Q2

# Scene Classification Based on Two-Stage Deep Feature Fusion

基于两阶段深度特征融合的场景分类

Yishu Liu, Yingbin Liu, and Liwang Ding

***Abstract*— In convolutional neural networks (CNNs), higher layer information is more abstract and more task specific, so people usually concern themselves with fully connected (FC) layer features, believing that lower layer features are less discriminative. However, a few researchers showed that the lower layers also provide very rich and powerful information for image representation. In view of these study findings, in this letter, we attempt to adaptively and explicitly combine the activations from intermediate and FC layers to generate a new CNN with directed acyclic graph topology, which is called the converted CNN. After that, two converted CNNs are integrated together to further improve the classification performance. We validate our proposed two-stage deep feature fusion model over two publicly available remote sensing data sets, and achieve a state-of-the-art performance in scene classification tasks.**

motivation

有向无环图 转化

***Index Terms*— 1 × 1 convolution, composite convolutional neural networks (CNNs), converted CNN, deep feature fusion, global average pooling (GAP).**

## I. Introduction

SCENE classification has been a hot research topic for decades. In recent years, convolutional neural networks (CNNs) have made great achievements in this field and frequently obtain a significantly better performance than the traditional hand-crafted features [1]–[5].

Most studies [2]–[5] adopted the outputs of fully connected (FC) layers of CNNs as the image representation, and people seem to believe that lower layer activations are less discriminative. However, Liu *et al.* [6] and Yue-Hei Ng *et al.* [7] demonstrated that lower layer activations produce comparable or in some cases better results than FC layer activations. This has proved that intermediate layers also provide very powerful and rich information for image representation. In fact, lower layers produce local and more generic features, while FC layers produce global and more data set specific ones, and they are complementary to some extent. Therefore, combining information from intermediate and FC layers can improve the classification performance, which has been justified by some researchers [1], [8], [9].

Hu *et al.* [1] concatenated the intermediate layer activations with FC layer activations after dimension reduction. Since they used pretrained CNNs as off-the-shelf feature extractors, their fusion scheme is rigid and most probably not optimal or even reasonable. Long *et al.* [8] proposed a multiresolution layer combination method for incorporating intermediate outputs during training. However, their fusion modules employ the simple "sum" rule, and hence, lack the flexibility in determining how much importance is attached to individual branches. Besides, their scheme needs a great many additional parameters. Furthermore, Yang and Ramanan [9] used directed acyclic graph (DAG)-CNNs to integrate different layers. Unfortunately, their fusion is performed at score level instead of at feature level, and score-level fusion is always less effective.

现成的

参考一篇ICCV的论文

创新点一：融合特征层

To address these issues, in this letter, we propose a novel feature-level fusion method for adaptively combining the information from lower layers and FC layers, in which the fusion coefficients are automatically learned from data, and not designed beforehand. Moreover, our model needs only a small number of additional parameters because of $1 \times 1$ convolution.

创新点二：自学习也就是加权

使用1x1卷积进行特征融合

In addition, Penatti *et al.* [5] empirically proved that fusing multiple CNNs leads to a better performance compared with the individual CNNs—they only concatenated the activations of the second FC layers of the pretrained OverFeat and CaffeNet, and over the well-known UC-Merced data set got an accuracy as high as 99.43%. Inspired by these promising results, we propose integrating two CNNs to further improve classification after the lower and higher layers of each CNN have been combined. However, our fusion is performed via a linear combination of feature vectors instead of feature concatenation, and the fusion weights are learned via training and fine-tuning. This has the following advantages: 1) the combined feature vector is shorter, making the subsequent classification easier; 2) more elasticity is brought into the fusion process due to variable fusion weights; and 3) our approach is more suitable for remote sensing (RS) scene classification because of fine-tuning over RS data sets—in fact, our model is tailor-made for RS images, while Penatti *et al.* [5] used the models aimed at everyday images.

CVPR

创新点三：特征融合是加权而不是直接连接

适应

In a nutshell, unlike the above one-stage fusion strategies [1], [5], [8], [9], a two-stage deep feature fusion approach is proposed in this letter: first, activations from intermediate layers and FC layers are automatically incorporated by individual CNNs, and second, the resulting converted CNNs are adaptively combined to generate one composite CNN. And then we conduct experiments over two publicly available RS data sets to validate our proposed method.

简而言之

## II. Two-Stage Deep Feature Fusion

Our proposed method can be illustrated in Fig. 1, in which C, P, FC, and global average pooling (GAP) mean

Manuscript received July 3, 2017; revised October 6, 2017 and November 13, 2017; accepted November 28, 2017. Date of publication December 22, 2017; date of current version January 23, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61673184. *(Corresponding author: Yishu Liu.)*

The authors are with the School of Geography, South China Normal University, Guangzhou 510631, China (e-mail: yishuliu_gz@hotmail.com; 2604245496@qq.com; 1021470364@qq.com).

华南师范

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2779469

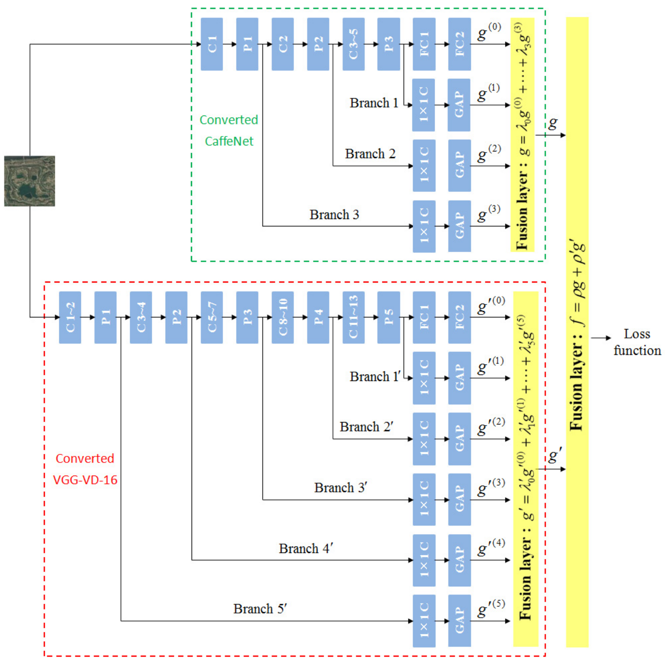


Fig. 1. Process of two-stage deep feature fusion.

*convolutional layer*, *pooling layer*, *FC layer*, and *GAP layer*, respectively. It should be pointed out that both nonlinear activation and normalization are omitted from Fig. 1 to simplify the illustration.

Since labeled RS images are usually limited, we refit the two existing CNN models, i.e., the well-known CaffeNet [10] and VGG-VD-16 [11] that have both been pretrained over the ILSVRC-2012 training set [12], instead of building new network architectures and training them from scratch. And then the two converted CNNs are fused.

### A. Combining Intermediate and FC Layers

In order to merge information from intermediate and FC layers, we insert one branch CNN after each pooling layer, as shown in Fig. 1. Each branch CNN is composed of three layers: $1 \times 1$ convolution, nonlinear activation ReLU, and GAP.

Suppose that a certain pretrained CNN (we call it *trunk CNN*) sends forth $B$ branches, and the input of the $b$th branch ($b = 1, 2, \ldots, B$) has $K^{(b)}$ channels (i.e., feature maps) $X_k^{(b)}$, $k = 1, 2, \ldots, K^{(b)}$, then the output of $1 \times 1$ convolution in the $b$th branch is

$$Y_l^{(b)} = \sum_{k=1}^{K^{(b)}} w_{kl}^{(b)} X_k^{(b)}, \quad l = 1, 2, \ldots, L \tag{1}$$

where $L$ is the number of the output channels, $W^{(b)} = (w_{kl}^{(b)})_{K^{(b)} \times L}$ is all the $1 \times 1$ convolution weights in the $b$th branch (we omit biases to simplify the expression).

Instead of using usual convolution, we utilize $1 \times 1$ convolution thanks to the inspiration drawn from GoogLeNet [13]. The $1 \times 1$ convolution is a "feature pooling" technique, producing a linear combination of various channels in a given layer. And it is equivalent to cross-channel parametric pooling layer, allowing learnable interactions of cross-channel information. In comparison with usual convolution, it needs less weights and can avoid overfitting to some extent due to smaller kernel size ($1 \times 1$).

Every $1 \times 1$ convolution is succeeded by a nonlinear activation layer such as ReLU, which adds more nonlinearity and more expressive power to the data.

And then, GAP follows. Denoting by $y_{i,j,l}^{(b)}$ the $(i, j)$ entry of $Y_l^{(b)}$, GAP results in

$$g_l^{(b)} = \sum_i \sum_j y_{i,j,l}^{(b)}, \quad l = 1, 2, \ldots, L. \tag{2}$$

Let

$$g^{(b)} = \left(g_1^{(b)}, g_2^{(b)}, \ldots, g_L^{(b)}\right), \quad b = 1, 2, \ldots, B \tag{3}$$

this $L$-dimensional vector is the final output of the $b$th branch. And we denote the activation vector of the second FC layer of the trunk CNN as $g^{(0)}$.

The pretrained CNN into which several branches have been inserted is referred to as *converted CNN* in this letter. The converted CNN's final feature vector can be expressed as the following linear combination:

$$g = \sum_{b=0}^{B} \lambda_b g^{(b)}. \tag{4}$$

We stress that $g^{(b)}$, $b = 0, 1, \ldots, B$, should have the same length, $L$, to guarantee vector addition in (4) can be performed. Therefore, in the first-stage fusion, about $(L \sum_{b=1}^{B} K^{(b)} + B + 1)$ new weight coefficients must be added to each pretrained CNN. These weights are learned from the training samples themselves (see Section II-C), providing much adaptivity and flexibility for fusion.

Finally, because both $g^{(0)}$ and $g'^{(0)}$ are 4096-D, we set $L = 4096$ in this letter.

### B. Combining the Converted CNNs

Instead of simply concatenating the two feature vectors as most researchers did [1], [5], we perform linear combination again. Denote the outputs of the converted CaffeNet and VGG-VD-16 as $g$ and $g'$, respectively, then the two converted CNNs are integrated via the following operation:

既然你能用线性加权，为什么就不能用逻辑回归？

$$f = \rho g + \rho' g' \tag{5}$$

where $\rho$ and $\rho'$ are adaptively determined. We refer to the resulting networks as *composite CNN*.

In the second-stage fusion, only two weights, $\rho$ and $\rho'$, are added, so the computational workload is negligible.

### C. Training

The network structure in Fig. 1 is a DAG, and we can train the composite CNN using stochastic gradient descent (SGD). The backpropagation is similar to that of GoogLeNet [13], which is also a DAG.

For any $b \in \{1, 2, \ldots, B\}$, we randomly initialize the $1 \times 1$ convolution weights $W^{(b)}$ and initially set

$$\lambda_b = \frac{1}{B+1}, \quad b = 0, 1, \ldots, B \tag{6}$$

and

$$\rho = \rho' = \frac{1}{2}. \tag{7}$$

TABLE I
OVERALL ACCURACIES OF VARIOUS MODELS

| **Dataset** | **Model** | | | | | |
|---|---|---|---|---|---|---|
| | Fine-tuned CaffeNet | Fine-tuned VGG-VD-16 | Converted CaffeNet | Converted VGG-VD-16 | Composite CNN | Concatenation |
| AID | 91.08±0.32 | 92.37±0.33 | 92.17±0.31 | 93.48±0.34 | 94.65±0.33 | 93.35±0.33 |
| RSSCN7 | 88.87±0.65 | 89.21±0.82 | 90.02±0.68 | 91.45±0.78 | 92.37±0.72 | 91.28±0.80 |

Since training deep CNNs requires a significant quantity of labeled training samples and the existing RS data sets cannot provide sufficient data, we use the everyday image data set, ILSVRC-2012 [12], to pretrain the networks, and then use the RS data sets for fine-tuning.

training strategy

Training the composite CNN consists of the following steps (remember that the trunks of CaffeNet and VGG-VD-16 have both been pretrained).

1) Freeze the trunk of CaffeNet, and train its branches and fusion layer using ILSVRC-2012.
2) Unfreeze the trunk of CaffeNet, and train the whole converted CaffeNet using ILSVRC-2012.
3) For VGG-VD-16, do things analogous to steps 1)–2).
4) Train the composite CNN using ILSVRC-2012.
5) Fine-tune the composite CNN using an RS data set.

In Section III, we also analyze the converted CNNs' performances for comparison purposes. To train the converted CaffeNet, we only need to replace the above steps 3)–5) with step 3′): fine-tune the converted CaffeNet using an RS data set.

## III. NUMERICAL EXPERIMENTS

### A. Experimental Setup

*1) RS Data Sets:* The details about the everyday image data set ILSVRC-2012 used for pretraining can be found in [12]. Besides, two very challenging RS data sets are used for fine-tuning. The first one is the newly released AID [2], which has 10 000 images belonging to 30 aerial scene categories. The image size is $600 \times 600$ pixels, and the spatial resolution varies from 8 m to about half a meter. In the second data set RSSCN7 [14], there are seven classes, each containing 400 images with the size of $400 \times 400$ pixels. These images were sampled on four different scales with 100 images per scale.

*2) Data Set Division:* We adopt the same data augmentation strategy as that in [4]. Then the two RS data sets are both randomly and equally divided into two subsets for fine-tuning and testing. We conduct 10 repetitions of each experiment, each repetition having different data set division.

*3) Parameters of SGD:* We train our networks using SGD, with a momentum of 0.9 and a weight decay of 0.0005 all the time. The batch size is 256 for ILSVRC-2012 and 100 for the RS data sets. As for the learning rates, we use 0.01 for ILSVRC-2012 and 0.001 for the RS data sets. Finally, 20 000 iterations are run in every training step as explained in Section II-C.

### B. Classification Accuracy

We consider the following three kinds of networks: 1) CaffeNet and VGG-VD-16 fine-tuned over the RS data sets ($g^{(0)}$ or $g'^{(0)}$ is used as feature vector); 2) the converted CNNs trained as described in Section II-C ($g$ or $g'$ is used as feature vector); and 3) the composite CNN trained as described in Section II-C ($f$ is used as feature vector). For comparison purposes, we also investigate another feature fusion scheme, i.e., feature concatenation [5]: after fine-tuning CaffeNet and VGG-VD-16 over the RS data sets, we concatenate $g^{(0)}$ and $g'^{(0)}$.

The whole training process takes about 32 h on an AMAX workstation with two Intel CPUs and two NVIDIA Titan X GPUs. Then, Liblinear [15] is used for supervised classification. The overall accuracies are presented in Table I. And statistical tests are conducted for the three groups of methods, i.e., fine-tuned CaffeNet, converted CaffeNet, and composite CNN; fine-tuned VGG-VD-16, converted VGG-VD-16, and composite CNN; and composite CNN and concatenation. The three $p$-values are all less than 0.00007, which means that there are statistically significant differences among/between the methods in any group. From these results, we reach the following conclusions.

居然是这个

1) The converted CaffeNet/VGG-VD-16 has a higher accuracy than CaffeNet/VGG-VD-16, demonstrating that integrating lower and higher layers leads to a better performance compared with using higher layers only. This indicates that intermediate layers also provide rich and useful information.
2) The composite CNN outperforms both the converted CNNs, demonstrating that combining two CNNs leads to a better performance compared with using one CNN only. This indicates that different CNNs contain complementary information.
3) Feature concatenation results in a better performance than individual fine-tuned CNNs, but it is inferior to the composite CNN.
4) Points 1)–3) show that our proposed two-stage fusion method is feasible, effective, and better than the commonly used feature concatenation.

特征级联性能优于个人微调
但是次于CNN的整体性能

### C. Confusion Analysis

We make a confusion analysis of the composite CNN. Due to limited space, Fig. 2 presents only the confusion matrix for RSSCN7.

Over AID, the classes with an accuracy close to 1 are basically natural scenes, including bare land, beach, desert, forest,

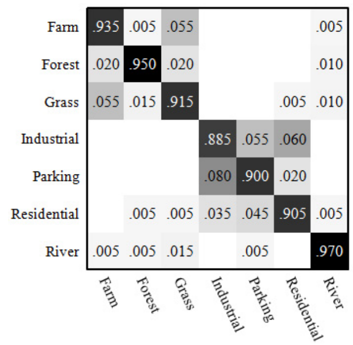


Fig. 2. Confusion matrix for RSSCN7.

TABLE II
PERFORMANCE COMPARISON OVER AID

| Method | Overall accuracy (%) |
|---|---|
| Pretrained GoogLeNet [2] | 86.39±0.55 |
| Pretrained CaffeNet [2] | 89.53±0.31 |
| Pretrained VGG-VD-16 [2] | 89.64±0.36 |
| salM$^3$LBP-CLM [16] | 89.76±0.45 |
| Combining 2 FC layers [3] | 91.87±0.36 |
| Two-stage deep feature fusion [this work] | **94.65±0.33** |

TABLE III
PERFORMANCE COMPARISON OVER RSSCN7

| Method | Overall Accuracy (%) |
|---|---|
| Pretrained GoogLeNet [2] | 85.84±0.92 |
| Hierarchical coding [17] | 86.4±0.7 |
| Pretrained VGG-VD-16 [2] | 87.18±0.94 |
| Pretrained CaffeNet [2] | 88.25±0.62 |
| Deep filter banks [18] | 90.4±0.6 |
| Two-stage deep feature fusion [this work] | **92.37±0.72** |

and mountain. And the classes with low accuracy contain school (0.60), square (0.64), resort (0.68), and center (0.72). The most noticeable confusion occurs between resort and park (about 10% resort images are mistakenly classified as park, and about 8% park images are mistakenly classified as resort), which may attribute to the common objects shared by resort and park scenes. And the confusion between school and dense residential is a close second.

Over RSSCN7, the class with the highest accuracy is river (0.97), followed by forest (0.95). These two kinds of scenes always have a regular textural and spatial structure. And the major confusion occurs between industrial region and parking lot, industrial region and residential region, farmland and grassland, resulting from their similar global structure or spatial layout.

D段 就将本文提出的方法比以往的方法精度更高

### *D. Comparison With State-of-the-Art Methods*

We compare the composite CNN with the existing methods (see Tables II and III). It can be seen that our proposed model outperforms all the other approaches, including quite a few ones based on deep learning, over both AID and RSSCN7, and the margins are generally rather large—our accuracies are higher than the second best results by 2.78% and 1.97%, over AID and RSSCN7, respectively.

## IV. CONCLUSION

Since in CNNs higher layers learn more abstract and more task-specific features, most people have a focus on FC layers and pay much less attention to lower layers. This letter attempts to adaptively and explicitly combine outputs from lower and higher layers, showing that intermediate layers also provide very rich and powerful information for image representation. Then the two resulting converted CNNs are adaptively integrated to further improve the classification performance. Experiments conducted over two challenging RS data sets demonstrate that our proposed two-stage deep feature fusion model outperforms the existing approaches. two-stage deep feature fusion 比现有的方法要优

## REFERENCES

[1] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015.

[2] G.-S. Xia, J. Hu, and F. Hu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[3] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.

[4] Y. Liu and C. Huang, "Scene classification via triplet networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 220–237, Jan. 2018.

[5] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, Boston, MA, USA, Jun. 2015, pp. 44–51. CVPR

[6] L. Liu, C. Shen, and A. van den Hengel, "The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4749–4757.

[7] J. Y.-H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., DeepVis. Workshop (CVPRW)*, Apr. 2015, pp. 53–61.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[9] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1215–1223. ICCV

[10] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 675–678.

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

[12] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 1–9.

[14] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[16] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2889–2901, Jun. 2017.

[17] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Hierarchical coding vectors for scene level land-use classification," *Remote Sens.*, vol. 8, no. 5, p. 436, May 2016.

[18] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Deep filter banks for land-use scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1895–1899, Dec. 2016.