HISTOGRAM FEATURE DEBLURRING

Takumi Kobayashi

National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Ibaraki, Japan

ABSTRACT

A histogram is an effective form for extracting various types of features and has been attracting keen attention in pattern recognition fields. In visual recognition, however, the histogram features suffer from smoothing due to the processes of quantizing continuous input patterns into discrete codes and pooling them. Such smoothing degrades discriminative power by blurring the distinctive features. In this paper, we propose a novel method to get rid of such blurriness and reveal the essential discriminative information from the histogram features. We first model the blurring process based on the graph Laplacian of discrete codes and then formulate a method for deblurring histogram features in a computationally efficient form which works as post-processing just after feature extraction. The proposed method is also shown to be related to unsharp masking of an image restoration technique. In the experiments on image classification using BoW histogram features, the proposed deblurring method favorably improves performance of the histogram features.

Index Terms— Histogram feature, feature transform, image classification

1. INTRODUCTION

Feature extraction is a basic building block for visual recognition and most of image features are formulated in a histogram form, such as bag-of-word (BoW) [1] and local binary pattern (LBP) [2]. The histogram form is such a general feature representation that we can also find it in the other fields than computer vision, *e.g.*, document analysis. However, the histogram features used in vision tasks suffer from smoothing mainly due to the processes of quantization and pooling.

In visual recognition, input signals, such as pixel values, are inherently continuous and it is required for extracting histogram features to quantize those continuous signals into discrete patterns (codes); on the other hand, in document analysis, text is naturally defined in a categorical form. Thus, although much research effort has been made in providing various types of code to effectively describe continuous input patterns, such as by gradient orientation bins, local binary patterns and visual words, the resultant histogram features inevitably contain quantization errors. To reduce the quantization errors, most methods employ soft assignment of continuous patterns onto discrete codes. Whereas, it increases smoothness of the histogram features, degrading the distinctiveness. So coded patterns are then aggregated (pooled) on the region of interest to construct histogram features which describe the region. In sum-pooling, all the codes are uniformly taken into account to provide statistically stable histogram features by sufficiently containing distinctive codes as well as erroneous ones. This process also enhances smoothness in histogram values.

Due to those soft coding and sum-pooling, the obtained histogram is subject to over-smoothing; we define such smoothing as *blurring*[1] in histogram features (Fig. 1a). In this paper, we propose a novel method to get rid of the blurriness and emphasize a discriminative features by *deblurring* histogram features (Fig. 1b). The blurring process is modeled by using the graph Laplacian of the discrete codes and then a computationally efficient method is presented for sharpening the distinctive codes in a histogram. In the framework of BoW, max-pooling [3, 4] has been proposed to pick up the most distinctive samples by applying $\max$ operator instead of sum operator in pooling codes, but it loses plenty of code information which may contain discriminative power. In contrast, sum-pooling thoroughly embeds them in histogram features. Therefore, we expect that it is more effective to apply the deblurring method to reveal the distinctive information buried by sum-pooling from the blurred histogram features. Note that the proposed method results in a simple form operating efficiently as post-processing just after the histogram feature extraction, and thus various types of transformation methods including normalization are applicable.

2. PROPOSED METHOD

For enhancing the discriminative power of the features in a histogram form, we propose a method to reveal the intrinsic features via deblurring the histogram. We first model the blurring process in extracting histogram features which causes performance degradation. Then, based on the model, we formulate a method to sharpen the blurred histograms in a com-

[1]Note that in this paper we mention blurring in the resultant histogram features, not on input images.

(a) Blurring process (b) Deblurring histogram

Fig. 1. Blurring/deblurring for (BoW) histogram features. (a) Histogram features are blurred due to soft coding and sum-pooling. (b) We define as *deblurring* the process to retrieve the intrinsic histogram features from the observed one. $\boldsymbol{P}$ indicates the transition probabilities among codes; see (1).

putationally efficient form.

In what follows, we proceed to describe a general formulation regarding the histogram blurring/deblurring, and it might be helpful for better understanding to refer to a BoW model which constructs a histogram of visual words via coding and pooling local descriptors (Fig. 1).

2.1. Blurring process in histogram feature extraction

The coding process has inherent difficulty in quantizing continuous patterns, such as real-valued pixels and local descriptors, into discrete codes; it is hard to decide which code is assigned to the continuous patterns at boundary. To mitigate miss quantization, soft coding is employed to represent a continuous pattern by (nearby) multiple codes with voting weights, though unfavorably smoothing the histogram features. On the other hand, it is also possible to statistically compensate the miss quantization by aggregating substantial codes via sum-pooling. Such aggregation embeds not only the essential code information but also noisy ones in the resultant histogram features. Consequently, the procedures of soft coding and sum-pooling lead to *blurring* histograms (Fig. 1a). In the blurred histogram, features are less discriminative since the distinctive feature patterns are buried.

Through the blurring, the codes which are close to each other tend to similarly get weights due to soft coding and/or miss quantization. Thereby, the intrinsic histogram values would spread on the similar codes (histogram bins). This can be mathematically modeled as follows. Given the pair-wise transition probabilities among C codes as $\boldsymbol{P} = \{P_{ij} = p(\mathtt{c}_i|\mathtt{c}_j)\} \in \mathbb{R}^{C\times C}$ $s.t. \sum_i P_{ij} = 1$, the code weights (histogram values) would leak according to the transition probability like random walks:

$$\boldsymbol{z} = \boldsymbol{P}\boldsymbol{z}^*, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}_+^C$ is the observed (blurred) histogram and $\boldsymbol{z}^* \in \mathbb{R}_+^C$ is the intrinsic (unblurred) one (Fig. 1b). The blurring deteriorates discriminativity of the histogram feature; in the extreme case, the strong blurring leads to a uniform histogram.

The transition probabilities $\boldsymbol{P}$ play a key role in this deblurring model (1). In practice, they can be computed based on pair-wise similarities $\boldsymbol{S}$, while the method of [5] straightforwardly provides the transition probabilities as described in Sec.3.1. However, probabilities for self-transitions P_{jj} can not be well defined since S_{jj} is inherently different measurement from $S_{ij}, i \neq j$. Thus, they are practically parameterized as $P_{jj} = p(\mathtt{c}_j|\mathtt{c}_j) = \lambda_j$ while the inter-transition probabilities are accordingly defined by $P_{ij} = \frac{(1-\lambda_j)S_{ij}}{\sum_{k\neq j} S_{kj}}, i \neq j$. As a result, the transition probabilities $\boldsymbol{P}$ are represented by

$$\boldsymbol{P} = \boldsymbol{\Lambda} + \tilde{\boldsymbol{P}} \quad \in \mathbb{R}^{C\times C}, \tag{2}$$

where $\boldsymbol{\Lambda} = \mathtt{diag}(\lambda_1, ..., \lambda_C)$ is a diagonal matrix to parameterize self-transition probabilities while $\tilde{\boldsymbol{P}}$ consists of inter-transition probabilities computed from pair-wise similarities $\tilde{P}_{ij} = \frac{(1-\lambda_j)S_{ij}}{\sum_{k\neq j} S_{kj}}, i \neq j$, with letting the diagonal zeros $\tilde{P}_{jj} = 0$. More practically, we can simply assume uniform self-transition probabilities as $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}$, resulting in

$$\boldsymbol{P} = \lambda\boldsymbol{I} + (1-\lambda)\boldsymbol{Q} \quad \in \mathbb{R}^{C\times C}, \tag{3}$$

where $\boldsymbol{Q}$ contains the inter-transition probabilities only; $Q_{ij} = \frac{S_{ij}}{\sum_{k\neq j} S_{kj}}, i \neq j$, and $Q_{jj} = 0$. The parameter λ controls blurriness in the model (1); the lower λ assumes higher blurriness in histogram features and vice versa. Note that the pair-wise similarities are usually computed not across all C codes but within a few similar codes, and thus the key property of the transition probability matrix $\boldsymbol{P}$ is sparseness.

2.2. Deblurring histogram

To dig out the discriminative and essential information from the blurred histograms, it is required to *deblur* histogram features. Based on the model (1), the deblurring operator is straightforwardly obtained as the inverse of $\boldsymbol{P}$;

$$\boldsymbol{z}^* = \max\left[\boldsymbol{P}^{-1}\boldsymbol{z},\, 0\right], \tag{4}$$

where we ensure non-negative histogram features $\boldsymbol{z}^*$ for consistency with the input histogram features $\boldsymbol{z}$ by simply cutting off negative features since non-negative least squares [6] for the large $\boldsymbol{P}$ requires huge computation cost. The inverse matrix $\boldsymbol{P}^{-1}$, however, is not necessarily sparse even for sparse $\boldsymbol{P}$, and thus (4) suffers from computational burden in matrix-vector multiplication. In this study, we derive a computationally efficient deblurring method by approximating (4) as follows. The inverse matrix $\boldsymbol{P}^{-1}$ is expanded as

$$\begin{aligned}\boldsymbol{P}^{-1} &= \{\lambda\boldsymbol{I} + (1-\lambda)\boldsymbol{Q}\}^{-1} = \lambda^{-1}(\boldsymbol{I} + \tfrac{1-\lambda}{\lambda}\boldsymbol{Q})^{-1}\\ &= \lambda^{-1}\{\boldsymbol{I} - \tfrac{1-\lambda}{\lambda}\boldsymbol{Q} + (\tfrac{1-\lambda}{\lambda}\boldsymbol{Q})^2 - \cdots\}\end{aligned} \tag{5}$$

The above expansion converges in the case that the maximum eigenvalue of $\frac{1-\lambda}{\lambda}\boldsymbol{Q}$ is less than 1. Such assumption holds by setting $0.5 < \lambda_j \leq 1$ and as a result we can obtain the first-order approximation;

$$\boldsymbol{P}^{-1} \approx \tfrac{1}{\lambda}\boldsymbol{I} - \tfrac{1-\lambda}{\lambda^2}\boldsymbol{Q} \propto \boldsymbol{I} + \eta(\boldsymbol{I} - \boldsymbol{Q}), \tag{6}$$

where η is re-parameterization of λ by $\eta = \frac{1-\lambda}{2\lambda-1} \geq 0$. Note that $\boldsymbol{I} - \boldsymbol{Q}$ is closely related to the graph Laplacian [7] since $\boldsymbol{Q}$ is practically derived from the similarity matrix. This approximation for $\boldsymbol{P}^{-1}$ is introduced into (4) to provide the approximated deblurring method;

$$\boldsymbol{z}^* = \max\left[\left\{\boldsymbol{I} + \eta(\boldsymbol{I} - \boldsymbol{Q})\right\}\boldsymbol{z},\ 0\right]. \tag{7}$$

This inherits sparseness from $\boldsymbol{P}$ and enables us to deblur the histogram feature with only a low computation cost. Note that the deblurred histogram features are usually fed into normalization in visual recognition.

2.3. Related works

The proposed method is closely related to image deblurring methods, especially unsharp masking [8]. The blurring model (1) implies that the transition probabilities work as a point spread function (PSF) from the viewpoint of image restoration. While recent deconvolution methods using the estimated PSF require substantial computational cost, unsharp masking [8] is formulated in a simple operator exploiting difference between an original image $\mathcal{I}$ and its blurred image $\mathtt{p} * \mathcal{I}$;

$$\mathcal{I}^* = \mathcal{I} + \hat{\eta}(\mathcal{I} - \mathtt{p} * \mathcal{I}) = \mathcal{I} + \hat{\eta}(\delta - \mathtt{p}) * \mathcal{I}, \tag{8}$$

where $*$ indicates convolution, $\mathtt{p}$ is a blurring PSF, *e.g.*, Gaussian kernel, δ is a delta function and $\hat{\eta}$ is a parameter how sharpening an image. The unsharp masking (8) can be extended for histogram features modeled in (1) as

$$\boldsymbol{z}^* = \{\boldsymbol{I} + \hat{\eta}(\boldsymbol{I} - \boldsymbol{P})\}\boldsymbol{z} = \{\boldsymbol{I} + \eta(\boldsymbol{I} - \boldsymbol{Q})\}\boldsymbol{z}, \tag{9}$$

where η is re-parameterized from $\hat{\eta}$ and λ; $\eta = \hat{\eta}(1-\lambda)$. This is the same form as (7) and thus we can say that the proposed method is regarded as the extension of the unsharp masking to general histogram features. It is noteworthy that although the deblurring by unsharp masking is a well-known technique in image restoration, it has never been applied to deblurring histogram features, to our best knowledge. According to the convention in image processing, we set $\eta = 1$, *i.e.*, $\lambda = \frac{2}{3}$, as a default value; this is empirically discussed in Sec.3.2.

From the statistical perspective, deblurring over histogram features is related to decorrelation of features, which can be realized such as by applying principal component analysis (PCA). Jégou and Chum [9] presented a method to whiten BoW histograms for decorrelation via PCA projection. Our method deblurs the histogram, not statistically, but based on the inherent structure among the codes, *e.g.*, words in BoW, which is represented by the graph Laplacian. Thus, it works as post-processing just after feature extraction and can be followed by other feature transform methods including the statistical decorrelation. Besides, the deblurring method exploits sparse projection performed efficiently, while PCA decorrelation based on a dense projection matrix is time-consuming especially for large-dimensional histograms.

In a BoW framework, Gao *et al.* [10] have proposed a method to assign words to local descriptors consistently according to the pair-wise similarities between the descriptors. That method would render deblurring effect to some extent in the sense that similar descriptors are coded into similar words, though it requires a considerable computation load to optimize the consistent coding. On the other hand, the proposed method operates on the resultant BoW histogram without going into detail of the relationships among the local descriptors.

3. EXPERIMENTAL RESULTS

We apply the proposed method to BoW histogram features [1] on image classification tasks. Bag of features is obtained by densely extracting SIFT local descriptors [11] at spatial grid points in 4 pixel step with three scales of $\{16, 24, 32\}$ pixels. To construct visual words $\{\mathtt{c}_i\}_{i=1}^C$, we apply k-means clustering to a million of SIFT descriptors randomly drawn from the training images; the dictionary learning [10, 4, 3] is out of our scope and thus k-means is simply employed in this study. An image is partitioned into sub-regions in three levels of spatial pyramid as $1 \times 1, 2 \times 2$ and 3×1; the word histograms are computed on the respective sub-regions via sum-pooling. The respective histograms are deblurred by the proposed method (7) and then concatenated into the image feature vector followed by L_2 normalization and linear SVM classifier [12].

3.1. Word coding by kernel-based transition probability

In a BoW model, local descriptors extracted in an image are coded into visual words. In this study, we apply the kernel-based transition probability (KTP) [5] to produce (soft) weights on the words, as well as to directly construct the transition probabilities $\boldsymbol{Q}$ among words.

The KTP method was proposed in [5] to produce transition probabilities among samples. In this case of word coding, the transition probability $p(\mathtt{c}_i|\boldsymbol{x})$ from the descriptor $\boldsymbol{x} \in \mathbb{R}^d$ to the i-th word $\mathtt{c}_i \in \mathbb{R}^d$ is regarded as a voting weight to the i-th word. Suppose we have C visual words and denote the transition probabilities by $\boldsymbol{w}(\boldsymbol{x}) = [p(\mathtt{c}_1|\boldsymbol{x}), \cdots, p(\mathtt{c}_C|\boldsymbol{x})]^\top$. The KTP is computed as follows;

$$\boldsymbol{w}(\boldsymbol{x}) = \arg \min_{\boldsymbol{w}|\boldsymbol{w}\geq 0, \sum_i^C w_i = 1} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{K}\boldsymbol{w} - \boldsymbol{w}^\top \boldsymbol{k}(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{K} \in \mathbb{R}^{C\times C}$ is the kernel Gram matrix of the C words, $K_{ij} = \mathtt{k}(\mathtt{c}_i, \mathtt{c}_j)$, and $\boldsymbol{k}(\boldsymbol{x}) \in \mathbb{R}^C$ is the kernel vector at the descriptor $\boldsymbol{x}$, $k_i(\boldsymbol{x}) = \mathtt{k}(\mathtt{c}_i, \boldsymbol{x})$; we employ Gaussian kernel $\mathtt{k}(\mathtt{c}_i, \mathtt{c}_j) = \exp(-\frac{\|\mathtt{c}_i - \mathtt{c}_j\|^2}{2\sigma^2})$ for the kernel function $\mathtt{k}$. As suggested by [5], (10) can be efficiently solved by applying SMO [13] to k-nearest neighbor words; we set $k = 30$ and $\sigma = 0.5$. The KTP is favorably sparse and robust to both the number of the nearest neighbors k and the bandwidth parameter in the Gaussian kernel [5]. This method is also applied in a leave-one-out manner to compute the inter-transition probabilities $\{Q_{ij} = p(\mathtt{c}_i|\mathtt{c}_j)\}_{i\neq j}$ among words; this is precomputed off-line for the deblurring method.

3.2. Performance analysis

We analyze the performances of the proposed method from various aspects by using PASCAL VOC2007 dataset [14].

Coding. Before proceeding to deblurring, we first compare the KTP coding method (Sec.3.1) with the other methods, RBF, LLC [4] and SAC [15]; the latter two coding methods are applied with max-pooling as suggested in those papers. The performance results at 8,192 words are shown in Table 1. The method of KTP coding works well with sum-pooling, compared with the other methods, by favorably exploiting local structure of the descriptors on the visual words.

Deblurring. Next, we further improve the performance by applying the proposed deblurring method to the BoW histogram features computed via sum-pooling. In Sec.2.2, we have presented two deblurring methods; the exact method (4) using inverse matrix and the approximated one (7) based on unsharp masking. Fig. 2a shows the performance results of those two methods on various $\eta \in \{0, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\}$ where $\eta = 0$ indicates the standard method without deblurring. The deblurring of $\eta > 0$ improves the performance in both exact and approximated methods for both KTP and RBF coding with sum-pooling; interestingly, as in unsharp masking [8], the best performance is obtained around $\eta = 1$. Since there is no significant performance gap between the exact and approximated methods, we employ the approximated one for computational efficiency. Then, to validate the process of sum-pooling for deblurring, we compare it with max-pooling in Fig. 2b which shows the performance gain compared to that of $\eta = 0$. As expected, the deblurring fails to improve performance in max-pooling methods. The max-pooling picks up only one sample for each word, degrading the relationships among words and breaking the blurring model (1). And, the deblurring might unfavorably enhance the effect of noisy words caused by the max operator. In contrast, it successfully digs out the discriminative features from sum-pooled histograms. As a result, we apply BoW of KTP coding with sum-pooling followed by the approximated deblurring method of $\eta = 1$ in the following experiments.

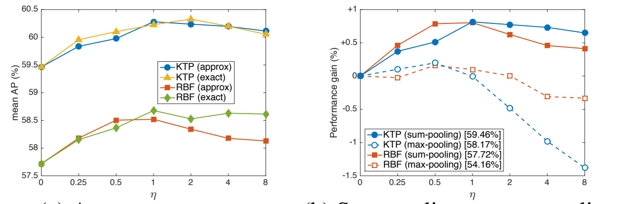
Number of words. Finally, we show the performance results on various numbers of word in Fig. 3. The proposed deblurring method favorably improves the performance, outperforming the methods of LLC [4] and SAC [15]. The performance gain by the deblurring becomes larger as the number of words increases. Due to the curse of dimensionality, in a high dimensional space of local descriptors, the code weights leak to substantial numbers of words and thus the histograms of larger number of words would be more blurred. The deblurring method works well to get rid of such high blurriness and extract the discriminative information.

3.3. Performance comparison

The deblurring method applied to BoW of 16,384 words is compared to the other methods on MIT scene [16] and Caltech256 [17] datasets as shown in Table 2. The proposed method improves the original (unblurred) features, outperforming the other methods as well.

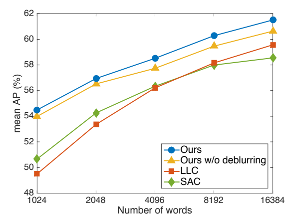
Table 1. Comparison in coding methods at 8,192 words.

	LLC [4]	SAC [15]	RBF	KTP
mAP(%)	58.16	57.99	57.72	**59.46**



(a) Approx. vs. exact (b) Sum-pooling vs max-pooling

Fig. 2. Comparison in deblurring. (a) The approximated deblurring method is compared to the exact one. (b) The approximated deblurring method is applied to sum- and max-pooling methods. The numbers in the brackets indicate the performances without deblurring ($\eta = 0$).



Method	Acc.
Ours	**61.5**
Ours w/o delur	60.6
LLC [4]	59.6
SaC [15]	58.6
VOC winner	59.4

at 16,384 words

Fig. 3. Comparison on various numbers of word.

Table 2. Comparison to the other methods.

MIT scene [16]

Method	Acc.
Ours	**60.2**
Ours w/o delur	58.4
LLC [4]	55.1
SaC [15]	55.2
Bo *et al*. [18]	50.5
Juneja *et al*. [19]	63.1

Caltech-256 [17]

#training sample	15	30	45	60
Ours	**40.9**±0.3	**48.3**±0.3	**52.7**±0.1	**55.5**±0.2
Ours w/o deblur	39.3±0.1	46.8±0.4	51.3±0.3	54.1±0.2
LLC [4]	38.8±0.1	46.1±0.4	50.2±0.1	53.0±0.2
SAC [15]	37.6±0.3	45.2±0.2	49.1±0.3	51.7±0.4
Bo *et al*. [18]	40.5 ±0.4	48.0 ±0.2	51.9 ±0.2	55.2 ±0.3
Sánchez *et al*. [20]	38.5±0.2	47.4±0.1	52.1±0.4	54.8±0.4

4. CONCLUSION

We have proposed a novel method to improve discriminative power of features defined in a histogram form. The histogram features used in visual recognition are subject to blurring through the processes of coding and pooling, which pollutes distinctive features. By modeling the blurring process based on graph Laplacian of codes, we formulate a histogram deblurring method which gets rid of the blurriness and extracts distinctive features. The experiment results on image classification using various datasets show that the proposed deblurring method effectively enhances the discriminative power and improves the performance of the histogram features. It is noteworthy that our method works as post-processing with a low computation cost just after feature extraction and can be followed by other types of feature transformation method.

5. REFERENCES

[1] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[2] M. Pietikäinen, G. Zhao, A. Hadid, and T. Ahonen, *Computer Vision Using Local Binary Pattern*, Springer, 2011.

[3] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009, pp. 1794–1801.

[4] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010, pp. 3360–3367.

[5] T. Kobayashi, "Kernel-based transition probability toward similarity measure for semi-supervised learning," *Pattern Recognition*, vol. 47, no. 5, pp. 1994–2010, 2014.

[6] R. Bro and S.D. Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, pp. 393–401, 1997.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[8] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice Hall, 2007.

[9] H. Jégou and O. Chum, "Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening," in *ECCV*, 2012, pp. 774–787.

[10] S. Gao, I.-H. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely - laplacian sparse coding for image classification," in *CVPR*, 2010, pp. 3555–3561.

[11] D.G. Lowe, "Distinctive image features from scale invariant features," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[12] V.N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[13] J.C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds., pp. 185–208. MIT Press, Cambridge, MA, USA, 1999.

[14] "The PASCAL Visual Object Classes Challenge 2007 (VOC2007)," http://www.pascal-network.org/challenges/VOC/voc2007/index.html.

[15] Y. Huang, K. Huang, Y. Yu, and T. Tan, "Salient coding for image classification," in *CVPR*, 2011, pp. 1753–1760.

[16] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009, pp. 413–420.

[17] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, Caltech, 2007.

[18] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *NIPS workshop on deep learning*, 2012.

[19] M. Juneja, A. Vedaldi, C.V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013, pp. 923–930.

[20] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.