21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Group Encoding of Local Features in Image Classification

Zifeng Wu, Yongzhen Huang, Liang Wang, and Tieniu Tan
National Lab of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{*zfwu, yzhuang, wangliang, tnt*}*@nlpr.ia.ac.cn*

Abstract

Saliency is an important factor in feature coding, based on which saliency coding (SaC) has been proposed for image classification recently. SaC is both effective and efficient in case of a moderate-scale codebook. However, empirical studies show that SaC will lose its superiority as the codebook size increases. To address this problem, we propose a group coding strategy, wherein the latent structure information of a codebook is explored by grouping neighboring codewords into a group-code. We apply group coding to SaC and derive the group saliency coding (GSC) scheme. Thorough experiments on different datasets show that GSC consistently performs better than SaC, and also outperforms other popular coding schemes, e.g., local-constrained linear coding, in terms of both accuracy and speed.

1. Introduction

Image classification has been one of the most active research areas in computer vision in the recent literature. Among various approaches to this purpose, the bag-of-words (BoW) model [2] is probably the most widely-used one. As illustrated in Figure 1 (a), there are basically four steps in the BoW model, i.e., feature extraction, coding, pooling, and classification. Among these steps, coding has become one of the hottest topics in image classification recently, and various coding schemes have been proposed. Probabilistic schemes such as hard voting (HV) [2] and soft voting (SV) [9] work well with a small-scale codebook. High-dimensional schemes, such as Fisher kernel coding [8] and super-vector coding [12] achieve impressive performance. However, a large quantity of memories are required for them. Reconstruction-based schemes also achieve high performance, but usually bear inefficiency in calculation, e.g., sparse coding [11]. Some of them cut down the computational cost by approximate algorithms, e.g., local-constrained linear coding (LLC) [10], which however inevitably introduces reconstruction errors, as detailed in [5].

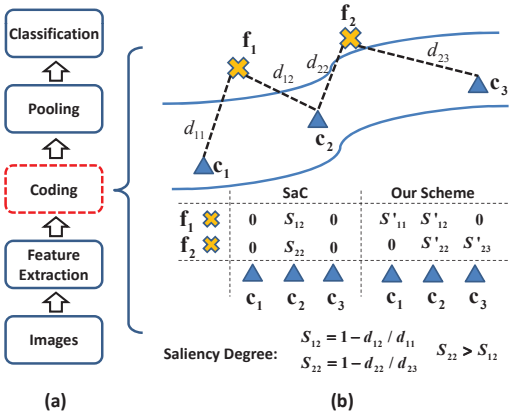

Figure 1. (a) Basic pipeline of the BoW model. (b) Motivation of this work.

The idea behind saliency coding (SaC) [5] is that a visual codeword should receive a strong response if it is much more similar with a feature than other codewords. SaC performs much better than classic probabilistic schemes and runs much faster than reconstruction-based schemes [5]. Nevertheless, SaC will lose its superiority in performance when the codebook size is relatively large. We attribute this problem to the hard-assignment strategy adopted in SaC, i.e., a feature is represented only by its nearest neighboring codeword. As a result, the feature's representation tends to be suppressed during max pooling. Consider the example in Figure 1 (b), wherein $\mathbf{f}_i$ denotes a local feature, $\mathbf{c}_j$ denotes a visual codeword, d_{ij} denotes the distance from $\mathbf{f}_i$ to $\mathbf{c}_j$, S_{ij} denotes the response of $\mathbf{f}_i$ to $\mathbf{c}_j$ in SaC, and S'_{ij} denotes the response in our scheme to be described in this paper. The response of $\mathbf{f}_1$ to $\mathbf{c}_2$ (S_{12}) is suppressed by the response of $\mathbf{f}_2$ to $\mathbf{c}_2$ (S_{22}) (since $S_{12} < S_{22}$), which results in the absence of $\mathbf{f}_1$'s representation. If multiple representations are generated for each feature, such side effect of suppression can be al-

leviated. For example in Figure 1 (b), even though the response of $\mathbf{f}_1$ to $\mathbf{c}_2$ (S'_{12}) is suppressed by the response of $\mathbf{f}_2$ to $\mathbf{c}_2$ (S'_{22}), the representation of $\mathbf{f}_1$ can still be found on $\mathbf{c}_1$ (S'_{11}).

It seems like that the above problem can easily be solved with soft-assignment. However, according to the original definition of SaC, the saliency degree of a feature only makes sense on its nearest neighboring codeword. As a result, soft-assignment is not applicable in SaC. To cope with this problem, we propose to treat a group of neighboring codewords as a single codeword, i.e., *group-code*, and perform group coding. A feature can thus be represented by the codewords included in group-codes. Experimental results on various datasets show that GSC outperforms SaC and LLC, which verifies the effectiveness of group coding.

2. Methods

In this section, we suppose that there are N codewords in the codebook $\mathbf{C}$, denoted by $\mathbf{c}_j$ respectively, and that M local features, denoted by $\mathbf{f}_i$, are extracted densely from an image.

2.1 Saliency coding

The main idea of saliency coding (SaC) is to encode local features according to the relative positions of features and codewords, as explained in Section 1. Let $\mathbf{s}_i$ denote $\mathbf{f}_i$'s coding result, and $\phi(\mathbf{f}_i)$ denote the saliency degree. The original definition of SaC can be written as [5]:

$$\mathbf{s}_i(j) = \begin{cases} \phi(\mathbf{f}_i) & \text{if } j = \arg\min_j \|\mathbf{f}_i - \mathbf{c}_j\|_2 \\ 0 & \text{else} \end{cases} \quad (1)$$

$$\phi(\mathbf{f}_i) = \frac{\sum_{t=2}^{K_S} (\|\mathbf{f}_i - \tilde{\mathbf{c}}_t\|_2 - \|\mathbf{f}_i - \tilde{\mathbf{c}}_1\|_2)}{\sum_{t=2}^{K} \|\mathbf{f}_i - \tilde{\mathbf{c}}_t\|_2} \quad (2)$$

wherein K_S denotes the number of codewords involved in calculating the saliency degree for each feature, and $\tilde{\mathbf{c}}_t$ denotes $\mathbf{f}_i$'s t-th nearest neighboring codeword, e.g., $\tilde{\mathbf{c}}_1$ is the nearest one.

Previous studies [5] show that SaC holds superiority in both effectiveness and efficiency. Nevertheless, there exists a limitation resulted from the hard-assignment strategy, as mentioned in Section 1. Moreover, it is impossible to derive a soft-assignment version considering that the saliency degree of a feature in SaC only makes sense on the feature's nearest neighboring codeword. Consequently, it is necessary to introduce a new strategy to enable each feature to vote for multiple codewords.

2.2 Group coding

The main idea of group coding is to treat several neighboring codewords as a single codeword, namely, a group-code. As illustrated in the second row of Figure 2, $\mathbf{c}_1$ and $\mathbf{c}_2$ comprise a group-code for $\mathbf{f}_i$, i.e., $\mathbf{c}_{\mathbf{f}_i,2}$, where the subscript 2 means that there are two codewords embedded here. Codewords in the same group-code will act as an integrated unit and receive the same response during a normal coding process. Such coding process will be repeated several times to cover different group-code sizes, and multiple coding results are obtained. Finally, we integrate these results to generate the output of the coding stage.

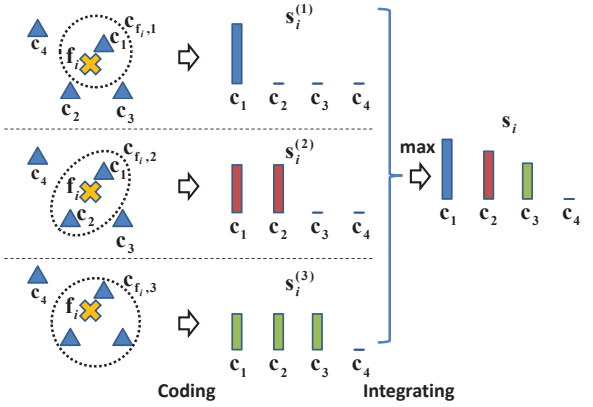

Figure 2. Illustration of group coding.

We mathematically obtain $\mathbf{f}_i$'s final coding output $\mathbf{s}_i$ with:

$$\mathbf{s}_i = \max_{k=1,\ldots,K} \mathbf{s}_i^{(k)} \quad (3)$$

wherein K denotes the maximum group-code size that should be considered, $k = 1, \ldots, K$ denotes different group-code sizes, and $\mathbf{s}_i^{(k)}$ denotes a normal coding result obtained with the group-code size k (to be detailed in the next subsection).

It is noteworthy that group coding is not an analogue of soft-assignment, even if they look similar. In the case of group coding, different number of neighboring codewords are grouped together and act as an integrated unit. It means that we are exploring the latent structure information that some neighboring codewords are potentially synonymous with each other. This structure information can not be sufficiently reflected by the soft-assignment strategy.

2.3 Group saliency coding

As mentioned in Section 1, we perform group coding for SaC to alleviate the side effect of suppression. Let

$\mathbf{s}_i^{(k)}$ denote $\mathbf{f}_i$'s coding result obtained with the group-code size k, and $\phi^{(k)}(\mathbf{f}_i)$ denote the revised saliency degree. The group saliency coding (GSC) scheme is defined as:

$$\mathbf{s}_i^{(k)}(j) = \begin{cases} \phi^{(k)}(\mathbf{f}_i) & \text{if } \mathbf{c}_j \in g(\mathbf{f}_i, k) \\ 0 & \text{else} \end{cases} \quad (4)$$

$$\phi^{(k)}(\mathbf{f}_i) = \sum_{t=1}^{K+1-k} (\|\mathbf{f}_i - \tilde{\mathbf{c}}_{k+t}\|_2^2 - \|\mathbf{f}_i - \tilde{\mathbf{c}}_k\|_2^2) \quad (5)$$

wherein $g(\mathbf{f}_i, k)$ denotes $\mathbf{f}_i$'s k nearest neighboring codewords, and K is the maximum group-code size.

The main idea of GSC is to measure the relative positions between $\mathbf{f}_i$'s group-codes and other codewords. With different k, there are consistently $K+1$ neighboring codewords considered for each feature. The k nearest ones of them are taken as the group-code, and the k-th nearest codeword is the representative of the group-code for calculating the saliency degree. To ensure the comparability among K different $\phi^{(k)}(\mathbf{f}_i)$'s ($k = 1, \ldots, K$), we remove the normalization operation. We will obtain K coding results with Eq.(4), and integrate them with Eq.(3) to get the final coding output of $\mathbf{f}_i$.

3. Experimental results

3.1 Datasets and experimental settings

We perform a series of experiments on three datasets, i.e., 15 Scenes [6], Caltech 101 [4] and PASCAL VOC 2007 [3]. On 15 Scenes, we randomly pick out 100 images from each category for training, and keep the remaining images for testing [6]. On Caltech 101, we randomly pick out different number (10, 20 and 30) of images from each category for training, and pick out at most 50 images from each category for testing [4]. The experiments are repeated for 10 times. Average classification accuracy and standard deviation are reported. On VOC 2007, we follow the official experimental settings [3] and report the mean AP (average precision).

SIFT descriptors [7] are densely extracted every four pixels on three scales, i.e., 16×16, 24×24 and 32×32 in pixels. Codebooks are trained by k-means clustering. SPM [6] is performed on three levels, i.e., 1×1, 2×2 and 3×1. Linear SVMs are trained as classifiers. We re-implement the coding schemes in the same framework to achieve comprehensive comparison. As a result, there might be some slight differences between our results and those reported by the original authors.

3.2 Experimental results and analysis

The maximum group-code size K in Eq.(5) is an important parameter. We first conduct experiments with different K on 15 Scenes, as reported in Figure 3. In an overall view, GSC achieves better performance as K increases, until $K = 5$. This tendency is basically the same on the other two datasets. Consequently, in the remaining experiments, we keep the setting that $K = 5$.

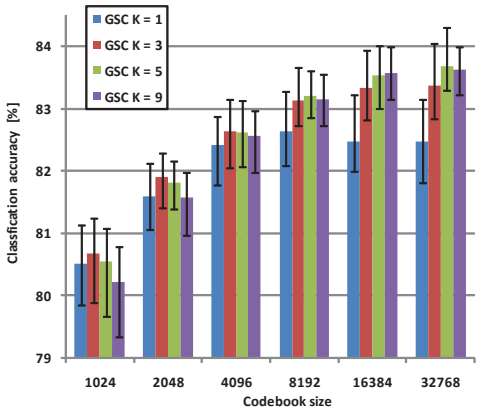

Figure 3. Classification results obtained with different K on 15 Scenes.

There are many coding schemes in the recent literature. LLC was one of the state-of-the-art schemes [10], and SaC is our baseline. The recent state-of-the-art schemes such as FK [1] and SVC [12] are not considered here. They are memory-consuming and dense, while LLC and GSC are compact and sparse. Therefore, we compare GSC with LLC and SaC on 15 Scenes, Caltech 101 and VOC 2007 respectively. In SaC, K_{S} denotes the number of codewords involved in measuring the saliency degree, while in LLC, K_{L} denotes the number of codewords that each feature is reconstructed with. Empirically, we keep $K_{\mathrm{S}} = 5$ and $K_{\mathrm{L}} = 5$ in our experiments [5, 10].

The results on different datasets obtained with different coding schemes are listed in Table 1. The numbers in parenthesis stand for different sizes of the training set for Caltech 101 [4]. GSC consistently outperforms SaC. In particular, GSC performs better than SaC by 2.0% at most on VOC 2007. These results verify the effectiveness of the proposed group coding strategy. Besides, GSC also performs better than LLC. We further increase the codebook size to 65,536 and perform experiments on the VOC 2007 dataset. The category-wise results are listed in Table 2, from which we can see that in terms of a large-scale codebook, GSC can achieve even better performance. As the codebook size increases, the reconstruction errors of LLC are alleviated. In this sit-

Category	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	
LLC	71.7	63.3	48.9	68.1	27.4	67.1	77.0	59.9	**55.4**	47.2	
SaC	70.8	63.4	45.7	66.5	26.6	64.0	76.3	56.4	53.8	46.3	
GSC	**73.2**	**65.7**	**51.2**	**68.3**	**30.3**	**67.6**	**78.1**	**60.7**	55.1	**49.2**	
Category	dinningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	mean AP
LLC	**51.9**	44.6	76.8	66.4	83.6	27.4	**47.9**	54.5	76.1	**54.2**	58.5
SaC	46.6	43.4	75.6	64.4	82.6	26.4	43.5	50.9	75.3	51.7	56.2
GSC	49.4	**46.1**	**76.9**	**67.6**	**83.9**	**28.4**	45.6	**54.7**	**76.5**	53.3	**59.1**

Table 2. Class-wise comparison on PASCAL VOC 2007. The codebook size is 65,536.

Dataset	Codebook size	LLC	SaC	GSC
15 Scenes	2048	81.4 ± 0.4	81.5 ± 0.3	$\mathbf{81.8 \pm 0.4}$
	8192	83.0 ± 0.3	82.5 ± 0.5	$\mathbf{83.2 \pm 0.4}$
Caltech 101 (10)	2048	58.4 ± 0.6	56.9 ± 0.5	$\mathbf{58.9 \pm 0.7}$
	8192	60.8 ± 0.7	57.9 ± 0.4	$\mathbf{61.0 \pm 0.7}$
Caltech 101 (20)	2048	66.1 ± 0.7	65.3 ± 0.9	$\mathbf{66.8 \pm 0.7}$
	8192	69.0 ± 0.8	66.6 ± 0.8	$\mathbf{69.2 \pm 0.9}$
Caltech 101 (30)	2048	70.0 ± 1.5	69.4 ± 1.0	$\mathbf{71.0 \pm 1.2}$
	8192	72.6 ± 1.4	71.1 ± 1.1	$\mathbf{73.4 \pm 1.2}$
VOC 2007	2048	49.3	49.8	**51.6**
	8192	55.2	54.8	**56.2**
	32768	57.9	56.4	**58.4**

Table 1. Comparison of different coding schemes on different datasets.

uation, we can perfectly recover original local features from an encoded LLC representation. In contrast, GSC is not able to do so since it is derived based on saliency. This is probably one of the reasons why there are four categories on which LLC outperforms GSC in Table 2.

Efficiency. The computational complexity of GSC is $\mathcal{O}(K)$, which is the same as SaC, i.e., $\mathcal{O}(K_{\mathrm{S}})$ [5], while the one of LLC is $\mathcal{O}(K_{\mathrm{L}}^2)$ [10].

4. Conclusion and future work

In this paper, we have proposed the group coding strategy and applied it to SaC. The resulting group saliency coding (GSC) scheme has shown its superiority to SaC. GSC also outperforms other popular coding schemes such as LLC, requiring lower computational cost.

It should be noted that group coding can also cooperate with other coding schemes such as hard voting, and even with multiple voting schemes such as soft voting. Besides, the local structure of codebooks might be better explored by methods other than the K neighbors strategy. We will cover these aspects in our future work.

Acknowledgement. This work is jointly supported by National Natural Science Foundation of China (61135002, 61135003), Hundred Talents Program of CAS, National Basic Research Program of China (2012CB316300), Tsinghua National Laboratory for Information Science and Technology Cross-discipline Foundation (Y2U1011MC1), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030300).

References

[1] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[4] 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. In *CVPR*, 2004.

[5] Y. Huang, K. Huang, Y. Yu, and T. Tan. Salient coding for image classification. In *CVPR*, 2011.

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

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

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

[9] 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*, 32:1271–1283, 2010.

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

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

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