LEARNING A COMPACT LATENT REPRESENTATION OF THE BAG-OF-PARTS MODEL

Xiaozhi Chen, Huimin Ma

Department of Electronic Engineering
Tsinghua University, Beijing 100084, China
chenxz12@mails.tsinghua.edu.cn, mhmpub@tsinghua.edu.cn

ABSTRACT

The Bag-of-Parts (BoP) model, which employs distinctive parts to represent images, has shown superior performance in vision recognition tasks. Our work is motivated by the need of reducing redundancy in tens of thousands parts. We propose a novel method to learn a compact latent representation from redundant part responses. We address this problem by employing spectral clustering and a multi-column coding scheme. The BoP model is viewed as a multi-scale convolutional model and additional sparse autoencoders are used to infer the latent patterns embedded in high-dimensional part-based representations. Spatial and semantic information is preserved by sparse learning on multiple spatial regions individually. Experiments demonstrate that the learnt representation achieves competitive performance with state-of-the-art methods on PASCAL VOC 2007 dataset.

Index Terms— mid-level representation, BoP, spectral clustering, multi-column sparse autoencoders

1. INTRODUCTION

Mid-level representations for images have recently gained much interest in visual recognition. They try to fill the semantic gap between low-level features (e.g. SIFT) and meanings of objects/scenes. Mid-level features could correspond to parts, objects, visual phrases, etc. For example, mid-level features correspond to recognizable clusters in appearance and configuration in Poselets [1], object categories in Object Bank [2] and object parts in Bag of Parts (BoP) [3, 4]. In this paper, we focus on the BoP representation.

Motivation. Our work is motivated by two problems. First, in part-learning algorithms [3, 4], the number of candidate part detectors can easily grow into tens of thousands as the number of object categories increases, thus the need of learning a compact representation from high-dimensional data arises. Second, part detectors learnt in weakly supervised setting (only class labels provided) are redundant and little information is gained even if more parts are used as shown in Fig. 3(a). Therefore, there is a need for reducing the redundancy while utilizing as much information provided by parts as possible. To address these problems, Singh et al. [4] ranks part detectors by measuring the purity and discriminativeness using SVM detection scores. Juneja et al. [3] introduces Entropy-Rank to select distinctive parts that are informative for a small proportion of classes. However, these methods do not explore latent structure in the part responses and they simply select a small subset of candidate parts. In this paper, we propose a novel algorithm to automatically learn a compact latent representation from a large set of part detectors. Our method differs from previous works. First, we employ spectral clustering to select representative part detectors which serve as mid-level primitives and perserve context information. Second, we view the BoP model as a multi-scale convolutional model conducted in Histogram of Oriented Gradients (HOG) [5] space and extent the model by stacking multi-column sparse autoencoders on top of part responses, which significantly reduces the original high-dimensional part responses while preserving the spatial information. We call the proposed compact BoP representation **C-BoP**.

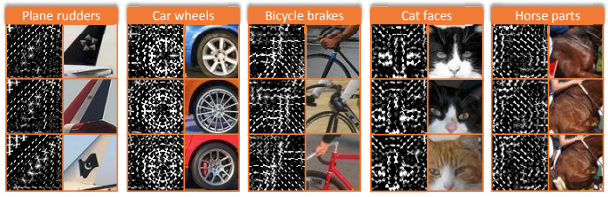

Fig. 1. Examples of part clusters obtained by spectral clustering. For each group, the left column shows some part templates within the same cluster and the right column shows corresponding patches that give maximal responses of the part detectors in validation set. Selected representative parts are used to learn a compact BoP representation.

This paper proceeds as follow. First, a parts selection method using clustering is introduced and a comparison with [3] is made in Sec.2.1. Then the compact BoP representation is described in Sec.3. Finally, experiments and analysis are demonstrated in Sec.4, and conclusion is made in Sec.5.

This work was supported by National Natural Science Foundation of China (No. 60502013 and No. 61171113).

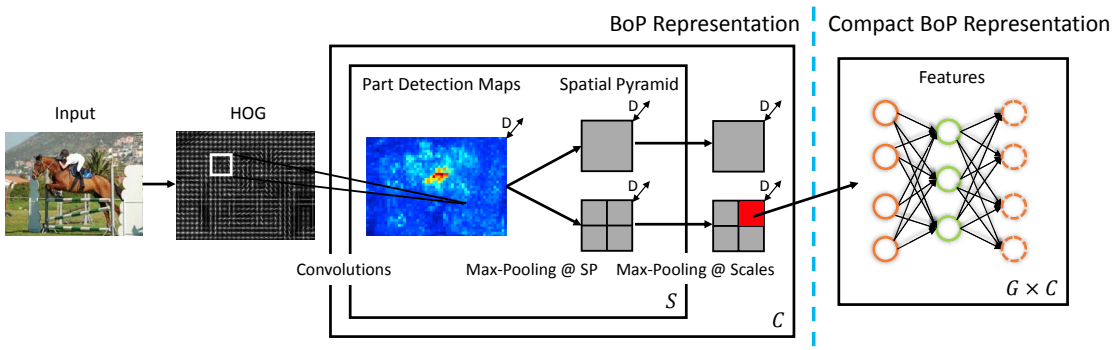

Fig. 2. Pipeline of the proposed model demonstrated using "plate-like" notation. The left part is the construction of the BoP representation from the view of multi-scale convolutional model. The right part is the compact BoP representation learnt by additional multi-column sparse autoencoders. The letter (i.e. S, C, or $G \times C$) in the corner of the plate indicates the number of repetitions of the subgraph. Refer to Sec.3 for more details.

2. PARTS SELECTION

We choose the BoP model introduced by Juneja et al. [3] to demonstrate our algorithm. In this section, we first introduce our method for selecting representative parts, then compare it with Entropy-Rank used in [3].

2.1. Clustering for Parts Selection

The goal of parts selection is to preliminarily reduce redundancy in tens of thousands part detectors obtained by the part-learning algorithm [3]. Let $\mathbf{w} \in \mathbb{R}^F$ (F is the dimension of HOG template, typically $F = 8 \times 8 \times 31$) denote a part template, which is actually the weights of a linear discriminant analysis (LDA) classifier learnt in the part-learning procedure. It is given by

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\mu_1 - \mu_0), \tag{1}$$

where $\mathbf{\Sigma}$ and μ_0 are the covariance matrix and the mean of the HOG features of the whole training set respectively, and μ_1 is the mean of HOG features of the positive part samples.

To reduce parts redundancy, a natural way is to use clustering to select representative parts. Notice that it is terrible to perform clustering in HOG space because the distribution of patches in HOG space is very non-uniform and low-level distance metrics (e.g. Euclidean, cross-correlation) cannot reflect visual similarity of parts. Fortunately, the term $\mathbf{\Sigma}^{-1}$ actually acts as a "whitening" operation to remove correlations in HOG space [6] and hence metrics in this whitened HOG space are more meaningful. We cluster the learnt part detectors into D clusters using spectral clustering [7]. The similarity graph is constructed using the exponential of cosine similarity $\langle \mathbf{w}_i / \|\mathbf{w}_i\|, \mathbf{w}_j / \|\mathbf{w}_j\| \rangle$ of two part templates. Then the part detectors nearest to the mean of each cluster are selected as representative parts. Fig. 1 shows some clusters obtained by spectral clustering. Notice that the resulting clusters categorize visually similar part detectors quite well. After this clustering step, the number of part detectors reduces to hundreds from about 10∼20 thousands per category produced originally by the part-learning algorithm.

2.2. Comparison with Entropy-Rank

Entropy-Rank introduced by [3] is trying to select parts that are discriminative while our method select representative parts without considering their discriminativeness. We argue that parts selection is actually a feature selection problem and that parts exhibit discriminativeness individually does not necessarily mean their combination is a good feature for classification. For example, Entropy-Rank may miss some parts containing context information which may not be discriminative itself but can be distinctive when combined with other parts. On the contrary, clustering ensures that all the potentially useful information including context will be utilized. Trival information will be filtered out in the process of learning latent representation, which will be specified in next section. As will be seen in Sec.4, our simple clustering method leads to a better representation for classification than Entropy-Rank with a much lower computational cost.

3. C-BOP REPRESENTATION

In this section, we first summarize the BoP representation from the view of multi-scale convolutional model. Then we present the coding scheme and learning procedure of the C-BoP representation respectively.

3.1. BoP Representation

Let C denote the number of categories in the dataset and D denote the number of selected part detectors per category. Then the total number of part detectors is $D \times C$. Convolutional kernels are obtained by these selected part templates. Given an input image, a HOG map is first computed to be fed into the model. A set of part detection maps is then obtained by applying the part filters to convolution operations

on HOG maps at multiple scales. Then max-pooling is performed twice over the detection maps: the first is conducted at multiple grids similar to Spatial Pyramid Matching (SPM) [8] to achieve translation-invariance and the second is conducted at multiple scales to achieve scale-invariance. Let S denote the number of scales and G denote the total number of grids in spatial pyramid. A BoP representation is obtained by concatenating a set of $G \times C$ grid- and category-wise sub-vectors $\{\mathbf{x}_{g,c}\}$, where $\mathbf{x}_{g,c} \in \mathbb{R}^D$ denotes the max responses of the set of D part detectors from category c in grid g. Thus the overal dimensionality of the BoP representation is $D \times G \times C$.

3.2. Coding on Part Responses

To uncover the correlations among responses of different parts, we employ a multi-column coding scheme on top of the BoP codes as shown in Fig. 2. "Multi-column" means that learning/coding for each grid- and category-wise sub-vector $\mathbf{x}_{g,c}$ are performed individually. By this means structural information is preserved and the learning becomes easier. It is notable that coding is performed category-wise because part detectors are learnt within images of a single category, thus the correlations among different sets of part detectors can be ignored. Without loss of generality, we ignore the subscript of $\mathbf{x}_{g,c}$ and let $\mathbf{x} \in \mathbb{R}^D$ denote the responses of D part detectors. Given a set of $\left\{\mathbf{x}^{(i)}, i = 1, 2, ..., M\right\}$ from training images, we train a two-layer autoencoder with K hidden nodes. Let $\mathbf{W}^{(l)}$ denote the weights associated with the connections between layer l and layer $l+1$, and $\mathbf{b}^{(l)}$ denote the biases of the $(l+1)$-th layer. The latent code $\mathbf{s}^{(i)} \in \mathbb{R}^K$ corresponding to $\mathbf{x}^{(i)}$ is defined by

$$\mathbf{s}^{(i)} = \sigma(\mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)}), \tag{2}$$

where $\sigma(\mathbf{z}) = 1/(1 + exp(-\mathbf{z}))$ is the sigmoid function, applied component-wise to the vector $\mathbf{z}$. The reconstruction from the latent code is obtained by another nonlinear mapping

$$h_\theta(\mathbf{x}^{(i)}) = \sigma(\mathbf{W}^{(2)}\mathbf{s}^{(i)} + \mathbf{b}^{(2)}), \tag{3}$$

where $\theta = (\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)})$ are model parameters to be estimated. Obviously, the output layer has the same number of nodes as the input layer.

3.3. Learning Compact Latent Representation

In the learning procedure, sparsity is imposed on the latent code, so as to discover some latent patterns embedded in the part responses. The learning of a sparse autoencoder is to minimize a squared reconstruction error with a sparsity penalty term contrained on the latent code. Formally, the algorithm solves the following problem:

$$\min_\theta \sum_{i=1}^{M} \|\mathbf{x}^{(i)} - h_\theta(\mathbf{x}^{(i)})\|_2^2 + \beta\Omega(\mathbf{s}^{(i)}, \rho) + \lambda\Phi(\theta). \tag{4}$$

The first term in the definition is the square error between input features and their reconstructions. The second term is the spasity penalty $\Omega(\mathbf{s}^{(i)}, \rho) = \sum_{j=1}^{K} \mathrm{KL}(\rho|\hat{\rho}_j)$ based on Kullback-Leibler divergence $\mathrm{KL}(\rho|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$ where $\hat{\rho}_j = \sum_{i=1}^{M} s_j^{(i)}$ is the average activation of the j-th hidden unit and ρ is the target sparsity (typically a small value close to zero, say $\rho = 0.05$). This sparsity term encourages the latent code to maintain an average activation close to the target sparsity ρ. The third term is a regularization term using "entrywise" 2-norm $\Phi(\theta) = \|\mathbf{W}^{(1)}\|_2^2 + \|\mathbf{W}^{(2)}\|_2^2$. β and λ control the relative importance of the three terms.

To solve the problem, backpropagation [9] with L-BFGS[1] algorithm is used to optimize the sparse autoencoder. We note that the learning process is unsupervised and thus the resulting latent features can be used for any tasks not limited to classification. In practice, pre-processing is performed before the unsupervised learning step. Each input vector $\mathbf{x}^{(i)}$ is normalized by subtracting the mean, truncating to ± 3 standard deviation of its elements, and rescaling to the range $[0, 1]$.

Finally, the C-BoP representation of the i-th image is constructed by concatenating the latent codes $\left\{\mathbf{s}_{g,c}^{(i)}\right\}$ across all grids and categories. Thus, the overall dimensionality of the C-BoP representation is $K \times G \times C$, (K/D)-fold reduction from the original BoP representation.

4. EXPERIMENT

We evaluate our method on the classification benchmark in PASCAL VOC 2007 dataset [10]. The dataset contains objects of 20 categories with 5,011 training images (`train` + `val` sets) and 4,952 test images (`test` set). Left-right flipped images are added as additional training set to avoid overfitting. We compare the proposed C-BoP representation with the BoP representation and VOC2007 winner [10]. For the C-BoP representation, we employ clustering to select part detectors. For the BoP representation, we compare the following two alternatives:

- BoP(ER): the BoP representation using Entropy-Rank for parts selection proposed by [3].
- BoP(Clustering): the BoP representation using spectral clustering for parts selection proposed in this paper.

Parameters setting. HOG features are extracted at 4 scales $(2^{-\frac{i}{3}}, i = 0, 1, 2, 3)$, each part is described by a block of 8×8 HOG cells, the image is divided into 1×1 and 2×2 grids, obtaining a total of 5 spatial pooling regions. The BoP and C-BoP codes are fed into linear SVM after l^1 normalization and the χ^2 explicit feature map [11]. All hyper-parameters (β, λ, ρ, and C in SVM) are determined by cross-validation.

[1] We used L-BFGS in minFunc by Mark Schmidt.

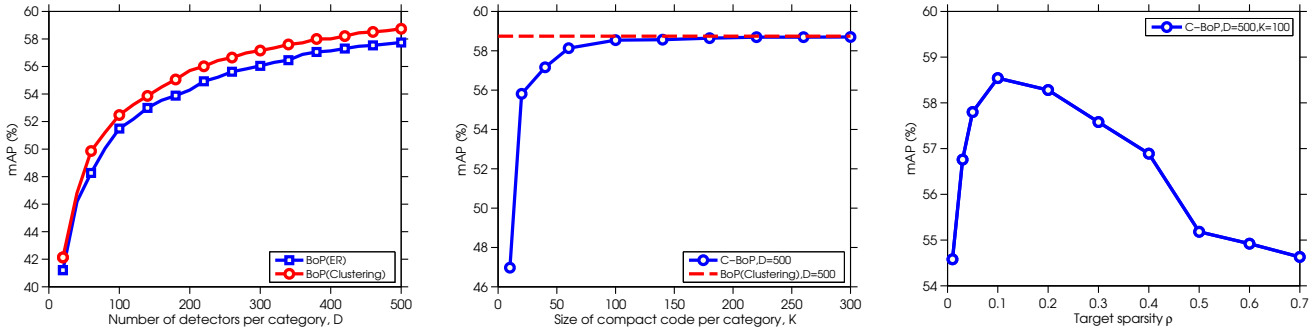

Fig. 3. Performance on PASCAL VOC 2007 dataset. (a) BoP(Clustering) outperforms BoP(ER). (b) Variation of the performance of C-BoP with K increases. It shows that C-BoP obtains a simlar result with BoP by compressing the BoP code into a much more compact representation. (c) Variation of the performance of C-BoP with different values of sparsity.

Table 1. Comparison with the BoP representation and VOC2007 winner.

category	BoP(ER)[3]		winner[10]	C-BoP
	D=100	D=500		D=1000,K=100
aeroplane	67.1	72.3	**77.5**	74.5
bicycle	71.5	73.9	63.6	**74.0**
bird	32.5	42.2	**56.1**	46.4
boat	53.4	61.0	**71.9**	65.0
bottle	22.9	27.9	**33.1**	29.9
bus	55.9	65.2	60.6	**69.3**
car	72.7	76.5	78.0	**79.9**
cat	62.0	**66.7**	58.8	65.9
chair	48.8	52.6	53.5	**55.6**
cow	33.7	41.8	42.6	**43.2**
diningtable	41.3	56.1	54.9	**57.2**
dog	45.4	**48.9**	45.8	48.0
horse	73.5	77.1	77.5	**79.1**
motorbike	62.1	65.8	64.0	**67.7**
person	83.7	85.9	85.9	**86.0**
pottedplant	20.9	25.6	**36.3**	28.4
sheep	35.8	44.6	44.7	**47.5**
sofa	45.6	54.4	50.6	**55.7**
train	62.5	71.0	**79.2**	73.9
tvmonitor	38.2	45.3	**53.2**	51.3
mAP	51.5	57.7	59.4	**59.9**

BoP(Clustering) v.s. BoP(ER). We first compare the BoP representations using two parts selection techniques. As Fig. 3(a) shows, BoP(Clustering) performs better than BoP(ER) consistently with number of parts increases. The mAP saturates at around 500 parts per category with BoP(ER) achieving 57.7% and BoP(Clustering) achieving 58.7% respectively. Thus the proposed method selects parts that are more useful for classification.

Analysis of C-BoP. We analyze the C-BoP representation from two aspects: size and sparsity of the compact code. We learn a compact representation from BoP(clustering) with $D = 500$. Fig. 3(b) shows that C-BoP maintains a performance similar to BoP(Clustering) with around 50-dimensional features per category, approximately 10-fold reduction from the orginal code. Sparsity also plays a vital role in compressing the BoP representation. The performance of C-BoP varies much with different values of target sparsity and high sparsity (smal ρ) is needed to achieve a good result as shown in Fig. 3(c). Thus sparsity leads to a more semantic-lossless reconstruction of part responses.

Final classification results. Finally, we compare the C-BoP representation with the BoP(ER) representation and VOC2007 winner. For the C-BoP representation, we cluster candidate parts into 1000 clusters and learn 100-dimensional latent codes. For the BoP(ER) representation, we use 500 parts per category for the limitation of memory. Besides, we compare with BoP(ER) that uses 100 parts per category because it produces codes with the same dimensionality as the C-BoP codes. As Table 1 shows, C-BoP achieves 59.9%, exceeding BoP(ER) with $D = 100$ by 8.4% and BoP(ER) with $D = 500$ by 2.2%. We owe this superiority to that our compact representation can utilize a larger set of parts than the BoP representation, while the latter is limited by its high dimensionality. This demonstrates the scalability of the C-BoP representation. Besides, the C-BoP representation is competitive to VOC2007 winner by beating it on 13 out of 20 categories, thus the C-BoP representation effectively works for object recognition.

5. CONCLUSION

We have proposed a novel method to learn a semantic-lossless compact representation from high-dimensional BoP features. The method effectively reduces redundancy in tens of thousands parts using spectral clustering and multi-column sparse autoencoders. The proposed representation exhibited superior performances over the original BoP representation in classification owing to its scalability. The compact code is learnt without supervision and hence also applicable for other tasks, which is included in our future work.

6. REFERENCES

[1] D. B. Lubomir and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *IEEE ICCV*, 2009, pp. 1365–1372.

[2] Li-Jia Li, Hao Su, E. P. Xing, and Fei-Fei Li., "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *NIPS*, 2010.

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

[4] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *IEEE ECCV*, 2012, pp. 73–86.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, 2005, pp. 886–893.

[6] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *IEEE ECCV*, 2012, pp. 459–472.

[7] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, December 2007.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE CVPR*, 2006, vol. 2, pp. 2169–2178.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, Winter 1989.

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

[11] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *IEEE CVPR*, 2010, pp. 480–492.