COMPACT CONVOLUTIONAL NEURAL NETWORK TRANSFER LEARNING FOR SMALL-SCALE IMAGE CLASSIFICATION

Zengxi Li[⋆] *Yan Song*[⋆] *Ian Mcloughlin*[†] *Lirong Dai*[⋆]

[⋆] National Engineering Laboratory of Speech and Language Information Processing, USTC
[†] School of Computing, University of Kent

ABSTRACT

Transfer learning methods have demonstrated state-of-the-art performance on various small-scale image classification tasks. This is generally achieved by exploiting the information from an ImageNet convolution neural network (ImageNet CNN). However, the transferred CNN model is generally with high computational complexity and storage requirement. It raises the issue for real-world applications, especially for some portable devices like phones and tablets without high-performance GPUs. Several approximation methods have been proposed to reduce the complexity by reconstructing the linear or non-linear filters (responses) in convolutional layers with a series of small ones.

In this paper, we present a compact CNN transfer learning method for small-scale image classification. Specifically, it can be decomposed into fine-tuning and joint learning stages. In fine-tuning stage, a high-performance target CNN is trained by transferring information from the ImageNet CNN. In joint learning stage, a compact target CNN is optimized based on ground-truth labels, jointly with the predictions of the high-performance target CNN. The experimental results on CIFAR-10 and MIT Indoor Scene demonstrate the effectiveness and efficiency of our proposed method.

Index Terms— CNN, Transfer Learning, Image Classification

1. INTRODUCTION

Recently, deep convolutional neural networks (CNN) have achieved outstanding performance in large scale visual recognition competitions [1]. Generally, the deep CNN structure can be decomposed into (1) convolutional layers, which perform non-linear feature extraction via convolution, rectified linear units (ReLU), and max-pooling operations, and (2) fully connected layers, which map the extracted features into posterior probabilities. It is known that the powerful modeling capability of deep CNN mainly comes from its complex structure with millions of parameters tuned with large-scale labeled dataset like ImageNet [2].

However, for small-scale datasets, *e.g.* MIT Indoor Scene [3], the complexly structured CNN may be prone to over-fitting, leading to reduced performance. In such cases, several recent works indicate that it is preferable to transfer a previous well-trained CNN rather than to train a new CNN with limited labeled data. For example, Razavian *et.al.* conducted a series of experiments for various recognition tasks using CNN features as generic image representation [4]. Chatfield *et.al.* compared the results of using CNNs with various structures, *e.g.* CNN-F, CNN-M and CNN-S [5]. In [6], Girshick *et.al* showed that *CNN fine-tuning* scheme can yield a significant performance boost. In [7], the transferability of features from different layers has been comprehensively evaluated. The effectiveness of CNN fine-tuning schemes has been validated on similar tasks.

Despite the superior performance of transferred CNNs, the high computational complexity and storage requirement make it difficult to apply them in real-world systems, especially for some portable devices, such as mobile phones and tablets without high-performance GPUs. So, it is of practical importance to improve CNN efficiency without reducing performance. Several approximation methods were developed to reconstruct linear filters or responses with a series of smaller ones [8, 9]. In [10], Zhang *et.al* proposed to minimize the reconstruction error of non-linear responses, which is subject to a low-rank constraint. These methods mostly focus on the convolutional layers of CNNs.

In this paper, we propose a compact transfer learning scheme for small-scale recognition tasks, as shown in Fig 1. Given a pre-trained CNN for source task (*i.e.* ImageNet), the transferring process can be decomposed into fine-tuning and joint learning stages. In the fine-tuning stage, a high-performance CNN model on the target dataset, such as MIT Indoor Scene and CIFAR-10, is fine-tuned by transferring the parameters of internal layers from a pre-trained CNN. In the joint learning stage, a compact CNN model that satisfies the complexity and storage requirement is firstly designed, and then optimized with an objective function which exploits the information lying in the output probabilities from the high-performance CNN. This may enforce the compact

We acknowledge the support of the following organizations for research funding: National Nature Science Foundation of China (Grant No. 61273264 and No. 61172158), Science and Technology Department of Anhui Province (Grant No. 15CZZ02007), Chinese Academy of Sciences (Grant No. XDB02070006).

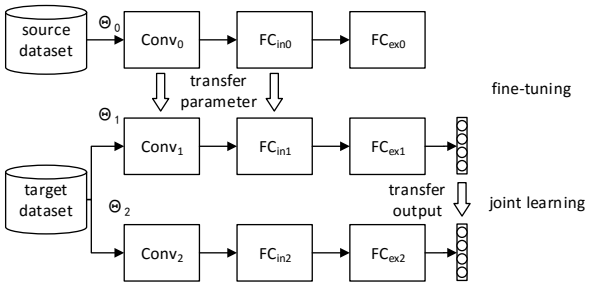

Fig. 1. **The illustration of our CNN transfer learning method.** (i) **fine-tuning stage.** A target network Θ_1 is configured and optimized by transferring all internal layers from source network Θ_0. This network is then fine-tuned on the small-scale target dataset. (ii) **joint learning stage.** A compact target network Θ_2 is optimized with an objective function which exploits the information lying in the output probabilities from network Θ_1.

CNN model to make the prediction similar as that of the high-performance CNN. To evaluate the effectiveness of the compact CNN transfer learning method, we conducted extensive experiments on CIFAR-10 and MIT Indoor Scene [11], achieving competitive image classification performance with low complexity model.

2. METHOD

This section introduces our compact CNN transfer learning method, which consists of fine-tuning and joint learning stages as shown in Fig. 1. We assume that there is an L layer CNN pre-trained on large-scale ImageNet (ImageNet CNN), which is denoted as $\Theta_0 = [\theta_0^1, \ldots, \theta_0^l, \ldots, \theta_0^L]$, where θ_0^l denotes the parameters of l-th layer.

2.1. Fine-tuning

In the fine-tuning stage, our goal is to adapt the pre-trained ImageNet CNN Θ_0 to the target dataset $X = \{\mathbf{x}_n\}_{n=1}^N$ with labels $Y = \{\mathbf{y}_n\}_{n=1}^N$. Let us define the target network as $\Theta_1 = [\theta_1^1, \ldots, \theta_1^l, \ldots, \theta_1^L]$. The internal parameters $\theta_1^1, \theta_1^2, \ldots, \theta_1^{L-1}$ are initialized by transferring parameters from Θ_0, while the output layer θ_1^L is randomly initialized. The model parameters Θ_1 are optimized according to the loss function $\mathcal{L}(g(X;\Theta_1), Y)$, where $g(X;\Theta_1)$ is the prediction of target network, $\mathcal{L}(g(X;\Theta_1), Y)$ is the cross entropy loss:

$$\begin{aligned}\mathcal{L}(g(X;\Theta_1), Y) &= \sum_{n=1}^{N} \mathcal{L}(g(\mathbf{x}_n;\Theta_1), \mathbf{y}_n) \\ &= -\sum_{n=1}^{N} \mathbf{y}_n \cdot \log \mathbf{p}_n \end{aligned} \quad (1)$$

where $\mathbf{p}_n = g(\mathbf{x}_n, \Theta_1)$ is the posterior probability vector $[p_n^1, \ldots, p_n^i, \ldots, p_n^C]$, C is the number of classes. The probability p_n^i of input $\mathbf{x}_n$ on i-th class is typically computed using "softmax" operation:

$$p_n^i = \frac{exp(v_n^i)}{\sum_{j=1}^{C} exp(v_n^j)} \quad (2)$$

where v_n^i is the input of "softmax" layer, corresponding to the i-th class in target task. The gradient of the "softmax" layer can be computed as

$$\frac{\partial \mathcal{L}(g(\mathbf{x}_n;\Theta_1), \mathbf{y}_n)}{\partial \mathbf{v}_n} = \mathbf{p}_n - \mathbf{y}_n \quad (3)$$

The gradients of the other layers can be calculated in the conventional way as illustrated in [12]. In Section 3, it is shown that this fine-tuning method can achieve excellent performance on small-scale target image classification tasks. However, the computational complexity and storage consumption are too high for resource-constrained applications.

It is straightforward to reduce the complexity of Θ_1 by transferring less parameters from network Θ_0. Specifically, a compact target CNN model can be defined as $\Theta_2 = [\theta_2^1, \theta_2^2, \ldots, \theta_2^L]$, in which the first k layers $[\theta_2^1, \theta_2^2, \ldots, \theta_2^k], k < L-1$ are configured by transferring the corresponding parameters in Θ_0. The remaining layers are configured with much less node weights than those of Θ_1. In implementation, we set $k = 5$. In this case, θ_0^{k+1} is the first fully connected layer which contains most parameters in ImageNet CNN Θ_0.

Furthermore, a less complex ImageNet CNN can be used for CNN fine-tuning, for instance, CNN-F in [13] with 4 pixel stride in the first layer has a reduced number of convolutional layers. In Section 3, we evaluate the performance of different compact target CNNs.

With the fine-tuned compact target CNN model, the computational complexity and model size can be greatly reduced at the cost of a significant performance degradation. To address this issue, we propose a joint learning method to further improve the performance of this compact target CNN.

2.2. Joint learning

As aforementioned, we aim to improve the performance of the compact target model Θ_2 by jointly training with ground truth labels and predictions of the target model Θ_1. The objective function of joint learning is re-formulated as

$$(1-\alpha)\mathcal{L}(g(X;\Theta_2), Y) + \alpha\mathcal{L}(g(X;\Theta_2), g(X;\Theta_1)) \quad (4)$$

In addition to the cross entropy with ground truth labels, a cross entropy with the soft targets (predictions) from the model Θ_1 is added as a regularization term, where the parameter α controls the degree of regularization.

This may force the model Θ_2 to generate similar predictions as those from the model Θ_1. Similar as eqn.(1), the

regularization term can be defined as

$$\begin{aligned}\mathcal{L}(g(X;\Theta_2),g(X;\Theta_1)) &= \sum_{n=1}^{N}\mathcal{L}(g(\mathbf{x}_n;\Theta_2),g(\mathbf{x}_n;\Theta_1)) \\ &= -\sum_{n=1}^{N}\mathbf{p}_n\cdot\log\mathbf{q}_n \end{aligned} \quad (5)$$

where $\mathbf{p}_n = g(\mathbf{x}_n;\Theta_1)$ and $\mathbf{q}_n = g(\mathbf{x}_n;\Theta_2)$ are posterior probability vectors computed using a "softmax" operation. The gradient of the regularization term can then be written as

$$\frac{\partial\mathcal{L}(g(\mathbf{x}_n;\Theta_2),g(\mathbf{x}_n;\Theta_1))}{\partial\mathbf{z}_n} = \mathbf{q}_n - \mathbf{p}_n \quad (6)$$

where $\mathbf{z}_n = [z_n^1, z_n^2, \ldots, z_n^C]$ is the input vector of "softmax" layer in model Θ_2. By combining eqn.(6) and similar "softmax" gradient equation of $\mathbf{z}_n$, the gradient of objective function in eqn.(4) can be obtained as

$$(1-\alpha)(\mathbf{q}_n - \mathbf{y}_n) + \alpha(\mathbf{q}_n - \mathbf{p}_n) \quad (7)$$

In [14], an extra parameter *T* was introduced as the temperature to control the softness of probability distribution over classes. With the revised "softmax" operation, eqn.(6) can be rewritten as

$$\frac{\partial\mathcal{L}(g(\mathbf{x}_n;\Theta_2),g(\mathbf{x}_n;\Theta_1))}{\partial\mathbf{z}_n} = \frac{1}{T}(\mathbf{q}_n - \mathbf{p}_n) \quad (8)$$

Eqn.(8) adjusts the gradient by changing *T*. After the gradient is acquired, we use back propagation (BP) and stochastic gradient descent (SGD) algorithms to update model Θ_2.

3. EXPERIMENT AND ANALYSIS

3.1. Datasets

In our experiments, we evaluate the performance of our proposed compact CNN transfer learning method on CIFAR-10 [15] and MIT Indoor Scene [3].

CIFAR-10 Dataset is a small-scale object classification dataset. This dataset contains 60000 images with 10 object categories. Each class consists of 6000 images, including 5000 training images and 1000 test images. The size of each image is 32×32.

MIT Indoor Scene Dataset contains 6700 images with 67 scene categories. For each category, the standard training/test set consists of 80 training and 20 test images. This dataset is quite challenging since most scenes are collections of objects organized in a highly variable layout, with some subtle cross-category differences. Furthermore, the difference between this dataset and ImageNet is greater than that between CIFAR-10 and ImageNet, which may help to assess the generalization capability of our proposed method.

3.2. Experiment Settings

Computing Environment. All of our experiments were performed on a server with Intel Xeon E5-2650 and NVIDIA Tesla K40m installed. The MatConvNet toolbox [12] is used for evaluation in experiments.

Data Augmentation. The images are augmented as follows. For MIT Indoor Scene, 224×224 patches are cropped from random positions in images. These images are resized by original aspect ratios and their minimum dimension is 256. For CIFAR-10, we do not crop images. The patches and images are randomly flipped before feeding into the model.

Network Structures. We use ImageNet CNN structures **CNN-S** and **CNN-F** in [13]. CNN-S is an "accurate" model related to OverFeat [16]. CNN-F is a "fast" model similar to the structure of Krizhevsky *et.al* [1], but with larger stride and fewer convolutional layers.

Training Methods. In our experiments, we implement and compare the following methods:

1) baseline (**Base**): Training on small-scale dataset directly. 2) fine-tuning1 (**FT1**): Fine-tuning a CNN with all internal layers transferred from ImageNet CNN. 3) fine-tuning2 (**FT2**): Fine-tuning a compact CNN with all convolutional layers transferred from ImageNet CNN and smaller fully connected layers. 4) fine-tuning2+joint learning (**FT2+JL**): Optimizing a compact CNN trained by Fine-tuning2 using the objective function in eqn.(4). We use the "CNN-S FT1" model as the target model Θ_1.

Evaluation. The performance on the target task is evaluated in terms of accuracy. For CIFAR-10, one prediction of each image is used for evaluation, while for the MIT Indoor Scene database, the average of the predictions on 18 cropped patches is used. Furthermore, storage consumption is evaluated in terms of network size (MB) and computational complexity in terms of test speed (images/sec).

3.3. Experimental results on CIFAR-10 dataset

The experimental results are shown in Table 1. Compared to **CNN-S FT1**, our method **CNN-S FT2+JL** and **CNN-F FT2+JL** reduce its model size by 83% and 93%, while accelerate test speed by 43% and 138% at the cost of slight accuracy decrease (less than 1% for both).

Furthermore, we compare the proposed method with other reported systems, as shown in Table 2. Deeply-Supervised Net [17] introduces "companion" objective functions at each individual hidden layer to improve network training. Network in Network [18] uses mlpconv layers to enhance model discriminability for local patches. DropConnect [19] sets a randomly selected subset of weights to zero to regularize large fully-connected layers. These leading methods also benefit from complex data augmentations like cropping, flipping, scaling and rotation. In comparison, our method achieved the best performance only with flipping operation. Further im-

provement can be obtained by a more complex data augmentation.

Table 1. Performance comparison of different CNN models on CIFAR-10 datset, in terms of accuracy (%), size (MB), and speed (images/sec)

method	accuracy	size	speed
CNN-S Base	87.78	377	122
CNN-S FT1	94.64		
CNN-S FT2	93.74	62	174
CNN-S FT2+JL	94.05		
CNN-F Base	87.57	217	244
CNN-F FT1	93.27		
CNN-F FT2	92.86	27	290
CNN-F FT2+JL	93.74		

Table 2. Performance comparison with other state-of-the-art methods on CIFAR-10 in terms of accuracy(%)

method	accuracy(%)
CNN-S FT1	94.64
CNN-S FT2+JL	94.05
CNN-F FT2+JL	93.74
Deeply-Supervised Net [17]	92.03
Network in Network [18]	91.19
DropConnect [19]	90.68
Maxout Network [20]	90.62

3.4. Experimental results on MIT Indoor Scene

The experimental results on MIT Indoor Scene dataset shows the same trend on MIT Indoor Scene, as shown in table 3. Compared to **CNN-S FT1**, our method **CNN-S FT2+JL** and **CNN-F FT2+JL** reduce model size by 51% and 74%, while accelerate test speed by 20% and 105%. Meanwhile, these two methods achieve accuracy improvements by 25% and 24.8%, compared to baseline method.

A comparison of our method with other leading methods on MIT Indoor Scene is shown in table 4. The performance of our proposed compact CNN transfer learning, *i.e.* 71.4% for CNN-S FT2+JL and 68.2% for CNN-F FT2+JL, is competitive with them. The best performance is achieved by FC8 FV method [21], which uses a bag of semantic Fisher Vector multi-scale patches extracted from a pre-trained ImageNet CNN. It is with high computational complexity and storage for Fisher Vector embedding on extracted CNN features. Furthermore, the performance of our compact CNN transfer learning can be improved by using multi-scale patches.

Table 3. Performance comparison of different CNN models on MIT Indoor Scene datset, in terms of accuracy (%), size (MB), and speed (images/sec)

method	accuracy	size	speed
CNN-S Base	46.42	378	122
CNN-S FT1	73.13		
CNN-S FT2	63.88	185	147
CNN-S FT2+JL	71.42		
CNN-F Base	43.43	217	244
CNN-F FT1	67.84		
CNN-F FT2	61.87	97	250
CNN-F FT2+JL	68.21		

Table 4. Performance comparsion with other reported state-of-the-art methods on MIT Indoor Scene in terms of accuracy(%)

method	accuracy(%)
CNN-S FT1	73.13
CNN-S FT2+JL	71.42
CNN-F FT2+JL	68.21
FC8 FV [21]	72.86
FC7 FV [21]	69.7
FC7 VLAD [22]	68.88
ImageNet finetune [6]	63.9
OverFeat + SVM [4]	69
FC6 + Sparse Coding [23]	68.2
Decaf [24]	59.5

4. CONCLUSIONS

In this paper, we presented a compact CNN transfer learning method for small-scale image classification tasks. The compact CNN transfer learning method includes fine-tuning and joint learning stages. In fine-tuning, a high-performance CNN trained on the target dataset is fine-tuned by transferring the parameters of internal layers from a pre-trained CNN. In the joint learning stage, a compact CNN that satisfies the complexity and storage requirement is firstly designed with reduced fully connected layers, and then optimized with an objective function according to posterior probabilities from the high-performance CNN model.

The experimental results clearly demonstrate the effectiveness and efficiency of our method. The classification accuracy of 94.1% on CIFAR-10, and 71.4% on MIT Scene have been achieved by using the transferred compact CNN model with more than 1.2x speedup. With a more compact target CNN model (i.e. CNN-F FT2+JL), the classification accuracy may slightly degrade to 93.7% on CIFAR-10, and 68.2% on MIT Indoor Scene, with more than 2x speed up.

5. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of NIPs*, 2012, pp. 1106–1114.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, "Imagenet: A large-scale hierarchical image database," in *Proc. of CVPR*, 2009, pp. 248–255.

[3] Ariadna Quattoni and Antonio Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 413–420.

[4] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014.

[5] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *Proc. of ECCV*, 2014.

[6] Ross B. Girshick and Jeff Donahue, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of CVPR*, 2014, pp. 580–587.

[7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Proc. of NIPs*, Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, Eds., pp. 3320–3328. 2014.

[8] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Proc. of NIPs*. IEEE, 2014.

[9] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proc. of BMVC*, 2014.

[10] Xiangyu Zhang, Jianhua Zou, Xiang Ming, Kaiming He, and Jian Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *Proc. of CVPR*, 2015.

[11] Ariadna Quattoni and Antonio Torralba, "Recognizing indoor scenes," in *Proc. of CVPR*, 2009, pp. 413–420.

[12] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," *CoRR*, vol. abs/1412.4564, 2014.

[13] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. ICLR*, 2014.

[17] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, "Deeply-supervised nets," *arXiv preprint arXiv:1409.5185*, 2014.

[18] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400v3*, 2014.

[19] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1058–1066.

[20] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[21] Mandar Dixit, Si Chen, Dashan Gao, Nikhil Rasiwasia, and Nuno Vasconcelos, "Scene classification with semantic fisher vector," in *Proc. of CVPR*, 2015.

[22] Yunchao Gong, Liwei Wang, and Ruiqi Guo, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. of ECCV*, 2014, pp. 392–407.

[23] Lingqiao Liu, Chunhua Shen, Lei Wang, Anton van den Hengel, and Chao 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*, 2014, pp. 1143–1151.

[24] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2014.