Robust Feature Encoding with Neighborhood Information for Image Classification

Bingyuan Liu[†], Jing Liu[†], Chunjie Zhang[‡], Hanqing Lu[†]
[†]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[‡] School of Computer and Control Engineering, University of Chinese Academy of Sciences
[†]{byliu, jliu, luhq}@nlpr.ia.ac.cn, [‡]cjzhang@jdl.ac.cn

Abstract

The bag of visual words (BoW) model is one of the most successful model in image classification task. However, the major problem of the BoW model lies in the determination of visual words, which consists of codebook training and feature encoding phases. The traditional K-means and hard-assignment method completely ignore the structure of the local feature space, leading to high loss of information. To alleviate the information loss, we propose to incorporate the neighborhood information of the features into the codebook training and feature encoding process. We firstly propose a model to roughly measure the influence of the distribution of the neighboring features. Then we combine the proposed model with the traditional K-means method in a probability perspective to train the visual codebook. Finally, in the feature encoding phase, both the hard-assignment and soft-assignment method are improved with the proposed neighborhood information term. We investigate our method on two popular datasets: 15-Scenes and Caltech-101. Experimental results demonstrate the effectiveness of our proposed method.

1. Introduction

Image classification remains to be one of the most significant but challenging task in the computer vision and machine learning community. In recent years the bag of visual words(BoW)[3] has been extremely popular in image classification systems. The BoW model usually starts from well-engineered local features such as SIFT[11] or HOG[4], a visual codebook is then trained and local features are encoded into an overcomplete representation. A compact histogram representation is calculated as the global image representation. Finally, a classifier, usually SVM or logistic regression is trained to predict the semantic label of the image.

The major problem of the BoW model is the determination of visual words which consists of two phases: codebook learning and feature encoding. In the learning phase, an overcomplete codebook is trained in an unsupervised manner. Local features are then encoded by assigning the feature vectors to the visual words. The traditional BoW method applies K-means algorithm to train the codebook and vector quantization(VQ) to encode the features to construct high level image representations. The simplicity of such a quantized codebook representation would come with a high information loss and the discriminative information is considerably reduced. While the traditional VQ method is found too restricted to encode the local features, [17] proposed the visual word ambiguity model to soft-assign the feature into several nearest codewords. Yang *et al.*[19] and Wang *et al.*[18] proposed to relaxed the restrictive constraint by sparsity or locally-constrained linearity regularization. However, all the methods try to use some centroids to represent the whole feature space without considering the local structure of the feature space. As the toy exam-

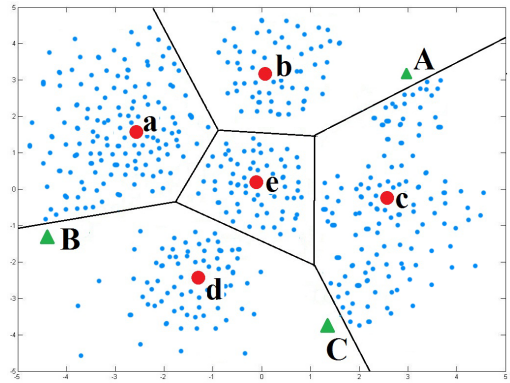

Figure 1. A toy example illustrating the neighborhood information in the codebook model. The blue dots are the feature samples, the labeled red circles are the learned codewords, and the green triangles are the data sample to be encoded. It is shown that the local distributions of the feature space should be considered to assign the triangle data sample into a reasonable codeword.

ple shown in Fig.1, the labeled red circles are codewords trained by unsupervised clustering method. By the traditional VQ method, the triangle data sample A is assigned to the codeword b. However, considering the local distribution of the features, most of the similar neighbors of A are assigned into the word d, it seems more reasonable to encode A into the word d even if A is in the area of the codeword b. These cases can be more common if the feature vector is very high-dimensional (*e.g.* SIFT) and the codebook is highly over-complete. While the traditional method can only capture the global distribution and statistic property of the feature space, we believe the local distribution of the features should also be incorporated.

To address above issues, this paper proposes to incorporate the neighborhood information to improve the K-means and VQ algorithm making the codebook model more robust and semantical. Firstly, we propose a model to roughly measuring the influence of the distribution of the neighborhood, as a complement to the centroid based assignment scheme. Then, we combine the proposed model with the traditional K-means method in a probability perspective to train the visual codebook. In the feature encoding phase, we improve both the hard-assignment and soft-assignment method with the proposed neighborhood information term. We evaluate the effect of the proposed model on the scene dataset 15-Scenes and object dataset Caltech-101 and the improvement over the baselines show the effectiveness and efficiency of the proposed method.

The rest of the paper is organized as follows. Section 2 reviews the related work of BoW models. In Section 3.1, we revisit the traditional BoW model, especially the coding phase. We elaborate our proposed model of incorporating the neighborhood information into the coding phase in Section 3.2. The experimental evaluation is given in Section 4, and we conclude in Section 5.

2. Related Work

Over the years the BoW[3] image representation model has been proven effective and widely used in image classification due to its invariance to illumination, object translation, and rotation. Once local features(*i.e.* SIFT[11], HOG[4], or LBP[14]) are extracted, the codebook training and feature encoding phase would be the most important and govern the quality of image representation. Traditional BoW applies the K-means method to generate the codebook which minimizes the variance between the clusters and the data. However, the simplicity and compactness of such a quantized codebook representation comes with a high cost and the discriminative information is considerably reduced[1]. The discriminative power may be improved by alternative clustering algorithms[8][10] or incorporating some supervised information[12][13][7]. The feature encoding process can be regarded as assigning each feature to the trained codewords. Traditional BoW adopts the vector quantization which is also regarded as the hard assignment scheme. The hard assignment method may induce severe information loss by assigning each feature to only one codeword. To relax the too restricted sparsity, [17] proposed the soft-assignment scheme by assigning each local feature to several nearest codewords.

Another problem of the BoW model is the ignorance of the spatial information as the model describes an image as an orderless collection of local features. To overcome this problem, one popular extension, called as Spatial Pyramid Matching(SPM)[9], has been shown effective by exploiting the absolute spatial information of location regions. More specially, the SPM model requires to first partition each image into a sequence of increasingly finer uniform grids (*i.e.* $1 \times 1, 2 \times 2, 4 \times 4$) and then concatenate the BoW features in each grids to form a high dimensional image feature.

More recently, Yang *et al.*[19] extended the SPM model using Sparse Coding(ScSPM), and showed obvious improvement in image classification. By replacing Kmeans and hard-assignment with sparse coding, their method automatically learn the codebook and search for the optimal weight to assign each local feature into the corresponding codewords. Wang *et al.*[18] proposed to extent the SPM model with the locality-constrained linear coding (LlcSPM), which considers the locality information in the codebook training and feature encoding process. Fisher encoding[15] and Super vector encoding[20] are proposed to capture the average first and second order differences between local features and their distribution centres modeled by Gaussian Mixture Models. However, all the above methods only use the cluster centers to represent the feature distribution, which is limited in the high-dimensional feature space, and the local structures of the features is totally neglected in the feature encoding phase. In this paper, we mainly consider to incorporate some local structure information of the feature space to improve the discrimination of the BoW model.

3. Method

3.1. BoW Model Revisited

In this section, we briefly review the image classification pipeline based on the BoW model, which mainly consists of two procedures of coding and spatial pooling. Specifically, we will focus on the coding section.

3.1.1 Coding

Starting from the raw images, the local features such as SIFT are extracted firstly, usually in the densely sampling scheme. Then two phases are needed to encode them into distinct visual words: codebook learning and feature encod-

ing.

For learning the codebook, a sampled feature set is needed by randomly sampling from the features of the whole images. In standard BoW model, the K-means algorithm is applied to cluster the sampled features into V clusters, which is considered as the visual codebook. We denote the sampled feature set as $X = [x_1, x_2, \ldots, x_N](N >> V)$, and the centroids of the clusters as $C = [c_1, c_2, \ldots, c_V]$. In the Kmeans algorithm, the features are hard-assigned to the nearest centroid:

$$f_v(x) = \begin{cases} 1 & if\ v = argmin_i D(x, c_i) \\ 0 & otherwise \end{cases} \quad (1)$$

The mapping function $f_v(x)$ here is also known as the standard 1-of-V hard-assignment coding scheme.

With the learned over-complete ($V >>$the dimension of feature) codebook, the local patches of each image are then encoded into the visual codewords using the 1-of-V hard-assignment scheme, which is a maximally sparse representation that has been most frequently used. It is generally believed that having an over-complete codebook while keeping the activations sparse helps classification. However, the hard-assignment may be too restricted with too much information loss and discriminative power reduced.

3.1.2 Spatial Pooling

After encoding the local patches into sparse codes, the global BoW representations are then generated by the max pooling or average pooling operation. The BoW model thus describes an image as an orderless statistics of local features, while the spatial layout of the features is completely neglected. To embed the spatial information, [9] proposed the Spatial Pyramid Matching(SPM) model, which has been shown effective for image classification. The SPM model requires to first partition each image into a sequence of increasingly finer uniform grids ($i.e. 1 \times 1, 2 \times 2, 4 \times 4$). The spatial pooling operation is then applied within each grid to get the representation for each region. The BoW representation for each grids are concatenated to form a high dimensional image feature.

Finally, a classifier, usually SVM is trained using the global image feature to predict the final label of the image. The most commonly used kernel functions are linear, χ^2 or histogram intersection.

3.2. Codebook Learning with Neighborhood Information

The traditional dictionary learning methods all try to learn several cluster centroids as the codebook to encode each feature, which is only able to capture the global structure or statistic property of the feature space. As shown in Fig.1, the local structure or distribution is very important to assign the feature into a reasonable visual codeword especially in the very high-dimensional and complex feature space. To this end, we propose to incorporate the neighboring information into the feature assignment scheme to roughly indicate some local structure of the feature distribution. To relax the too restricted 1-of-V hard-assignment scheme, we also adopt the soft-assignment scheme, which assigns a certain feature into several nearest clusters.

We firstly reformulate the 1-of-V hard-assignment scheme in a probability perspective. To a feature point x, the probability to assign it into the v-th cluster is:

$$p_{centroid}(x \to c_v) = \frac{K_\sigma(D(x, c_v))}{\sum_v^V K_\sigma(D(x, c_v))} \quad (2)$$

where K is a kernel function to smooth the local neighborhood of the data sample. In this paper, we use the SIFT descriptor that draws on the Euclidean distance as the distance function D, which assumes a Gaussian distribution. Hence, we adopt the Gaussian-shaped kernel $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$ as the kernel function. The Gaussian-shaped kernel has a smoothing parameter σ representing the size of the kernel, which determines the degree of similarity between feature samples, dependent on the data set, the feature dimensionality and the range of the feature values. Therefore, we tune the parameter discriminatively by cross validation in our experiments.

With the probability defined above, the hard-assignment is:

$$f_v(x) = \begin{cases} 1 & if\ v = argmax_i\ p_{centroid}(x \to c_i) \\ 0 & otherwise \end{cases} \quad (3)$$

the feature is assigned to the codeword with the maximum probability, which is equivalent to the Eq.1.

We propose to incorporate some local structure information into the feature assignment process, as a complement to the hard-assignment based on the centroids of the feature clusters. Since it is difficult to exactly model the local structure in the high-dimensional space, we only consider the neighborhood of the certain feature for simplicity, to roughly indicate some local information. Given a feature x, we firstly find out the k nearest neighbors in the sampled feature set, which is denoted as $\{x_{n_1}, x_{n_2}, x_{n_3}, \ldots, x_{n_k}\}$, and if x_{n_i} is belonged to the j-th cluster we denote it as $x_{n_i,j}$. Thus we define the probability function to assign a certain feature x into the v-th cluster considering the influence of the k nearest neighbors as:

$$p_{knn}(x \to c_v) = \frac{\sum_i K_\sigma(D(x, x_{n_i,v}))}{\sum_j K_\sigma(D(x, x_{n_j}))} \quad (4)$$

In this probability function, we assume that the assignment of the certain feature x is determined by the distribution of

the k nearest neighbors. We both consider the number and the distance of the neighboring features in the certain cluster. The feature is tended to be encoded into the cluster in which there are more neighboring features assigned into the cluster and the feature is more close to these features. We call this definition as the neighborhood information term.

In order to train a more robust and semantical visual codebook, we combine the neighborhood information with the traditional K-means algorithm. In the probability reformulation, it is very nature to combine the centroid-based assignment term and the neighborhood information term. Thus the combined probability to assign the feature into the v-th cluster is defined as:

$$p(x \to c_v) = p_{centroid}(x \to c_v) + \lambda p_{knn}(x \to c_v) \quad (5)$$

where λ is the tradeoff coefficient.

To relax the too restricted sparsity of the hard assignment in K-means, we adopt the soft-assignment scheme which assigns the feature into several clusters according to the probability, regarding the probability as the weights of the feature point in the cluster:

$$f_v(x) = p(x \to c_v) \quad (6)$$

We conclude our codebook training procedure in Algorithm 1. Note that in the algorithm, we firstly run several standard k-means iterations to initialize the centroids of the clusters which we find helpful to improve the performance in our experiments.

Algorithm 1 K-means with neighboring information

Input:

feature set X, codebook size V, tradeoff coefficient λ, number of soft assignment θ, max iterations I

Output:

codebook $C = \{c_1, c_2, \ldots, c_v\}$

1: Init C using standard k-means
2: **for** $iter = 1 : I$ **do**
3: Update the soft assignment for each feature x using Eq.5 and Eq. 6
4: Update the codebook C: $c_v = \frac{1}{N_v} \sum_i f_v(x_{v_i}) x_{v_i}$
5: Stop the process when the update of new centroids reach convergence criteria
6: **end for**

3.3. Feature Encoding with Neighborhood Information

Given the visual codebook, we then encode the local features of the images according to the mapping function. As the centroids of the codebook is too limited to represent the structure of the codebook, we combine the neighborhood information as described in Eq.5. The training feature set X should be remained in the coding phase, since the k nearest neighbors in the feature set should be found out firstly for a given feature, which is different from the traditional coding method. Note that our model can be applied to both hard-assignment and soft-assignment scheme, we will both investigate the two coding scheme in our experiments. In the hard-assignment scheme, we assign the feature point to the codebook with the maximum probability according to Eq. 5 while keeping others zero, while the soft-assignment assigns the feature to several codeword with max assignment probability. A normalization operation is usually needed in the soft-assignment scheme.

4. Experiments and Results

In the experiments, we mainly compare our method with the popular kernel SPM[9] on two popular datasets, 15-Scenes and Caltech-101. We start our experiments with an in-depth analysis of our methods on the dataset of 15-Scenes, after which we transpose these findings to the experiments on the Caltech-101. First, we evaluate the effectiveness our method in the codebook training phase. Then we investigate the hard-assignment and soft-assignment scheme with the proposed neighborhood information term incorporated. For our experimental setup we closely follow Lazebnik *et al.*[9] for fair comparison. We use a single local descriptor, the popular SIFT descriptor, by densely extracting local patches of 16×16 pixels computed over a grid with spacing of 8 pixels. When training SVM classifier, we apply the histogram intersection kernel and use the well implemented LIBSVM[2] package. The SVM regularization terms are chosen via 5-flod cross validation on the training data. The detailed comparisons and analysis are presented in the following subsections.

4.1. Results on 15-Scenes Dataset

We firstly experiment with a popular scene classification benchmark, 15-Scenes dataset, complied by several researchers[6][9]. The dataset is composed of 15 scene classes (*e.g.* kitchen, coast, highway), with each class containing 200 to 400 images and there are 4485 gray-scale images in total. Following the experiment setup of [9], we take 100 images per class for training and the rest for testing, and the size of the visual codebook is fixed to 400. The final SVM classifier are trained in the one-versus-others scheme and the image is classified to the category with the max score.

4.1.1 The Effect of the Neighborhood Information on the Ccodebook Learning

We first evaluate the effect of the neighborhood information term in Eq.4 on the codebook training phase, while we apply the traditional 1-of-V hard-assignment scheme in

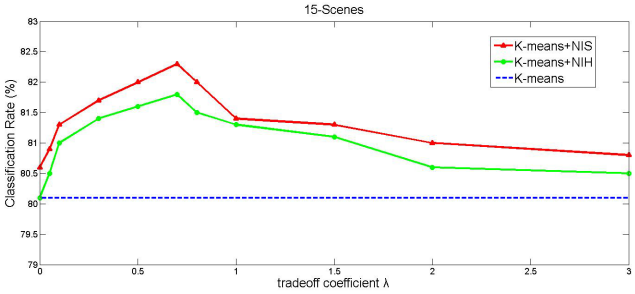

Figure 2. Performance comparisons on the 15-Scenes to investigate the effect of the neighborhood information on the codebook training with varying tradeoff coefficient λ.

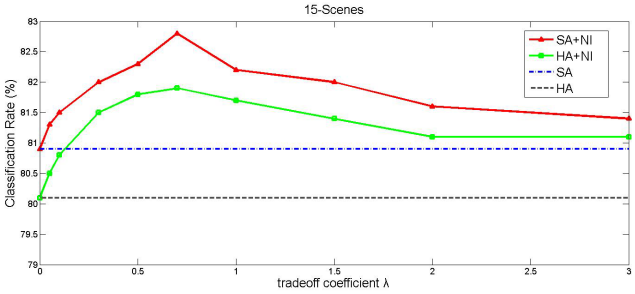

Figure 3. Performance comparisons on the 15-Scenes to investigate the effect of different coding scheme varying tradeoff coefficient λ.

the coding phase for fair comparison. We randomly sample $10,000$ SIFT features from the features extracted from the whole images. The tradeoff term λ in the Eq.5 controls the proportion of the neighborhood information term: the bigger λ is, the more the neighborhood information contribute to the assignment probability. Thus we evaluate the effect of the neighborhood information term by varying the tradeoff term λ, as shown in Fig.2. K-means+NIH denotes the proposed method of the improved K-means with the neighborhood information and hard-assignment scheme, while K-means+NIS denotes the method of the improved K-means with the neighborhood information and soft-assignment scheme. Note that when λ equals 0, the algorithm degenerates to the standard K-means, and when λ is large enough the assignment is almost dependent on the neighborhood term. It is shown that incorporating the neighborhood term does help to improve the performance. The K-means+NIS outperforms the other two methods and the best performance are achieved with well balanced coefficient. Empirically, we found that keeping the tradeoff term λ to be around 0.7 yields good results.

4.1.2 The Effect of the Neighborhood Information on the Feature Encoding

Then we investigate the performance of various types of feature encoding scheme with the codebook training method fixed to K-means in our experiments, which is shown in Fig.3. Similar to the above experiment, we vary the tradeoff coefficient λ to evaluate the effect of the neighborhood information term. In Fig.3, HA denotes the traditional hard-assignment method, and SA denotes the traditional soft-assignment method. HA+NI and SA+NI denotes the hard-assignment and soft-assignment scheme combined with the proposed neighborhood information term respectively. It is shown that the soft-assignment scheme outperforms the hard-assignment in general. The neighborhood information term really helps to improve the classification accuracy, which is not very sensitive to the tradeoff parameter λ. The best performance is achieved when the parameter λ is around 0.8. As shown in the figure, when λ is large, the performance is still slightly better than the traditional method, further indicating the importance of the neighborhood term.

Table 1. Classification rate (%) comparison on 15-Scenes.

Method	Codebook	Encoding	Classification Rate
KSPM[9]	K-means	HA	81.40 ± 0.50
ScSPM[19]	SC	SC	80.28 ± 0.93
KSPM	K-means	HA	80.10 ± 0.71
KSPM	K-means	SA	80.90 ± 0.36
Ours	K-means	SA+NI	82.81 ± 0.31
Ours	K-means+NIS	HA	82.32 ± 0.23
Ours	**K-means+NIS**	**SA+NI**	$\mathbf{83.23 \pm 0.22}$

Finally, we summarize the comparisons of different methods we implemented in Table 1. We repeat every method 5 times and report the mean and standard deviation of the mean class accuracy. Note that our implementations of KSPM are not able to reproduce the results reported in [9] probably due to the SIFT descriptor extraction and normalization process. The best performance is achieved by using K-means+NIS in the codebook training phase and SA+NI in the feature encoding phase. Our method outperforms the KSPM by more than 3% according to our implementations. It is noted that our method also outperforms ScSPM[19] which adopts the Sparse Coding (SC) to learn the codebook and encode the local patches. The improvement of our simple scheme to roughly incorporate the neighborhood information has shown the importance of the local structure of the features in the codebook training and feature encoding phase.

4.2. Results on Caltech-101 Dataset

We conduct our second set of experiments on the Caltech-101 dataset[5]. The Caltech-101 dataset contains 9144 images totally from 102 different categories, including 101 object categories and 1 additional background category with high shape variability. The number of images per cat-

Table 2. Classification rate (%) comparison on Caltech-101.

Method	15 training	30 training
KSPM[9]	56.40	64.6 ± 0.80
KC[16]	—	64.14 ± 1.18
KSPM	56.13 ± 0.30	63.70 ± 0.35
Ours	$\mathbf{58.53 \pm 0.23}$	$\mathbf{66.52 \pm 0.36}$

egory varies from 31 to 800, and most images are medium resolution, *i.e.* about 300×300 pixels. We follow the experiment setup of [9], namely, training on 15 and 30 images per class and test on the rest. For efficiency, we limit the number of test images to 50 per class. We randomly sample from the whole local features to get a feature set containing $10,000$ local features to tain the codebook with the size fixed to 1000. We repeat every method 5 times and report the mean and standard deviation of the mean class accuracy.

The performance comparison results are shown in Table 2. We use the empirical parameter values in the 15-Scenes experiments. The best performance of our method is achieved by using Kmeans+NIS to train the codebook and SA+NI to encode the features. As shown, our method outperforms the baseline by more than 2 percent for both 15 training and 30 training per category.

5. Conclusion

In this paper, we address the issue of the determination of visual words in the BoW image classification model. As the local structure of the features are neglected in the existing algorithms, we propose a neighborhood information model to roughly indicate the local distribution. Then we combine the proposed model with the standard K-means method to improve the process of the codebook learning. In the feature encoding phase, our model is also able to incorporated into the hard-assignment and soft-assignment method to improve the robustness and discrimination. The experiments on 15-Scenes and Caltech-101 datasets have shown the effect of the incorporated neighborhood information term, and our method outperforms the traditional BoW model. Possible future work involves more carefully engineered local structures of the feature space, and unsupervised learning of such information. Another interesting direction is to incorporate the neighborhood information into the framework of the sparse coding or local coordinate coding.

References

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on statistical learning in computer vision*, 2004.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 106(1):59–70, 2007.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[7] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, 2011.

[8] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[10] J. Liu and M. Shah. Scene modeling using co-clustering. In *ICCV*, 2007.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2008.

[14] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[15] F. Perronnin, J. Snchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[16] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[17] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[19] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[20] X. Zhou, K. Yu, T. Zhang, T. S. Huang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010.