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

Adjacent Coding for Image Classification

Yueming Wang, Xinggang Wang, Shaojun Zhu, Xiang Bai, Wenyu Liu
Dept. of Electronics and Information Engineering, Huazhong Univ. of Science and Technology
{*yueminghust2007,wxghust,vincentzhu122,xiang.bai*}*@gmail.com, liuwy@hust.edu.cn*

Abstract

The locality and sparsity constrained encoding methods have shown the good image classification performance in recent papers. Among these methods, the common strategy is encoding one descriptor into one code by a learned codebook and then applying SPM and Pooling strategy to get the final image representation. However, the ignorance of local spatial context has been a barrier to improve their discriminative power. To address this problem, we propose the so called Adjacent Coding (AC), which employs the adjacency of one descriptor to express the local spatial context. Different from traditional coding methods, Adjacent Coding encodes one descriptor and its adjacent neighbors together. In this paper, we further show that AC also keeps the properties of locality and sparsity. Finally, our experiments on the standard benchmarks (Scene 15 and Caltech 101) show our method can outperform the state-of-the-art feature coding methods.

1. Introduction

Image classification is a very important problem in computer vision which has gained significant improvement in the last few years. Most of the advanced image classification system take the advantage of modern feature coding methods, in which low-level image features, e.g., SIFT [9], HOG [5], are encoded into high dimensional codes to obtain better discriminative power. Recently, the most successful feature coding methods includes [12] and [10] which keep the sparsity and locality of codewords.

Typical framework for image classification consists of 5 steps: (1) densely extracting low-level image descriptors (e.g., SIFT); (2) unsupervised learning codebook (e.g., Kmeans); (3) encoding image descriptors into image features (e.g., VQ [4]); (4) building SPM (Spatial Pyramid Matching [8]) and pooling multi-scaled sub-region features together; (5) training linear or non-linear classifiers (e.g., SVM [2]) for image classification. The recent research [3] has already shown that different encoding method has quite different performance under the same framework. Specially, Yang et al. [12] applied patch-based Sparse Coding (SC) of densely extracted SIFT features to represent image instead of the traditional Vector Quantization (VQ) method. The results show that SC provides a significant improvement than VQ. Wang et al. [10] proposed Localily-constraint Linear Coding (LLC) to project one descriptor into its local-coordinate system, which outperforms SC in many cases and is more efficient than SC.

All the above methods have one thing in common that encodes one descriptor in the way that transform one descriptor into one code. Hence, we don't know the relationship between one descriptor and its adjacent ones. In fact, even if we reorganize the positions of descriptors, the encoding result will always be the same, which is obviously unreasonable. Spatial relationship has very strong discriminative power for human perception of one image. Our idea is that besides encoding with sparsity and locality, we should also encode with spatial correspondence, more specific, with adjacency. Although SPM has already provided some rough spatial correspondence, adjacency will bring more locality, which will improve the discriminative power for every code. We propose the Adjacent Coding (AC). Compared with SC and LLC, AC encodes one descriptor with its adjacent neighbors together. AC encodes one descriptor according to its local spatial context meanwhile AC keeps local and sparse.

Our main contribution is that we add adjacency into encoding which has a strong discriminative power and also has sparsity and locality. We have examined our idea on the popular Scene 15 [8] dataset and Caltech 101 [6] dataset, which proves the adjacency is quite effective for image representation.

The rest of this paper is organized as follows: In Section 2, we will revisit the Sparse Coding and Locality-constrained Linear Coding. In Section 3, we introduce our proposed Adjacent Coding model. In Section 4, we

978-4-9906441-1-6

will evaluate our proposed method on image classification benchmark datasets. Finally, Section 5 presents the conclusion of this paper.

2. Encoding with Sparsity and Locality

Sparse Coding and Locality-constraint Linear Coding both encodes one descriptor into a sparse representation.They both aim at finding a small subset of codewords and reconstructing image descriptor with the codewords by solving a ℓ_1 or ℓ_2 minimization problem. Given a set of D-dimensional descriptors densely extracted from an image, i.e. $\{X_1, X_2, \ldots, X_N\}$ and a codebook B with M entries, i.e. $B = \{b_1, b_2, \ldots, b_M\}$, where each codeword is also D-dimensional.

2.1. Sparse Coding

Codebook is created from a set of training images by standard clustering algorithm, e.g., Kmeans. Codebook represents the centroids of the large set of input descriptors. Usually, codebook size is quite large to get high performance, i.e, codebook is usually over-complete. Sparse Coding attempts to find a small subset of codebook to reconstruct the original descriptor by combing the codewords in a weighted way. For one descriptor, the idea of Sparse Coding is that projecting the descriptor only onto a limited number bases. That is

$$\arg\min_{C} \sum_{i=1}^{N} \|x_i - Bc_i\|^2 + \lambda \|c_i\|_{l^1} \qquad (1)$$

where c_i denotes the code of descriptor x_i, λ denotes the sparsity regulation term. Hence, each descriptor is encoded by the strong activation of relatively small set of codewords. In this way, one descriptor can be seen as being generated by a small number of distinct codewords. Sparse Coding captures the sparse structure of distribution of codewords meanwhile keeps a low reconstruction error against Vector Quantization.

2.2. Locality-constraint Linear Coding

LLC incorporates the locality constraint instead of sparsity. LLC encodes one descriptor by projecting it onto the k-nearest neighbor codewords. k is usually between 3 and 10. As suggested by [10], locality is more essential than sparsity, as locality must lead to sparsity but not vice versa. The locality constraint can be written as follows:

$$\min_{C} \sum_{i=1}^{N} \|x_i - kNN(x_i)c_i\|^2 \qquad (2)$$
$$s.t. \mathbf{1}^T c_i = 1, \forall i$$

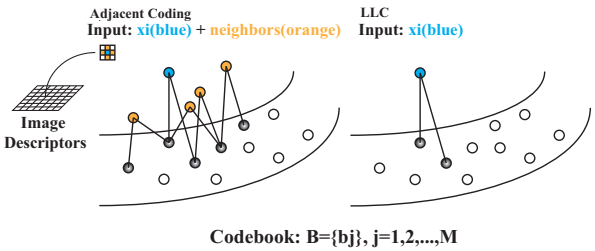

Figure 1: Comparison between AC and LLC. Both AC and LLC encodes one descriptor(blue). The adjacent neighbor descriptors are in orange. The Selected bases for representation are highlighted.

where kNN(x_i) denotes the k nearest neighbors of x_i. Different from Sparse Coding, LLC keeps the code invariant for one descriptor while sparse coding may change the code due to the over-completeness of the codebook. LLC generates similar codes for similar descriptors. Thus, LLC outperforms Sparse Coding in many cases.

3. Adjacent Coding Model

3.1 Spatial Adjacency

SC and LLC represents one descriptor by projecting it onto limited number bases or just local bases to keep the code sparse. Although sparse representation has shown to be quite effective, the spatial correspondence of descriptors is not included. This may cause the loss of spatial structure information of image representation, which obviously will influence the final classification performance.

Thus, we propose the Adjacent Coding (AC) to model the spatial relationship. The spatial relationship of descriptors depicts the relationship of image patches. Thus, AC actually reveals the local structure. Here, we constrain the spatial relationship into adjacency which is a quite strong description of local correspondence. Given one descriptor x_i, its 4-adjacent descriptors are noted as $N_4(x_i)$. In AC, we simply define $\{x_i, N_4(x_i)\}$ as the spatial adjacency of x_i.

3.2 Weighting by heat kernel

LLC applies a least square solution to get each code. The efficiency will be lower if the descriptor has a higher dimensions. We propose heat kernel to address this problem. This is inspired by the classical perceptron and Radial Basis Coding [11]. For descriptor x_i and codeword b_j, We define the activation strength between these two as follows:

$$W(x_i, b_j) = e^{-\frac{\|x_i - b_j\|^2}{t}} \tag{3}$$

where $t \in R$ is a normalization parameter. In contrast to SC and LLC, we neither solve ℓ_2 nor any other ℓ_p norm problem. In our framework, we simply encode x_i by heat kernel.

3.3 Adjacent Coding

Assume $\text{kNN}(x_i, N_4(x_i)) = \{b_{i1}, b_{i2}, \ldots, b_{i\alpha}\}$, where α denotes the total number of the kNN union. The final formulation of AC should be as follows:

$$c(x_i) = \begin{cases} e^{-\frac{\|x_i - b_j\|^2}{t}} & \text{for } b_j \in \{b_{i1}, b_{i2}, \ldots, b_{i\alpha}\} \\ 0 & \text{elsewhere} \end{cases} \tag{4}$$

$$AC(x_i) = \frac{c(x_i)}{\|c(x_i)\|} \tag{5}$$

We show the encoding process of AC in Figure 1. AC has several features. AC represents the local spatial context in spatial adjacency. Meanwhile, AC keeps the sparse property by set most elements to 0 in one code. AC encodes one descriptor by searching its k-nearest neighbors and its adjacent neighbors' kNN. Hence, AC also keeps the locality property. The union of kNN reveals the salient pattern of local spatial context. Besides, AC is also smooth sparse. It encodes similar descriptors into similar codes. One descriptor is similar to another one only if it locates in the similar spatial context, which is considerably discriminative for classification.

The overall procedure for Adjacent Coding is shown in Algorithm 1.

Algorithm 1 Adjacent Coding Algorithm

1: Dense extracting SIFT descriptors
2: Finding adjacent neighbors of one descriptor
3: Building the kNN union of one descriptor and its adjacent neighbors
4: Calculating Weights by heat kernel
5: Repeat step 2-4 until every descriptor is encoded
6: Pooling multi-scale descriptors together
7: Train linear SVM Classifiers

4. Experiments and discussions

In this section, we report classification results on benchmark Scene 15 and Caltech 101. We used the SIFT descriptors densely extracted from image at every 5 pixels as the low level features. Codebook was constructed by clustering the input descriptors by Kmeans. For fair comparisons, we conducted the experiment in the same manner as LLC. We assigned normalization parameter $t = 0.15$. We applied the same max-pooling strategy as LLC because max-pooling captures the most activate component within a spatial region and is robust to small translations. In our experiment, we used max-pooling combined with ℓ_2 normalization.

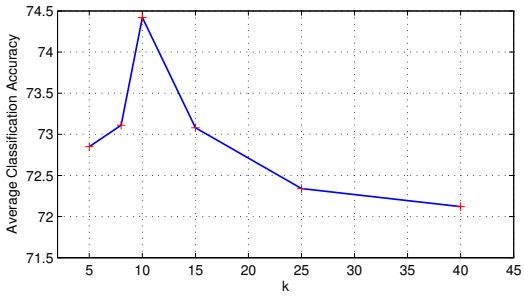

Figure 2: Performance of Adjacent Coding under different k

4.1 Scene15

Scene 15 dataset consists of 4,485 images from 15 categories such as mountains, forest, coast and so on. Each category contains 200 to 400 images. For each class, we randomly select 30 images as training examples and treat others as test examples. We assigned $k = 10$ which got the best result from our trials. While LLC keeps $k = 5$ which performs best according to their experiments. We summarize the results in Table1 as follows:

Table 1.Image classification results on Scene 15 dataset

codebook size	256	512
LLC	70.88	73.97
AC + least square	71.09	73.63
AC + heat kernel	**71.83**	**74.42**

Our experiment suggested that heat kernel has the comparable performance with least square for encoding. Besides, we also test how heat kernel speed up encoding procedure by comparing least square-based and heat kernel based AC. Encoding 33,840 descriptors by least square-based AC cost 4.036 seconds while heat kernel cost 3.441 seconds. Given that dense extraction strategy usually leads to a huge number of descriptors, AC adapts heat kernel instead least square for calculating weights. We then evaluate how performance will change by assigning different values to k as show in Figure 2, where codebook size is fixed to 512.

Figure 2 shows that the average classification accuracy of AC has a maximum when $k = 10$. As indicated above, in AC, k is a tradeoff between spatial context and sparsity representation. The larger k will lead to a less sparse code while lead to a better mixture of spatial context. However, we keep k=10 in all of our experiments for fair comparison with LLC method.

4.2 Caltech101

Caltech 101 dataset consists of 9,144 images of 101 categories such as airplanes, cameras, vehicles and animals, as well as one background category with significant variation. The number of images from each category varies from 31 to 800. Caltech 101 is more variant in size, location, pose, etc., than those of Scene 15 dataset. We followed the standard experimental settings from above.

We report the results of our proposed method and other existing method in Table 2. The results suggest AC performs a little better than SC and LLC under different training images. Although improvement is quite small under training image set to 30, performance has an obvious raise under other cases. For training image set to 20, the performance has raised 2.4%. Adjacent Coding combines the local spatial structures into code and thus has a more discriminative power, which improves the final classification performance. If we recover the 4 adjacent neighbors into patches in image, we can see that adjacent coding is actually a combination of local spatial context of one patch. We further observe that one descriptor and its adjacent neighbor is quite different if they are in the region of large variation but quite similar if the region is smooth.

Overall, our experiment confirms that Adjacent Coding is an effective encoding method and outperforms both LLC and Sparse Coding on Scene 15 and Caltech 101.

Table 2.Image classification results on Caltech 101 dataset

training images	5	10	15	20	25	30
Griffin [7]	44.2	54.5	59.0	63.3	65.8	67.6
Boiman [1]	-	-	65.00	-	-	70.40
SC [12]	-	-	**67.00**	-	-	73.20
LLC [10]	51.15	59.77	65.43	67.74	70.16	73.44
Ours	**52.15**	**62.03**	66.83	**70.15**	**72.05**	**74.16**

5. Conclusion

In this paper, we proposed the Adjacent Coding. Adjacent Coding is designed to capture the local spatial context of descriptors. Meanwhile we proved that Adjacent Coding also keeps locality and sparsity. We also introduced heat kernel to calculate coefficients in a more fast way instead of least square. Our experiments have shown that Adjacent Coding has a better performance than both LLC and Sparse Coding on Scene 15 and Caltech 101 datasets. Overall, we have proven that Adjacent coding has a more discriminative power and achieves state-of-the-art benchmark results.

Acknowledgements. This work was supported by the National Natural Science Foundation of China #60903096 and #61173120.

References

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR'08*, 2008.

[2] C. Chang and C. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011.

[3] 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.

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

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, 2005.

[6] L. Fei-fei. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. 2004.

[7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR'06*, 2006.

[9] D. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, 1999.

[10] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR'10*, 2010.

[11] X. Wang, X. Bai, W. Liu, and L. Latecki. Feature context for image classification and object detection. In *CVPR'11*, 2011.

[12] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR'09*, 2009.