Salient Coding for Image Classification

Yongzhen Huang, Kaiqi Huang, Yinan Yu and Tieniu Tan
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China
{yzhuang, kqhuang, ynyu, tnt}@nlpr.ia.ac.cn

Abstract

The codebook based (bag-of-words) model is a widely applied model for image classification. We analyze recent coding strategies in this model, and find that saliency is the fundamental characteristic of coding. The saliency in coding means that if a visual code is much closer to a descriptor than other codes, it will obtain a very strong response. The salient representation under maximum pooling operation leads to the state-of-the-art performance on many databases and competitions.

However, most current coding schemes do not recognize the role of salient representation, so that they may lead to large deviations in representing local descriptors. In this paper, we propose "salient coding", which employs the ratio between descriptors' nearest code and other codes to describe descriptors. This approach can guarantee salient representation without deviations. We study salient coding on two sets of image classification databases (15-Scenes and PASCAL VOC2007). The experimental results demonstrate that our approach outperforms all other coding methods in image classification.

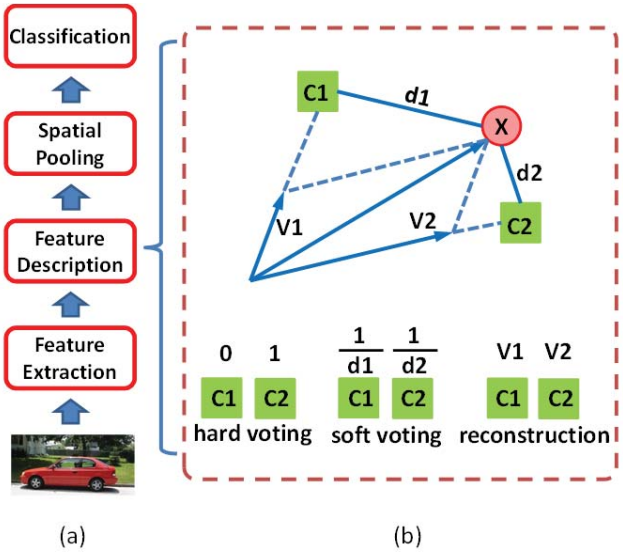

Figure 1. (a) The framework of the codebook based model. (b) A demonstration of three kinds of classic coding strategies. The red ball is a local descriptor and the green rectangles are codes.

1. Introduction

Image classification is an important problem in computer vision and pattern recognition. It plays a key role in many applications such as video surveillance, image retrieval and web content analysis. There are many approaches for image classification. The codebook based (bag-of-words) model [14] and its extensions achieve the state-of-the-art performance in many famous databases (e.g., Caltech101 [1] and Caltech256 [2]) and competitions (e.g., PASCAL VOC [3] and TRECVID [4]).

Figure 1 (a) shows the framework of the original codebook based model. Firstly, it extracts images' local features by detectors or dense sampling and then calculates their descriptors. For local feature detection, classic detectors include Harris detector [17] and its extension [25], maximally stable extremal region detector [21], affine invariant salient region detector [22]. For local feature description, we usually use local descriptors such as Haar descriptor [27], scale-invariant feature transform (SIFT) descriptor [20], gradient location and orientation histogram (GLOH) descriptor [23], rotation-invariant feature transform (RIFT) descriptor [18], shape context [10], histogram of gradients (HOG) descriptor [15]. After obtaining local features (i.e., descriptors), this model uses a codebook to represent them. The codebook is a group of codes usually obtained by clustering over all descriptors. This process is usually called "coding". We will detail various coding strategies in Section 2.1. Afterwards, the responses on each code are integrated into one value by spatial pooling operation, e.g., maximum or average pooling. Therefore, an image is described by a histogram whose length is equal to the size of the codebook. Finally, the histogram is sent to a classifier,

e.g., Boosting [24],[16] or Support Vector Machine (SVM) [11], [13] for classification.

In this paper, we focus on the process of representing descriptors using the codebook. Figure 1(b) illustrates three kinds of classic coding strategies. **Hard voting** [14] is adopted in the original codebook model. It reflects the occurrence frequency of codes. In hard voting, each descriptor is represented by its nearest code. It is simple and fast, but limited in representing descriptors. To improve it, **soft voting** [26] is developed, wherein a descriptor is represented by multiple codes. Besides, frequency is replaced by Euclidean distance between descriptors and codes. This scheme enriches descriptors' information and increases the final classification accuracy. **Reconstruction** based method, e.g., sparse coding [29] is another idea to enhance hard voting. It chooses a group of codes to reconstruct [1] descriptors plus a constraint to the number of codes. In further researches (e.g., LCC [30] and LLC [28]), the locality constraint of codes' spatial location is integrated. Reconstruction based coding as well as sparse and locality constraints achieves very good performance compared with hard voting and soft voting [12],[30],[28].

We take LLC as an example for further analysis. LLC uses K nearest codes to encode a descriptor. If there is a code that is much closer to the descriptor than other codes, the response in this code will be much stronger than others. We call this code and process as a ***salient code*** and ***salient representation*** respectively. The salient representation is robust because the salient code can independently describe the descriptor. Due to maximum pooling operation used in LLC, only the strongest response on each code is preserved. Weak responses are discarded although they are used for reconstruction. These weak responses are unstable because they need combining with other responses to represent a descriptor. Therefore, LLC plus maximum pooling operation is not to obtain exact description but salient representation to descriptors. However, the least square optimization adopted in LLC cannot guarantee salient representation in all cases. When K is smaller than the dimensionality of descriptors, the least square optimization is an under-determine problem which may lead to a non-zero solution. Thus, LLC may produce large deviation in descriptor reconstruction.

To resolve the above large deviation problem, we reconsider the definition of saliency, and propose a salient coding based method for image classification. Specifically, we apply the difference between a descriptor's nearest code and other codes to represent the descriptor. This strategy can stably obtain salient representation and avoid the large deviation problem that exists in reconstruction based coding methods. We study salient coding on the 15 natural scenes and PASCAL VOC 2007. In these databases, our method performs very well.

We have three main contributions in this paper:

1. Comprehensively analyze various coding schemes in the codebook based model, including their advantages and limitations. Based on the analysis, we present, for the first time to our knowledge, that saliency is the fundamental characteristic of coding.

2. Propose a novel salient coding algorithm based on the above analysis. It can stably extract saliency representation to descriptors without the large deviation problem that exists in reconstruction based coding methods.

3. Conduct a number of experiments on two kinds of image classification databases: 15 natural scenes and PASCAL VOC2007. In these experiments, we study the influence of the parameters in salient coding, and compare salient coding with other coding strategies. Our method achieves very good performance, comparable to the state-of-the-art algorithms on both the 15 natural scenes dataset and the PASCAL VOC2007 database. Moreover, it runs much faster than reconstruction based coding strategies.

The rest of this paper is organized as follows. In Section 2, we analyze various coding schemes including their advantages and limitations. Based on the discussion, we propose salient coding. Section 3 provides experimental studies on the 15 natural scenes dataset and the PASCAL VOC2007 database. Finally, we conclude the paper in Section 4.

2. Our method

In this section, we introduce classic coding strategies, analyze their limitations and propose our solution.

2.1. Classic coding methods

Let x be a descriptor, e.g., 128 dimensional SIFT descriptor, $B = [b_1, b_2, ..., b_M]$ be a codebook with M cluster centers and $V = [v_1, v_2, ..., v_M]$ be the responses of the codebook.

To represent x, hard voting assigns 1 to the nearest code and 0 to others:

$$v_i = \begin{cases} 1, & \text{if } i = \arg\min_j(||x - b_j||_2) \\ 0, & otherwise \end{cases} \quad (1)$$

Hard voting [14] only uses frequency information of codes and is limited to describe descriptors.

[1] The reconstruction is implemented via resolving a least square optimization problem. We detail it in Section 2.

In soft voting [26], a descriptor is encoded by multiple codes using the kernel function (e.g., Gaussian function K_σ) of distance between descriptors and code:

$$v_i = K_\sigma(||x - b_i||_2), \qquad i = 1, 2, ..., M \tag{2}$$

Although soft voting achieves better performance than hard voting, it is not sufficient to obtain more complete representation to descriptors. In this sense, reconstruction based sparse coding contains more information than soft voting. To better understand sparse coding, we borrow the explanation from [29] and rewrite hard voting as:

$$\begin{aligned} V &= \arg\min ||x - VB^T||_2 \\ s.t.\ &||V||_0 = 1,\ \sum_i^M v_i = 1 \end{aligned} \tag{3}$$

where $||\bullet||_0$ denotes the l^0-norm, which counts the number of nonzero entries in a vector.

Given the above form, it is clear that the l^0-norm to V is too strong which leads to limited description to x. In sparse coding, the l^0-norm is replaced by the l^1-norm which is integrated into the optimization object function:

$$\begin{aligned} V &= \arg\min ||x - VB^T||_2 + \lambda||V||_1 \\ s.t.\ &\sum_i^M v_i = 1 \end{aligned} \tag{4}$$

where $||\bullet||_1$ denotes the l^1-norm.

Further studies [30] found that the locality constraint is more important than the sparse constraint. In LCC, the locality constraint is obtained by minimizing the Euclidean distance between a descriptor and codes used to reconstruct it:

$$\begin{aligned} V &= \arg\min ||x - VB^T||_2 + \lambda \sum_i ||v_i||_1\ ||x - b_i||_2 \\ s.t.\ &\sum_i^M v = 1 \end{aligned} \tag{5}$$

In this manner, LCC forces distant codes away from descriptor reconstruction and focuses on near codes. A more detailed explanation can be found in [30]

The computation cost of sparse coding and LCC is relatively high. Recently, LLC [28], a simplified and fast version of LCC, is proposed, wherein the constraint of $\sum_i ||v_i||_1\ ||x - b_i||_2$ is replaced by using K nearest codes to reconstruct each descriptor:

$$\begin{aligned} V &= \arg\min ||x - \tilde{V}\tilde{B}^T||_2 \\ s.t.\ &\sum_i v = 1\ ,\ \forall i = 1, 2, ..., M \end{aligned} \tag{6}$$

where $\tilde{B}$ is K nearest codes to x.

Because $\tilde{B}$ is the nearest K codes, LLC achieves locality representation. In addition, as K is a small number compared with the size of codebook, LLC also achieves sparse representation. This optimization problem has an analytical solution. It reduces the computation complexity from $O(M^2)$ in sparse coding and LCC to $O(M + K^2)$.

2.2. Salient coding

The work of sparse coding/LCC/LLC achieves surprising good performance using very simple local features and classifiers. On many image classification databases, they greatly outperform hard voting. On the image classification competition of PASCAL VOC2009, It ranks first on 18 out of 20 object classes. Why SC/LCC/LLC performs so excellently on image classification? We consider that the reason is **salient representation** under maximum pooling operation.

We take LLC as an example. Every local descriptor leads to a representation using K nearest codes chosen from the codebook. After encoding all descriptors, each code may obtain multiple responses. In the later maximum pooling operation, low responses on each code are suppressed and only the maximum response is preserved. Therefore, the meaningful responses are those largest responses on each code.

Next, we firstly analyze some numeric examples on these preserved responses, and then provide the mathematical and physical explanations.

Table 1 shows two examples of the data in LLC. x_1 and x_2 are two 128 dimensional SIFT descriptors. $[b1, b2, ..., b5]$ denotes five codes. $||x - b||_2$ indicates the Euclidean distance between x and b. $[v1, v2, ..., v5]$ is the responses of codes calculated by Eq. (6).

Table 1. Two examples showing that LLC reflects saliency.

$\|\|x_1 - b_1\|\|^2$	$\|\|x_1 - b_2\|\|^2$	$\|\|x_1 - b_3\|\|^2$	$\|\|x_1 - b_4\|\|^2$	$\|\|x_1 - b_5\|\|^2$
2.4333	3.3371	4.4999	4.5241	4.6679
v_1	v_2	v_3	v_4	v_5
1.1022	0.4160	-0.084	0.0427	-0.4760

$\|\|x_2 - b_1\|\|^2$	$\|\|x_2 - b_2\|\|^2$	$\|\|x_2 - b_3\|\|^2$	$\|\|x_2 - b_4\|\|^2$	$\|\|x_2 - b_5\|\|^2$
2.5518	2.9176	3.2088	3.3748	3.3834
v_1	v_2	v_3	v_4	v_5
0.3881	0.2975	0.0308	0.0938	0.1824

It is obviously that LLC reflects the saliency properties. That is, if a code (b_1) is much closer to the descriptor (x_1) than other codes (b_2 to b_5), this code obtains a much larger response (v_1) than other codes' responses (v_2 to v_5). The large response v_1 is highly possible to win the later maximum pooling competition. If no code is much closer to a descriptor than other codes (the case of the bottom part of Table 1), there will be no large response for all codes. In this case, all responses may be suppressed and discarded. Therefore, the preserved largest responses are salient representation to descriptors.

Figure 2 (a) illustrates the mathematical explanation of reconstruction based coding strategy in the case of $K = 2$ in a 2-dimensional feature space. The red balls and the green rectangles denote descriptors and codes respectively. In LLC, the reconstruction by least square optimization in Eq. (6) is equal to vector composition by the parallelogram law [5]. When the descriptor (F2) is very close to a code (C2) and far away from the other one (C3), LLC produces a large response (V2') for C2 and a small response (V3') for C3. When a descriptor (F1) locates in the middle of two codes (C1 and C2), the responses (V1 and V2) for both of them are not strong enough. For the case with larger K, the analysis is similar, i.e., using the parallelogram law multiple times.

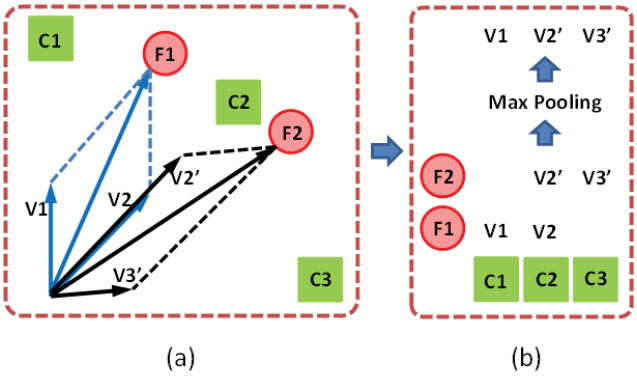

Figure 2. (a) An illustration of descriptors reconstruction by the parallelogram law. (b) An illustration of maximum pooling operation.

What's the physical meaning of the salient representation under the framework of the reconstruction based coding strategy and MAX pooling? When a code obtains a very strong response, i.e., this code is much closer to the descriptor than others, this code can **independently** describes the descriptor (salient representation). This is like the case of C2, C3 and F2 in Figure 2(a), where V2' can approximately represent F2 without V3'. When all responses in representing a descriptor are weak (not salient representation, i.e., codes are similarly close to the descriptor), the descriptor needs multiple codes to be described. This is like the case of C1, C2 and F1. In this case, any single code **cannot independently** represent the descriptor, and its response is unstable because weak response may be discarded, e.g., the response (V2) of C2 on F1 is suppressed by the one (V2') on F2.

In the later maximum pooling operation, high responses tend to be preserved, each of which independently describes a pattern of a descriptor. When the size of codebook is sufficient large, the salient representation of the codebook can effectively describes many patterns of descriptors which covers most areas of images. This leads to the final robust representation of images.

The least square based reconstruction is a good way to reflect the salient representation. It can exactly obtain salient representation in the low dimensional space guaranteed by the parallelogram low (see Figure 2(a)). However, if K is smaller than the dimensionality of descriptors, it is very possible that descriptors and codes are not in the same plane. In this case, it is not guaranteed that a descriptor can be reconstructed by K codes. Figure 3 illustrates an example wherein two codes C1 and C2, in a 3-dimensional feature space, cannot represent the descriptor x by the parallelogram law. The distance between x and x' reflects the deviation of reconstruction.

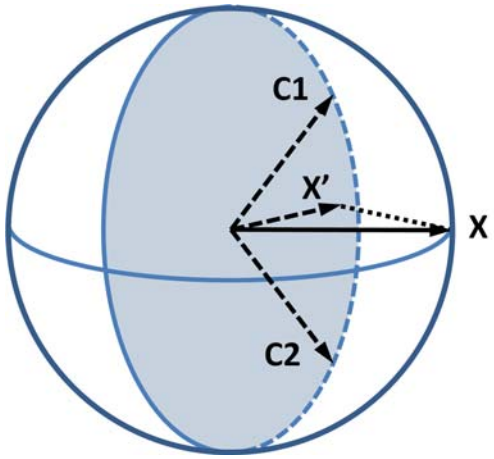

Figure 3. An example showing the deviation generated by descriptors reconstruction in LLC. The least square based optimization in LLC (Eq. (6)) is equal to reconstruct x' using C1 and C2, where x' is the project vector of x in the plane decided by C1 and C2.

For general application of the codebook based model, 128 dimensional SIFT descriptors are usually used as the feature. Due to sparse constraint, K is usually much smaller than 128. Thus, Eq. (6) is an under-determined problem. The reconstruction to a descriptor may lead to large devi-

ation so as not to correctly reflect the saliency properties. In this case, LLC produces abnormal values. We show an example in Table 2.

In Table 2, b_1 and b_2 are similarly close to x_3, but the response on b_1 is much smaller than the one on b_2, even smaller than the one on b_5.

Table 2. An example showing the failure of LLC in representing saliency.

$\|\|x_3 - b_1\|\|^2$	$\|\|x_3 - b_2\|\|^2$	$\|\|x_3 - b_3\|\|^2$	$\|\|x_3 - b_4\|\|^2$	$\|\|x_3 - b_5\|\|^2$
4.0504	4.0605	4.3845	4.7321	4.7405
v_1	v_2	v_3	v_4	v_5
0.2642	0.6194	-0.4171	0.1923	-0.3413

To solve the above problem, we should return to the definition or the nature of the saliency. In common sense, saliency indicates the most noticeable or important property. For coding operation in the codebook based model, saliency means that the nearest code is much closer to a descriptor than other codes. Thus, we can employ the difference between the nearest code and other $K-1$ codes to reflect saliency. Specifically, we use the ratio of them to define **saliency degree**:

$$\Psi(x, \tilde{b}_i) = \Phi\left(\frac{||x - \tilde{b}_i||_2}{\frac{1}{K-1}\sum_{j\neq i}^{K} ||x - \tilde{b}_j||_2}\right) \tag{7}$$

where $\Psi(x, \tilde{b}_i)$ denotes the saliency degree in the process of using $\tilde{b}$ to describe x, Φ is a monotonically decreasing function, and $[\tilde{b}_1, \tilde{b}_2, ..., \tilde{b}_k]$ is the set of K nearest codes to x.

With the definition of saliency degree, it is easy to introduce salient coding:

$$v_i = \begin{cases} \Psi(x, b_i), & \text{if } i = \arg\min_j(||x - b_j||^2) \\ 0, & otherwise \end{cases} \tag{8}$$

The salient coding has the following good properties:

1. Not limited by feature dimensionality. Our method starts from the intuitive definition of saliency. It does not rely on reconstruction, thus has no the underdetermine problem in LLC.

2. Easily to implement. There is no optimization in salient coding. The monotonically decreasing function can be any form. In our experiments, we use $\Phi(z) = 1 - z$ for convenience of normalization. In future work, we will study the influence of different forms.

3. Very fast. Our algorithm is much faster than sparse coding, LCC and LLC. The computation is nearly equal to hard voting.

3. Experimental results

In this section, we experimentally compare our method with other excellent algorithms in two sets of image classification databases: 15 natural scenes dataset [9] and Pascal VOC 2007 [6]. We first study our method in the 15 natural scenes dataset with an in-depth analysis, including parameters discussion, and then use a fixed set of parameters in another database.

We note that for some algorithms, we could not reproduce their performance, possibly due to engineering details, e.g., normalization of features, clustering techniques, dimensionality reduction and SVM parameters. Thus, we implement some of these algorithms which are exactly based on the same algorithm framework. These methods are: hard voting based codebook model, soft voting based codebook model and LLC based codebook model[2]. This kind of comparison makes more sense because it does not bias any methods by different implementations. In addition, for reader's convenience, we also quote the best results on these databases directly from the literature.

In our framework, we use the 128 dimensional SIFT descriptor [20] which densely extracted from images on a grid with step size of 4 pixels under three scales: 16×16, 24×24 and 32×32. We use the standard K-means clustering algorithm to generate codebook. The parameters of the codebook size and the number of nearest codes (K) will be discussed or stated in each experiment. Lib-linear SVM [8] is adopted for classification wherein the penalty coefficient is set to 1. In the 15 natural scene, we repeat the experiment 10 times with different random selected training and testing samples, and show the average accuracy and the standard deviation. All experiments are conducted in a server with an Intel E5520 CPU (2.27GHz and 16 cores) and 16G RAM.

3.1. 15 natural scenes dataset

The 15 natural scenes dataset consists of 4,485 images spread over 15 categories, each of which contains 200 to 400 images. The images categories vary from outdoor

[2]The LLC based codebook model is implemented via embedding the key part of the open LLC code [7] into our framework. In all experiments on LLC, we set $K = 5$ according to the best performance reported in [28].

scenes like mountains and forest to indoor environments like living room and kitchen. We follow the experimental setup of Lazebnik et al. [19] wherein 100 random images per class are chosen as training samples and the rest are used for testing.

On this dataset, we study the influence of K in Eq. (7) to our algorithm, and then compare the performance of various coding strategies under different size of codebook.

Figure 4 shows the performance when K = 2, 5, 10, 20 and 40 respectively under 4,096 codes. The experimental results indicate that a small K leads to good performance. As K increases to and more than 20, the performance decreases quickly. This is because the saliency degree defined in Eq. (7) tends to be equivalent when using too many neighboring codes. When K=5, our method performs best, thus we fix K to 5 in the rest of experiments.

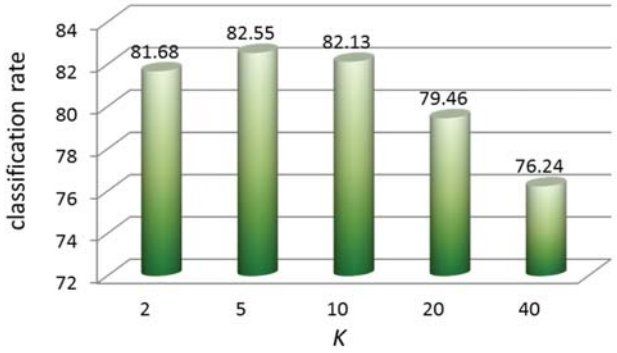

Figure 4. Performance of salient coding under different K on the 15 natural scene dataset.

Figure 5 shows the results of various coding strategies under different size of codebook. The salient coding outperforms other three coding schemes when the codebook size is large. When the codebook size is small, e.g., 256, our method is worse than hard voting and soft voting, because a small size of codebook can represent only a few local features of images in our method. But for hard voting and soft voting, the performance does not decrease so fast because the response of the codebook can still reflect the distribution of local features. When the codebook size is large (larger than 1,024), our method performs best in all coding schemes.

The performance LLC and our method are sensitive to the codebook size because these two methods rely on salient representation under the maximum pooling operation. A large number of codes can reflect many local descriptors of an image. In contrast, the performance of hard voting and soft voting is relatively stable because they adopt average pooling operation that has no such direct link with the codebook size. However, hard voting and soft voting have no much potential to further improve the classification

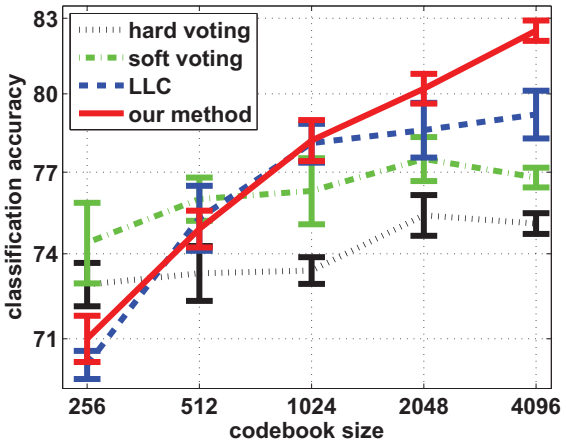

Figure 5. Performance comparison of various coding strategies under different sizes of codebook on the 15-scenes dataset.

accuracy since the codebook size has little influence to the average pooling operation.

Besides, we list some excellent results reported by other algorithms in table 3. Our method uses 4,096 codes and achieves very competitive performance.

Table 3. Several existing algorithm an their performances on the 15 natural scene dataset.

KSPM [19]	81.40 ± 0.50
KC [26]	76.67 ± 0.39
Yang [29]	80.28 ± 0.93
Boureau [12]	83.6 ± 0.4
Ours	82.55 ± 0.41

3.2. PASCAL VOC2007

In this section, we test our algorithm on PASCAL VOC2007, which is one of the most challenging databases for image classification. All images are obtained from Flicker with large variation on size, illumination, scale, viewpoint, deformation and clutter, as well as complex backgrounds. The performance measure is the average precision (AP), a standard metric used by PASCAL challenge [3].

In this experiment, the codebook size used in salient coding is 24,000, and we sets K=5 for Eq. (6) and Eq. (7). In Table 4, we list our scores for all 20 classes on VOC 2007. We also compare our approach with other three cod-

ing methods and the best performance of the challenge.

In PASCAL VOC 2007, our approach outperforms other three coding methods, and is comparable to the best performance by recently reported algorithms. It is worth noting that most of these algorithms apply the combination of multiple kinds of features, codebook learning and subspace learning. Our system only uses gray SIFT features and K-means algorithm, and does not employ any codebook learning and subspace learning techniques.

Table 4. Performance comparison on PASCAL VOC2007.

class	hard voting	soft voting	LLC	ours	winner VOC07
aero	62.6	68.5	70.6	71.3	**77.5**
bicyc	51.7	58.2	63.6	**64.2**	63.6
bird	38.1	40.6	46.5	45.5	**56.1**
boat	57.5	60.8	66.2	67.4	**71.9**
bottle	23.8	26.7	29.3	29.8	**33.1**
bus	50.8	60.6	63.5	**63.9**	60.6
car	67.7	72.2	76.6	**78.2**	78.0
cat	44.0	48.8	57.7	**59.2**	55.8
chair	45.3	51.2	**53.7**	53.6	53.5
cow	29.0	34.5	42.8	**43.3**	42.6
table	41.1	46.6	50.2	48.2	**54.9**
dog	35.5	36.6	43.3	43.8	**45.8**
horse	71.3	73.6	75.5	76.2	**77.5**
mbike	54.2	62.1	65.3	**66.4**	64.0
person	77.8	80.8	82.5	82.9	**85.9**
plant	17.2	24.5	27.4	29.1	**36.3**
sheep	35.0	38.8	46.1	**46.5**	44.7
sofa	38.5	46.0	52.2	**52.4**	50.9
train	60.9	71.4	75.9	76.1	**79.2**
tv	45.7	48.2	51.9	52.0	**53.2**
mean	47.8	52.6	57.0	57.5	**59.4**

4. Conclusion

In this paper, we have analyzed various coding strategies in the codebook based model and especially we discuss the LLC from the viewpoint of geometry, including its advancements and limitations, i.e., the reconstruction deviation. We have demonstrated that saliency is an fundamental property of coding. Based on this analysis, we have proposed a novel and fast coding strategy, called salient coding. It can effectively obtain salient representation to descriptors without the deviation problem in reconstruction based coding methods. Experiments on different kinds of databases (15 nature scenes dataset and PASCAL VOC 2007 database) demonstrate that salient coding achieves better classification accuracy than previous coding schemes. At the same time, salient coding largely reduces the computation cost compared with various reconstruction based coding schemes.

In future, we will study the influence of different choices of the form of salient coding, and experimentally analyze its performance in more kinds of image classification database.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No.60875021,60723005), NLPR 2008NLPRZY-2, National Hi-Tech Research and Development Program of China (2009AA01Z318), Key Project of Tsinghua National Laboratory for Information Science and Technology.

References

[1] www.vision.caltech.edu/Image_Datasets/Caltech101/.

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

[3] http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

[4] http://trecvid.nist.gov/.

[5] http://en.wikipedia.org/wiki/Parallelogram_law.

[6] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007.

[7] http://www.ifp.illinois.edu/ jyang29/.

[8] http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

[9] http://www.cs.unc.edu/~lazebnik/research/scene_categories.zip/.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[11] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992.

[12] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. *CVPR*, 2010.

[13] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.

[14] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. *ECCV*, 2004.

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

[16] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[17] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[18] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *Technical Report, Beckman Institute, University of Illinois*, 2004.

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

[20] D. G. Lowe. Distinctive image features from dcale-invariant key-points. *International Journal of Computer Vision*, 2(60):91–110, 2004.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[22] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[23] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[24] R. E. Schapire. The strength of weak learnability. *Machine Learing*, 5(2):197–227, 1990.

[25] T. Tuytelaars and L. V. Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[26] J. van Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. *ECCV*, 2008.

[27] P. Viola and M. Jones. Robust real-time object detection. *Proc. of IEEE Workshop on Statistical and Computa-tional Theories of Vision*, 2001.

[28] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei. Content based image indexing and searching using daubechies wavelets. *Journal of Digital Libraries*, 1(4):311–328, 1998.

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

[30] K. Yu, T. Wang, and Y. Gong. Nonlinear learning using local coordinate coding. *NIPS*, 2009.