2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Understand Scene Categories by Objects: A Semantic Regularized Scene Classifier Using Convolutional Neural Networks

Yiyi Liao[1], Sarath Kodagoda[2], Yue Wang[1], Lei Shi[2], Yong Liu[1]

***Abstract*— Scene classification is a fundamental perception task for environmental understanding in today's robotics. In this paper, we have attempted to exploit the use of popular machine learning technique of deep learning to enhance scene understanding, particularly in robotics applications. As scene images have larger diversity than the iconic object images, it is more challenging for deep learning methods to automatically learn features from scene images with less samples. Inspired by human scene understanding based on object knowledge, we address the problem of scene classification by encouraging deep neural networks to incorporate object-level information. This is implemented with a regularization of semantic segmentation. With only 5 thousand training images, as opposed to 2.5 million images, we show the proposed deep architecture achieves superior scene classification results to the state-of-the-art on a publicly available SUN RGB-D dataset. In addition, performance of semantic segmentation, the regularizer, also reaches a new record with refinement derived from predicted scene labels. Finally, we apply our model trained on SUN RGB-D dataset to a set of images captured in our university using a mobile robot, demonstrating the generalization ability of the proposed algorithm.**

Fig. 1. Scene classification demonstration. The examples are captured in our university using the mobile robot. Our SS-CNN trained on SUN RGB-D dataset gives the predict labels below each image, without retraining for the completely new environment. Labels in black means correct classification. Two misclassified image are given with labels in red, while the predicted labels are in accord with human recognition to some extent.

I. Introduction

Today's robotics face many perception challenges such as scene classification (Figure 1), semantic segmentation, object recognition and detection. For object-level tasks, a series of new performance standards are set with the recently successful deep Convolutional Neural Networks (CNN) [1]–[3], while the performance on scene-level perception based on deep CNN did not reach the same level of success before the work of Place-CNN [4]. As pointed out in [4], scene-level task is more challenging for feature learning due to the larger diversity of scene images compared to iconic object images. For Place-CNN, it overcame this diversity and reached state-of-the-art by training on 2.5 million scene images. However, it is very expensive to collect and label training images in such a large scale. Furthermore, enhancing the performance by increasing the number of training samples is not preferable in most robotic applications, especially for those tasks with insufficient samples. In this paper, we focus on constructing a scene classifier with competitive performance, while automatically learning feature with less amount of training images using the deep CNN.

It is more likely that the human beings understand scene classes mainly according to the object-level information, as scene classes are naturally defined at a higher level than the objects. For example, we incline to recognize the scene as "bedroom" as we find objects "bed" and "night stand" in it. Intuitively, understanding the scene classes involving object-level information would suppress the large diversity of scene images and lead to better generalization ability. This hypothesis is validated with a baseline experiment by using object existence as feature vector to classify the scene classes — with a much lower dimension, the object existence feature allows a similar performance to the Place-CNN features. This result reveals that object-level information has the potential to improve scene classification. Inspired by human way of scene classification, we encourage the deep CNN to understand objects in early stage. Specifically, we develop a *scene classification* model with regularization of *semantic segmentation* based on the well-known CNN architecture, Alexnet [1], named SS-CNN. An example of our model structure is shown in Figure 2, where the features learned for scene classification in SS-CNN automatically involve object-level information. On SUN RGB-D dataset [5], we train our SS-CNN and show it significantly outperforms the original Alexnet, which further validate our hypothesis that the semantic regularization enhances the generalization ability. Besides, SS-CNN achieves superior results compared

*This work is supported by the National Natural Science Foundation Project of China (U1509210), the Natural Science Foundation Project of Zhejiang Province (LR13F030003), and the Joint Centre for Robotics Research (JCRR) between Zhejiang University and the University of Technology, Sydney.

[1]Yiyi Liao, Yue Wang and Yong Liu are with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China.

[2]Sarath Kodagoda and Lei Shi are with the Centre for Autonomous Systems (CAS), The University of Technology, Sydney, Australia.

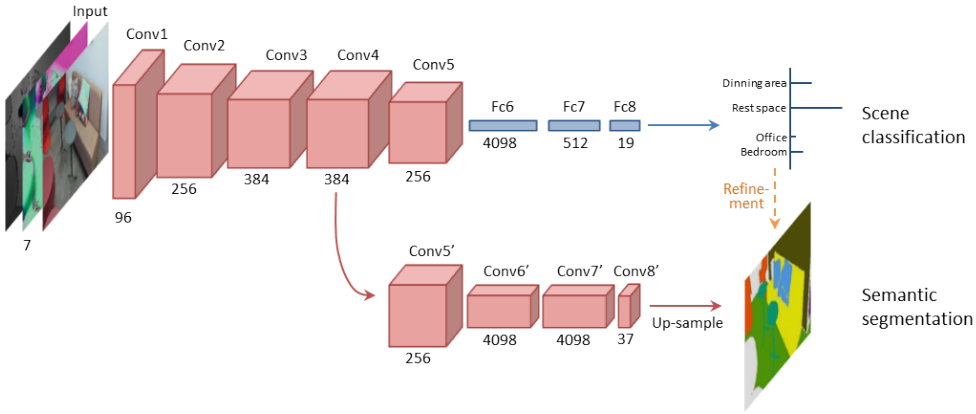


Fig. 2. An example architecture of the proposed SS-CNN, which is composed of a main branch for scene classification and a regularizer branch for semantic segmentation. The semantic regularization is imposed to the beginning 4 layers in this figure. The main branch outputs 1-D probability prediction for each image, where the regularization branched outputs 2-D probability prediction for each pixel. A refinement process denoted as the dashed orange line is implemented in the test process to promote the performance of the semantic segmentation using the predicted scene labels.

to the state-of-the-art Place-CNN, which is also based on Alexnet but gains its power with 2.5 million training images, while SS-CNN is only trained with 5 thousand images.

In addition, we develop a refinement method for semantic segmentation with the predicted scene classes, which is based on scene-object co-occurrences learned from training data. For instance, knowing the scene as a "bedroom" could prevent us from misclassifying the object "cabinet" as a "fridge". As a result, the performance of semantic segmentation also reaches the state-of-the-art on SUN RGB-D dataset.

After training and validation on SUN RGB-D dataset collected in the US, we further apply our SS-CNN to a mobile robot to classify scenes in a building of our university, in Australia. The mobile robot and some RGB images with its predicted results are given in Figure 1. The promising performance of SS-CNN in the completely new environment reveals its potential capability in robotics applications.

The remainder of the paper is organized as follow: Section II gives a review of related works. The proposed model and refinement method are given in Section III, and the experimental results are shown in Section IV. Finally, we conclude the paper with future direction of research in Section V.

II. Related works

Some previous works have demonstrated that interaction between scene and objects has the capability to promote each other [6]–[10]. The typical idea is to build the relationship between scenes and objects using a graphical model such as Markov Random Field or Conditional Random Field [11]. Though these works have achieved superior results, they are based on hand-crafted features, which means the feature extraction and classification in these works are not in a unified optimization framework. Compared to these works focusing on simultaneously labeling, our work is more close to Object Bank [12] since we focus on regularizing scene classification with semantic segmentation. Object Bank proposed a high-level representation for scene classification by encoding the images with combination of a large amount of object detectors. However, the feature extraction and scene classification in Object Bank are still optimized separately, and it requires pre-training a large number of object detectors. Recently, the superior results achieved with deep learning methods suggest that learning features with a fully trainable architecture may be a better choice. In this paper, we implemented the scene classifier using a fully trainable deep architecture with a single semantic segmentation branch encoding all object-level information.

As for the conventional deep learning methods, the most successful CNN model in scene classification is Place-CNN [4], which is trained on 2.5 million labeled images belonging to 476 scene classes using the well-known architecture Alexnet [1]. Before [4], the performance of CNN on scene classification was within the range of performances of some hand-crafted features based implementations [13]. As pointed out in [4], one reason of the relatively poor result on scene classification of CNN is due to the larger diversity of scene-centric images compared to object-centric images, which means scene classification has higher requirement on generalization ability. A further investigation to the Place-CNN [14] shown that object-level information emerges from the scene-centric trained model, which also gave the inspiration that the CNN learns to classifier scenes by understanding object-level information. By explicitly encouraged the CNN to classify scenes through understanding of object existence, we developed a scene classifier also based on Alexnet and achieves better generalization ability than Place-CNN with only 5 thousand training images.

To the best of our knowledge, considering of multiple tasks is less exploited in deep learning methods. In DeepID-Net [15], a refinement scheme is conducted to refine the object detection using the image classification result. More specifically, they introduced another separated network for image classification and concatenated the estimated image probability with the estimated object probability for a further classification, which means the information of two tasks

are only combined after independent training, instead of simultaneous training as implemented in this paper.

III. Model design

With the aim of learning scene features involved object information, we construct our SS-CNN for *scene classification* with regularization of *semantic segmentation*. In this section, the network architecture of our SS-CNN is introduced in detail, followed by the model learning and input construction. On top of that, we implement refinement for semantic segmentation with the predicted scene labels.

A. CNN for scene classification with semantic segmentation regularization

Notation. We first clarify the symbols used in this paper. Assume there are M_s scene classes in scene classification and M_o object classes in semantic segmentation. Let's denote the data structure of a single sample as $(\boldsymbol{X}, \boldsymbol{y_s}, \boldsymbol{Y_o})$, where $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$ is input image with H as height, W as width and C as number of channels, $\boldsymbol{y_s} \in \mathbb{Z}^{1\times M_s}$ is the ground truth of a scene label encoded in 1-of-K encoding scheme, i.e. $y_s^k = 1$ if $\boldsymbol{X}$ belongs to k^{th} scene class, otherwise $y_s^k = 0$. $\boldsymbol{Y_o} \in \mathbb{Z}^{H\times W\times M_o}$ is the ground truth of semantic segmentation label having the same height and width with $\boldsymbol{X}$. Analogously, $y_o^{ijk} = 1$ denotes the pixel (i, j) belonging to k^{th} object class.

Network architecture. For model construction of SS-CNN, a conventional CNN model is employed as the basic model to predict the scene classes with input pair $(\boldsymbol{X}, \boldsymbol{y_s})$. Then the major contribution of this paper is to add another fully convolutional branch [16] to the basic model, with the aim of estimating $\boldsymbol{Y_o}$ for semantic segmentation. The fully convolutional branch can be added to the main branch on arbitrary layer, we further define SS-CNN-Rn to denote the different configurations of SS-CNN as follow:

Given an original CNN for scene classification with N_l layers in all, denote SS-CNN-Rn as the SS-CNN with the previous n layers regularized by semantic segmentation, n is ranging from 1 to N_l.

In this paper, we take the well-known Alexnet architecture [1] as our main branch for scene classification. In Alexnet, we have $N_l = 8$ and there are 8 invariants of SS-CNN. The detailed network configuration of some typical networks are given in Figure 3.

Intuitively, how many layers are regularized by semantic segmentation would influence the performance of SS-CNN. If n is small, then the regularization is only imposed to a few layers of the scene classification. Considering the extreme case with $n = 0$, then two separate neural networks are constructed for scene classification and semantic segmentation respectively. As n getting larger, the semantic segmentation regularizes more layers in the main branch.

It is to be noted that the main branch keeps its original structure from SS-CNN-R1 to SS-CNN-R5 with 5 convolutional layers and 3 fully connected layers. Beginning from SS-CNN-R6, the structure of the main branch is slightly different, as the fully connected layers in main branch are also casted into convolutional layers one by one. When $n = 8$, `fc6` and `fc7` are both casted into convolutional layer, thus two additional fully connected layers are built for scene classification.

Model learning. As can be seen from the SS-CNN architectures, the loss function of our SS-CNN is composed of two parts, one is the loss of scene classification and the other is the semantic segmentation. In this paper, we use the multinomial logistic loss on a softmax layer. The loss function of scene classification is:

$$L_{scene} = -\sum_{k=1}^{M_s} y_s^k log(p_s^k) \tag{1}$$

where p_s^k is the probability of estimating $\boldsymbol{X}$ in class k, which is obtained with the final softmax layer taking $\boldsymbol{f}$ as input:

$$p_s^k = \frac{e^{\boldsymbol{f}^T\boldsymbol{\theta_k}}}{\sum_{i=1}^{M_s} e^{\boldsymbol{f}^T\boldsymbol{\theta_i}}} \tag{2}$$

Analogously, we can obtain the probability of each pixel p_o^{ijk} in semantic segmentation branch and define the loss function as:

$$L_{object} = -\sum_i \sum_j \sum_{k=1}^{M_o} y_o^{ijk} log(p_o^{ijk}) \tag{3}$$

Then the loss of the whole network is composed of these two losses as:

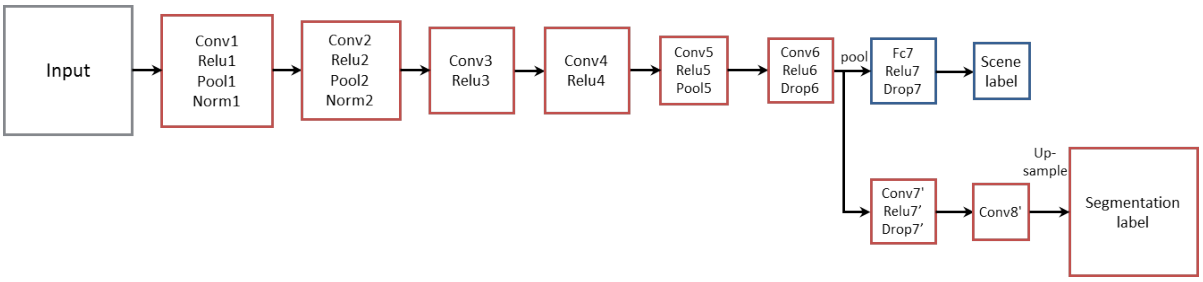
$$L_{ss} = L_{scene} + \alpha L_{object} \tag{4}$$

where α is the weight of the regularization term L_{object}. Notice that each image is corresponding to a single cost for scene classification, while the cost of semantic segmentation is summarized over all pixels (not normalized in our model). In this paper we choose $\alpha = 1/1000$ based on experiments.

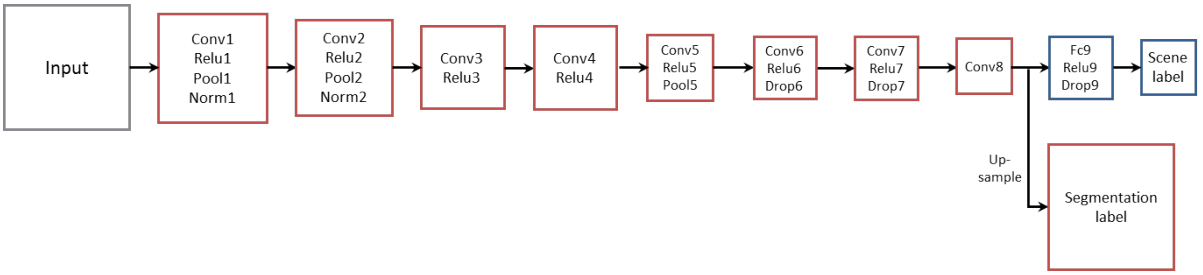
We use stochastic gradient descent with momentum for model training. Note that given SS-CNN-Rn, only weights from layer 1 to layer n are regularized with semantic segmentation, i.e. tuned with respect to the partial gradient of L_{ss}. From layer $n+1$, the weights in scene classification branch is tuned with only respect to L_{scene}, and the same for the semantic segmentation branch as being tuned with respect to αL_{object}.

Depth representation. Depth information is important in scene understanding. Many successful models are built on RGB-D inputs captured by the affordable RGB-D sensors such as Kinect and X-tion, especially in indoor environments [17], [18]. In this paper, we also explore the effective ways to encode the depth information in deep CNN.

The most direct way of considering depth information in deep CNN is to add a depth channel in the input layer. The depth image we use is linearly rescaled to $[0, 255]$, which is in the same range as the RGB image. Since depth image only provides information of distances, we also consider using the knowledge of normal vectors. For estimation of normal vectors, the depth image is first applied to a bilateral filter for smoothing. And then the smoothed depth image is transformed into a point cloud with the camera intrinsic

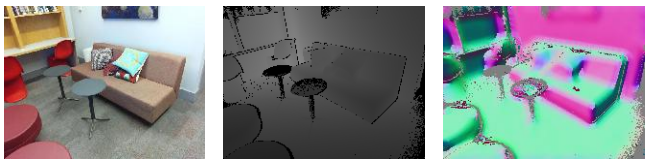


(a) SS-CNN-R6



(b) SS-CNN-R8

Fig. 3. Examples of SS-CNN-Rn with $n = 6, 8$. Note that the structure in Figure 2 is SS-CNN-R4. The main branch in SS-CNN-R4 has the same structure with Alexnet with 5 convolutional layers and 3 fully connected layers, while SS-CNN-R6 have 6 convolutional layers and 2 fully connected layers. SS-CNN-R8 is more special with its 8 convolutional layers and 2 additional fully connected layers in the main branch.



(a) RGB Image (b) Depth Image (c) Normal Vector

Fig. 4. An example of the RGB image in SUN RGB-D dataset, with its corresponding depth image and normal vector image.

parameters, on which normal vector is estimated. The normal vector is also rescaled to $[0, 255]$ and represented in an image with 3 channels. An example of the RGB image and its corresponding depth image and normal vector are given in Figure 4.

In this paper, we encode the depth representation as a combination of depth image and normal vector image, and then the RGB-D input has 7 channels for each image as shown in Figure 2.

B. Refinement of semantic segmentation with scene classification

Intuitively, scene classes can provide prior information about object occurrences. This idea could be used to further refine the performance of the semantic segmentation. For example, if a robot recognizes an environment correctly as a bedroom, then it is fair to expect a bed in the image, rather than a shower curtain. Based on the architecture of SS-CNN, we can conveniently incorporate the estimated scene probability to refine the performance of semantic segmentation.

As pointed out in (2), the softmax layer generates the estimated probability of scene classification. Let's denote $\boldsymbol{p_s} \in \mathbb{R}^{1\times M_s} = [p_s^1, \cdots, p_s^{M_s}]$ as the probability vector. Similarly, the probability of semantic segmentation is denoted as $\boldsymbol{p_o} \in \mathbb{R}^{H\times W\times M_o}$. Then the refinement process can be represented as follow:

$$\begin{aligned} \boldsymbol{p_{so}} &= \boldsymbol{p_s} \times \boldsymbol{W_{so}} \\ \widetilde{\boldsymbol{p}}_{\boldsymbol{o}} &= \boldsymbol{p_{so}} \otimes \boldsymbol{p_o} \end{aligned} \tag{5}$$

where $\boldsymbol{W_{so}} \in \mathbb{R}^{M_s\times M_o}$ is the refinement matrix learned from training data, $\boldsymbol{p_{so}}$ represents the prior probability of objects learned from estimated scene classes, which is propagated to $\boldsymbol{p_o}$ through multiplication with broadcasting (denoted as $\otimes$ in this paper), i.e. broadcast the M_o values in $\boldsymbol{p_{so}}$ to each score map in $\boldsymbol{p_o}$ respectively. The refinement process is illustrated in Figure 5.

For the refinement matrix $\boldsymbol{W_{so}}$, it is constructed based on the scene-object co-occurrence distribution in training dataset. Rather than directly deciding the refinement matrix from the object frequency, we propose to construct $\boldsymbol{W_{so}}$ in a way similar to term frequency-inverse document frequency (tf-idf). Inspired by the inverse document frequency term in tf-idf, how important an object is in a scene is also considered. For example, the object classes "wall" and "floor" are most common ones and almost appear in every scene. When we want the robot to finish a certain task such as "find the bowl in the kitchen", these common classes are less meaningful in the context of semantic segmentation, while the training process actually pays more attention to these classes because of their large amount of training samples.

Let's first construct the original term frequency matrix $\boldsymbol{f} \in \mathbb{R}^{M_s\times M_o}$, where f_{ij} denotes the count of object j occurs in scene i. And then the term frequency is normalized as:

$$tf_{ij} = log(1 + f_{ij}) \tag{6}$$

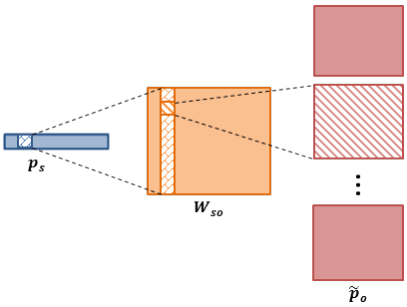


Fig. 5. Illustration of refinement process.

In this paper the inverse document frequency is constructed as:

$$idf_j = \frac{M_s}{\sum_{i=1}^{M_s} tf_{ij}} \tag{7}$$

Different from the common choice which constructs the idf term taking the number of scene classes containing the jth object as denominator, we measure the idf of each object as the inverse of mean term frequency of this object. For example, in the SUN RGB-D dataset, both classes "floor" and "bag" appear in almost of the scenes in training dataset, but the mean term frequency of "bag" is apparently smaller than that of "floor" as the appearance times of "bag" in some scene classes is tiny. Thus we still regard the "bag" as the object we need pay attention to.

Finally, the w_{ij} in weight matrix $\boldsymbol{W_{so}}$ is constructed by the multiplication of these two terms with normalization as:

$$w_{ij} = log(1 + tf_{ij} \times idf_j) \tag{8}$$

where $i = 1, \cdots, M_s$, $j = 1, \cdots, M_o$. If $w_{ij} = 0$, we set $w_{ij} = 1e^{-2}$ in case that the training dataset cannot exactly represent scene-object occurrences of the test dataset.

IV. Experiments and Results

We first train and validate our SS-CNN on the SUN RBG-D dataset [5], which is an indoor dataset with 10335 RGB-D images in all. In [5], the benchmark of scene classification is conducted on a subset of the dataset, which is composed of 19 scene classes with more than 80 samples, while the benchmark of semantic segmentation is conducted on the whole dataset with 45 scene classes. For these two tasks, their corresponding datasets are named as S_{19} and S_{45} respectively as described in Table I, where the split configuration is provided in the toolbox of SUN RGB-D dataset[1]. To make a fair comparison, we also validate the scene classification performance on S_{19} and validate the semantic segmentation performance on S_{45} in this paper. For both cases, SS-CNN is trained with only the training images in SUN RGB-D dataset, without other data augmentation.

On top of the model trained and validated on SUN RGB-D dataset, we experimentally test the performance of SS-CNN on a set of test images collected in a building of our university using a mobile robot.

[1]http://rgbd.cs.princeton.edu.

TABLE I
Summary of SUN RGB-D dataset.

Task	Dataset	#Train	#Test	#All
Scene classification	S_{19}	4845	4659	9504
Semantic segmentation	S_{45}	5285	5050	10335

A. Experimental setup

During the training process, we resize both input images and semantic segmentation ground truth to 210×158 for computation efficiency. Let's denote the resized image datasets as $\hat{S}_{19}$ and $\hat{S}_{45}$ respectively.

To predict the pixel-wise labels in the semantic segmentation branch, we construct our SS-CNN based on a slightly modified Alexnet. The receptive field of the original Alexnet is 224×224, with pixel stride 32. Intuitively, large stride leads to coarse semantic segmentation results. Smaller stride is obviously required for semantic segmentation in our work since the image size we use is 210×158. In [16], the author implemented a fusion technique named "deep jet" for finer segmentation results. Instead of fusing results from multiple layer such as using "deep jet", we choose to slightly modify the configuration of Alexnet to directly get a network with stride 16 and receptive field 81×81. The rationale is this paper focuses on validating the effectiveness of semantic regularization on scene classification rather than obtaining a finer semantic segmentation. Besides, the length of `fc7` is reduced to 512 while the original length in Alexnet is 4096, which is also illustrated in Figure 2.

Our network is implemented on Caffe [19], a popular deep learning framework. For model learning, we use stochastic gradient descent with momentum to train the randomly initialized network, and the size of each minibatch is 20. The learning rate is fixed as 10^{-4} during the training process, and the momentum is fixed as 0.9. Similar to the common configuration in training deep neural networks, we use a weight decay of 5^{-4}, and double the learning rate of biases. We also employ dropout in the fully connected layers. We are planning to release our model in the near future.

B. Evaluation of semantic regularization

To evaluate the effectiveness of our semantic regularization, we first make a comparison between our SS-CNN-Rn and the basic Alexnet, where layer 1 to layer n in SS-CNN-Rn are regularized by semantic segmentation cost as introduced in Section III-A. Both SS-CNN-Rn and the original Alexnet are trained with the same training data in $\hat{S}_{19}$, and only RGB images are considered in this evaluation experiment.

The models we compared include SS-CNN-R2, R4, R6 and R8. Comparison result is shown in Figure 6, which demonstrates SS-CNN-Rn considerably outperforms the original Alexnet for each n. It reveals the generalization ability on scene classification is significantly improved with the regularization of semantic segmentation.

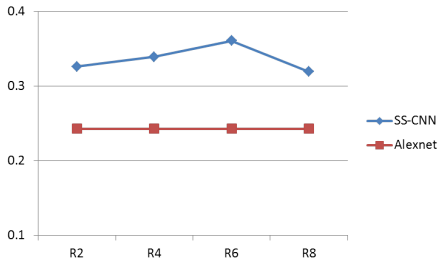
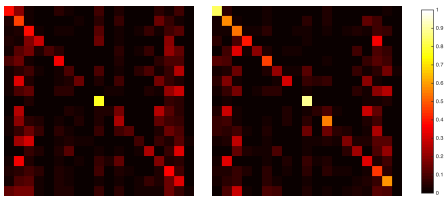


Fig. 6. Comparison of SS-CNN-Rn on scene classification with $n = 2, 4, 6, 8$. The performance on the original Alexnet is given as a baseline.



(a) Alexnet (b) SS-CNN-R6

Fig. 7. Confusion matrices of Alexnet and SS-CNN-R6, both are trained with only training images in SUN RGB-D dataset.

By analyzing the influence of n in SS-CNN-Rn, we can further gain insights in how the generalization ability is enhanced with the regularization. Figure 6 shows the performance of SS-CNN-Rn experiences slight promotion with increased n from 2 to 6. It can be explained that when n is small, the regularization is added only to the early layers of main branch, which means the low-level features are regularized. As n is increasing, the features being regularized become more abstract, and even object-level features would start to emerge in higher layers with the regularization on semantic segmentation. However, it can be seen that the performance of SS-CNN-R8 has an apparent drop. One possible reason is that SS-CNN-R8 directly classifies the scene based on semantic segmentation results, in which the performance of scene classification would suffer from the misclassification of semantic segmentation. For better illustration, the confusion matrices of our best model SS-CNN-R6 and Alexnet are given in Figure 7, which demonstrate the scene classification result is considerably improved with the semantic regularization.

C. Validation on scene classification

In this section, we make comparison between our SS-CNN and the benchmark methods of scene classification on S_{19} [5], with 4845 training samples and 4659 validation samples as shown in Table I. Two additional baseline experiments are conducted, one takes object occurrence as feature vector, and the other is based on the original Alexnet. It is to be noted that our SS-CNN is trained and evaluated on the resized $\hat{S}_{19}$, which does not compromise the fairness in the task of scene classification. Following describes the techniques those are used for the comparisons.

- GIST [20] + SVM. GIST is a famous descriptor for modeling a scene image, which summarizes the gradient information of a given image. An RBF kernel SVM is employed for classification.
- Place-CNN [4] + SVM. As introduced in Section II, Place-CNN is pre-trained on 2.5 million scene images using Alexnet. Because of its pre-trained structure, Place-CNN is usually employed as a feature extraction method in scene classification applications and an additional classifier is utilized for classification. In [5], both Linear SVM and RBF Kernal SVM are considered to train and classify the Place-CNN features extracted from S_{19}, and the later achieves the state-of-the-art performance.
- Object occurrence + SVM. Assuming the ground truth of object occurrences is known in every image, each image can be represented by a binary encoded vector with length M_o, with 1 denotes the object is contained in the image and 0 otherwise, and M_o is the number of object classes in the whole dataset. A linear SVM is employed for classification.
- Alexnet. As both Place-CNN and SS-CNN are based on Alexnet, the performance of the original Alexnet is also evaluated as a baseline. The Alexnet we implemented is trained with only training images in $\hat{S}_{19}$ from randomly initialized weights. Unlike the separate feature extraction and classification required in Place-CNN model, we trained Alexnet directly to classify scene using the softmax classifier within the network architecture.
- SS-CNN. As suggested in Figure 6, SS-CNN-R6 is the best configuration and thus is employed in this comparison. Our SS-CNN-R6 is also trained with only training images in $\hat{S}_{19}$ and the softmax classifier is employed for classification.

Comparison results are given in Table II, where the accuracy is calculated as the mean accuracy of 19 scene classes. Except for the classification based on object occurrence, both RGB input and RGB-D input are considered for other methods. For the depth information, [5] adopt the HHA [18] representation in GIST feature and Place-CNN feature. HHA is composed of horizontal disparity, height above ground, and the angle information. As HHA requires inferring of the ground and the gravity direction, our depth representation in Section III-A is a more compact and effective choice as shown in Table II.

Seen from the baseline experiment based on object occurrences, experimental results show that the binary occurring feature significantly outperforms the hand-crafted feature GIST, and reaches a similar level to the Place-CNN and SS-CNN. It is to be noted that the dimension of the object occurrence feature is much lower than the feature extracted using the other methods. This experiment validates our hypothesis that the knowledge on object level has the potential to promote the performances of scene classification. On the other hand, the superior results achieved by Place-CNN and SS-CNN demonstrate the power of automatic feature learning.

Furthermore, it can be seen that Place-CNN gains a considerable promotion with pre-training on 2.5 million

TABLE II

SCENE CLASSIFICATION COMPARISON ON SUN RGB-D DATASET.

Model	Input	Acc (%)
GIST + RBF Kernel SVM [5]	RGB	19.7
	RGB + D	23.0
Place-CNN + Linear SVM [5]	RGB	35.6
	RGB + D	37.2
Place-CNN + RBF Kernel SVM [5]	RGB	38.1
	RGB + D	39.0
Object occurrence + Linear SVM	–	33.1
Alexnet	RGB	24.3
	RGB + D	30.7
SS-CNN-R6	RGB	36.1
	RGB + D	**41.3**

TABLE III

SEMANTIC SEGMENTATION COMPARISON ON SUN RGB-D DATASET.

Dataset	Model	Input	Acc (%)
$\hat{S}_{45}$	SS-CNN-R6	RGB	27.77
		RGB + D	37.03
		RGB + D refined	**41.76**
S_{45}	NN [5]	RGB + D	8.97
	SIFT Flow [5]	RGB + D	10.05
	KDES [5]	RGB + D	36.33
	SS-CNN-R6	RGB + D refined	**40.66**

TABLE IV

EXPERIMENTAL VALIDATION RESULTS ON DATASET COLLECTED IN OUR UNIVERSITY.

Class	#Sample	Acc (%)
Computer room	41	19.5
Conference room	29	13.8
Corridor	38	47.4
Kitchen	14	35.7
Office	94	63.8
Rest space	14	57.1
All	230	39.6

scene images compared to the original Alexnet trained with only SUN RGB-D dataset. For SS-CNN-R6 which is also trained on SUN RGB-D dataset with only 5 thousand training images, it achieves superior results taking advantage of the regularization on semantic segmentation, which is slightly better than the Place-CNN with our RGB-D input. It is well known that deep neural networks usually require a large number of training data, while our SS-CNN achieves superior results with much less training samples, which reinforcements our hypothesis that the scene classification performance could be enhanced by involving object-level information.

D. Validation on semantic segmentation and its refinement

We also evaluate the performance of the semantic segmentation, the regularizer, and its refined results. The dataset we use is S_{45} as shown in Table I, which has 37 object classes. The comparing results are shown in Table III, the accuracy is calculated as the mean accuracy of all 37 objects.

In Tabel III, we first compare the performances of SS-CNN-R6 on $\hat{S}_{45}$, i.e. the resized dataset. Results show that depth information significantly increases the mean accuracy of semantic segmentation. Then it is further refined to increase the mean accuracy. In particular, the accuracies on "chair", "ceiling" and "bookshelf" are significantly increased with refinement.

To make a fair comparison to the benchmark methods mentioned in [5], our predicted results on $\hat{S}_{45}$ is directly resized to S_{45}, which slightly effects the mean accuracy. The comparing methods are listed as follow:

- Nearest neighbor. A nonparameteric method, [5] first extracts features using the trained Place-CNN to represent each image, and the test image directly takes the ground truth of the nearest neighbor in feature space as its segmentation label.
- SIFT Flow [21]. Also a nonparameteric method which takes the SIFT flow matching algorithm to search the match images from dataset with available semantic segmentation.
- Kernel Descriptors (KDES) [22]. A state-of-the-art method which encodes the input with kernel descriptors and the contextual information is considered with superpixel MRF and segmentation tree.

As can be seen in Table III, on dataset S_{45}, we also achieve the state-of-the-art performance on semantic segmentation with the SS-CNN-R6. We illustrate some examples of our predicted semantic segmentation labels with their refined results in Figure 8.

E. Experimental validation

The experiments on the publicly available SUN RGB-D dataset demonstrate the effectiveness of our SS-CNN. To further validate the performance of SS-CNN in robotics related application, we conducted an experiment using our mobile robot. The robot moved around in one of our university buildings and collected 230 RGB-D images with an on-board Kinect V2, belonging to 6 scene classes.

For scene classification, we use the SS-CNN-R6 training on SUN RGB-D dataset to predict the scene classes in the collected images without retraining the network with images in the new environment. To adapt to the SS-CNN-R6, each collected image is also represented as catenation of RGB image, depth image and normal vector image. The predicted results are given in Table IV, where the mean accuracy of all these 6 classes are given at the bottom row. Figure 1 gives some example RGB images with their predicted labels. It is to be noted that some images in this dataset are challenging even for humans since the boundary between some scene classes is not very clear. The last row in Figure 1 gives two examples in this situation, the ground truth of these two images is "computer room" and "rest space" respectively, while they are denoted with "office" and "discussion area".

As can be seen from Table IV, the predicted results are in similar order to the validation results on SUN RGB-D in the completely new environment, which further demonstrates

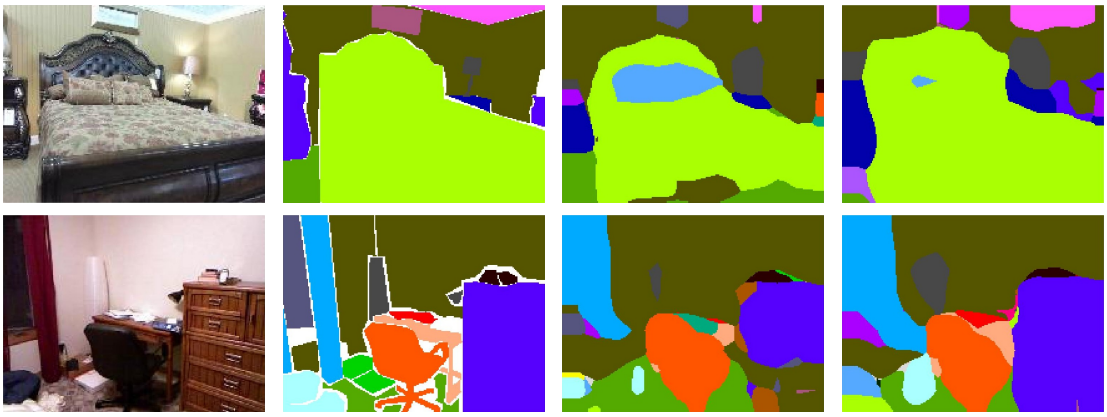


Fig. 8. Illustration of semantic segmentation and its refinement. From left to right: RGB input, ground truth of semantic segmentation, predicted results of SS-CNN-R6, refined predicted results of SS-CNN-R6. White color in the ground truth images denotes the background or confusing region and not considered either in training nor test. It can be seen that refinement not only plays the role of smoothing, but also "strengthens" some specific objects in the corresponding scene.

the generalization ability of our SS-CNN. Therefore, our SS-CNN has the potential to be implemented in real robotics applications without further training.

V. Conclusion

In this paper, we address the scene classification problem using deep learning methods with a much smaller amount of training images, by regularizing deep architecture with semantic segmentation. Experimental results validate the effectiveness of the regularization as SS-CNN achieved the state-of-the-art results on both scene classification and semantic segmentation on the publicly available SUN RGB-D dataset. Further experiments on our robot demonstrates the generalization ability of the proposed approach. For the future work, we would like to investigate the potential possibility in both horizontal and vertical dimensions, which means to couple more relevant tasks, and to find better architecture to incorporate the relations between these tasks.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[3] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pp. 512–519, IEEE, 2014.

[4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, pp. 487–495, 2014.

[5] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D : A RGB-D Scene Understanding Benchmark Suite," *CVPR*, 2015.

[6] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2036–2043, June 2009.

[7] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 702–709, IEEE, 2012.

[8] D. Lin, S. Fidler, and R. Urtasun, "Holistic scene understanding for 3d object detection with rgbd cameras," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1417–1424, IEEE, 2013.

[9] R. Luo, S. Piao, and H. Min, "Simultaneous place and object recognition with mobile robot using pose encoded contextual information," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 2792–2797, IEEE, 2011.

[10] J. G. Rogers III, H. Christensen, *et al.*, "A conditional random field model for place and object classification," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1766–1772, IEEE, 2012.

[11] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[12] 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 *Advances in neural information processing systems*, pp. 1378–1386, 2010.

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

[14] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[15] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412, 2015.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038*, 2014.

[17] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *arXiv preprint arXiv:1301.3572*, 2013.

[18] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Computer Vision–ECCV 2014*, pp. 345–360, Springer, 2014.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[20] A. Oliva and A. Torralba, "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, 2001.

[21] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pp. 1972–1979, IEEE, 2009.

[22] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2759–2766, IEEE, 2012.