGGNet-16 is 512 and size of feature map is dependent on size of input image fed to CNN. DSPMK operates over the given sets generates the kernel matrix. RVFs are generated by diagonalizing the kernel matrix. Datasets used for classification task are (i) Vogel-Schiele (VS) [19], (ii) MIT-8 [20], (iii) MIT-67 [7] and (iv) SUN-397 [6]. The description of these dataset is given below.

(i) *Vogel Schiele dataset* [19] consists of 6 semantic classes, namely, 'coast', 'river', 'forest', 'mountain', 'open-country' and 'sky-cloud' with total of 700 images. Results are produced in terms of average classification accuracy using 5-fold stratified divisions.

(ii) *MIT-8 scene dataset* [20] comprises of 8 scene classes, namely, 'tall building', 'street', 'inside-city', 'highway', 'coast', 'mountain', 'forest' and 'open-country' with total of 2688 images. We randomly selected 100 images per class as training and rest as testing examples in 5-fold to obtain average classification accuracy.

[2] we consider m blocks for c classes, where each class may have one or more blocks

Table 1. Classification accuracies using our proposed approach (DSPMK + RVFs + BSRC) on different datasets. Base features for the proposed method are extracted using VGGNet-16 [13] which is pre-trained network on ImageNet [14], Places-205 [15] and Places-365 [16] datasets. d : RVF dimension, N: total training examples. Results are shown for BSRC with ℓ_q norm (q = 1, 2).

VGGNet-16 architecture pre-trained using	**Vogel-Schiele**		**MIT-8 scene**		**MIT-67**		**SUN-397**	
	d = 300, N = 559		d = 300, N = 800		d = 1000, N = 5360		d = 2000, N = 19850	
	q = 1	q = 2	q = 1	q = 2	q = 1	q = 2	q = 1	q = 2
ImageNet dataset [14]	84.22	84.16	94.06	94.39	73.16	74.82	51.15	52.67
Places-205 dataset [15]	84.23	**84.64**	94.82	95.00	78.81	**80.01**	58.92	59.73
Places-365 dataset [16]	83.56	83.65	94.90	**95.11**	77.41	78.92	59.81	**60.63**

Table 2. Comparison of classification accuracies with state-of-the-art methods. (SIFT: Scale invariant feature transform, IFK: Improved Fisher kernel, BoP: Bag of part, MOP: Multiscale orderless pooling, FV: Fisher vector, DSP: Deep spatial pyramid)

Method	**Vogel-schiele**	**MIT-8 scene**	**MIT-67**	**SUN-397**
SIFT + BOVW [1]	67.49	79.13	45.86	24.82
IFK + BoP [2]	73.23	85.76	63.18	-
MOP-CNN [3]	76.81	89.45	68.88	51.98
Places-CNN-fc7 [15]	76.02	88.30	68.24	54.32
Hybrid-CNN-fc7 [15]	78.56	91.23	70.80	53.86
fc8 + FV [4]	79.56	88.43	72.86	54.40
VGGNET-16 + DSP [8]	81.34	92.34	76.34	57.27
DSPMK + RVFs + SVM	83.45	94.16	78.52	58.82
Proposed approach (DSPMK + RVFs + BSRC)	**84.64**	**95.11**	**80.01**	**60.63**

(iii) *MIT-67 dataset* [7] is an indoor scene dataset with total of 15620 scene images having 67 classes. This is quite challenging dataset as interclass variation is very less. Classification results are reported on the standard split available with approx 80 training and 20 testing examples per class.
(iv) *SUN-397 dataset* [6]- is a very huge dataset for scene classification with 397 classes including nature, indoor and urban categories. Dataset split is publicly available with 50 training and 50 testing images per class. Classification results are reported in terms of average accuracy of 3-fold.

Here we compare the performance of SVM and BSRC for classifying RVFs. However, SVM has following drawbacks: (1) we need to choose a kernel function which generates kernel matrix of size ($N \times N$) hence increases memory complexity, (2) high parameter tuning is required according to chosen kernel. So we propose to use BSRC classifier which uses dictionary of size ($d \times N$) formed from RVFs obtained from training data. Memory complexity of BSRC ($d \times N$) is less than that of SVM ($N \times N$) as ($d << N$). Table 1 shows the classification results obtained using our approach over different datasets. No parameter tuning is required in our proposed framework except reconstruction error in BSRC which is set to 0.0005. We observe that VGGNet-16 pre-trained on Places datasets gives better classification accuracies in comparison to the one trained over ImageNet which is expected since the former are scene datasets and later is object dataset. It is also evident that performance of block sparsity constraint with ℓ_2 *norm* is better than ℓ_1 *norm*. Comparison of classification accuracies with state-of-the-art methods is shown in Table 2. Results show that classification accuracy of BSRC with ℓ_2 *norm* using RVFs as dictionary atoms is better than that of SVM classifier with linear kernel over RVFs.

Recent papers which use multi-resolution or complex deep architectures [21, 22] with specific training yield better results. But our approach does not require any training and parameter tuning which we are projecting as advantage of our approach.

6. CONCLUSION

In this paper, we have proposed scene image classification in sparse framework using block sparsity constraint. Sets of feature maps are generated using pre-trained CNN from last convolutional pooling layer by passing images to CNN in true size. A novel dynamic kernel known as deep spatial pyramid matching kernel (DSPMK) is also proposed to generate kernel matrix. Reduced virtual features (RVFs) representation is obtained by diagonalizing the kernel matrix. Dictionary is built using the RVFs obtained from training images as atoms. Classification of test image is performed in sparse framework by imposing block sparsity constraint. The results obtained are better despite reduced size with the added advantage that no training is required.

7. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 923–930.

[3] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European conference on computer vision*. Springer, 2014, pp. 392–407.

[4] Mandar Dixit, Si Chen, Dashan Gao, Nikhil Rasiwasia, and Nuno Vasconcelos, "Scene classification with semantic Fisher vectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2974–2983.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[6] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010, pp. 3485–3492.

[7] Ariadna Quattoni and Antonio Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 413–420.

[8] Bin-Bin Gao, Xiu-Shen Wei, Jianxin Wu, and Weiyao Lin, "Deep spatial pyramid: The devil is once again in the details," *arXiv preprint arXiv:1504.05277*, 2015.

[9] Ehsan Elhamifar and René Vidal, "Block-sparse recovery via convex optimization," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4094–4107, 2012.

[10] Svetlana Lazebnik, Cordelia Schmid, and Jean 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*. IEEE, 2006, vol. 2, pp. 2169–2178.

[11] Kai Zhang, Liang Lan, Zhuang Wang, and Fabian Moerchen, "Scaling up kernel svm on limited resources: A low-rank linearization approach," in *Artificial Intelligence and Statistics*, 2012, pp. 1425–1434.

[12] Bernhard Schölkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[15] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[16] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[17] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[18] Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry, "Feature selection in face recognition: A sparse representation perspective," *submitted to IEEE Transactions Pattern Analysis and Machine Intelligence*, 2007.

[19] Julia Vogel and Bernt Schiele, "Natural scene retrieval based on a semantic modeling step," in *CIVR*. Springer, 2004, pp. 207–215.

[20] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[21] Zhe Wang, Limin Wang, Yali Wang, Bowen Zhang, and Yu Qiao, "Weakly supervised patchnets: Describing and aggregating local patches for scene recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2028–2041, 2017.

[22] Limin Wang, Sheng Guo, Weilin Huang, Yuanjun Xiong, and Yu Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2055–2068, 2017.