IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2020, accepted April 18, 2020, date of publication April 23, 2020, date of current version May 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989863

FOSNet: An End-to-End Trainable Deep Neural Network for Scene Recognition

HONGJE SEONG, JUNHYUK HYUN, (Member, IEEE), AND EUNTAI KIM, (Member, IEEE)
School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Euntai Kim (etkim@yonsei.ac.kr)

This work was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT under Grant NRF-2017M3C4A7069370.

ABSTRACT Scene recognition is a kind of image recognition problems which is aimed at predicting the category of the place at which the image is taken. In this paper, a new scene recognition method using the convolutional neural network (CNN) is proposed. The proposed method is based on the fusion of the object and the scene information in the given image and the CNN framework is named as FOS (fusion of object and scene) Net. To combine the object and the scene information effectively, a new fusion framework named CCG (correlative context gating) is proposed. In addition, a new loss named scene coherence loss (SCL) is developed to train the FOSNet and to improve the scene recognition performance. The proposed SCL is based on the idea that the scene class does not change all over the image. The proposed FOSNet was experimented with three most popular scene recognition datasets, and their state-of-the-art performance is obtained in two sets: 60.14% on Places 2 and 90.30% on MIT indoor 67. The second highest performance of 77.28% is obtained on SUN 397.

INDEX TERMS Scene recognition, convolutional neural network, fusion network, scene coherence, end-to-end trainable.

I. INTRODUCTION

Scene recognition is one of the most spotlighted topics in image recognition, applied to image retrieval, autonomous robot, and drone. Many studies have explored the scene recognition. Most of the early studies, however, have a drawback that they consider scene recognition as a simple image recognition problem; and they applied the general CNN or image recognition methods to scene recognition [1]–[4].

In the last few years, some studies have used the scene image traits to improve scene recognition. In particular, the scene image traits that a scene image consists of a combination of several objects and the objects in the image possesses much information about the category of the scene were used in [5]–[8]. For example, MetaObject-CNN was developed in [8]. In the paper, a region proposal technique developed to generate a set of discriminative patches potentially containing objects for scene recognition. Multi-scale CNN architectures were developed to reduce scale bias between scene dataset and object dataset in [5] and [6]. Discriminative objects which frequently appears in scene images were selected using the posterior probability of scene images in [7]. A high-level deep representation of objects was extracted using YOLOv2 [9] for scene recognition in [10].

Unfortunately, however, we believe that most of the existing methods do not fully exploit the valuable traits of scene images for scene recognition: (P1) First, previous fusion method which combines object and the scene information are ineffective: Most of the previous works focused on the extraction of object and scene features, and how to combine these two kinds of information effectively is not fully addressed in the previous works. The two kinds of information were simply fused by summation or concatenation. The domain difference between object and scene is not taken into consideration. (P2) Second, the standard cross-entropy loss function is not enough for scene recognition, since scene recognition is quite different from general image recognition: A scene spreads all over the image, and the class of the scene does not change over the entire image. This is contrary in the object images, where an object appears only at specific locations in an image, as shown in Fig. 1. In the figure, we visualize the objectness or sceneness which is a region where the main object or scene appears, respectively. Thus, the classes of the objects change from patch to patch in the same image. This

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang.

trait should be reflected in the loss function for the scene recognition.

In this paper, a new scene recognition framework is proposed. Then proposed network is named as FOSNet since it is based on the effective fusion of the object and the scene information in the given image. To solve the problem (P1), a new object-scene fusion framework named correlative context gating (CCG) is developed. The CCG combines the object and scene information effectively by matching the domains of two kinds of information and applying the notion of attention to the fusion. To solve the problem (P2), a scene coherence loss (SCL) is developed to train the FOSNet. This SCL is based on the idea that sceneness spreads all over the image, and the class of the scene does not not change from patch to patch in a given image.

The contributions of FOSNet are as follows:

1) A new fusion framework named CCG is proposed to combine the object and scene features from the image. Unlike the previous fusion methods in which the two features are simply concatenated and the classifier is designed for the features, the CCG selects important features and fuses the two sets of features effectively for training.
2) The traits of scene coherence (SC) in a scene image are defined, and a new loss SCL is developed based on the trait. The SCL is the first loss specialized for scene recognition.

The rest of the paper is organized as follows: Section II provides a brief review of the related studies. Section III explained FOSNet in detail. Section IV applies the FOSNet to three benchmark problems, and the performance of the FOSNet is demonstrated through experimentation. Section V conducts some ablation study to verify the value of our proposed SCL and CCG. Section VI concludes the paper.

II. RELATED WORKS

In this section, we review previous works on scene recognition with an emphasis on (1) a combination of the object and scene information, and (2) the application of other scene traits.

A. OBJECT-SCENE FUSION

Using object information in an image is the most utilized scene traits for scene recognition [5]–[8]. When a particular object appears in an image, the chance of the image belonging to a certain category associated with the object increases. For example, if a TV is detected, the chance of the image being in a living room increases. In the previous works, the object features were used for scene recognition instead of detecting the objects directly. To extract the object features, large image datasets for object recognition are used and they are ImageNet [13], PASCAL visual object classes (VOC) [14], or Microsoft COCO [15].

In order to combine scene and object information for scene recognition, an effective fusion framework is of great importance; and several fusion methods have been reported. For example, the two features extracted by two different CNNs were combined at feature level by summing or concatenating the features in [5]–[8], [16]–[23]. Then, the classical classifiers such as support vector machine (SVM) [24] was applied to the fused features. Unfortunately, these methods have some drawbacks that they cannot be trained in an end-to-end manner. Moreover, the simple summation or concatenation might degrade the recognition performance owing to redundancy in two feature sets.

In this paper, a new fusion method named correlative context gating (CCG) is proposed. The CCG is an extended version of our previous work CCM [25]. In the CCG, a relationship between object and scene is trained, and the domains of two kinds of information are matched by converting the object domain into scene domain. By doing so, the two kinds of information in different domains are transformed into the same domain, and the fusion is performed through the element-wise multiplication. Further, the attention is applied to the fusion and a new scene feature which attends to the object is generated.

B. OTHER TRAITS IN SCENE IMAGE

Other scene traits were also used in previous studies: The analysis of object scales in the scene images was utilized in [5] and [6]. In [7], [8], [19]–[21], the number of CNN input patches was adjusted by considering several objects in the scene image. To capture recurring visual elements and salient objects in scene recognition, the deformable part-based model (DPM) was utilized in [30]. In addition, the traits that features appearing in each image region within scene images are all similar was used in [31]. A super category was proposed to solve the problem that the scene categories have label ambiguity in [17]. A deep gaze shifting kernel was developed to distinguish sceneries from different categories in [19]. As such, the traits of the scene image are very diverse, and there seem to be still many available unused scene traits in scene recognition studies. In this work, another scene trait is used to improve the scene recognition performance. Unlike objectness which appears on specific parts of the image in general image recognition problem, whereas sceneness spreads all over an image in scene recognition problem. This trait is named as scene coherence (SC) in this paper. In Table 1, the previous and proposed methods for scene recognition which use deep convolutional neural networks are compared.

III. PROPOSED METHODS

In this section, a new scene recognition network named FOSNet is proposed. The overall FOSNet structure is shown in Fig. 2. As shown in the figure, FOSNet has two input streams. In the upper stream named ObjectNet, the features of the objects in the scene images are extracted. In the lower stream named PlacesNet, scene features are extracted. In a trainable fusion module, two streams of features are fused into a combined feature for scene recognition. The FOSNet

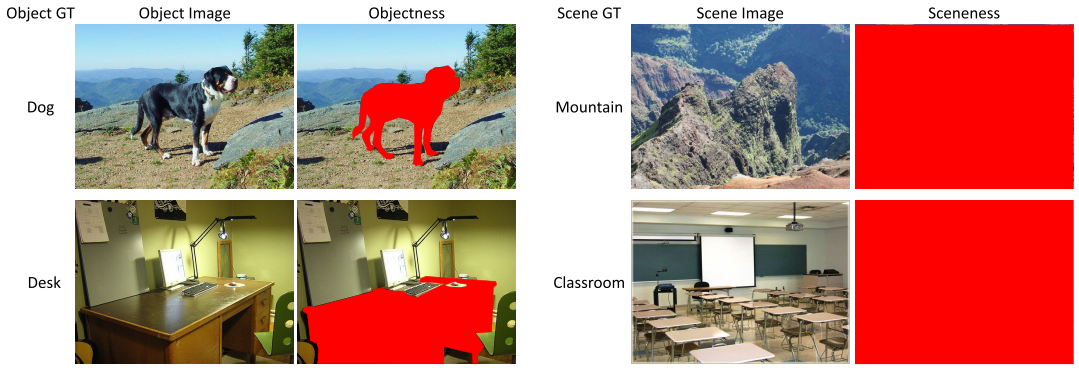


FIGURE 1. Object recognition vs. scene recognition: In object recognition, objectness indicated in red focuses on specific parts of the image, whereas sceneness indicated in red spreads all over a scene image. The object and scene images are taken from ImageNet [11] and Places 2 [12], respectively.

TABLE 1. Comparison between our works and existing methods for scene recognition using deep convolutional neural networks.

Category	Method	Strength	Weakness
Intra-class variations	Setting super category [17]	Can improve scene recognition performance without any additional prediction time.	Have a heavy training process to train a teacher network.
Human-centric analysis	Important patches extraction using gaze shifting [19]	Have a high interpretability.	Have a low performance.
Discriminative region detection	Multiple patch feature extraction [19]–[21], [26]–[29]	Can expect a high performance by analyzing tremendous object patches from a single image.	Take extremely long prediction time because all of the detected or sliding window patches should be passed to CNN independently.
Using object information	Object features extraction from multiple patches [5]–[8]		Using two or more independent deep CNNs.
	Feature fusion method of object and scene (ours)	Can adapt to any object feature extraction method.	Using two independent deep CNNs.
Scene coherence	Formulating loss about scene coherence (ours)	Can improve scene recognition performance without requiring any model changes, additional prediction time, or additional module.	Need to find a hyper-parameter about the ratio between classification loss and scene coherence loss.

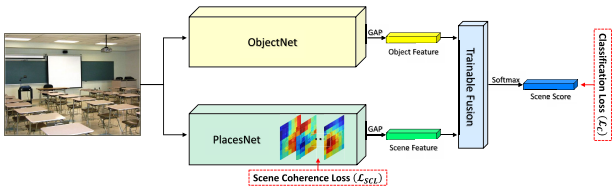


FIGURE 2. An overall architecture of FOSNet.

consists of ObjectNet, PlacesNet, and trainable fusion modules, and all networks can be trained in an end-to-end manner. The three subnets are explained in detail in the subsequent subsections.

A. ObjectNet

Based on the scene traits (P1), information about the objects that appear in the scene is exploited in FOSNet. To obtain a highly discriminating object descriptor, ObjectNet is utilized in the upper stream of Fig. 2 to extract a feature of the objects in a scene image. As the ObjectNet, the popular CNN models [3], [32]–[34] pre-trained on ImageNet [13] are used, as shown in Fig. 3. An object feature extracted through ObjectNet is fed into the trainable fusion module. In the structure given in Fig. 3, not only the object feature but also the object score can be fed into the trainable fusion module. Detailed description of the fusion level is given in Section III-C.

B. PlacesNet

PlacesNet is another CNN model and it extracts a scene feature from an image. The PlacesNet is pre-trained using Places 2 [12] and its structure is the same as that of the ObjectNet, as shown in Fig. 4(a). To train the PlacesNet,

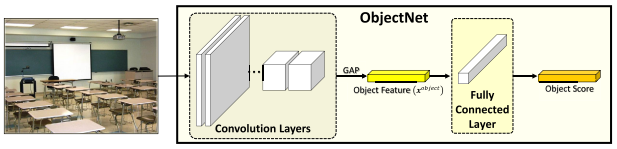


FIGURE 3. Structure of ObjectNet.

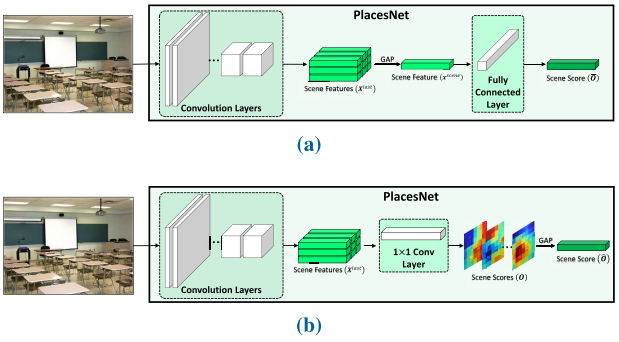


FIGURE 4. Structures of PlacesNet. (a) Vanilla CNN structure; PlacesNet should be applied multiple times to compute the SCL. (b) A new structure in which SCL can be computed by applying PlacesNet only once.

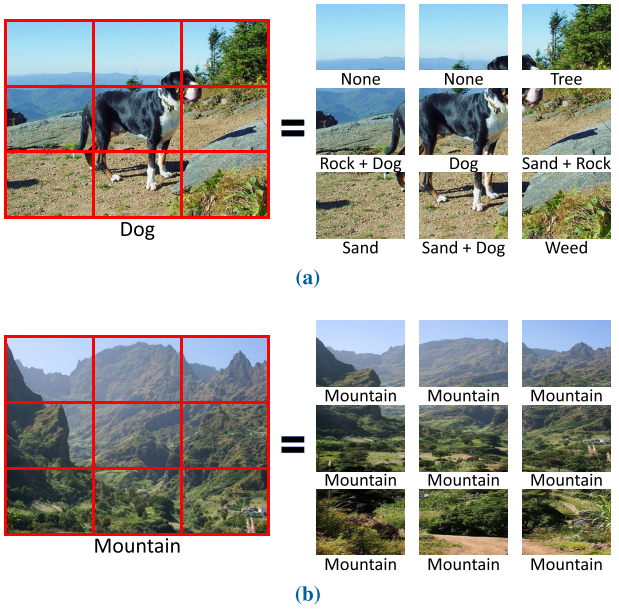


FIGURE 5. (a) Object recognition vs. (b) Scene recognition: In (a) object image, the class of the object can change from patch to patch. In (b) scene image, however, even if a scene image is divided into multiple grids, each grid cell represents the same class of scene, which is scene coherence.

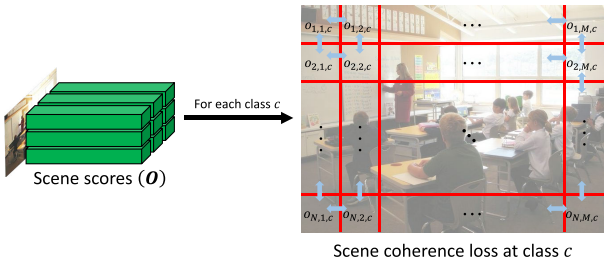


FIGURE 6. Visualization of scene coherence loss (SCL).

scene coherence loss (SCL) is developed in this paper. The SCL is a new loss tailored for scene recognition, and it is based on the scene trait that *objectness focuses on specific parts of an image, whereas sceneness it unfocused on specific parts but it spreads all over the image. In particular, the class of the scene is unchanged over the image.* This trait is named the coherence in the scene of an image, and the SCL embodies this trait into a single loss. For example, let us consider two images in Fig. 5. In general object recognition problem, an object appears in a specific part of an image and the object class can change from the patch to patch, as shown in Fig. 5(a). In scene recognition problem, however, the scene class is coherent all over the image. When the whole image is divided into nine grids, all nine grids cannot have different scene classes and all of them have the same scene class of a mountain, as shown in Fig. 5(b). This also implies that all of the patches in the scene image should be used in the scene recognition; and the region activated in the class activation map (CAM) [35] should be wider than that in the object recognition.

The scene coherence is a unique trait of the scene image and it is formulated into a new loss SCL:

$$\mathcal{L}_{SCL} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{(N-1)M + N(M-1)} \times \left(\sum_{n=1}^{N-1}\sum_{m=1}^{M}\left(o_{n+1,m,c} - o_{n,m,c}\right)^2 + \sum_{n=1}^{N}\sum_{m=1}^{M-1}\left(o_{n,m+1,c} - o_{n,m,c}\right)^2\right) \quad (1)$$

where N and M are the numbers of grid cells in the vertical and horizontal directions, respectively; C is the number of classes; $o_{n,m,c}$ denotes the classification result for the class c in the grid cell (n, m), as shown in Fig. 6. As stated, the SCL defined in (1) favors the case in which all the grids have the same scene class, whereas it penalizes the case in which the adjacent grids have the different scene classes.

Here, when we apply SCL in (1) to the PlacesNet training, a difficulty arises; The PlacesNet should be applied to $N \times M$ grid cells separately and repeatedly and it leads to the waste of computation time. To resolve this inefficiency, the PlacesNet in Fig. 4(a) is converted into the form of a fully convolutional network, as shown in Fig. 4(b). The conversion is motivated by class activation map (CAM) [35] and it can be applied to any CNNs, in which the last layers are global average pooling (GAP) followed by fully connected (FC) layers. In the PlacesNet, the input image with the size of 224 × 224 is reduced to 7 × 7 feature map after going through five pooling operations in convolutional layers. Then, the scene scores for each 7 × 7 grid cell is obtained by replacing the last GAP-FC sequence with 1 × 1 convolution. Then, a scene score for the

entire image is computed by applying the GAP to the tensors obtained from 1×1 convolution.

Interestingly, it can be shown that the PlacesNet with the GAP followed by FC shown in Fig. 4(a) outputs the same result with the converted version with the 1×1 convolution followed by GAP shown in Fig. 4(b). With slightly relaxed notation, the feature tensor extracted from the PlacesNet in Fig. 4(a) is represented into

$$\begin{aligned}\mathbf{X}^{last} &= \left(x^{last}_{n,m,d}\right) \\ &= \begin{pmatrix} \mathbf{x}^{last}_{1,1,1:D} & \mathbf{x}^{last}_{1,2,1:D} & \cdots & \mathbf{x}^{last}_{1,M,1:D} \\ \mathbf{x}^{last}_{2,1,1:D} & \mathbf{x}^{last}_{2,2,1:D} & \cdots & \mathbf{x}^{last}_{2,M,1:D} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}^{last}_{N,1,1:D} & \mathbf{x}^{last}_{N,1,1:D} & \cdots & \mathbf{x}^{last}_{N,M,1:D} \end{pmatrix} \\ &\in \mathbb{R}^{N\times M\times D} \end{aligned} \tag{2}$$

where $N \times M$ is the feature map size extracted from convolution layers in Fig. 4(a); D is the number of output channels of last convolution layer; $\mathbf{x}^{last}_{n,m,1:D} \in \mathbb{R}^D$ denotes a feature vector at position (n, m) of $\mathbf{X}^{last}$; $1 : D$ in the third axis of $\mathbf{x}^{last}_{n,m,1:D}$, is a collection of all the elements accumulated over D channels and it is actually a vector. Let the trainable parameters $\mathbf{W} = \left(w_{c,d}\right) \in \mathbb{R}^{C\times D}$ and $\mathbf{b} = (b_c) \in \mathbb{R}^C$ be weight and bias, respectively, for the FC layer of the model in Fig. 4(a), and $\overline{\mathbf{O}} \in \mathbb{R}^C$, $\widehat{\mathbf{O}} \in \mathbb{R}^C$ be the classification results of the models in Figs. 4(a) and (b), respectively. Then, we can prove that $\overline{\mathbf{O}}$ and $\widehat{\mathbf{O}}$ are the same by

$$\begin{aligned}\overline{\mathbf{O}} &= \mathrm{FC}\left(\mathrm{GAP}\left(\mathbf{X}^{last}\right), \mathbf{W}, \mathbf{b}\right) \\ &= \mathrm{FC}\left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{x}^{last}_{n,m,1:D}, \mathbf{W}, \mathbf{b}\right) \\ &= \mathbf{W}\left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{x}^{last}_{n,m,1:D}\right) + \mathbf{b} \\ &= \left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\mathbf{W}\mathbf{x}^{last}_{n,m,1:D} + \mathbf{b}\right)\right) \\ &= \left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathrm{Conv}^{1\times 1}\left(\mathbf{X}^{last}, \mathbf{W}, \mathbf{b}\right)_{n,m}\right) \\ &= \mathrm{GAP}\left(\mathrm{Conv}^{1\times 1}\left(\mathbf{X}^{last}, \mathbf{W}, \mathbf{b}\right)\right) \\ &= \widehat{\mathbf{O}} \end{aligned} \tag{3}$$

where

$$\begin{aligned}&\mathrm{Conv}^{1\times 1}\left(\mathbf{X}^{last}, \mathbf{W}, \mathbf{b}\right) \\ &= \begin{pmatrix} \mathbf{W}\mathbf{x}^{last}_{1,1,1:D} + \mathbf{b} & \cdots & \mathbf{W}\mathbf{x}^{last}_{1,M,1:D} + \mathbf{b} \\ \vdots & & \vdots \\ \mathbf{W}\mathbf{x}^{last}_{N,1,1:D} + \mathbf{b} & \cdots & \mathbf{W}\mathbf{x}^{last}_{N,M,1:D} + \mathbf{b} \end{pmatrix} \\ &\in \mathbb{R}^{N\times M\times C} \end{aligned} \tag{4}$$

is a tensor obtained by applying 1×1 convolution with weights $(\mathbf{W}, \mathbf{b})$ to input $\mathbf{X}^{last}$, and it is also the classification results for each grid cell shown in Fig. 4(b). Since $\overline{\mathbf{O}}$ and $\widehat{\mathbf{O}}$ have the same values, the model in Fig. 4(b) performs the same classification as the one in Fig. 4(a) and it has an advantage of obtaining classification results $\mathbf{O} \triangleq \mathrm{Conv}^{1\times 1}\left(\mathbf{X}^{last}, \mathbf{W}, \mathbf{b}\right) \in \mathbb{R}^{N\times M\times C}$ for all grid cells without applying PlacesNet to all grid cells repeatedly. Here, classification error is defined using the cross-entropy loss and it is denoted by

$$\mathcal{L}_C = -\sum_c y_c \log\left(\frac{\exp\left(\widehat{o}_c\right)}{\sum_c \exp\left(\widehat{o}_c\right)}\right), \tag{5}$$

where

$$\begin{aligned}\widehat{\mathbf{O}} &= \left(\widehat{o}_c\right) \\ &= GAP\left(o_{n,m,c}\right) \\ &= \left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}o_{n,m,c}\right) \in \mathbb{R}^C \end{aligned} \tag{6}$$

is a vector of classification results for C classes and it is obtained by applying GAP to the result of (4); $\mathbf{Y} = (y_c) \in \{0, 1\}^C$ denotes the ground truth of the class of the given scene image and it is represented by a one-hot vector.

Then, the total training loss $\mathcal{L}_{total}$ is defined as a summation of the proposed SCL $\mathcal{L}_{SCL}$ and classification loss $\mathcal{L}_C$, and it is represented by

$$\mathcal{L}_{total} = \mathcal{L}_C + \gamma\mathcal{L}_{SCL} \tag{7}$$

where γ denotes the SCL rate and controls the relative weight between SCL and the classification loss.

Another key feature of PlacesNet is that partial convolution [36] is applied to all convolution layers. In vanilla convolution with zero padding, boundary of the image is filled with zeros and the vanilla convolution is applied, as shown in Fig. 7. Using the padded input $\mathbf{x}^{l-1,pad}_{n,m,1:D^l} = \left(x^{l-1,pad}_{n,m,d^l}\right) \in \mathbb{R}^{D^l}$, the output vector $\mathbf{x}^{l}_{n,m,1:D^{l+1}}$ of the l-th layer of vanilla convolutions at the position (n, m) is computed as follows:

$$x^l_{n,m,d^l} = \sum_{i=1}^{H^l}\sum_{j=1}^{W^l}W^l_{i,j,d^{l-1},d^l}x^{l-1,pad}_{n+i,m+j,d^l} + b^l_{d^l} \tag{8}$$

where $\mathbf{W}^l = \left(W^l_{i,j,d^{l-1},d^l}\right) \in \mathbb{R}^{H^l\times W^l\times D^{l-1}\times D^l}$ and $\mathbf{b}^l = \left(b^l_{d^l}\right) \in \mathbb{R}^{D^l}$ are the filter weights of the l-th layer; H^l and W^l are height and width of filter size respectively; D^l is the number of output channels of the l-th layer. Here, it can be seen that vanilla convolution $\mathbf{X}^l = \left(\mathbf{x}^l_{n,m,1:D^l}\right)$ around the boundary of the feature might not be as accurate as that inside the feature since the vanilla convolution should include many zero paddings. Recently, a lot of convolution layers are connected sequentially, and then the performance deterioration of the boundary of the given image becomes worse. For other general classification CNNs, the accuracy degradation around the boundary of the image might not be important because GAP is used before the classification. In FOSNet,

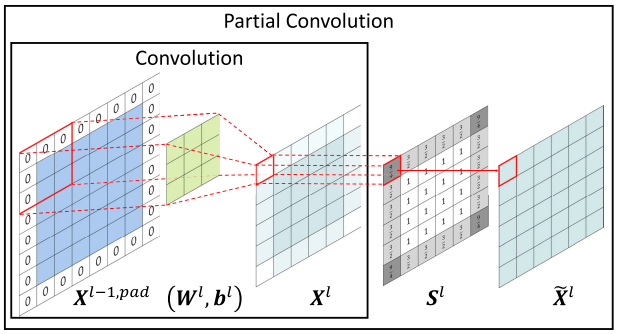


FIGURE 7. Illustration of convolution with zero padding and partial convolution.

however, the SCL is used as a loss and the classification accuracy around the boundary of the image is as important as that inside the image. Thus, the partial convolution proposed in [36] is used in FOSNet.

The structure of partial convolution [36] is given in Fig. 7. The scaling mask $\mathbf{S}^l = \left(S^l_{n,m}\right)$ is multiplied with the convolution $\mathbf{X}^l$, and the output of the partial convolution is computed by

$$\tilde{x}^l_{n,m,d^l} = S^l_{n,m} \sum_{i=1}^{H^l} \sum_{j=1}^{W^l} W^l_{i,j,d^{l-1},d^l} x^{l-1,pad}_{n+i,m+j,d^l} + b^l_{d^l}, \tag{9}$$

where

$$S^l_{n,m} = \frac{H^l W^l}{H^l W^l - \sum_{i=1}^{H^l} \sum_{j=1}^{W^l} \mathbb{1}^{l;pad}_{n+i,m+j}}, \tag{10}$$

where $S^l_{n,m}$ is scaling factor of output feature $\tilde{x}^l_{n,m,d^l}$; $\mathbb{1}^{l;pad}_{n,m}$ is 1 if position (n, m) of the input feature $x^{l-1,pad}_{n,m,d}$ is zero padded position, otherwise 0. Therefore, partial convolution adjusts for the varying amount of valid inputs by scaling, and it likely increases the accuracy of the near image boundary.

The partial convolution is a good match with the SCL and it will be shown that the combination enhances the classification accuracy significantly. The analysis will be given in Section V.

C. FUSION OF OBJECT FEATURE AND SCENE FEATURE

In this subsection, a new fusion module CCG is proposed. The CCG combines object feature $\mathbf{x}^{object}$ containing information of objects in the image with scene feature $\mathbf{x}^{scene}$. $\mathbf{x}^{object}$ is extracted from ObjectNet in Fig. 3, while $\mathbf{x}^{scene}$ is extracted from PlacesNet that is trained using SCL in Fig. 4(b). The CCG is based on a scene traits that *when a specific object in an image is found, the scene is very likely to belong to a particular class associated with the object*. The CCG is inspired by context gating [37] and the CCM [25]. The concept of CCG is depicted in Fig. 8.

Using CCM [25], CCG converts an object feature into a scene feature and outputs a pseudo scene feature

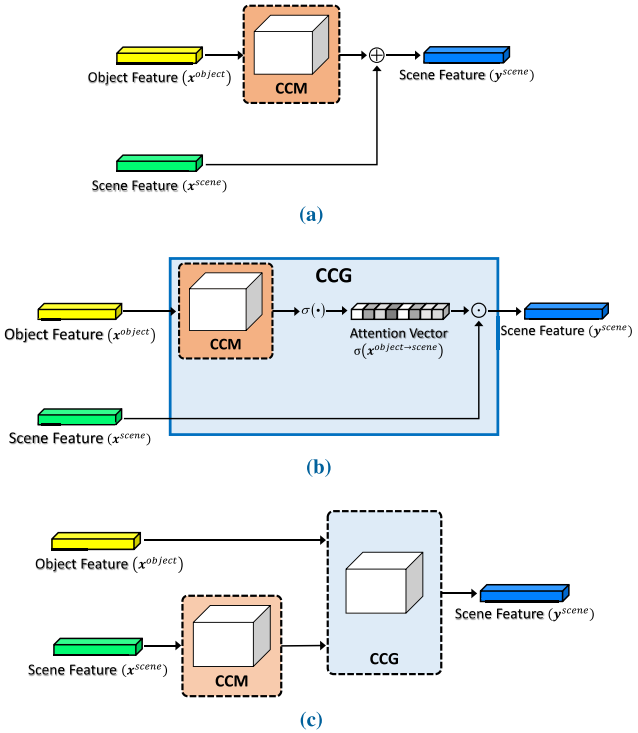


FIGURE 8. Trainable fusion modules with object feature and scene feature. (a) CCM; (b) CCG; (c) mixed CCM-CCG.

$\mathbf{x}^{object \to scene}$. Then, an attention map is generated by applying a sigmoid function to $\mathbf{x}^{object \to scene}$. The scene feature $\mathbf{x}^{scene}$ from PlacesNet is multiplied by the generated attention map $\sigma\left(\mathbf{x}^{object \to scene}\right)$ in element-wise manner, and a new scene feature $\mathbf{y}^{scene}$ is obtained by

$$\begin{aligned} \mathbf{y}^{scene} &= \sigma\left(\mathbf{x}^{object \to scene}\right) \odot \mathbf{x}^{scene} \\ &= \sigma\left(\mathbf{W}\mathbf{x}^{object} + \mathbf{b}\right) \odot \mathbf{x}^{scene} \end{aligned} \tag{11}$$

where $\odot$ denotes element-wise multiplication; $\mathbf{W}$ and $\mathbf{b}$ are the trainable parameters; $\mathbf{x}^{object \to scene}$ is a pseudo scene feature obtained by converting the object feature into the scene feature through CCM, and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is a sigmoid function.

The structure of CCG is motivated by context gating [37]. The context gating transforms the input feature into a new feature using a self-gating mechanism, and it demonstrated significant improvements in video understanding tasks. Motivated by context gating, CCG selectively activates the channels of scene feature $\mathbf{x}^{scene}$. The selective activation is carried out by applying a gating mechanism at the object feature $\mathbf{x}^{object}$, which are relevant to scene recognition. Here CCM [25] is applied to the object feature $\mathbf{x}^{object}$, and it converts $\mathbf{x}^{object}$ into a pseudo scene feature $\mathbf{x}^{object \to scene}$ to modify the context gating concept of the self-gating mechanism into the correlative-gating mechanism. The structure of CCG is shown in Fig. 8(b). As applied in the batch normalization (BN) [38] at the CCM in [25], batch normalization can be

applied to CCG as in

$$\mathbf{y}^{scene} = \sigma\left(\mathrm{BN}\left(\mathbf{W}\mathbf{x}^{object}\right)\right) \odot \mathbf{x}^{scene}. \quad (12)$$

Another variation, a mixed CCM-CCG, can also be considered. Since PlacesNet is pre-trained using Places 2 dataset [12], performance degradation might occur when PlacesNet is applied to scene recognition datasets other than Places 2 (e.g., SUN397 [39], MIT 67 [11]). To obtain $\mathbf{x}^{scene \rightarrow scene^{target}}$, CCM converts the scene feature extracted from the PlacesNet to the feature suitable for the target scene dataset. Then, the converted $\mathbf{x}^{scene \rightarrow scene^{target}}$ and object features are fused using CCG. In this case, the mixed CCM-CCG proceeds as follows:

$$\begin{aligned}\mathbf{y}^{scene} &= \sigma\left(\mathbf{x}^{object \rightarrow scene}\right) \odot \mathbf{x}^{scene \rightarrow scene^{target}} \\ &= \sigma\left(\mathbf{W}^1\mathbf{x}^{object} + \mathbf{b}^1\right) \odot \left(\mathbf{W}^2\mathbf{x}^{scene} + \mathbf{b}^2\right) \quad (13)\end{aligned}$$

The structure of the mixed CCM-CCG is depicted in Fig. 8(c).

Fusion can be conducted at two levels: feature level and score level, as carried out in [25]. For score level fusion, an object score in Fig. 3 and a scene score in Fig. 4 are fed into the trainable fusion module in Fig. 2. In this case, we do not apply softmax on each score vector. For feature level fusion, we use an object feature in Fig. 3 and a scene feature in Fig. 4 as input features to be fused. This previous study [25] provides a more detailed explanation.

IV. EXPERIMENTS

The proposed FOSNet is applied to three popular scene recognition datasets, and its performance is compared with that of the previous works. The three scene datasets for the experiment are Places 2 [12], SUN 397 [39], and MIT indoor 67 [11]. ImageNet dataset [13] is also used for the training of ObjectNet.

A. DATASETS

Places 2 dataset [12] is the largest dataset for scene recognition. It is an upgraded version of Places 1 [40], and it is also the latest of all the scene recognition datasets. This dataset has 365 scene categories; consisting of two versions of datasets: Places365-Challenge dataset and Places365-Standard dataset. Both versions of datasets share the same validation images and only differ in the number of training images. The Places365-Challenge dataset provides 8 million training images, whereas the Places-Standard dataset provides 1.8 million training images.

SUN 397 dataset [39] was the most popular scene dataset before the Places dataset [12], [40] was released. This dataset consists of 397 scene categories. Each category has at least 100 different numbers of images. The entire set has a total of 108,754 images. For fair comparison with other methods using this dataset, 10 subsets each of which has 50 training images and 50 validation images per class were used to evaluate the competing methods. The average validation accuracy over the 10 subsets were used as the overall accuracy of each method.

MIT indoor 67 dataset [11] is a scene recognition dataset consisting of 67 indoor scene categories, and it comprises a total of 15,620 indoor scene images. All the experiments with the MIT indoor 67 dataset were performed according to the standard evaluation protocol: A subset that has 80 training images and 20 testing images per scene category is used for evaluation.

ImageNet dataset [13] is one of the most commonly used datasets for object recognition task, and it consists of 1.2 million object images and 1000 object categories. A number of popular CNN structures were trained in the dataset, which include AlexNet [41], ResNet [32], DenseNet [33], ResNeXt [34], SE-Net [3], and others [42]–[44].

B. IMPLEMENTATION DETAILS

The FOSNet is comprised of neural networks and it was trained from scratch. All models were trained for 130 epochs. The initial learning rate was 0.15 when the mini-batch size was 256. For different mini-batch sizes, the learning rate was adjusted using the linear scaling rule [45] to achieve a similar performance. The learning rate was dropped by 0.1 times every 30 epochs. The synchronous stochastic gradient descent with a momentum of 0.9 was used as the optimization method. The training data were augmented by random rescaling, cropped randomly into 224 × 224 [43], [46] and horizontally flipped with a 0.5 chance. The input image was normalized by the per-color mean and standard deviation [46]. In addition, the data balancing strategy [2] was adopted for mini-batch sampling [3]. PlacesNet was trained using Places 2 dataset, and experiments were performed using transfer learning [47] on other datasets such as SUN 397 and MIT indoor 67.

A hyper-parameter γ in (7) is set to 1. Detailed explanation about γ is discussed in Section V-A. For a backbone network, the SE-ResNeXt-101 model, which is a combination of ResNeXt [34] with SE-Network [3], was used for ObjectNet and PlacesNet in FOSNet. The standard 10-crop testing method [17] is used for comparison with other methods, and an evaluation measurement is the average classification accuracy of 10 crops.

C. EXPERIMENTAL RESULTS ON THE PLACES 2

The FOSNet is compared with other scene recognition methods using the validation set of the Places 2 [12]. Previous methods [5], [10], [12], [17], [20], [25], [29] trained their networks on the Places365-Standard or Places365-Challenge dataset. The FOSNet is trained using the Places365-Challenge data for a fair comparison with the current state-of-the-art [17].[1] The comparison with other methods is summarized in Table 2.

All the methods listed in Table 2 use CNN: Adi-Red [5], CCM [25], and CNN-SMN [20], used information of the objects which appear in scene images. To obtain the object

[1]Note that [17] uses Places401 dataset which contains more than 10 million images for training a teacher network.

TABLE 2. Comparison with other scene recognition methods on Places 2 [12] validation set.

Methods	Publication	# of Patches	Network Input Size	Accuracy (100%) top-1	top-5
CNN-SMN [20]	TIP 2017	276	224×224	57.1	-
Multi-Resolution CNNs [17]	TIP 2017	40	512×512	58.3	87.3
Adi-Red [5]	ACM MM 2018	≈7	224×224	41.87	-
Places365-ResNet [12]	TPAMI 2018	10	224×224	54.74	85.08
Places365-VGG [12]	TPAMI 2018	10	224×224	55.24	84.91
SOSF+CFA+GAF [10]	TCSVT 2018	1	608×608	57.27	-
LGN [29]	Arxiv 2019	≈1500	224×224	56.50	86.24
CCM [25]	IJCNN 2019	1	224×224	56.82	86.92
FOSNet (ours)	-	10	224×224	**60.14**	**88.86**

TABLE 3. Comparison with other scene recognition methods on SUN 397 [39].

Methods	Publication	# of Patches	Network Input Size	Accuracy (100%)
DAG-CNN [1]	ICCV 2015	10	224×224	56.2
MetaObject-CNN [8]	ICCV 2015	128	227×227	58.11
Three [6]	CVPR 2016	≈120	899×899	70.17
Hybrid CNN [18]	TCSVT 2017	100	224×224	70.69
Sparse Representation [23]	TIP 2017	6907	224×224	71.08
Multi-Resolution CNNs [17]	TIP 2017	30	336×336	72.0
CNN-SMN [20]	TIP 2017	276	224×224	72.6
PatchNet [22]	TIP 2017	1800	224×224	73.0
Places365-VGG-SVM [12]	TPAMI 2018	10	224×224	63.24
SDO [7]	PR 2018	≈200	224×224	73.41
Adi-Red [5]	ACM MM 2018	≈7	224×224	73.59
SOSF+CFA+GAF [10]	TCSVT 2018	1	608×608	**78.93**
VS-CNN [21]	IEEE Access 2019	3	227×227	43.14
Gaze Shifting-CNN+SVM [19]	IEEE Trans. Cybernetics 2019	5	not fixed	56.2
Deep Patch Representations [26]	ACM Trans. MM 2019	1	448×448	57.47
NNSD [27]	IEEE Trans. MM 2019	352	224×224	64.78
Context Modeling with BiLSTM [28]	SIBGRAPI 2019	>1000	224×224	71.81
LGN [29]	Arxiv 2019	≈1500	448×448	74.06
FOSNet (ours)	-	10	224×224	77.28

information, they used the CNN pre-trained on the object recognition dataset. Multi-Resolution CNN [17] created a super category by considering the label ambiguity of scene categories to train a teacher network. Places365-VGG and Places365-ResNet [12] used the vanilla CNN architecture and the scene trait was not taken into consideration. In our FOSNet, two independent CNNs were used as backbone networks. Here, it should also be noted that the proposed FOSNet is also computationally more efficient than other methods: The number of patches used for the evaluation of the compared methods is usually larger than 100. The number of patches used in [20] is even almost 300, whereas the number of patches used in FOSNet is only 10. Further, the input size of FOSNet is 224 × 224 and it is less than a fifth of the size of Multi-Resolution CNNs. From the figure, our FOSNet achieves state-of-the-art accuracy of 60.14% on the Places 2, and it is the first time that the accuracy exceeds 60% on the dataset. This implies that FOSNet demonstrates the best performance among other compared methods while spending reasonably small computation.

D. EXPERIMENTAL RESULTS ON THE SUN 397

FOSNet is applied to the SUN 397 dataset [39]. An average validation accuracy of 10 subsets provided in the dataset is carried out to compare the competing scene recognition methods, and the comparison results are summarized in Table 3.

From the table, it can be seen that the FOSNet outperforms most of the previous methods except [10]. Among the competing methods, FOSNet achieves the second best accuracy of 77.72%, slightly lower than that of the state-of-the-art SOSF + CFA + GAF method [10]. Here, it should be noted that it is unfair to directly compare the results of the two methods, considering that SOSF + CFA + GAF

TABLE 4. Comparison with other scene recognition methods on the MIT 67 [11] validation set.

Methods	Publication	# of Patches	Network Input Size	Accuracy (100%)
DPM+GIST+SP [30]	ICCV 2011		None	43.1
RBoW [31]	CVPR 2012		None	37.93
DAG-CNN [1]	ICCV 2015	10	224×224	77.5
MetaObject-CNN [8]	ICCV 2015	256	227×227	78.9
Three [6]	CVPR 2016	≈120	899×899	86.04
PatchNet [22]	TIP 2017	1800	224×224	86.2
CNN-SMN [20]	TIP 2017	276	224×224	86.5
Multi-Resolution CNNs [17]	TIP 2017	30	336×336	86.7
Sparse Representation [23]	TIP 2017	3886	224×224	87.22
Hybrid CNN [18]	TCSVT 2017	100	224×224	85.97
SOSF+CFA+GAF [10]	TCSVT 2018	1	608×608	89.51
ResNet-152-DFT^{+} [4]	ECCV 2018	1	224×224	76.5
Places365-VGG-SVM [12]	TPAMI 2018	10	224×224	76.53
SDO [7]	PR 2018	≈200	224×224	86.76
Gaze Shifting-CNN+SVM [19]	IEEE Trans. Cybernetics 2019	5	not fixed	75.1
Deep Patch Representations [26]	ACM Trans. MM 2019	1	448×448	79.63
VS-CNN [21]	IEEE Access 2019	3	227×227	80.37
NNSD + ICLC [27]	IEEE Trans. MM 2019	224	224×224	84.3
LGN [29]	Arxiv 2019	≈1500	448×448	88.06
Context Modeling with BiLSTM [28]	SIBGRAPI 2019	>1000	224×224	88.25
GP+AP+V67 [50]	CMES-COMP 2020	1	224×224	70.46
FOSNet (ours)	-	10	224×224	**90.30**

TABLE 5. Ablation study of SCL using ResNet-18. This experiment is performed on the Places365-Standard dataset [12].

Base Model	ResNet-18									
SCL Rate (γ)	0 (Baseline)		10^{-2}		10^{-1}		10^{0}		10^{1}	
	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
SCL with Vanilla Conv	54.438	84.912	54.548	84.715	54.770	84.901	54.942	**85.074**	53.775	84.151
SCL with Partial Conv	54.718	84.866	54.710	84.841	54.901	85.038	**55.090**	85.027	53.688	84.099

[10] includes YOLOv2 [9] and 4-directional long short-term memory (LSTM) [48]. To train YOLOv2, an object detection dataset Object177 [49] was additionally used. Unlike the dataset used to train ObjectNet, the object detection dataset Object177 includes not only the class labels but also bounding box information for the location of objects in an image, which is quite difficult to annotate. Furthermore, the method SOSF + CFA + GAF requires more computation than our method. The FOSNet uses input images of size 224 × 224, whereas SOSF + CFA + GAF uses input images of size 608×608, and employs 4-directional LSTM, which is obviously very computationally expensive. Further, it also should be noted that Sparse Representation [23], Context Modeling with BiLSTM [28], PatchNet [22], and LGN [29] use more than thousands of patches. The compuation time increases linearly with the number of patches used in test. Our FOSNet uses a 10-crop testing method.

E. EXPERIMENTAL RESULTS ON THE MIT INDOOR 67

In this subsection, the FOSNet is applied to the validation set of the MIT indoor 67 [11], and Table 4 presents a comparison result of the scene recognition for MIT 67.

In Table 4, all the existing methods except RBoW [31] and DPM+GIST+SP [30] use CNN. The two methods use the handcraft features. From Table 4, of all the competing methods, our FOSNet offers the best accuracy. In particular, the FOSNet outperforms Context Modeling with BiLSTM [28] which is the current state-of-the-art method on MIT indoor 67. As a result, our FOSNet achieves state-of-the-art accuracy of 90.30% on the MIT indoor 67, and this is the first time that the accuracy exceeds 90% on the dataset.

V. ABLATION STUDY

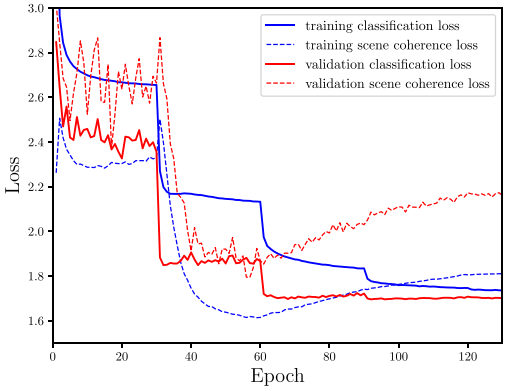
Additional experiments are performed to gain a better understanding of the effects of our proposed SCL and CCG. All ablation studies are performed using the Places365-Standard dataset [12] for various experiments with fast training. Standard 224 × 224 single-crop evaluation is employed, and ResNet-18 and ResNet-50 [32] are used as the backbone architectures.

A. ANALYSIS ON SCENE COHERENCE LOSS (SCL)

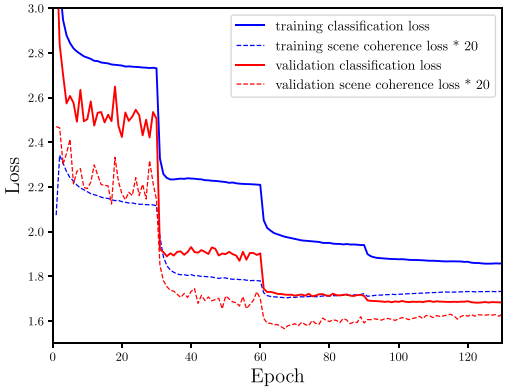
In this subsection, the effects of SCL on scene recognition are demonstrated. In the experiment, only PlacesNet is used and all models in the experiments are trained from scratch

TABLE 6. Ablation study of SCL using ResNet-50. This experiment is performed on the Places365-Standard dataset [12].

Base Model	ResNet-50			
SCL Rate (γ)	0 (Baseline)		10^0	
	top-1	top-5	top-1	top-5
SCL with Vanilla Conv	55.888	86.123	56.285	**86.288**
SCL with Partial Conv	56.227	86.099	**56.337**	86.195



(a)



(b)

FIGURE 9. Classification loss and SCL curves of ResNet-18 trained (a) with only classification loss and (b) with classification loss and SCL. The blue line denotes the loss of the training set, and the red line denotes the loss of the validation set. The solid line represents classification loss, and the dotted line represents SCL.

for a fair comparison. The results of SCL ablation studies are shown in Tables 5 and 6.

In Table 5, accuracy is computed while varying the SCL rate (γ) in (7). When $\gamma = 0$, only classification loss is used as a total loss in (7). This case is a baseline. When $\gamma \leq 1$, accuracy is improved from the baseline in all cases, whereas when $\gamma = 1$, the PlacesNet in FOSNet achieves the best top-1 accuracy. When $\gamma = 10$, the accuracy is degraded, revealing that too much emphasis on SCL is an obstacle to minimizing classification errors. Table 6 shows the results of the same experiment using ResNet-50, and similar results are obtained regarding the effects of the SCL with Table 5. The models with SCL always outperform the ones without SCL regardless of which CNN backbone is used. Experimental results regarding the effects of partial convolution [36] on SCL are given in Tables 5 and 6. This partial convolution improves the performance of the baseline, and its effect on the performance is higher when it is combined with SCL. Through the ablation studies, it can be noted that scene recognition performance is improved by using SCL. Since the best performance is obtained when $\gamma = 1$, the value is used to train PlacesNet.

Another experiment is performed to show the validity of the SCL. In Fig. 9(a), the SCL is monitored, and it is unused (not propagated backward) for the training. As shown in Fig. 9(a), the SCL decreases until reaching 60 epochs even when SCL is unused for the training. After 60 epochs in Fig. 9(a), the SCL increases rapidly; the validation loss is almost saturated but the training loss decreases rapidly, revealing that the PlacesNet is overfitted. From the observation, the overfitting in the scene recognition is highly related to SCL. Thus, if the PlacesNet is trained to force the SCL to be reduced, the overfitting of the PlacesNet will be relaxed and its generalization performance will be improved. Fig. 9(b) shows the result when ResNet-18 is trained with SCL. In this case, SCL converges quickly and almost vanishes. Thus, SCL is magnified 20 times for visualization in Fig. 9(b). After 60 epochs in Fig. 9(b), both training and validation errors decrease gradually but consistently, implying that the PlacesNet overfitting is relaxed.

Fig. 10 provides the results of class activation map (CAM) [35] using ResNet-18. The first row shows the input images. Using ResNet-18 trained without and with SCL, the second and third rows show the CAM images, respectively. In the figures, red parts denote the region which is relevant and makes a contribution to the scene classification, whereas the blue parts denote the region that offer no information and no contribution to the scene classification. When trained with SCL, the red region becomes bigger than trained without the SCL. This shows that the region which would be ignored if trained without SCL is fully exploited with SCL. In the entire experiments, the SCL is an effective loss for the scene classification that enables the FOSNet to fully exploit the *sceneness* all over the image.

B. ANALYSIS ON CORRELATIVE CONTEXT GATING (CCG)

The proposed feature fusion method CCG is analyzed through experimentation. For a fair comparison, in PlacesNet, CNN models without partial convolution were used for scene feature extraction, and the experimental results are presented in Table 7.

All the models in Table 7 are trained from scratch. Compared with the existing fusion methods, such as sum, concatenation or CCM [25], our fusion method CCG improves performance in most models regardless of whether SCL is added. Although the fusion by concatenation achieves the best top-5 performance in ResNet-50, the simple fusion delivers limited performances in a new dataset that PlacesNet did not train, as explained in Sections IV-D and IV-E.

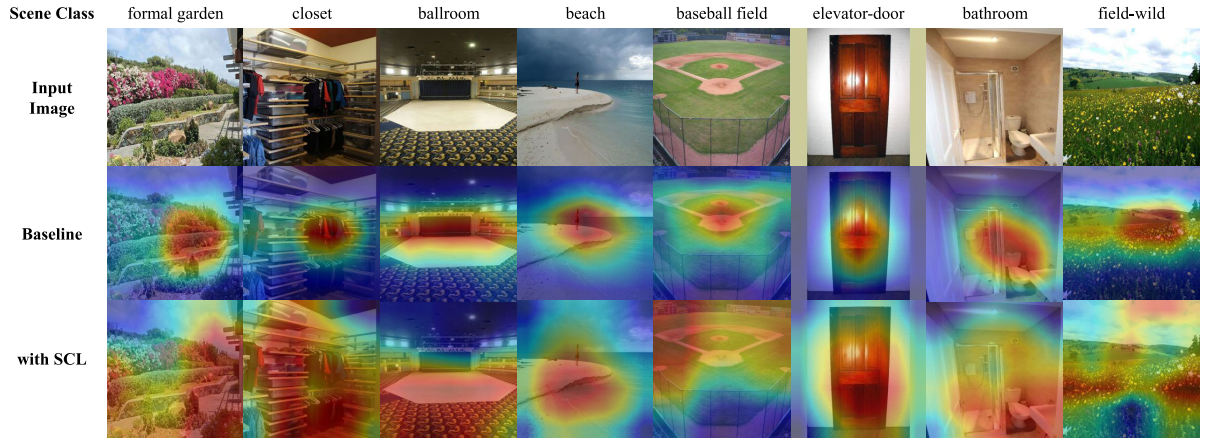


FIGURE 10. The class activation map (CAM) [35] results using ResNet-18. The ground truth about the scene class of the image is on top of the image. The first row shows the input image. The second row shows the CAM result using ResNet-18 trained without SCL. The third row shows the CAM result using ResNet-18 trained with SCL.

TABLE 7. Ablation study of CCG on the Places365-Standard dataset [12].

Fusion Level	Method	ResNet-18		ResNet-18 – SCL		ResNet-50		ResNet-50 – SCL	
		top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
	Baseline	54.438	84.912	54.942	85.074	55.888	86.123	56.285	86.288
Feature Level	Sum	53.485	84.123	54.570	84.680	56.115	86.285	56.630	86.345
	Concatenate	54.701	85.071	55.164	85.145	56.230	**86.411**	56.685	86.441
	CCM with ReLU [25]	54.756	85.107	55.076	85.090	56.334	86.375	56.663	86.529
	CCM-BN with ReLU [25]	54.786	85.181	55.129	85.230	56.269	86.395	56.726	**86.573**
	CCG	54.575	84.888	54.907	84.921	56.060	86.186	56.469	86.233
	CCG-BN	**54.934**	**85.206**	55.153	85.088	**56.367**	86.395	56.729	86.397
	mixed CCM-CCG-BN	54.504	84.997	54.959	85.132	56.060	86.373	**56.800**	86.466
Score Level	CCM [25]	54.477	84.869	55.055	85.107	56.096	86.233	56.581	86.315
	CCM-BN [25]	54.600	84.979	55.104	85.126	56.110	86.238	56.690	86.343
	CCG	54.562	84.901	55.071	85.071	56.030	86.192	56.600	86.238
	CCG-BN	54.677	85.156	**55.211**	**85.233**	56.203	86.375	56.685	86.348

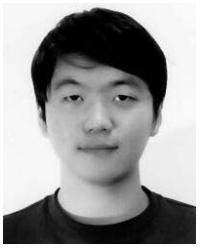
VI. CONCLUSION

In this paper, a new scene recognition framework named FOSNet has been proposed, in which the object and the scene information have been combined in a trainable fusion module named CCG. The entire system was trained using SCL, which is a new loss developed for the scene recognition. SCL is based on the unique property of the scene, e.g., the '*sceneness*' spreads and the scene class does not change all over the image. The proposed FOSNet was experimented with three most popular scene recognition datasets, and the state-of-the-art performance is obtained in Places 2 and MIT indoor 67.

REFERENCES

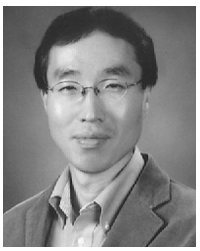
[1] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1215–1223.
[2] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *Proc. ECCV*, 2016, pp. 467–482.
[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, Jun. 2018, pp. 7132–7141.
[4] J. Ryu, M.-H. Yang, and J. Lim, "DFT-based transformation invariant pooling layer for visual classification," in *Proc. ECCV*, 2018, pp. 84–99.
[5] Z. Zhao and M. Larson, "From volcano to toyshop: Adaptive discriminative region discovery for scene recognition," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1760–1768.
[6] L. Herranz, S. Jiang, and X. Li, "Scene recognition with CNNs: Objects, scales and dataset bias," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 571–579.
[7] X. Cheng, J. Lu, J. Feng, B. Yuan, and J. Zhou, "Scene recognition with objectness," *Pattern Recognit.*, vol. 74, pp. 474–487, Feb. 2018.
[8] R. Wu, B. Wang, W. Wang, and Y. Yu, "Harvesting discriminative meta objects with deep CNN features for scene classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1287–1295.
[9] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
[10] N. Sun, W. Li, J. Liu, G. Han, and C. Wu, "Fusing object semantics and deep appearance features for scene recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1715–1728, Jun. 2019.
[11] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.
[12] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[16] A. Bayat and M. Pomplun, "Deriving high-level scene descriptions from deep scene CNN features," in *Proc. 7th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2017, pp. 1–6.
[17] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2055–2068, Apr. 2017.
[18] G.-S. Xie, X.-Y. Zhang, S. Yan, and C.-L. Liu, "Hybrid CNN and dictionary-based models for scene recognition and domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1263–1274, Jun. 2017.
[19] X. Sun, L. Zhang, Z. Wang, J. Chang, Y. Yao, P. Li, and R. Zimmermann, "Scene categorization using deeply learned gaze shifting kernel," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2156–2167, Jun. 2019.
[20] X. Song, S. Jiang, and L. Herranz, "Multi-scale multi-feature context modeling for scene recognition in the semantic manifold," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2721–2735, Jun. 2017.
[21] J. Shi, H. Zhu, S. Yu, W. Wu, and H. Shi, "Scene categorization model using deep visually sensitive features," *IEEE Access*, vol. 7, pp. 45230–45239, 2019.
[22] Z. Wang, L. Wang, Y. Wang, B. Zhang, and Y. Qiao, "Weakly supervised PatchNets: Describing and aggregating local patches for scene recognition," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2028–2041, Apr. 2017.
[23] G. Nascimento, C. Laranjeira, V. Braz, A. Lacerda, and E. R. Nascimento, "A robust indoor scene recognition method based on sparse representation," in *Proc. Iberoamer. Congr. Pattern Recognit.*, 2017, pp. 408–415.
[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[25] H. Seong, J. Hyun, H. Chang, S. Lee, S. Woo, and E. Kim, "Scene recognition via Object-to-Scene class conversion: End-to-End training," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.
[26] S. Jiang, G. Chen, X. Song, and L. Liu, "Deep patch representations with shared codebook for scene classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 1s, pp. 1–17, Feb. 2019, doi: 10.1145/3231738.
[27] L. Xie, F. Lee, L. Liu, Z. Yin, and Q. Chen, "Hierarchical coding of convolutional features for scene recognition," *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1182–1192, May 2020.
[28] C. Laranjeira, A. Lacerda, and E. R. Nascimento, "On modeling context from objects with a long short-term memory for indoor scene recognition," in *Proc. 32nd SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2019, pp. 249–256.
[29] G. Chen, X. Song, H. Zeng, and S. Jiang, "Scene recognition with prototype-agnostic scene layout," 2019, *arXiv:1909.03234*. [Online]. Available: http://arxiv.org/abs/1909.03234
[30] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *Proc. ICCV*, Nov. 2011, pp. 1307–1314.
[31] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb, "Reconfigurable models for scene recognition," in *Proc. CVPR*, Jun. 2012, pp. 2775–2782.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.
[34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[36] G. Liu, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, A. Tao, and B. Catanzaro, "Partial convolution based padding," 2018, *arXiv:1811.11718*. [Online]. Available: http://arxiv.org/abs/1811.11718
[37] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," 2017, *arXiv:1706.06905*. [Online]. Available: http://arxiv.org/abs/1706.06905
[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 1–11.
[39] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3485–3492.
[40] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. NIPS*, 2014, pp. 487–495.
[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.
[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[45] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*. [Online]. Available: http://arxiv.org/abs/1706.02677
[46] S. Gross and M. Wilber. (2016). *Training and Investigating Residual Nets*. [Online]. Available: https://github.com/facebook/fb.resnet.torch
[47] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[49] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. NIPS*, 2010, pp. 1378–1386.
[50] L. Chen, K. Bo, F. Lee, and Q. Chen, "Advanced feature fusion algorithm based on multiple convolutional neural network for scene recognition," *Comput. Model. Eng. Sci.*, vol. 122, no. 2, pp. 505–523, 2020.



HONGJE SEONG received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2018, where he is currently pursuing the combined master's and Ph.D. degrees. He has studied computer vision, machine learning, and deep learning.

JUNHYUK HYUN (Member, IEEE) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2014, where he is currently pursuing the combined master's and Ph.D. degrees. He has studied computer vision, machine learning, and deep learning.



EUNTAI KIM (Member, IEEE) was born in Seoul, South Korea, in 1970. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, in 1992, 1994, and 1999, respectively. From 1999 to 2002, he was a Full-Time Lecturer with the Department of Control and Instrumentation Engineering, Hankyong National University, Kyonggi-do, South Korea.

Since 2002, he has been with the Faculty of the School of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. He was also a Visiting Researcher with the Berkeley Initiative in Soft Computing, University of California, Berkeley, CA, USA, in 2008. His current research interests include computational intelligence, statistical machine learning and deep learning and their application to intelligent robotics, autonomous vehicles, and robot vision.