SPATIAL PYRAMID ALIGNMENT FOR SPARSE CODING BASED OBJECT CLASSIFICATION

Joonsoo Kim, Khalid Tahboub and Edward J. Delp

Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

ABSTRACT

The bag of visual words (BOW) model is widely used for image representation and classification. Spatial pyramid based feature pooling utilizes the BOW model and is the most popular approach to capture the spatial distribution (layout) of local image features. It makes the assumption that the center of an object is aligned with the center of an image, which can lead to misalignment and degradation in performance. In this paper, we propose a method to utilize max pooled features to estimate objects centers and align the spatial pyramid accordingly. We also propose an image representation descriptor robust to misalignments and objects deformations. The experimental results demonstrate that our spatial pyramid alignment method is simple yet efficient in handling misalignments and achieves high object classification accuracy.

Index Terms— object classification, spatial pyramid, feature coding, spatial pyramid alignment

1. INTRODUCTION

The bag of visual words (BOW) [1] model has been widely used for image representation. Using the BOW model an image is represented based on its local image features. Local image features are high dimensional vectors representing visual content in an image. The goal of local features extraction and using the BOW model is to obtain an image representation that can be used to match similar objects. Dense SIFT (Dense Scale Invariant Feature Transform) [2–4] is an example of a method used to extract local features. In Dense SIFT, single SIFT descriptor is generated on each of densely sampled pixel locations. The single SIFT descriptor is used as a local image feature.

In the BOW model, each local feature is coded using a set of predefined codewords. The set of codewords is referred to as the codebook. Coding features using the codebook is analogous to representing vectors using a set of basis vectors. The codebook is usually constructed using a set of local image features randomly sampled from the training images. The feature coding vector is the output of the coding process. It is comprised of a set of coefficients where each coefficient is the contribution of a particular codeword in representing the feature. Vector quantization (VQ) [5, 6] simplifies the coding process by assuming that a feature can be represented by a single codeword. Therefore, all the elements of the coding vector are zeros except for a single element corresponding to the codeword closet to this feature. However, vector quantization is not adequate to represent the variation of features. This causes degradation in the performance of image representation and classification. Sparse coding [2, 7–16] has been utilized to address this problem. In sparse coding each local image feature is represented by a combination of a small number of codewords.

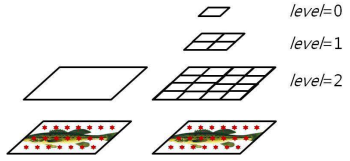


Fig. 1: Example of the BOW model with the spatial pyramid

The final image representation is based on the coding vectors of all the local image features. To combine multiple coding vectors into a single vector, average or max pooling [17] is utilized. In average pooling, the final image representation vector is computed by averaging all the coding vectors, whereas max pooling uses the maximum value of each element among all the coding vectors separately. Figure 1 shows an example where dense SIFT local features (represented by red stars) are extracted from an image. The basic BOW model combines the features coding vectors (left side in Figure 1 without considering the spatial distribution (layout) of the local image features. This means that the spatial locations of the local images features are not used in the BOW model. This drawback of the BOW model limits the descriptive power of the final image representation.

To address this problem, spatial pyramid feature pooling (SPP) [2, 5, 7, 10–15] has been proposed and incorporated in most feature coding methods. To construct a spatial pyramid, an image is partitioned into $2^l \times 2^l$ subregions at different l^{th} levels (l=0,1,2), as shown in Figure 1 (right). The first level consists of 16 subregions, whereas the second level contains 4 subregions and the third one is a single region. Instead of using max or average pooling on the entire image, SPP is done on each subregion. The final image representation of SPP is the concatenation of all the subregions representation vectors. Existing methods which are based on SPP assume that the center of an object is aligned with the center of the image. Therefore, the center of the image is used as the center of the spatial pyramid. However, the center of most images is not aligned with the center of objects correctly. This misalignment propagates in feature pooling results in several subregions at multiple pyramid levels.

In this paper, we propose a method to estimate the center of various objects and propose to align the spatial pyramid center ac-

This work was funded by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001. Address all correspondence to Edward Delp (ace@ecn.purdue.edu).

cordingly. Our proposed method is based on the spatial layout of the max pooled features. We also propose a final image representation descriptor robust to the misalignment of the spatial pyramid center. The experimental results demonstrate that our method is simple yet efficient in handling misalignments issues.

2. REVIEW OF EXISTING METHODS

Codebook construction is essential to accurate image representation and classification. Many sparse coding methods have been proposed to learn accurate codebooks. In [15] a sparse representation for face recognition is shown to perform better than the conventional face recognition representations. In [16] the importance of an accurate codebook generation is addressed. K-SVD is also introduced to learn a codebook accurately. In [9] discriminative K-SVD is proposed. K-SVD learns a codebook and a classifier jointly in an unsupervised approach, which results in an improved image classification accuracy. In [7] a supervised codebook learning method based on K-SVD is proposed. In [18] a kernel function is used to generate a codebook, and is shown to improve the classification accuracy. In [13], the hierarchical sparse coding is proposed. Feature coding and pooling are used on local image features sequentially to generate subregion representation vectors. The same feature coding method is used on the subregion representation vectors. This hierarchical coding achieves image representation robust to objects deformations. In [14] multiple image patches are used as local image features. These features are encoded using K-SVD based spare coding in combination with a hierarchical sparse coding scheme.

In [2, 3, 5] the combination of the feature coding and the spatial pyramid is investigated. The classification performance is improved by combining the vector quantization with the spatial pyramid in [5]. In [2] the sparse coding is used with spatial pyramid instead of vector quantization. It shows that sparse coding with spatial pyramid can represent an image with more discriminative power. In [3] the importance of locality constrained feature coding (LLC) is addressed. The locality constraint is shown to be more powerful than the sparsity constraint in coding features.

In [10, 11] geometric feature pooling methods are described. Based on the assumption that the same class image shares similar spatial layouts, the weighted spatial pooling function is learned. In [19] the regions for feature pooling are learned. Instead of dividing the entire image using a spatial pyramid grid, a hierarchical structure of subregions is learned and is shown to improve the classification accuracy. In [12] a spare coding method based on an ensemble of classifiers is proposed. This method is shown to be robust against overfitting.

Most existing methods described here use spatial pyramids and feature coding methods to improve the classification accuracy, However, the proper alignment between the center of an object and the center of the spatial pyramid is not addressed.

3. CENTER-ALIGNED SPATIAL PYRAMID (CASP) BASED OBJECT CLASSIFICATION

Figure 2 shows the block diagram of our proposed object classification method. First, the dense local image features are extracted from an image. Coding vectors for all the local image features are generated using sparse coding. Instead of generating a regular spatial pyramid grid, we generate a Center-Aligned Spatial Pyramid (CASP) by estimating the center of the object in the image and aligning the center of the pyramid accordingly. Max pooling of all the feature coding vectors of the entire image is used to estimate the center of the object. Final image descriptor is found using max pooling in our CASP. Furthermore, to make our image descriptor robust to object deformations, we generate multiple CASP based image descriptors. Each descriptor is obtained by shifting the estimated center in a pre-defined margin. The maximum value of each element among all the image descriptors is computed to generate our final image descriptor. For image classification, the linear support vector machine (SVM) [20, 21] is used.

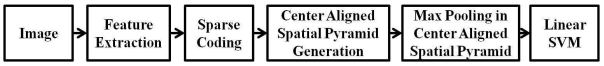


Fig. 2: System Overview

3.1. Sparse Coding and Max Pooling

To generate a feature coding vector for a local image feature, we use sparse coding in [2]. Once the local image feature is extracted from an image, the feature coding and codebook learning process is done by minimizing the following formula.

$$\begin{aligned} &\underset{\mathbf{u_m},\mathbf{V}}{\operatorname{argmin}} \sum_{m=1}^{M} \|\mathbf{f_m} - \mathbf{V u_m}\|^2 + \lambda\|\mathbf{u_m}\| \\ &s.t. \|\mathbf{v_c}\| \leq 1 \quad m = 1, 2, .., M \end{aligned} \tag{1}$$

where $\mathbf{f_m} \in \mathbb{R}^{\mathrm{d}\times 1}$ is the m^{th} local image feature vector, $\mathbf{u_m} = [u_{1m}, u_{2m}, .., u_{cm}, ..]^T \in \mathbb{R}^{\mathrm{C}\times 1}$ is a feature coding vector for $\mathbf{f_m}$,$\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, .., \mathbf{v_C}] \in \mathbb{R}^{\mathrm{d}\times C}$ is a codebook, $\mathbf{v_c}$ is the c^{th} codeword in the codebook, M is the total number of the local image features in an image, and C is the size of the feature coding vector. For minimizing this , "feature-sign search" [22] is used. Once all the feature coding vectors for all the local image features are generated, max pooling is used to generate a subregion descriptor using all the feature coding vectors within the subregion in spatial pyramid. Let z_c^{ijl} be the max value of the c^{th} element of all the feature coding vectors within the $(i, j)^{th}$ subregion at the l^{th} pyramid level.

$$\begin{aligned} z_c^{ijl} &= max(w_1^l|u_{c1}|, w_2^l|u_{c2}|, .., w_m^l|u_{cm}|, .., w_M^l|u_{cM}|) \\ w_m^l &= \begin{cases} 1 & if \quad \mathbf{p_m} \in \mathbf{R}_{ij}^l \\ 0 & otherwise \end{cases} \qquad i, j = 1, 2, .., 2^l \end{aligned} \tag{2}$$

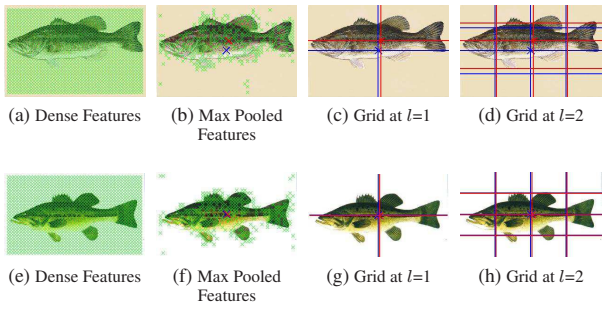
where $\mathbf{R}_{ij}^l$ is $(i, j)^{th}$ subregion at l^{th} level in pyramid, $\mathbf{p_m}$ is the spatial coordinate of $\mathbf{f_m}$, and $\mathbf{z^{ijl}} = [z_1^{ijl}, z_2^{ijl}, .., z_C^{ijl}]$ is the subregion representation vector for $\mathbf{R}_{ij}^l$.

3.2. Center-Aligned Spatial Pyramid Generation

BOW models using dense local image features have demonstrated higher accuracy in object recognition compared to spare features (i.e. interest points) [1, 4, 23]. One of the famous dense local image features is a Dense SIFT (Dense Scale Invariant Feature Transform). In Dense SIFT, single SIFT descriptor is generated on each of densely sampled pixel locations. The single SIFT descriptor is used as a local image feature. Since the center of an object is difficult to estimate from dense local image features, the center of the spatial pyramid is assumed to be the center of the image. However, when considering multiple images containing the same object, the misalignments between the centers of the spatial pyramids and the object center cause parts of the object to lie in different subregions within the spatial

pyramids. This misalignment propagates in feature pooling results in several subregions at multiple pyramid levels.

In Figure 3 two images for the same class (fish) illustrate this problem. The blue lines represent the grid of the original spatial pyramid and the red lines represent the grid of our CASP. As shown in Figure 3, our CASP splits the same parts of two object into the same subregions, but the original spatial pyramid does not. To find



(a) Dense Features (b) Max Pooled Features (c) Grid at l=1 (d) Grid at l=2

(e) Dense Features (f) Max Pooled Features (g) Grid at l=1 (h) Grid at l=2

Fig. 3: Example of Center-Aligned Spatial Pyramid Generation: Blue color represents the original spatial pyramid grid and red color represents our CASP grid

the center of an object, we describe a simple but efficient method based on max pooled features on the entire image. Since the max pooled features indicates the most salient parts of an object, the spatial layout of the max pooled features can be used to estimate the center of an object. We find the index of the most salient feature, m' that has the maximum response to the c^{th} codeword in $\mathbf{V}$:

$$m' = \underset{m}{\text{argmax}} \; (\; w_m^0 |u_{cm}| \;) \quad m = 1, 2, .., M \tag{3}$$

where $m' = 1, 2, .., M'$, M' is the number of the max pooled features in the entire image. Note that we repeat this process from c=1,..,C, and we do not find the max pooled feature index, m' if the maximum response for c^{th} codeword is zero. Therefore, the number of max pooled features, M' cannot be larger than the feature coding vector size, C. Then, the center of an object, $\mathbf{p_{ct}}$ is estimated as follows:

$$\mathbf{p_{ct}} = \sum_{m'=1}^{M'} \mathbf{p_{m'}} \qquad where \quad \mathbf{p_{ct}} = (x_{ct}, y_{ct}) \tag{4}$$

Once the center of a spatial pyramid is estimated, our CASP is constructed based on subregions of it defined as:

$$\begin{aligned}
\mathbf{R_{ij}^l} &= \{(x,y) | x_l < x \leq x_h \; and \; y_l < y \leq y_h\} \\
x_l &= \tfrac{x_{ct}}{2^{l-1}}(i-1) \\
x_h &= \tfrac{x_{ct}}{2^{l-1}} i \qquad\qquad i \leq 2^{l-1} \\
x_l &= \tfrac{\mathrm{w}-\mathrm{x_{ct}}}{2^{l-1}} i + 2x_{ct} - \mathrm{w} \\
x_h &= \tfrac{\mathrm{w}-\mathrm{x_{ct}}}{2^{l-1}}(i+1) + 2x_{ct} - \mathrm{w} \qquad i > 2^{l-1} \\
y_l &= \tfrac{y_{ct}}{2^{l-1}}(j-1) \\
y_h &= \tfrac{y_{ct}}{2^{l-1}} j \qquad\qquad j \leq 2^{l-1} \\
y_l &= \tfrac{\mathrm{h}-\mathrm{y_{ct}}}{2^{l-1}} j + 2y_{ct} - \mathrm{h} \\
y_h &= \tfrac{\mathrm{h}-\mathrm{y_{ct}}}{2^{l-1}}(j+1) + 2y_{ct} - \mathrm{h} \qquad j > 2^{l-1}
\end{aligned} \tag{5}$$

where h and w are the image height and width, respectively. The final image descriptor is the concatenation of all the max pooled feature vector over all the subregions in the CASP centered at $\mathbf{p_{ct}}$

$$D_{\mathbf{p_{ct}}} = [\mathbf{z^{ijl}}] \qquad i, j = 1, 2, ..2^l, l = 0, 1, 2 \tag{6}$$

Note that the c^{th} element of $D_{\mathbf{p_{ct}}}$ is obtained by finding the maximum value among all the c^{th} elements of all the feature coding vectors within the $(i,j)^{th}$ subregion at the l^{th} pyramid level. We also propose to add new subregions. These subregions are formed by connecting the centers of the four nearest subregions at each level of CASP. Figure 4 illustrates this concept. In Figure 4 left one is our previous CASP and right one is our modified CASP (MCASP). Note that the newly added subregions (shown in red) describe the parts of object more in detail.

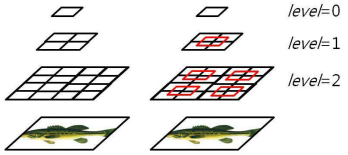


Fig. 4: Our Modified Center-Aligned Spatial Pyramid (MCASP) Structure

Let the five subregion descriptors additionally added at l =1,2 be $\mathbf{z_{ad1}^1}$, $\mathbf{z_{ad1}^2}$, $\mathbf{z_{ad2}^2}$, $\mathbf{z_{ad3}^2}$, and $\mathbf{z_{ad4}^2}$ and the concatenation of these descriptors be $D_{\mathbf{p_{ct}}}^{add}$. Then, our image descriptor is generated by concatenating $D_{\mathbf{p_{ct}}}$ and $D_{\mathbf{p_{ct}}}^{add}$.

$$D'_{\mathbf{p_{ct}}} = [D_{\mathbf{p_{ct}}}, D_{\mathbf{p_{ct}}}^{add}] = [d_1^{\mathbf{p_{ct}}}, d_2^{\mathbf{p_{ct}}}, ..] \tag{7}$$

3.3. Image Descriptor Robust to Object Deformation

Even though our proposed CASP is aligned with the center of the object of interest, object deformations might still occur and cause degradation in performance. When an object is deformed, its parts might be shifted from one subregion to an adjacent one. This is referred to spatial quantization error in spatial pyramid. To solve this spatial quantization error, we shift $\mathbf{p_{ct}}$ multiple times, and for each time we find the image descriptor, $D'_{\mathbf{p_{ct}}}$ centered on the shifted $\mathbf{p_{ct}}$. The range of shifting is referred to the margin of deformation. Then, the final image descriptor is generated based on max pooling of each element of the multiple image descriptors, $D'_{\mathbf{p_{ct}}}$.

$$FD_{\mathbf{p_{ct}}} = [max(d_1^{\mathbf{p_{ct}}+\mathbf{p_{margin}}}), max(d_2^{\mathbf{p_{ct}}+\mathbf{p_{margin}}}), ...] \tag{8}$$

where $\mathbf{p_{margin}} = (x_{margin}, y_{margin})$, $-t \leq x_{margin} \leq t$, and $-t \leq y_{margin} \leq t$. This enables us to capture the similar salient parts of an object in the same subregions of the same class images even when there are small shape variations between them.

4. EXPERIMENTAL RESULTS

To evaluate the performance of our method, we implemented our method on two different feature coding methods: SC [2] and LLC [3]. In this experiment three different datasets are used: Caltech-101 [24], Caltech-256 [25], and Evil Tattoo dataset [26]. First two datasets are popular for testing object recognition, and the third dataset is suitable to evaluate the robustness of an image descriptor against object deformations.

We compared our methods against several existing methods that use the BOW model combined with a spatial pyramid. Additionally,

Table 1: Classification Accuracy (%) on Caltech-101 and Caltech-256 Datasets

Method	Caltech-101	Caltech-256
KC [18]	64.16	27.17
VQ [5]	64.60	29.51
SRC [15]	70.70	33.33
D-SVD [27]	73.00	32.67
SC [2]	73.20	34.02
LLC [3]	73.44	41.19
LC-KSVD [7]	73.60	34.42
HSC [13]	74.00	N/A
MHMP [14]	82.50	50.70
LLC (our evaluation)	71.95	35.96
Our method (LLC+CASP)	73.21	36.53
Our method (LLC+MCASP)	74.02	37.55
SC (our evaluation)	72.33	34.75
Our method (SC+CASP)	74.50	36.30
Our method (SC+MCASP)	75.07	37.09

for SC [2] and LLC [3] we reported their results based on a local evaluation using the software they provide under the exact same experimental setup: the same sets of images randomly chosen for training and test, and the same codebook. These experimental results are referred to SC (our evaluation) and LLC (our evaluation). Note that the accuracies of our evaluation for SC and LC is slightly different from what they reported in their publications [2, 3]. For fair comparison, we compare our proposed methods against SC (our evaluation) and LLC (our evaluation) since the same experimental setup is used.

We refer to the method based on SC for feature coding and CASP for spatial pyramid as SC+CASP. SC+MCASP is the method that uses SC with our modified CASP as shown in Figure 4. LLC+CASP is the method that uses LLC as feature coding and CASP as spatial pyramid. LLC+MCASP is the method that uses LLC with our modified CASP as shown in Figure 4. In both methods, the final image descriptor is generated by shifting the estimated center within a pre-defined margin, $t = 3$. For local image features, dense SIFT in [2, 3] are used for all the datasets. For Caltech-101 and Caltech-256, we followed the same experimental setting as in [2, 3]. We randomly select 30 images per class for training and use the remaining images for testing. After computing classification accuracy for each object class, the mean classification accuracy over all classes is computed. We repeat this process 5 times and compute the average accuracy.

4.1. Caltech-101 Dataset

The Caltech-101 dataset [24] contains 9144 images from 102 different class, including 101 object class and 1 additional background class. The number of image per class ranges from 31 to 800. For this dataset, we followed the same experiment setting as [2]. The size of a codebook for SC and LLC is set to $C = 1024$. Table 1 shows the experimental results. With SC, our method (SC+MCASP) outperforms other existing methods except MHMP. Our method (SC+MCASP) outperforms the original method, SC by 2.74%. Also, our method (LLC+MCASP) outperforms the original method, 2.08%. These results show that our MCASP can be combined with two different feature coding methods and it improves both of original methods.

4.2. Caltech-256 Dataset

The Caltech-256 dataset [25] contains 29,780 images from 257 different class, including 256 object class and 1 additional background class. The number of image per class ranges from 80 to 827. The size of a codebook for SC and LLC is set to $C = 1024$ and $C = 4096$ as in [2, 3]. Table 1 also shows the experimental results for Caltech-256. With SC and LLC, our method (SC+MCASP) and our method (LLC+MCASP) outperforms other existing methods except MHMP as well. Our method (SC+MCASP) outperforms the original SC by 2.34%, and our method (LLC+MCASP) outperforms the original LLC by 1.59%.

Table 2: Classification Accuracy (%) on Evil Tattoo Dataset

Method	Accuracy (%)
LLC (our evaluation)	51.57
Our method (LLC+CASP)	51.93
Our method (LLC+MCASP)	52.33
SC (our evaluation)	54.12
Our method (SC+CASP)	55.81
Our method (SC+MCASP)	56.56

4.3. Evil Tattoo Dataset

The evil tattoo dataset [26] contains 1,477 images in total from 27 different class. All the images in this dataset are acquired from evil-tattoo.com. The number of image per class ranges from 14 to 180. For this dataset, we randomly select 10 images per class for training and use the rest of images for testing. The size of a codebook for SC and LLC is set to $C = 1024$. Table 2 also shows the experimental results for the evil tattoo dataset. Since there are no published results on this dataset, we compared our method against our local evaluations of SC and LLC. With SC, our method (SC+MCASP) outperforms the original SC by 2.44%. With LLC, our method (LLC+MCASP) outperforms the original LLC by 0.76%.

4.4. Discussion

We tested our method with two different feature coding methods on three different dataset. Our experimental results show that by combining our proposed spatial pyramid (MCASP) with the feature coding methods the classification accuracies are improved. This was demonstrated using the three challenging datasets. We also investigated the impact of using the proposed new subregions within MCASP. This was achieved by comparing the improvement achieved using CASP versus MCASP. The experimental results demonstrated that the improvement is equally attributed to CASP and MCASP. For example, LLC+MCASP improved LLC+CASP by 1% on the Caltech-256 dataset, where as LLC+CASP improved LLC by 0.6%. Also, SC+MCASP improved SC+CASP by 0.8% on the Caltech-256 dataset, where as SC+CASP improved SC by 1.55%. This means that both contributions: spatial pyramid alignment and the introduction of new subregions, are significant to the improvement in classification accuracy.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a simple but efficient spatial pyramid alignment method that can be combined with the existing feature coding methods. By using max pooled features, we estimate an object center and align the spatial pyramid accordingly. We also propose an image representation descriptor robust to misalignment and object deformations using max pooling on multiple image descriptors generated by shifting the pyramid center in a pre-defined margin. We tested the modified center-aligned spatial pyramid with two different feature coding methods on three different datasets. Our experimental results show that by combining our proposed spatial pyramid (MCASP) with the feature coding, classification accuracy is improved. In the future, we will investigate methods to estimate an object center when there is background clutter.

6. REFERENCES

[1] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 524–531, June 2005, San Diego, CA.

[2] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1794–1801, June 2009, Miami, FL.

[3] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360–3367, June 2010, San Francisco, CA.

[4] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, June 2006, New York, NY.

[6] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1458–1465, October 2005, Cambridge, MA.

[7] Z. Jiang, Z. Lin, and L. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[8] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," *Proceedings of the International Conference on Machine Learning*, pp. 921–928, June 2011, Bellevue, WA.

[9] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2691–2698, June 2010, San Francisco, CA.

[10] C. Weng, H. Wang, and J. Yuan, "Learning weighted geometric pooling for image classification," *Proceedings of the IEEE International Conference on Image Processing*, pp. 3805–3809, September 2013, Melbourne, Australia.

[11] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric l_p-norm feature pooling for image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2609–2704, June 2011, Colorado Springs, CO.

[12] Y. Quan, Y. Xu, Y. Sun, Y. Huang, and H. Ji, "Sparse coding for classification via discrimination ensemble," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5839–5847, June 2016, Las Vegas, NV.

[13] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1713–1720, June 2011, Colorado Springs, CO.

[14] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 660–667, June 2013, Portland, OR.

[15] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K -SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] M. Ranzato, Y. Boureau, and Y. Cun, "Sparse feature learning for deep belief networks," *Proceedings of Neural Information Processing Systems*, pp. 1185–1192, December 2007, Vancouver, BC,Canada.

[18] Gemert J, J. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," *Proceedings of the European Conference on Computer Vision*, vol. 5304, pp. 696–709, October 2008.

[19] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3370–3377, June 2012, Providence, RI.

[20] E. Boser, M. Isabelle, and V. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, July 1992, Pittsburgh, PA.

[21] Corinna C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," *Proceedings of Neural Information Processing Systems*, pp. 801–808, December 2006, Vancouver, BC,Canada.

[23] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2559–2566, June 2010, San Francisco, CA.

[24] F. Li, R. Fergus, and P. Peronai, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[25] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," *Technical Report*, April 2007, California Institute of Technology .

[26] J. Kim, A. Parra, J. Yue, H. Li, and E. J. Delp, "Robust local and global shape context for tattoo image matching," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2194–2198, September 2015, Quebec city, Quebec, Canada.

[27] Z. Jiang, Z. Lin, and L. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1697–1704, June 2011, Colorado Springs, CO.