Multi-layer Feature Extractions for Image Classification – Knowledge from Deep CNNs –

Kazuya Ueki, Tetsunori Kobayashi
Faculty of Science and Engineering, Waseda University,
Room 40-701, Waseda-machi 27, Shinjuku-ku, Tokyo, 162–0042 Japan
ueki@pcl.cs.waseda.ac.jp

***Abstract*—Recently, there has been considerable research into the application of deep learning to image recognition. Notably, deep convolutional neural networks (CNNs) have achieved excellent performance in a number of image classification tasks, compared with conventional methods based on techniques such as Bag-of-Features (BoF) using local descriptors. In this paper, to cultivate a better understanding of the structure of CNN, we focus on the characteristics of deep CNNs, and adapt them to SIFT+BoF-based methods to improve the classification accuracy. We introduce the multi-layer structure of CNNs into the classification pipeline of the BoF framework, and conduct experiments to confirm the effectiveness of this approach using a fine-grained visual categorization dataset. The results show that the average classification rate is improved from 52.4% to 69.8%.**

Keywords—Deep learning; Feature extraction; Bag-of-Features; Generic object recognition; Fine-grained visual categorization

I. Introduction

Recently, deep learning has received increased attention in the fields of image recognition, automatic speech recognition, and natural language processing. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) benchmark [1], the deep convolutional neural network (CNN) demonstrated superior performance to extensions of conventional methods using local descriptors and Bag-of-Features (BoF) [2]. This result introduced many researchers to the benefits of deep neural networks. The network used in ILSVRC 2012 has a deep structure and consists of eight trainable layers. The structure includes five convolution layers, some of which are followed by max-pooling layers, and three fully connected layers [3].

For some time, deep CNN has been used to recognize faces [4] and handwriting [5]. However, the difficulty of training deep multi-layer neural networks, problems with overfitting, and excessive computation times saw neural network research enter a hiatus in the 1990s. Recently developed techniques, such as pre-training [6], [7] and dropout [8], can deal with the above-mentioned problems, and have been employed to successfully train networks without overfitting. Thus, researchers have once again begun to pay attention to deep CNNs.

Simultaneously, improvements in recent computer hardware and GPU implementations have accelerated the training of deep networks, making it relatively easy to handle large numbers of training samples (for example, there were more than one million training images in the ILSVRC benchmark). Moreover, the new rectified linear unit (ReLU) activation function [3] has led to faster training convergence and high classification performance. Using these new techniques, deep CNN has demonstrated overwhelmingly positive performance compared with several recently proposed methods. However, the basic network structure is almost the same as those used in the 1990s, and various factors, such as how deep CNN works and how many parameters should be used, remain to be clarified.

Recently, there are several new approaches and techniques to understand the operation of CNN. One of the novel techniques is feature visualization proposed in [9]. They also show how those visualizations can be used to find problems in the model and obtain the better classification. Another network architecture called ”Network In Network” (NIN) [10] also demonstrated the state-of-the-art performance. This result indicates that a non-linear model is more suitable: because the data for the same class often lies on a non-linear manifold, therefore the representations that capture these classes are generally non-linear function of the input. The recent trend has been to use deeper networks such as Visual Geometry Group’s CNN architectures (16 and 19 layers) [11] and GoogLeNet (22 layers) [12]. However there is no clear understanding of why the performance of very deep networks is so well, or how many layers should be used.

In this paper, to validate deep CNN’s excellent characteristics, we introduce deep CNN’s distinctive characteristics to mainstream solutions for conventional object recognition, namely, a combination of the scale invariant feature transform (SIFT) feature extractor [13], [14], [15] and BoF encoding [2].

This paper is organized as follows. Section II describes our proposed method. We present experimental results that demonstrate the effectiveness of our method in Section III, and give our conclusions and suggestions for future research in Section IV.

II. Proposed method

A. The difference between SIFT+BoF and deep CNN-based methods

The typical classification pipeline for object recognition proceeds as follows:

1) Extraction of a large number of SIFT features
2) Unsupervised dimensionality reduction by principal component analysis (PCA)
3) Encoding by BoF, Fisher vector [16], or similar.
4) Classification by support vector machine, logistic regression, or similar.

978-1-4673-8353-0/15/$31.00 ©2015 IEEE

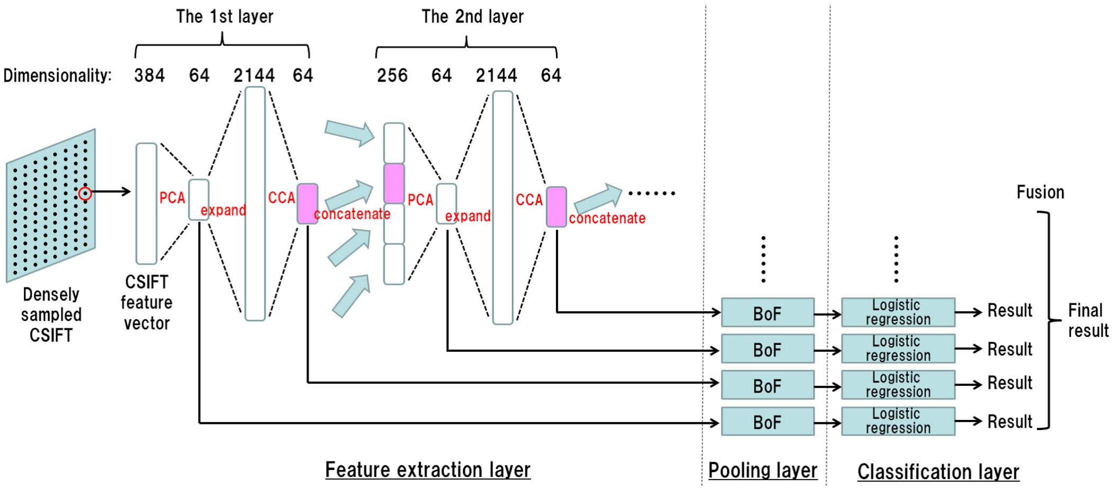

Fig. 1. Proposed method: multi-layer discriminative feature extraction.

The main differences between SIFT+BoF and deep CNN-based methods are as follows. First, the dimensionality reduction and pooling in the classification pipeline of SIFT+BoF are fully unsupervised, whereas deep CNNs can automatically obtain discriminative local filters trained with class labels using the back-propagation algorithm. Second, deep CNNs have a multi-layer structure that can gradually increase the size of local filters by repeatedly combining neighboring features and finally obtaining more discriminative features.

In this paper, we implement the effective characteristics of deep CNNs in the SIFT+BoF classification pipeline, and confirm the effectiveness of this approach by comparing its performance with that of conventional methods.

B. Introducing the multi-layer structure to SIFT+BoF

The outline of our proposed method is shown in Figure 1. Out method is based on the BoF approach, with two modifications: discriminative feature extraction and a multi-layer structure.

Here, we explain the first modification: changing the early stage of feature extraction in the SIFT+BoF classification pipeline to a supervised technique using class labels. After extracting local features (such as SIFT features), we reduce the dimension using PCA, as in the original method. Next, each vector's dimension is expanded by computing the product of all vector elements, and then reduced again by a supervised dimensionality reduction. Let $\boldsymbol{v}_{\mathrm{PCA}}$ be the d-dimensional local feature compressed by PCA. The feature expansion can be described as follows:

$$\boldsymbol{v}_{\mathrm{Ex}} = \begin{pmatrix} \boldsymbol{v}_{\mathrm{PCA}} \\ upperVec(\boldsymbol{v}_{\mathrm{PCA}} \boldsymbol{v}_{\mathrm{PCA}}^T) \end{pmatrix}, \tag{1}$$

where $upperVec(\boldsymbol{v}_{\mathrm{PCA}} \boldsymbol{v}_{\mathrm{PCA}}^T)$ is the flattened vector of components in the upper-triangular part of a symmetric matrix. For example, when a 64-dimensional feature compressed by PCA is expanded, the dimensionality of $\boldsymbol{v}_{\mathrm{Ex}}$ becomes $64 + (1 + 2 + \cdots + 64) = 2,144$.

For the supervised dimensionality reduction, we use canonical correlation analysis (CCA). CCA finds the linear combination with the maximum correlation between two sets of multi-dimensional variables. First, we prepare the sets of feature vectors and label vectors, $\left\{\left(\boldsymbol{v}_{\mathrm{Ex}}^{(i)}, \boldsymbol{l}^{(i)}\right)\right\}_{i=1}^{n}$, where n is the number of training samples, $\boldsymbol{v}_{\mathrm{Ex}}^{(i)}$ is the expanded feature from the training set, and $\boldsymbol{l}^{(i)}$ is a label vector whose k-th element is 1 if the image belongs to category k, and 0 otherwise. We can find the linear projections $\boldsymbol{s} = \boldsymbol{A}^T \boldsymbol{v}_{\mathrm{Ex}}$ and $\boldsymbol{t} = \boldsymbol{B}^T \boldsymbol{l}$, and then project multiple views into a common canonical space where their correlation is maximized. The CCA transformation matrix can be computed by solving the eigenvalue problem [17]. Finally, a low-dimensional discriminative vector can be obtained by mapping $\boldsymbol{v}_{\mathrm{Ex}}$ using the CCA matrix $\boldsymbol{A}$.

We consider this feature extraction process as the first layer of our multi-layer network, and term each individual procedure as *the first PCA layer*, *first expansion layer*, and *first CCA layer*.

After applying CCA to reduce the dimensionality, the local region is enlarged by combining neighborhood features. Specifically, one vector can be obtained by simply concatenating four adjacent local features. In the case of 64-dimensional vectors compressed by CCA, we obtain 256-dimensional (= 64×4) vectors.

Thereafter, by repeating the concatenation, PCA, expansion, and CCA process, a larger region is gradually considered. The second iteration is termed *the second PCA layer*, *second expansion layer*, and *second CCA layer*. Hereafter, the N-th iteration is called *the N-th PCA layer*, *N-th expansion layer*, and *N-th CCA layer*.

III. Experiments

A. Dataset

To validate the effectiveness of our proposed method, we conducted experiments using the Oxford Flowers 102 dataset [18], which contains 8,189 images divided into 102 flower classes. Twenty samples per class were used for training (giving a total of 2,040 (= 20×102) training images), and the others were used for testing. First, we cropped all images

TABLE I. AVERAGE CLASSIFICATION RATES WITH THE OXFORD FLOWERS 102 DATASET. LEADING APPROACH AMONG SINGLE CLASSIFIERS IS SHOWN IN BOLD. THE UNDERLINED NUMBER INDICATES THE BEST PERFORMANCE IN ALL THE EXPERIMENTS.

Single/Fusion	Layer used for classification								Average classification rates
	1st layer		2nd layer		3rd layer		4th layer		
	PCA	CCA	PCA	CCA	PCA	CCA	PCA	CCA	
Single classifiers	✓								52.4%
		✓							63.1%
			✓						64.4%
				✓					**67.1%**
					✓				66.9%
						✓			66.3%
							✓		65.3%
								✓	65.1%
Multiple classifier fusion		✓		✓					68.5%
		✓		✓		✓			69.1%
		✓		✓		✓		✓	69.4%
		✓	✓	✓					68.8%
		✓	✓	✓	✓	✓			**69.8%**
		✓	✓	✓	✓	✓	✓	✓	69.6%
	✓	✓	✓	✓	✓	✓	✓	✓	69.6%

using the bounding box provided, and then normalized the size of these images to have 150 pixels on their shortest side.

B. Experimental setup

We extracted local descriptors from the size-normalized images. We presumed that the color information would boost the classification performance of flower categorization. Thus, in all the experiments, we used colored scale invariant feature transform (CSIFT) features [19]. These were densely extracted using regular grids with a spacing of five pixels. We did not extract features from multiple scales, but only from one scale. As a result, an average of less than 1,000 local CSIFT descriptors were extracted from each image.

After randomly extracting 1,000,000 CSIFT features from the training images, we computed a PCA transformation matrix using these 1,000,000 feature vectors to reduce the dimensionality. All of the descriptors were reduced from 384 to 64 dimensions by this PCA matrix.

For the encoding, we chose the most basic BoF method, and used the same 1,000,000 features to create the visual dictionary. The codebook size was fixed to $k = 5,000$, because this value produced relatively good performance in a preliminary experiment.

In terms of classifiers, after considering the methods to combine outputs from multiple classifiers, we decided to use logistic regression in the LIBLINEAR package [20] to output probabilities for each class.

As the number of images in the classes was unbalanced, we used the average classification rate to assess the effectiveness of our method.

C. Experimental results and discussion

The results of our experiments using the Oxford Flowers 102 dataset are summarized in Table I. The simplest method, using features from the first PCA layer, produced an average classification rate of 52.4%. In contrast, the features from the first CCA layer, after dimensionality expansion and supervised dimensionality reduction, resulted in significantly improved performance, with a classification rate of 63.1%.

Using features from the second PCA layer, i.e., after concatenating four adjacent feature vectors, further improved the performance to 64.4%, and those from the second CCA layer achieved a classification rate of 67.1%.

Features from deeper layers resulted in slightly poorer performance (e.g., those from the third PCA and CCA layers gave classification rates of 66.9% and 66.3%, respectively, and features from the fourth PCA and CCA layers produced slightly lower accuracy again).

A small region is treated in the layers close to the input, and a relatively large region is taken into account in the layers close to the output. Therefore, it is very likely that we can obtain different types of features from each layer. We expect that further improvement could easily be achieved by fusing multiple classifiers. For this reason, we combined the results obtained by multiple classifiers, as shown in the pooling and classification layer on the right of Figure 1. Finally, because we used logistic regression, we can obtain combined probability outputs by taking the average of each classifier's probability. The classification results of this fusion process are given in the lower half of Table I.

From this table, we can see that combining new discriminative CCA layers improves the classification performance: from 63.1% to 68.5%, 69.1%, and 69.4%. Moreover, when the outputs from PCA layers are also integrated, the performance is much more stable. Finally, the optimum performance was achieved by combining the output from five layers (first CCA layer, second PCA layer, second CCA layer, third PCA layer, and third CCA layer), which resulted in a classification accuracy of 69.8%. Figure 2 shows the difference between the baseline and the optimum performance. We can see that the classification performance is improved in most of the classes.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have described how the superior performance of deep CNNs could be adapted to SIFT+BoF-based approaches to improve the results of image classification.

In deep CNNs, large numbers of training samples are generally needed to estimate the network weights, but one of the advantages of our method is that it can achieve excellent

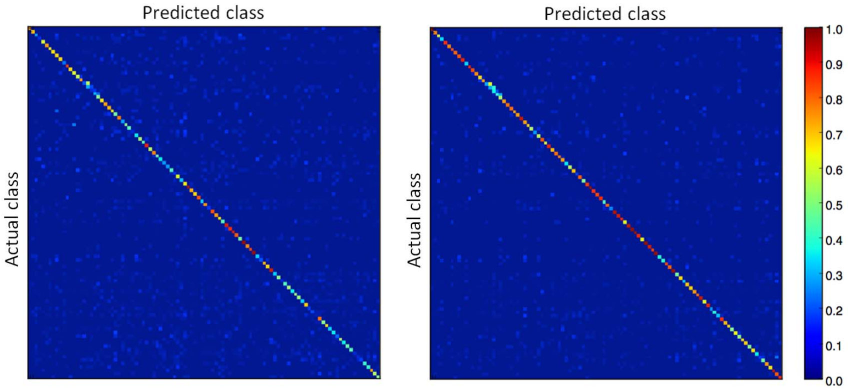

Fig. 2. Confusion matrices of multi-class classification by different methods: Left: using features only from the first PCA layer; Right: using features from five layers (first CCA layer, second PCA layer, second CCA layer, third PCA layer, and third CCA layer).

performance with a relatively small number of training samples.

Our approach is also very versatile and flexible. For instance, it is not only applicable to SIFT or CSIFT features, but also HOG (Histogram of Oriented Gradients) [21] and LBP (Local Binary Patterns) [22] features, and can utilize other pooling methods such as LLC (Locality-constrained Linear Coding) [23], super vector coding [24], or Fisher vector encoding [16].

In future research, we aim to make use of the multiple filters used in deep CNN, and to add a pooling function (max pooling or average pooling) to improve the robustness of our approach against distortions and geometric transformations.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 15K00249.

References

[1] "ImageNet Large Scale Visual Recognition Challenge 2012," http://www.image-net.org/challenges/LSVRC/2012/.

[2] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Proceedings of ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1106–1114, 2012.

[4] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] S. O. G. E. Hinton and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv e-prints*, 2012.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[10] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[13] D. G. Lowe, "Object recognition from local scale invariant features," in *Proceedings of IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157.

[14] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[17] F. P. A. Gordo, J. Rodriguez and E. Valveny, "Leveraging category-level labels for instance-level image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3045–3052.

[18] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of Indian Conf. Computer Vision Graphics and Image Processing*, 2008, pp. 722–729.

[19] A. E. Abdel-Hakim and A. A. Farag, "Sift descriptor with color invariant characteristics," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1978–1983.

[20] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[22] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of the IAPR International Conference*, vol. 1, 1994, pp. 582–585.

[23] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.

[24] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 141–154.