Computer Modeling in Engineering & Sciences CMES, vol.122, no.2, pp.505-523, 2020

Advanced Feature Fusion Algorithm Based on Multiple Convolutional Neural Network for Scene Recognition

Lei Chen[1, #], Kanghu Bo[2, #], Feifei Lee[1, *] and Qiu Chen[1, 3, *]

Abstract: Scene recognition is a popular open problem in the computer vision field. Among lots of methods proposed in recent years, Convolutional Neural Network (CNN) based approaches achieve the best performance in scene recognition. We propose in this paper an advanced feature fusion algorithm using Multiple Convolutional Neural Network (Multi-CNN) for scene recognition. Unlike existing works that usually use individual convolutional neural network, a fusion of multiple different convolutional neural networks is applied for scene recognition. Firstly, we split training images in two directions and apply to three deep CNN model, and then extract features from the last full-connected (FC) layer and probabilistic layer on each model. Finally, feature vectors are fused with different fusion strategies in groups forwarded into SoftMax classifier. Our proposed algorithm is evaluated on three scene datasets for scene recognition. The experimental results demonstrate the effectiveness of proposed algorithm compared with other state-of-art approaches.

Keywords: Scene recognition, deep feature fusion, multiple convolutional neural network.

1 Introduction

Deep convolutional neural network (DCNN) has proved to provide better feature representation compared with low-level manually extracted features [Seong, Hyun and Kim (2019); Liu, Chen, Chen et al. (2018)]. Primary learning of the CNN is end-to-end, so that make training convenient. In DCNN, the low-level convolutional layer is used as the Gabor filters and color blob detectors [Yosinski, Clune, Bengio et al. (2014)] which can extract the information such as edges and texture, and fully-connected (FC) layers encode abstractive feature information that reduce the influence of the local information change. Recently, researchers have found that high-level features which are extracted form FC layers has better performance in image classification [Razavian, Azizpour,

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China.

[2] Algorithm Department, Unisoc, Shanghai, 201203, China.

[3] Major of Electrical Engineering and Electronics, Graduate School of Engineering, Kogakuin University, Sinjuku-ku, Tokyo, 163-8677, Japan.

[#] Both authors contributed equally to this work.

[*] Corresponding Authors: Feifei Lee. Email: feifeilee@ieee.org;

Qiu Chen. Email: q.chen@ieee.org.

Received: 24 August 2019; Accepted: 23 October 2019.

Sullivan et al. (2014); LeCun, Bengio and Hinton (2015); Guo, Huang, Wang et al. (2017); Liu, Ma, Zhou et al. (2019); Zhao and Larson (2018)]. It is significant to employ multiple such high-level features from distinct CNN architecture for more generalized representation of scene information. The reason is that using single modality high-level features is hard to distinguish the fine-grained difference in the field of scene recognition. So researchers focus on feature fusion methods, such as multi-stage feature fusion of the GoogLeNet models for scene recognition [Tang, Wang and Kwong (2017)], which improves the accuracy than the state-of-the-art CNN models. However, only a single CNN model, i.e., GoogLeNet is used, thus the features extracted from different CNN models cannot be combined, although which can express more semantic information.

These methods seek to optimize features acquisition are propitious to scene recognition. In previous period scene recognition algorithms use manual features like GIST [Oliva and Torralba (2001)], SIFT [Lowe (2004)] and CENTRIST [Wu and Rehg (2011)], which obtain promising results for certain tasks. But more discriminative information is ignored at higher levels that are critical for scene understanding. In addition, the manually features cannot be transferred to a new target domain because they only have better performance in the original domain. Because of the shortness of manually features, deep feature learning methods are proposed such as Deep Belief Net (DBN) [Hinton, Osindero and Teh (2006)], Deep Boltzmann Machines (DBM) [Salakhutdinov and Hinton (2012)] and Convolutional Deep Belief Network (CDBN) [Lee, Pham, Largman et al. (2009)] and et al. We can see that the features extracted from different layers by unsupervised CNN feature learning are still more prominent than these methods. On the other hand, fusion of multiple features has proved to achieve better results than many single methods for applications like classification and recognition [Arora, Bhaskara, Ge et al. (2014)]. Our work is to improve scene recognition performance by constructing a more general feature representation model via fusion of multiple deep features. Lavinia et al. fuse the features extracted from 2 or 3 CNN models [Lavinia, Vo and Verma (2016)], but train CNN models in the same dataset. Different from it, we train 3 single models in 3 different datasets in our method, which can get the more suitable parameters of the CNN models because of the diverse scale.

Inspired by CNN's applications using fusion strategy in other tasks, we propose in this paper an advanced feature fusion framework based on Multiple Convolutional Neural Network (Multi-CNN) for scene recognition. The contributions of our paper are concluded as follows:

1. An advanced feature fusion framework using multiple DCNN for scene recognition is proposed;
2. The fusion of 2 deep CNNs provides higher accuracy of scene recognition than a single model;
3. The fusion of 3 deep CNNs obtains a higher-level performance compared with the 2-CNN fusion;
4. The fusion coefficient shows the importance in the fusion CNN feature learning, which may enhance the distinction of the respective corresponding model in fusion models;
5. The training of different source domain makes an important contribution to the latter fusion CNN and the generalization.

The rest of our paper is organized as follows. Some related DCNN models will firstly introduced, and then proposed DCNN fusion framework be described in Section 2. Experimental results compared with conventional approaches will be presented and discussed in detail in Section 3. Finally, the conclusions will be given.

2 Proposed deep CNN fusion methodology

Our method is built on the idea of using high-level CNN representation for scene recognition. In this paper, a SoftMax classifier is trained by employing a number of CNN models with completely distinct structures and using their complementary cues.

2.1 Overview of deep learning models

Three CNN models are used as the basis of the fusion model in our proposed strategy. The deep CNN models we chose are AlexNet, GoogLeNet and VGG-16. Generally speaking, a smaller CNN is suitable for small datasets. Because the Alexnet has only eight layers, so we choose it as the first single model to extract low-level features such as edge features. GoogLeNet includes the 1*1 convolutional layers, which can reduce the channel dimensions of feature maps, and improve the computational effectiveness. GooLeNet is selected as the second single model to extract higher-level features such as some fine-grained local features. On the other hand, VGG-16 can balance the computation and efficiency. Although these 3 models contain some similar structures, they have their own characteristics. We train these 3 single models with 3 different datasets, which contain images of different scene and scale. Therefore, we can get better image features by the fusion strategy. Their convolutional layers are regarded as feature extractors to extract scene features, and the parameters of the full convolutional layers of all benchmark models are fixed. Our work mainly focuses on the full connection layers and the derived probabilistic layers. The architecture of each model is described as follows.

2.1.1 AlexNet [Krizhevsky, Sutskever and Hinton (2012)]

AlexNet is a typical Convolutional Neural Network (CNN) architecture, which contains 5 convolutional layers as well as 3 fully-connected (FC) layers. Some novel methods such as ReLU, Dropout are adopted first time in this CNN model for reducing overfitting in the FC layers. As the winner of the 2012 ILSVRC competition, AlexNet achieves a significant improvement with the top-5 error rate down to 16.4%, almost 10% ahead of the second place. In Alexnet the overlap adjacent pooling units certainly not lead to over fitting. For the training of the small dataset, it is undoubtedly good to choose the relatively shallow AlexNet as the first DCNN model, which structure is shown in Fig. 1.

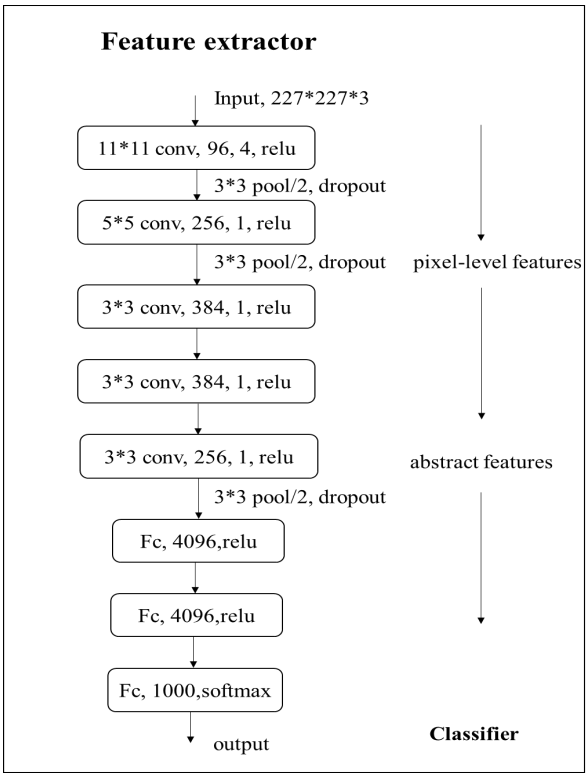


Figure 1: The architecture of AlexNet

2.1.2 GoogLeNet [Szegedy, Liu, Jia et al. (2015)]

As the winner of the 2015 ILSVRC competition, GooLeNet (Inception-v1) is a deep yet small scale network, which has improvements on performance and calculating. Its relatively low computation cost profits from the two aspects: firstly, convolutional neural network (CNN) is optimized by using sparseness. Secondly, the dimensionality is reduced through 1*1 convolutional layers as that in Network-in-Network (NIN) model [Lin, Chen and Yan (2013)].

The Inception-v1 module of GoogLeNet, block is shown in Fig. 2, in which the overall architecture is wide and deep (22 layers) as shown in Fig. 3. However, the decrease of the calculating on account of the first conventional convolutional layers and the 1*1 convolutional layers reduce the dimension. GoogLeNet depends on wider structure to boost capacity of network and training DCNN becomes easier than before. Thus, GoogLeNet is our second DCNN model.

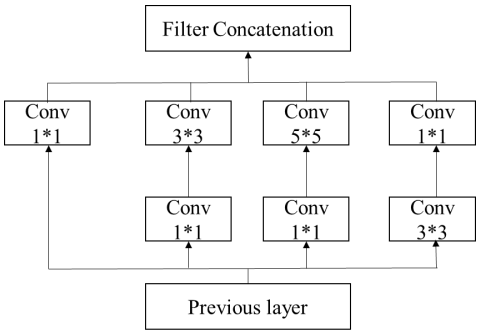


Figure 2: The structure of block

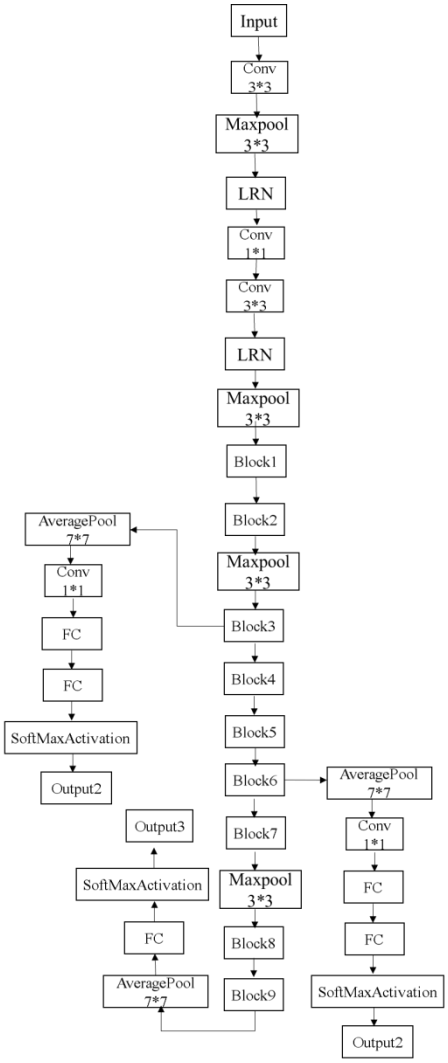


Figure 3: The architecture of GoogLeNet

2.1.3 VGG Net [Cui, Zhou, Wang et al. (2017)]

This deep CNN architecture employs 16 or 19 layers with small (3*3) convolutional kernel and the stride is 1 in the overall network. VGG-16 has 13 convolutional layers and 3 FC layers. The advantage can be summarized as two aspects: 1) More layers show the better performance in distinguish the rectified linear activation; 2) decrease the number of parameters. VGG is a large computation but good performance network. Considering the balance between performance and efficiency, we choose VGG16 as the third DCNN model and its architecture is shown in Fig. 4.

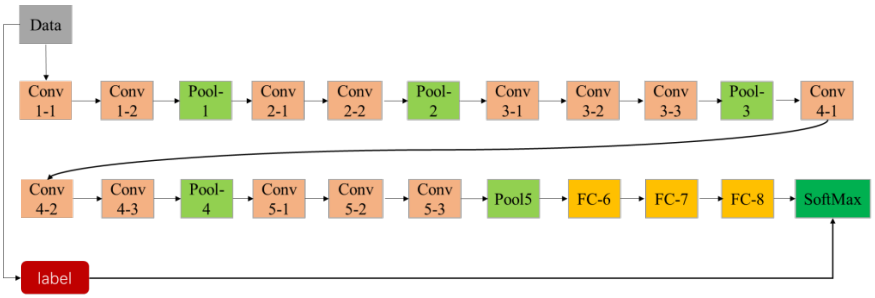


Figure 4: The architecture of VGG16

2.2 Proposed method

We make the assumption that extracted features from diverse CNN layers can draw on each other's strong points, after that the fusion layer extracted the uniform features among these features in order to make better discrimination between the scene classes.

As shown in Figs. 5-7, the process of feature fusion learning consists of 4 steps:

(1) Input Images & Preprocessing (segmentation) (Section 2.2.1)

(2) Feature transformation (high-level CNN representation) (Section 2.2.2)

(3) Fusion feature (with five rules) (Section 2.2.3)

(4) Classification (with SoftMax) (Section 2.2.4)

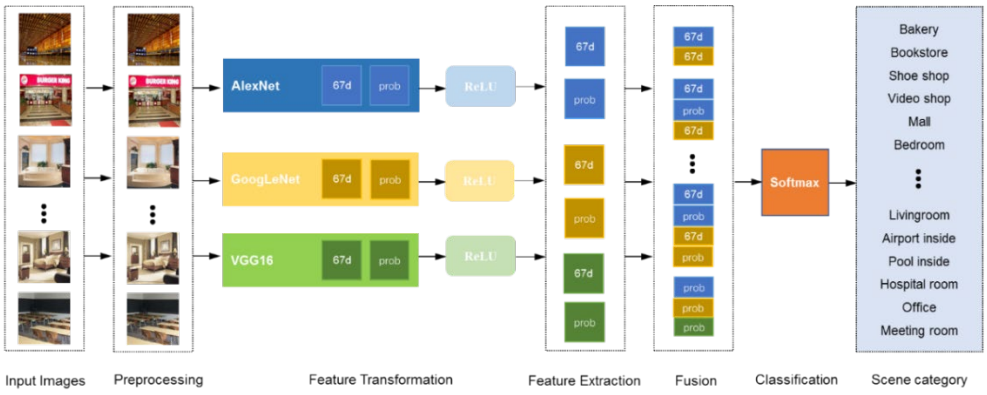


Figure 5: Fusion feature learning frame overview on the same dataset, e.g., MIT67-indoor dataset

As shown in Fig. 5, input images are resized and split in two directions and fed into 3 single CNN models. Then, extracted features from 67-dimensional(d) fully-connected (FC) layer and probabilistic layer on the 3 models. Finally, those vectors are fused with different fusion strategies in groups of two and three forwarded into SoftMax classifier.

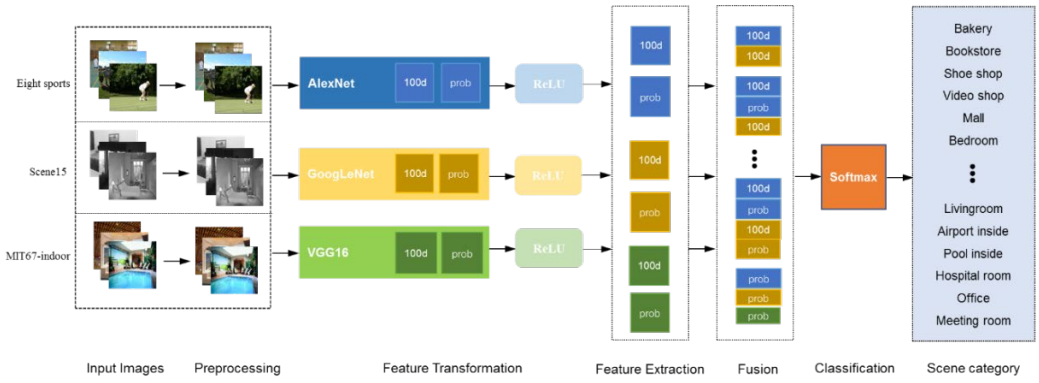


Figure 6: Fusion feature learning frame overview on different datasets

The process of learning on different datasets shown in Fig. 6 is approximately the same as the Fig. 5. We use 3 single CNN models to extract the features of the input images from different datasets. Eight sports is used for training Alexnet, Scene15 for GoogLeNet, and MIT67-indoor for VGG16. Then we get the trained models to extract features from images to achieve better results. The difference is that we insert a new 100- dimensional (d) FC layer as the penultimate FC layer to figure out whether the dimension of the penultimate FC layer can affect the experimental results and the 67-d is the optimal choice. The 100-d can reduce the dimension and regularization of the extracted features which have different dimension. The features are extracted from the 100-d layer and the probabilistic layer derived from distinct CNN models. Moreover, the target domain has not overlap with source domain in testing.

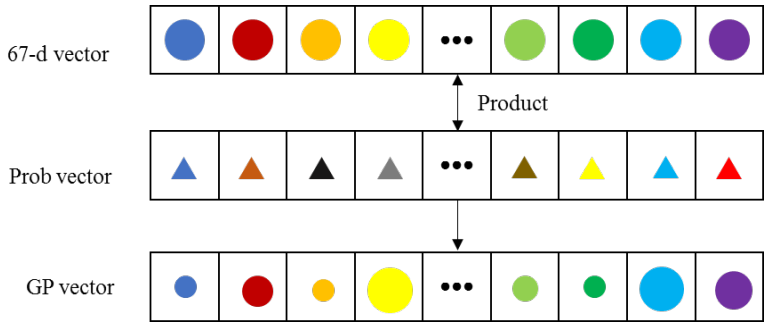


Figure 7: The GP vector generated from GoogLeNet on MIT67-indoor dataset

The Fig. 7 shows that the 67-dimensional(d) vector is the output of the last FC layer. The theory is to make the best of a higher dimensional feature vectors which will result in a better performance. The probabilistic vector is the output of this probabilistic layer which

we are interested in treating as the coefficient of fusion with other models' feature vectors and the GP vector can be obtained by computing the product of the corresponding feature vector element wise. The production process of AP, VP vector is the same as the GP vector.

During the preparation period, we train AlexNet, GoogLeNet, and VGG16 on the benchmark scene recognition datasets respectively to obtain each model's scene recognition accuracy. After receiving the accuracy of each single deep CNN model, we extract features from the last FC layer and probabilistic layer as candidate layers. We are prior to fuse and fine-tune these layers. The probabilistic layer is the output of SoftMax layer in CNN which is a vector stands for probability of each class. We treat it as the pixel-level coefficient of fusion with other models' feature vectors. A fusion function f: $V_I \rightarrow V_O$ fuses features:

$$\mathrm{V}_O = P_1 V_{I1} \oplus_1 P_2 V_{I2} \oplus_2 \ldots \oplus_n P_n V_{In} \quad (1)$$

where V_0 denotes the vector after fusion and V_{In} is extracted from the candidate pool. *P* is the coefficient of fusion came from the probabilistic layer, and *n* is the number of fusing CNN model's features. "$\oplus$" is the fusion strategy that will be described as the following.

We fuse 2 or 3 layers from this candidate pool. Finally, they are applied to some large-scale scene datasets by SoftMax classifier.

2.2.1 Input images & preprocessing

Because the size of scene images varies, we resize them to 256*256, which is later cropped to 224*224 in CNN. Furthermore, the spatial layout information of a scene image is vital for recognition. The DCNN can hardly obtain satisfied recognition results if ignoring the spatial information. In this paper, we split scene images into two parts equally in vertical direction, which can reduce the redundancy between subjects and prevent the over-fitting in training.

2.2.2 Feature transformation

As shown in Fig. 2 the high -level CNN features fusion through distinct hidden layers which are correlated with single CNN features. The input of this hidden layer is modified by rectilinear units (ReLU), which create sparse representation, decreasing computational work. Besides, it is a non-linear activation function, and the sparse representation shows the bionic characteristics. And it can avoid the gradient explosion and gradient disappearance. Hence, the particular CNN feature space X_i associated with the single-hidden layer can be rectified from the dimensionality *N* to *K* as

$$V_{i,K} = \sum_{n=1}^{N} \omega_{i,k,n} \bullet X_{i,n} + b_{i,k} \quad (2)$$

where *N* donates the dimension of i^{th} CNN featureX_i, $k \in \{1, \ldots, K\}$, *K* is the number of classes, $\omega$and *b* are the weight matrix and bias vector respectively.

2.2.3 Fusion strategy

To make the description more comprehensible, we now propose several methods to fuse the 3 extracted features. The function$f^*: V_1, V_2, V_3 \rightarrow V^*$ means the fusion strategy which fuses the three feature vectors, $V_1 \in \mathbb{R}^{H^1 \times W^1 \times D^1}$, $V_2 \in \mathbb{R}^{H^2 \times W^2 \times D^2}$, and $V_3 \in$

$\mathbb{R}^{H^3 \times W^3 \times D^3}$ into one feature vector $V^* \in R^{(H^* \times W^* \times D^*)}$. And W is the abbreviation of width, H means the height for short and D shows the number of the feature vectors channels respectively. In this paper, $H^1 = H^2 = H^3 = 1$, $D^1 = D^2 = D^3 = 1$, and $W^1 = W^2 = W^3$ =number of classes of scene datasets respective. So the fusion strategy we used in this paper is as follows.

(1) Product fusion: $V^{prod} = f^{prod}(V_1, V_2, V_3)$calculates the product of the 3 feature maps.

$$V^{prod} = \prod_{i=1}^{3} V_i$$

(2) Sum fusion: $V^{sum} = f^{sum}(V_1, V_2, V_3)$calculates the sum of the 3 feature maps.

$$V^{sum} = \sum_{i=1}^{3} V_i \quad (4)$$

(3) Max fusion: $V^{max} = f^{max}(V_1, V_2, V_3)$chooses the largest values among the feature maps.

$$V^{max} = \arg max(V_1, V_2, V_3) \quad (5)$$

(4) Average fusion: $V^{mean} = f^{mean}(V_1, V_2, V_3)$the sum of the three feature maps divided by three.

$$V^{mean} = \arg mean(V_1, V_2, V_3) \quad (6)$$

(5) Concatenation fusion: $V^{cat} = f^{cat}(V_1, V_2, V_3)$stacks the three feature maps across the feature channels d.

$$V_{2d}^{cat} = V_{(1,d)}, V_{(2d-1)}^{cat} = V_{(2,d)} \quad (7)$$

2.2.4 Classification

SoftMax is applied to a N-way classified function, which input the logits function and output the probability p of each class in vector Y^* to a vector of K elements in [0,1], shown as

$$\rho(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}}, \quad for\ j = \{1, \dots, K\} \quad (8)$$

where Z is the input vector with the dimensional K same as the output where the sum of all real values is 1. K is the number of different class in the scene datasets. We can use the largest probability to achieve the mean accuracy of the fused model. Beyond that, cross entropy (CE) loss computes the distance between predicted label and ground-truth label. We can denote the loss function of cross-entropy (CE) as below, which use the parameter φ and in the form of likelihood maximization.

$$\text{argmax } \mathcal{L}(\varphi|t, z) \quad (9)$$

where by the conditional distribution whose target is t, and input is z with the same parameter φ, we can get likelihood $\mathcal{L}$.

$$P(t, z|\varphi) = P((t|z,\varphi)P(z|\varphi) \quad (10)$$

where P(t,j|z) means the probability of the input z belong to class j. We can minimize the negative log-likelihood to get the cost estimation in the likelihood maximization, as

$$\xi(t, z) = -log\ \mathcal{L}(\theta|t, z) \quad (11)$$

where ξ means the CE loss function. Besides, we can use the predicted class probability's derivative $\partial\, \xi / \partial\, \omega$ of the cost function for the weight-update, as

$$\omega(t+1) = \omega(t) - \lambda \frac{\partial \xi}{\partial \omega(t)} \tag{12}$$

3 Experimental results and discussions

In this part, we will investigate proposed algorithm which fuses features exacted from different 5CNN models by 3 publicly available scene datasets: Scene15, MIT67-indoor and Eight sports.

3.1 Setup and implementation

3.1.1 Datasets

The Scene15 dataset [Li and Perona (2005)] has 15 natural scene categories, each of which has 216 to 400 images, 100 images per category are randomly selected out as the training set. With respect to the eight sports event categories dataset [Li and Li (2007)], it contains 8 sports event categories: rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock climbing, with the image number from 137 to 250 of each category. The MIT67 dataset [Liu, Liu, Wang et al. (2015)] has 67 Indoor categories containing 15620 jpg images totally. 80 images per category are used as training samples while the remained 20 images as testing samples. Fig. 8 shows four samples from target scene dataset respectively.

(a) Scene15 (b) MIT67-indoor (c) Eight sports

Figure 8: Four samples from target scene datasets

3.1.2 Initialization

We use Caffe [Jia, Shelhamer, Donahue et al. (2014)] to do experiments, which is a popular framework in using CNN in image processing. The batch size is 10, and the initial learning rate is set as 0.001 which is reduced at every iteration by the "poly" method. Moreover, to avoid the over-fitting we set a momentum as 0.9 to avoid local optimum, meanwhile, the weight decay, Dropout ratio, and iteration time are set as 0.005, 0.5, and 8000, respectively.

We set accuracy as the evaluation criterion because mean average precision (mAP) are more fit for image retrieval and accuracy show the results directly.

3.2 Experimental results

In order to ensure our experimental results are accurate, we repeat each experiment five times and take the average as the statistical illustration. The Tab. 1-1 shows the accuracy of our individual deep CNN model on scene datasets respectively. We can observe that GoogLeNet achieved the highest accuracy with 93.12%, 65.97% and 95.85% on the 3 datasets of Scene15, MIT67-indoor and Eight sports, respectively. GoogLeNet and VGG-16 provide almost similar accuracy, and the result of GoogLeNet is slightly higher.

Table 1-1: The accuracy of base model on Scene15 datasets (%)

Model	Alex Net	GoogLeNet	VGG16
Scene15	85.80	**93.12**	93.01
MIT67-indoor	56.33	**65.97**	65.40
Eight sports	91.82	**95.85**	95.50

(*The bolded values are marked as the best results.)

The Tab. 1-2 shows the accuracy of using high-level layers with different weight. The highest accuracy with GP is higher than the GoogLeNet on the Scene15 and MIT67-indoor with 94.13% and 67.20%, which is improved 1.01% and 1.23% respectively. The highest accuracy on Eight sports with VP is approximately the same with GoogLeNet, and in this datasets GP and VP achieve similar accuracy.

Table 1-2: The accuracy of using high-level (last fc) layers with different weight (%)

Model	AP	GP	VP
Scene15	90.25	**94.13**	91.33
MIT67	56.00	**67.20**	66.13
Eight sports	90.90	95.35	**95.80**

"P" stands for the probability vector obtained from the probabilistic (prob) layer of the VGG-16(V), GoogLeNet(G), and AlexNet(A) models, and the same naming are used over this paper.

Tabs. 2-5 show the scene recognition accuracy of our fusion strategy, where the target domain has overlap with source domain in testing. "15", "67" and "8" refer to 15-d layer, 67-d layer and 8-d layer. Hence, "V15", "G15", "A15" stand for the features from the 15-d layers of the VGG-16 (V), GoogLeNet (G), and AlexNet (A) models, and the rest can be done in the same manner

Table 2-1: The accuracy of fusion model on Scene15 (V+G, %)

Model	Sum	Max	Average	Concatenation
VP+G15	95.56	95.33	95.56	**95.67**
VP+GP	93.36	93.37	93.36	93.63
V15+GP	93.43	93.46	93.43	91.69
V15+G15	95.52	94.59	95.52	94.76

Table 2-2: The accuracy of fusion model on MIT67-indoor (V+G, %)

Model	Sum	Max	Average	Concatenation
VP+G67	64.27	65.33	64.27	62.67
VP+GP	66.40	66.47	66.40	**67.20**
V67+GP	64.73	64.60	64.73	64.60
V67+G67	63.33	64.60	63.33	64.60

Table 2-3: The accuracy of fusion model on Eights sports (V+G, %)

Model	Sum	Max	Average	Concatenation
VP+G8	94.60	94.40	94.60	94.20
VP+GP	94.20	94.40	94.20	94.40
V8+GP	**95.20**	**95.20**	**95.20**	**95.20**
V8+G8	95.00	**95.20**	95.00	**95.20**

Table 3-1: The accuracy of fusion model on Scene15 (V+A, %)

Model	Sum	Max	Average	Concatenation
VP+A15	91.93	92.20	91.93	94.21
VP+AP	89.73	82.97	89.73	85.90
V15+AP	87.20	89.46	87.20	77.20
V15+A15	**95.59**	91.49	90.59	91.87

Table 3-2: The accuracy of fusion model on MIT67-indoor (V+A, %)

Model	Sum	Max	Average	Concatenation
VP+A67	66.27	59.20	66.27	65.07
VP+AP	63.80	61.80	63.80	63.73
V67+AP	**69.53**	67.73	**69.53**	68.87
V67+A67	68.07	61.20	68.07	69.46

Table 3-3: The accuracy of fusion model on Eights sports (V+A, %)

Model	Sum	Max	Average	Concatenation
VP+A8	92.07	94.24	92.07	94.62
VP+AP	96.24	95.84	96.24	95.69
V8+AP	95.98	96.32	95.98	**96.59**
V8+A8	96.52	95.94	96.52	96.28

Table 4-1: The accuracy of fusion model on Scene15 (G+A, %)

Model	Sum	Max	Average	Concatenation
GP+A15	94.87	95.09	94.87	94.49
GP+AP	94.37	95.23	94.37	94.19
G15+AP	94.56	94.70	94.54	**95.85**
G15+A15	92.02	91.85	92.02	94.02

Table 4-2: The accuracy of fusion model on MIT67-indoor (G+A, %)

Model	Sum	Max	Average	Concatenation
GP+A67	56.47	56.13	56.47	57.27
GP+AP	65.36	62.23	64.36	67.26
G67+AP	65.24	68.56	66.46	**69.42**
G67+A67	68.56	67.68	64.25	66.25

Table 4-3: The accuracy of fusion model on Eights sports (G+A, %)

Model	Sum	Max	Average	Concatenation
GP+A8	95.73	97.73	95.76	**98.29**
GP+AP	97.50	97.56	97.49	97.13
G8+AP	97.49	96.49	97.49	97.56
G8+A8	96.25	96.12	96.25	95.64

Table 5-1: The accuracy of fusion model on Scene15 (V+G+A, %)

Model	Sum	Max	Average	Concatenation
VP+GP+A15	94.40	94.73	94.40	95.20
GP+AP+V15	**96.85**	94.69	96.39	94.80
VP+A15+G15	93.86	93.89	93.86	93.73
GP+V15+A15	94.29	93.27	94.29	91.79
G15+V15+A15	94.39	94.20	94.39	93.47

Table 5-2: The accuracy of fusion model on MIT67-indoor (V+G+A, %)

Model	Sum	Max	Average	Concatenation
VP+GP+A67	66.82	65.86	66.82	66.24
GP+AP+V67	65.87	66.58	65.87	**70.46**
VP+A67+G67	65.24	66.48	65.24	66.84
GP+V67+A67	68.56	67.24	68.56	68.54
G67+V67+67	66.76	66.56	66.76	67.28

Table 5-3: The accuracy of fusion model on Eights sports (V+G+A, %)

Model	Sum	Max	Average	Concatenation
VP+GP+A8	97.88	97.66	97.88	**98.56**
GP+AP+V8	96.25	96.56	96.25	96.54
VP+A8+G8	97.48	97.26	97.48	97.22
GP+V8+A8	97.84	97.88	97.84	97.44
G8+V8+A8	96.85	96.25	96.85	96.23

Besides the above experiments, we propose to use the 100-d FC layers features.

3.3 Discussions

From the fusion results on Tabs. 2-5, we can see that the highest layer combination accuracy on Scene15 has been increased from 93.12% to 96.85% (GP+AP+V15); the highest layer combination accuracy on MIT67-indoor has been increased from 65.97% to 70.46% (GP+AP+V67); the highest layer combination accuracy on Eight sports has been increased from 95.85% to 98.56% (VP+GP+A8). Note that these highest results of scene recognition is a fusion the probabilistic layers, if we look at Tabs. 2-5, VP, GP play an important role in the individual and achieve the best fusion results for VGG-16 and GooLeNet fusion model. Because of the linearity of the features, the accuracy of sum and mean are equal. As shown in Tabs. 6-8, we can know that this pattern fusion cannot make a better performance in fusion model.

Table 6: The accuracy of fusion model on Scene15 (V+A, %)

Model	Sum	Max	Average	Concatenation
V100+A100	89.84	91.46	89.46	**94.18**
V100+AP	92.24	92.89	92.24	93.22
VP+AP	92.45	93.20	92.45	92.85

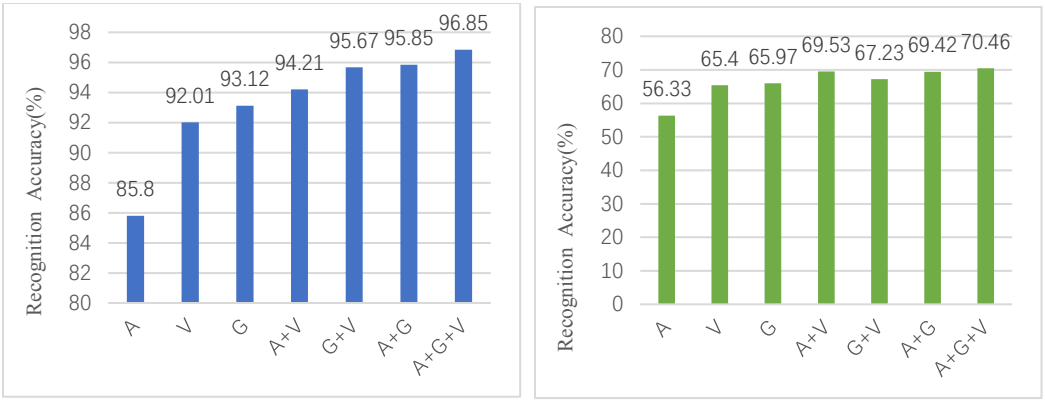
"100" refers to 100-d layer.

Table 7: The accuracy of fusion model on MIT67-indoor (G+A, %)

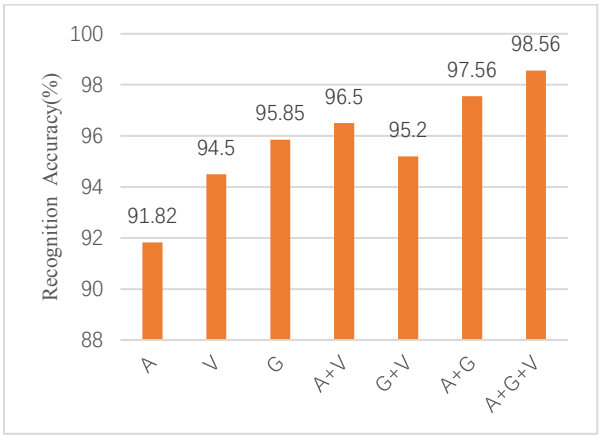
Model	Sum	Max	Average	Concatenation
G100+A100	66.24	67.21	66.24	66.88
G100+AP	65.46	66.42	65.46	**68.26**
VP+AP	67.01	66.85	67.01	66.42

Table 8: The accuracy of fusion model on Eight sports (V+G, %)

Model	Sum	Max	Average	Concatenation
V100+G100	95.88	95.84	95.88	95.24
V100+GP	96.21	96.76	96.21	97.21
VP+GP	96.04	97.26	96.04	**96.46**



(a) Scene15 (b) MIT67-indoor



(c) Eight sports

Figure 9: Accuracy (%) comparison between individual models with fusion models

Fig. 9 shows three single models (A, V, and G), the fusion of two models (A+G, A+V, and V+G), and the fusion of three models (A+V+G) accuracy results respectively. The algorithm by fusing all 3 features improves the accuracy of scene recognition almost by 2.7%~4.7%, which obtains the better results compared with other state-of-the-arts algorithms as shown in Tabs. 9-11.

The values shown here are same with bolded values in Tabs. 2-8.

Table 9: Comparison our proposed algorithm with other state-of-the-arts on Scene15

Methods	Accuracy (%)
M-ADDL [Zheng, Yi, Qi et al. (2018)]	80.40
SIFT+SPM [Lazebnik, Schmid and Ponce (2006)]	81.42
HOG+SPM3 [Xie, Lee, Liu et al. (2018)]	83.30
SIFT+ORSP [Jiang, Yuan and Yu (2012)]	83.93

HIK [Wu and Rehg (2009)]	84.15
Convnets+SVM [Zhou, Lapedriza, Xiao et al. (2014)]	84.26
SIFT+BRSP [Jiang, Yuan and Yu (2012)]	88.12
Convnets+RSP [Wu and Rehg (2009)]	89.48
LScSPM [Gao, Tsang, Chia et al. (2010)]	89.74
Place CNN [Zhou, Lapedriza, Xiao et al. (2014)]	90.19
G-MS2F [Tang, Wang and Kwong (2017)]	92.90
DUCA [Khan, Hayat, Bennamoun et al. (2016)]	94.50
NNSD+ICLC [Xie, Lee, Liu et al. (2019)]	95.1
Objectness [Cheng, Lu, Feng et al. (2018)]	95.80
SDO [Cheng, Lu, Feng et al. (2018)]	95.88
Our approach	**96.85**

Table 10: Comparison between the proposed algorithm and other state-of-the-arts on Eight sports

Methods	**Accuracy（%）**
ScSPM [Yang, Yu, Gong et al. (2009)]	79.06
LLC+SPM [Cheng, Lu, Feng et al. (2018)]	83.02
Wu et al. [Wu and Rehg (2009)]	84.38
HOG+SPM3 [Xie, Lee, Liu et al. (2018)]	84.88
BOWL+SVM [Banerji, Sinha and Liu (2013)]	87.72
Place CNN [Zhou, Lapedriza, Xiao et al. (2014)]	94.42
S^2ICA [Hayat, Khan, Bennamoun et al. (2016)]	95.80
Our approach	**98.56**

Table 11: Comparison between the proposed algorithm and other state-of-the-arts on MIT67-indoor

Methods	**Accuracy（%）**
RBoW [Parizi, Oberlin and Felzenszwalb (2012)]	37.93
DPM+GIST+SP [Pandey and Lazebnik (2011)]	43.10
D-Parts [Banerji, Sinha and Liu (2013)]	51.42
Convnets+FC	58.08
Convnets +SVM [Wu and Rehg (2009)]	58.45
IFV [Cheng, Lu, Feng et al. (2018)]	60.81
MLrep [Yang, Yu, Gong et al. (2009)]	64.03
CFA [Sun, Li, Liu et al. (2019)]	67.31
Our approach	**70.46**

4 Conclusions

In this paper, we have proposed an advanced fusion framework based on multiple DCNN for scene recognition. Experimental results using publicly available datasets demonstrate that proposed algorithm achieves higher recognition performance than the single model. Furthermore, three single CNN models increase the performance than fused two models. These fusion methods take advantage of complementary of multi-CNN models to get the features which are more in common use and can distinguish the fine-grained difference.

The future work is to make the research that applies this method to other many tasks like medical image classification. Besides that, it can be an interesting work that adding another deeper model such as ResNet is whether to have a better performance further.

Conflicts of Interest: The authors declare that we have no conflicts of interest to report regarding the present study.

References

Arora, S.; Bhaskara, A.; Ge, R.; Ma, T. (2014): Provable bounds for learning some deep representations. *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no.1, pp. 584-592.

Banerji, S.; Sinha, A.; Liu, C. J. (2013): A new bag of words LBP (BoWL) descriptor for scene image classification. *Lecture Notes in Computer Science*, vol. 8047, pp. 490-497.

Cheng, X. J.; Lu, J. W.; Feng, J. J.; Yuan, B.; Zhou, J. (2018): Scene recognition with objectness. *Pattern Recognition*, vol. 74, pp. 474-487.

Cui, Y.; Zhou, F.; Wang, J.; Liu, X.; Lin, Y. Q. et al. (2017): Kernel pooling for convolutional neural networks. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921-2930.

Gao, S. H.; Tsang, I. W. H.; Chia, L. T.; Zhao, P. L. (2010): Local features are not lonely-Laplacian sparse coding for image classification. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3555-3561.

Guo, S.; Huang, W. L.; Wang, L. M.; Qiao, Y. (2017): Locally supervised deep hybrid model for scene recognition. *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 808-820.

Hayat, M.; Khan, S. H.; Bennamoun, M.; An, S. (2016): A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4829-4841.

Hinton, G. E.; Osindero, S.; Teh, Y. W. (2006): A fast learning algorithm for deep belief nets. *Neural Computation*, vol. 18, no. 7, pp. 1527-1554.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J. et al. (2014): Caffe: convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675-678.

Jiang, Y. N.; Yuan, J. S.; Yu, G. (2012): Randomized spatial partition for scene recognition. *Proceeding of European Conference on Computer Vision*, pp. 730-743.

Khan, S. H.; Hayat, M.; Bennamoun, M.; Togneri, R.; Sohel, F. A. (2016): A

discriminative representation of convolutional features for indoor scene recognition. *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3372-3383.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): ImageNet classification with deep convolutional neural networks. *Proceedings of the NIPS*, pp. 1097-1105.

Lavinia, Y.; Vo, H. H.; Verma, A. (2016): Fusion based deep CNN for improved large-scale image action recognition. *2016 IEEE International Symposium on Multimedia*, pp. 609-614.

Lazebnik, S.; Schmid, C.; Ponce, J. (2006): Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169-2178.

LeCun, Y.; Bengio, Y.; Hinton, G. (2015): Deep learning. *Nature*, vol. 521, pp. 436-444.

Lee, H.; Pham, P.; Largman, Y.; Ng, A. Y. (2009): Unsupervised feature learning for audio classification using convolutional deep belief networks. *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pp. 1096-1104.

Li, F. F.; Perona, P. (2005): A Bayesian hierarchical model for learning natural scene categories. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 524-531.

Li, L. J.; Li, F. F. (2007): What, where and who? Classifying events by scene and object recognition. *Proceeding of IEEE 11th International Conference on Computer Vision*, pp. 1-8.

Lin, M.; Chen, Q.; Yan, S. (2013): Network in network. arXiv: 1312.4400v3.

Liu, B. Y.; Liu, J.; Wang, J. Q.; Lu, H. Q. (2015): Learning a representative and discriminative part model with deep convolutional features for scene recognition. *Proceeding of the Asian Conference on Computer Vision*, pp. 643-658.

Liu, W.; Ma, X.; Zhou, Y.; Tao, D.; Cheng, J. (2019): p-Laplacian regularization for scene recognition. *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2927-2940.

Liu, Y.; Chen, Q.; Chen, W.; Wassell, I. (2018). Dictionary learning inspired deep network for scene recognition. *Proceeding of the 32nd AAAI Conference on Artificial Intelligence*, pp. 7178-7185.

Lowe, D. G. (2004): Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110.

Oliva, A.; Torralba, A. (2001): Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145-175.

Pandey, M.; Lazebnik, S. (2011): Scene recognition and weakly supervised object localization with deformable part-based models. *Proceeding of the IEEE International Conference on Computer Vision*, pp.1307-1314.

Parizi, S. N.; Oberlin, J. G.; Felzenszwalb, P. F. (2012): Reconfigurable models for scene recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2755-2782.

Razavian, A. S.; Azizpour, H.; Sullivan, J.; Carlsson, S. (2014): CNN features off-the-

shelf: an astounding baseline for recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 806-913.

Salakhutdinov, R.; Hinton, G. (2012): An efficient learning procedure for deep Boltzmann machines. *Neural Computation*, vol. 24, no. 8, pp. 1967-2006.

Seong, H.; Hyun, J.; Kim, E. (2019). Fosnet: an end-to-end trainable deep neural network for scene recognition. arXiv:1907.07570v2.

Sun, N.; Li, W.; Liu, J.; Han, G.; Wu, C. (2019): Fusing object semantics and deep appearance features for scene recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1715-1728.

Szegedy, C.; Liu, W.; Jia, Y. Q.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Tang, P.; Wang, H.; Kwong, S. (2017): G-ms2f: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing*, vol. 225, pp.188-197.

Wang, J. J.; Yang, J. C.; Yu, K.; Lv, F. J.; Huang, T. et al. (2010): Locality-constrained linear coding for image classification. *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360-3367.

Wu, J. X.; Rehg, J. M. (2009): Beyond the Euclidean distance: creating effective visual codebooks using the histogram intersection kernel. *Proceeding of the 12th International Conference on Computer Vision*, pp. 630-637.

Wu, J. X.; Rehg, J. M. (2011): CENTRIST: a visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489-1501.

Xie, L.; Lee, F.; Liu, L.; Yin, Z.; Chen, Q. (2019): Hierarchical coding of convolutional features for scene recognition. *IEEE Transactions on Multimedia (Early Access)*.

Xie, L.; Lee, F.; Liu, L.; Yin, Z.; Yan, Y. et al. (2018): Improved spatial pyramid matching for scene recognition. *Pattern Recognition*, vol. 82, pp. 118-129.

Yang, J. C.; Yu, K.; Gong, Y. H.; Huang, T. (2009): Linear spatial pyramid matching using sparse coding for image classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1794-1801.

Yosinski, J.; Clune, J.; Bengio, Y, Lipson H. (2014): How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27*.

Zhao, Z.; Larson, M. (2018): From volcano to toyshop: adaptive discriminative region discovery for scene recognition. arXiv:1807.08624.

Zheng, C.; Yi, Y.; Qi, M.; Liu, F.; Bi, C. et al. (2018): Multicriteria-based active discriminative dictionary learning for scene recognition. *IEEE Access*, vol. 6, pp. 4416-4426.

Zhou, B. L.; Lapedriza, A.; Xiao, J. X.; Totralba, A.; Oliva, A. (2014): Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems 27.*