# Deep Bottleneck Feature for Image Classification

Yan Song
National Engineering Laboratory of Speech and Language Information Processing
Univ. of Sci.& Tech of China
Hefei, China
songy@ustc.edu.cn

Ian McloughLin
National Engineering Laboratory of Speech and Language Information Processing
Univ. of Sci.& Tech of China
Hefei, China
ivm@ustc.edu.cn

Lirong Dai
National Engineering Laboratory of Speech and Language Information Processing, USTC
P.O.Box 4
Hefei, China
lrdai@ustc.edu.cn

## ABSTRACT

Effective image representation plays an important role for image classification and retrieval. Bag-of-Features (BoF) is well known as an effective and robust visual representation. However, on large datasets, convolutional neural networks (CNN) tend to perform much better, aided by the availability of large amounts of training data. In this paper, we propose a bag of Deep Bottleneck Features (DBF) for image classification, effectively combining the strengths of a CNN within a BoF framework. The DBF features, obtained from a previously well-trained CNN, form a compact and low-dimensional representation of the original inputs, effective for even small datasets. We will demonstrate that the resulting BoDBF method has a very powerful and discriminative capability that is generalisable to other image classification tasks.

## General Terms

Theory, Machine Learning

## Keywords

Image Classification, Transfer Learning, BoF, CNN

## 1. INTRODUCTION

Effective image representation plays an important role in content based recognition and retrieval applications. Over the past decade, representations based on local features, known as bag-of-features (BoF) methods, were considered to be state-of-the-art, especially for SIFT [11] descriptors extracted from small patches. It has been shown that BoF methods may provide a degree of robustness to the changes caused by image scaling, translation and occlusion.

Recently, deep convolutional neural networks(CNN)[9] have achieved outstanding performance in large scale visual recognition competitions. This may be attributed to the high-capacity of CNN structures, with millions of parameters tuned from large scale dataset like Imagenet [4]. However, for some benchmark image recognition tasks, such as PASCAL VOC [6] and MIT Scenes [14], the performance of CNN is limited due to relative lack of training data. In such cases, a number of recent works [5, 7, 12, 15, 3] have indicated that it is preferable to transfer a previously well-trained CNN rather than to learn one directly using limited training data. For example, Razavian *et.al.* conducted a series of experiments for different recognition tasks, and demonstrated the superiority of the features extracted from a pre-trained CNN [15]. At the same time, Chatfield *et.al.* [3] report comprehensive comparisons between CNN and Improved Fisher Vector (IFV) leading to similar conclusions as [15].

Despite the proven superiority of CNN based features, they have been empirically shown to still be fairly sensitive to global translation, rotation and scaling in terms of classification accuracy [8]. Furthermore, it is argued that the images in various benchmark datasets may have significant different statistics [17]. The transferability of CNN activations decreases as the distance between source dataset and the target one increases [19]. It may therefore be interesting to use a pre-trained CNN for front-end feature extraction (to yield good features), while still exploiting a BoF framework to enjoy more robust image representation. The issue is on how to effectively encode and aggregate the higher dimensional CNN features for diverse tasks which contain limited training data.

In this paper, we thus propose using a structured CNN containing a bottleneck layer as the front-end feature extractor. As shown in Fig. 1, the number of hidden nodes in the bottleneck layer (*i.e.* FC7) is much smaller than other fully-connected layers. The CNN training will then force the activations in the bottleneck layer to form a low-dimensional compact representation of the original input. We denote the output vector of that constricted internal layer as the Deep Bottleneck Feature (DBF). Then, a bag-of-DBFs (BoDBF) method is proposed in which a second-order pooling scheme is applied on top of the DBFs. To evaluate the effectiveness of the proposed BoDBF method, we conduct extensive experiments on PASCAL VOC [6] and MIT Scenes [14]. It is shown that the proposed BoDBF can achieve state-of-the-art performance on these image classification problems, especially for the MIT Scenes dataset.

Section 2 will introduce a BoDBF framework for image classification. A general BoF pipeline is first briefly reviewed, and then a CNN with a bottleneck layer is detailed

*ICMR'15,* June 23–26, 2015, Shanghai, China.
Copyright © 2015 ACM 978-1-4503-3274-3/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2671188.2749314.

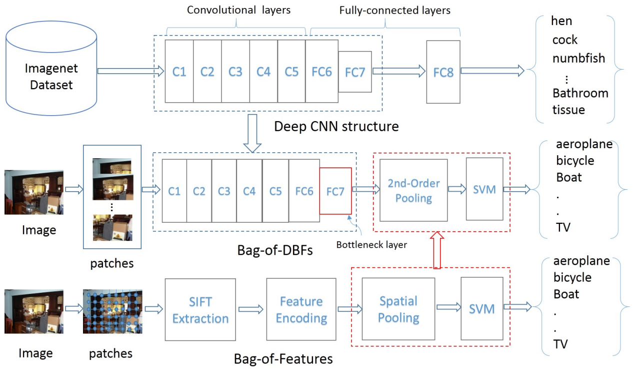

**Figure 1: The BoDBF classification framework.**

as the front-end feature extractor. After that, a second-order pooling scheme [2] is introduced for constructing robust image representation. Section 3 will evaluate the technique by reporting experimental classification results for PASCAL VOC 2007 and MIT Scene datasets. Section 4 presents conclusions and discusses the potential for future extension of this research.

# 2. DBF BASED CLASSIFICATION

The DBF based image classification framework is shown in the middle of Fig 1, which follows the traditional BoF pipeline. Generally speaking, BoF methods map the local features (*i.e.* SIFT) into a fixed-length histogram. The process mainly consists of two main phases: (i) *Feature Encoding* assigns every local feature to the nearest visual words in a dictionary. The dictionary would generally have been obtained off-line through a clustering process on a large local feature set. (ii) *Spatial pooling* counts occurrences of visual words in the image (or in spatial regions) to form a histogram representation.

From the perspective of convolution, there exists an intrinsic link between SIFT features and CNN activations. SIFT may be described as forming histograms (pooling results) of oriented edge filter responses, arranged in spatial blocks. Meanwhile, CNN activations are obtained by passing a raw image through multiple layers, each of which contain convolution and pooling operations, outputting a single high-dimensional (*i.e.* 4096) vector from subsequent fully-connected layers. In each layer, the input data is convolved with learned filters and then non-linearly transformed to output activations, and often a spatial pooling operation is applied before output. Some major differences between CNN activations and traditional SIFT features are:

**large/small:** SIFT features are extracted from either dense grids or detected Regions of Interest (ROIs), whereas CNN activations can preserve more spatial information over a larger patch size.

**deep/shallow:** From the perspective of CNN, SIFT features are the output of a shallow structure, whereas CNN activations can be derived from deeper layers, providing a compact mid-level representation of the original image. Zeiler *et.al.* showed that the reconstruction of the activations from deeper convolutional layers resembles the original image [20].

**learned/fixed:** The edge filters used for extracting SIFT features are generally hand-crafted, while in CNN, the filters are discriminatively learned from training data.

Due to the superior performance of CNN on large-scale visual recognition, we are interested in using the activations from deep CNN structure instead of SIFT features for effective image representation. However, these activations are generally with higher dimension, which is difficult to be modeled using traditional unsupervised learning methods. We thus introduce a special CNN structure with bottleneck layer as a front-end feature extractor.

## 2.1 CNN structure with Bottleneck Layer

The structure of CNN with bottleneck layer is shown on the top of Fig. 1. This is an 8-layer Zeiler and Fergus's (ZF) style CNN [20], consisting of five convolutional layers (C1-C5) and three fully-connected layers (FC6-FC8). The input image size is 224 × 224. The filter numbers (sizes) of the five convolutional layers are: $96(7 \times 7)$, $256(5 \times 5)$, $512(3 \times 3)$, $512(3 \times 3)$ and $512(3 \times 3)$ respectively. The first two convolutional layers have a stride of 2 pixels, and the rest have a stride of 1 pixel.

In our current implementation, the fully-connected layer FC7 is chosen as the bottleneck layer. This deep CNN is trained on the Imagenet dataset and has performance comparable to AlexNet [9] on the validation set with moderate training time.

In the BoDBF framework, we use this deep CNN structure following the BoF pipeline: the output of convolutional layers can be considered as the patch descriptor, and the bottleneck layer functions likes an encoder to produce compact codes. These codes are further pooled together to produce a global description of the image which is robust to slight transformations.

## 2.2 Second-order pooling of DBFs

Given a collection of $m$ patch codes $D = (\mathbf{X}, \mathbf{F})$, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m), \mathbf{x}_i \in \mathbb{R}^n$ are the DBFs extracted from patches centered at positions $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m), \mathbf{f}_i \in \mathbb{R}^2$.

The simplest pooling method consists in averaging the feature vectors within a spatial neighborhood, which we term first-order pooling (O1P): $\mathbf{P}_1 = \frac{1}{m}\sum_{i=1}^{m} |\mathbf{x}_i|$

In this paper, we introduce a more effective second-order method that can capture the pairwise correlations between DBFs extracted from an image. These correlations can be defined using an outer product of DBFs, which is symmetric positive definite (SPD). The second-order pooling (O2P) is

$$\mathbf{P}_2 = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i \cdot \mathbf{x}_i^\top \qquad (1)$$

From the perspective of computational differential geometry [1], the set of symmetric positive definite matrices forms a Riemannian manifold, a non-Euclidean space. Note that linear SVM generally follows pooling, for reasons of efficiency. However in this case it is not optimal for a linear SVM to be trained from second-order pooling outputs. To address this issue, a log-Euclidean metric is proposed to map the SPD matrix $\mathbf{P}_2$ to a tangent space, $\mathbf{P}_2 = log(\mathbf{P}_2)$. It is proven with strong theoretic guarantees that this type of map can preserve the intrinsic geometric relationships as defined in the original Riemannian manifold [1]. Furthermore, the signed-square-root and $l_2$-normalization, corresponding

**Table 1: Comparison of results on Pascal VOC 2007 dataset of each 20 classes in terms of mAP(%)**

| method/object class | aero | bicyc | bird | boat | bott | bus | car | cat | chair | cow |
|---|---|---|---|---|---|---|---|---|---|---|
| CNN-SVM [15] | 89.7 | 81.4 | 86.0 | 85.3 | 45.7 | 76.2 | 84.6 | 84.9 | 64.5 | 62.5 |
| SCFVC [10] | 89.5 | 84.1 | 83.7 | 83.7 | 43.9 | 76.7 | **87.8** | 82.5 | 60.6 | 69.6 |
| DBF-O2P | **91.7** | **86.0** | **88.8** | **88.0** | **55.0** | **80.5** | 86.4 | **88.4** | **65.3** | **73.2** |

| method/object class | table | dog | horse | m-bike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|
| CNN-SVM [15] | 70.0 | 80.3 | 83.0 | 78.0 | 87.7 | 50.1 | 75.8 | 59.8 | 88.8 | 72.5 |
| SCFVC [10] | 72.0 | 77.1 | **88.7** | 82.1 | **94.4** | 56.8 | 71.4 | **67.7** | 90.9 | 75.0 |
| BoDBF-O2P | **74.7** | **85.4** | 88.4 | **83.7** | 91.4 | **60.5** | **81.2** | 67.4 | **91.4** | **78.5** |

**Table 2: Comparison of results on Pascal VOC 2007. The mean average precision over 20 classes in terms of mAP(%)**

| Method | MAP | Comment |
|---|---|---|
| SCFVC [10] | 76.9 | with single scale |
| CNN-SVM [15] | 73.9 | on whole image |
| CNNaug-SVM [15] | 77.2 | with augmented data |
| CNN-SVM [3] | 78.6 | with augmented data |
| BoDBF-O1P | 75.4 | single scale patches |
| BoDBF-IFV | 76.1 | CNN for single scale patches |
| BoDBF-O2P | **80.6** | CNN for single scale patches |

to the Hellinger kernel map [18] is also applied before classification.

# 3. EXPERIMENT AND ANALYSIS

To evaluate the effectiveness of the proposed bag-of-DBFs method, we conduct extensive experiments on PASCAL VOC 2007 and MIT Scenes datasets.

## 3.1 Experimental Setting

The DBFs are extracted as follows: firstly the input images are resized so that their maximize dimension is 512 and their minimum dimension is at least 224. Next, patches of size $224 \times 224$ pixels are cropped from each image with a stride of 8 pixels. The patches are then fed into a pre-trained deep CNN, and the activations from the bottleneck layer (*i.e.* FC7) are used as DBFs. In our implementation, we exploit several deep CNN structures trained by [3] with FC7 dimensions of 128, 1024 and 2048. Use of these publicly available pre-trained structures is important in ensuring that our approach can be compared fairly with other methods. We implement the following alternative systems for comparison: 1. BoDBF-IFV: using the IFV method based on DBFs [16, 13]. 2. BoDBF-O1P: using first-order pooling, *i.e.* average pooling based on DBFs. 3. BoDBF-O2P: using second-order pooling based on DBFs. In addition, we also compare with several reported state-of-the-art methods with similar settings [8, 10].

## 3.2 Experiments on PASCAL VOC 2007

The PASCAL-VOC 2007 dataset [6] consists of 9963 images from 20 classes. These images include indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. The dataset is divided into three parts: (i) a training set of 2501 images, (ii) a validation set of 2510 images and (iii) a test set comprising 4952 images. Results are shown in Table 2. It is clear that the proposed BoDBF-O2P method outperforms our own BoDBF-01P and BoDBF-IFV by $5\% - 6\%$. Furthermore, we compare the results with other state-of-the-art CNN systems, CNN-SVM1 [15], SCFVC [10] and CNN-SVM2 [9]. CNN-SVMs using the same CNN structure can achieve 78.6%, which is the closest to the performance of our BoDBF-O2P system. SCFVC [10] uses sparse coding techniques for Fisher vector encoding, outperforming the BoDBF-O1P and BoDBF-IFV. However, the computational complexity of SCFVC may become expensive with high dimensional CNN features. Table.1 shows the detailed comparisons for the 20 categories, demonstrating that BoDBF-O2P achieves best performance in 16 categories, as well as being competitive in the remaining 4 categories.

## 3.3 Experiments on MIT Scenes dataset

The MIT scenes dataset contains 6700 images over 67 indoor scene categories. For each category, the standard training/test split consists of 80 training and 20 test images. This dataset is quite challenging due to the subtle cross-category difference. Moreover, the contents of this dataset are more different from Imagenet than PASCAL VOC 2007 is, which may help to evaluate the generalization capability of our proposed BoDBF.

Classification results are compared in Table 3. Again, the proposed BoDBF-O2P method significantly outperforms BoDBF-IFV and BoDBF-O1P. We notice that the CNN-SVM method [15, 3] perform quite well, and are closer to BoDBF on the PASCAL VOC dataset than they are on MIT Scenes. To the best our knowledge, the state-of-art performance on the MIT Scenes dataset is achieved by MOP-CNN [8], which concatenates the CNN activations using three different scales. SCFVC [10] can achieve similar performance. Compared to them, the performance of the proposed BoDBF-O2P is as high as 72.8%, which is by far the best on this dataset.

## 3.4 Discussion

We have evaluated the performance of the proposed BoDBF-O2P on two benchmark object datasets, i.e. PASCAL VOC and MIT Scenes. Results clearly show that the BoDBF framework can take advantage of both the robustness of BoF and the improved representation power of a deep CNN. Unlike existing methods including [8, 10], we use a fully-connected bottleneck layer as the feature encoder. Second-order pooling capture the fact that activations from the deep CNN layer are not Euclidean distributed.

Furthermore, we evaluated the performances BoDBF-O2P using activations from CNN structures with different bottleneck sizes of 128, 1024 and 2048. For 1024 and 2048 dimension DBFs, principal component analysis (PCA) was applied to reduce them to the same 128 dimensions. All

**Table 3: MIT Scenes performance reporting precision averaged over all 67 classes in terms of mAP(%)**

| Method | MAP | Comments |
|---|---|---|
| SCFVC [10] | 68.2 | with single scale |
| CNN-SVM [15] | 58.4 | CNN feature for whole image |
| MOP-CNN [8] | 68.9 | for patches with three scales |
| BoDBF-O1P | 59.8 | for patches with single scale |
| BoDBF-IFV | 66.2 | for single scale patches |
| BoDBF-O2P | **72.8** | for single scale patches |

results reported above used 1024 dimension DBFs reduced to 128 dimension using PCA. Thus the final dimension of BoDBF-O2P was $128 \times 129/2 = 8256$, which is less than MOP-CNN ($4096 \times 3 = 12288$) and SCFVC ($100 \times 1000 = 100,000$). Detailed comparisons for all dimension DBFs are shown in Table. 4, which indicates that all yield competitive performance and larger sizes may be unnecessary.

**Table 4: The effect of different layer size in terms of mAP (%)**

| Dataset/Dimension | 128 | 1024 | 2048 |
|---|---|---|---|
| PASCAL VOC | 80.3 | 80.6 | 79.1 |
| MIT 67 Scenes | 69.9 | 72.8 | 70.0 |

## 4. CONCLUSION AND FUTURE WORK

This paper has presented a novel image classification framework (*i.e.* BoDBF) based on a carefully designed CNN structure that contains a narrow bottleneck layer. The CNN, having been trained on a large image dataset, is used as the front-end feature extractor for other visual recognition tasks. In the proposed BoDBF framework, the output of convolutional layers are considered as the patch descriptors, and the bottleneck layer functions as an encoder to produce compact codes, *i.e.* DBFs. Second-order pooling scheme is further introduced to explore the correlations between DBFs. The BoDBF can take advantage of both the robustness of BoF and the improved representation power of a deep CNN, performance is excellent. In fact the experimental results, on benchmark datasets such as Pascal VOC 2007 and the MIT 67 Scenes dataset, reveal that performance exceeds that of current state-of-the-art approaches. The current results are for a relatively untuned network. In future, we would like to experiment with different CNN structures, including assessing the effects of the dimension and location of the bottleneck layer (e.g. moving it to FC6 or FC8).

## 5. REFERENCES

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Analysis Applications*, 29(1):328–347, 2006.

[2] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Proc. of ECCV 2012*, pages 430–443, 2012.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *Proc. of ECCV*, 2014.

[4] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*, pages 248–255, 2009.

[5] J. Donahue, Y. Jia, and O. Vinyals. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. of ICML*, pages 647–655, 2014.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[7] R. B. Girshick and J. Donahue. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of CVPR*, pages 580–587, 2014.

[8] Y. Gong, L. Wang, and R. Guo. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. of ECCV*, pages 392–407, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of NIPs*, pages 1106–1114, 2012.

[10] L. Liu, C. Shen, L. Wang, A. van den Hengel, and C. Wang. Encoding high dimensional local features by sparse coding based fisher vectors. In *Advances in Neural Information Processing Systems 27, 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1143–1151, 2014.

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

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. of CVPR*, pages 1717–1724, 2014.

[13] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. of ECCV*, pages 143–156, 2010.

[14] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. of CVPR*, pages 413–420, 2009.

[15] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[16] J. Sanchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[17] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proc. of CVPR*, pages 1521–1528, 2011.

[18] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):480–492, 2012.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proc. of NIPs*, pages 3320–3328, 2014.

[20] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. of ECCV*, pages 818–833, 2014.