Hindawi
Journal of Robotics
Volume 2019, Article ID 8591035, 12 pages
https://doi.org/10.1155/2019/8591035

Research Article

An Indoor Scene Classification Method for Service Robot Based on CNN Feature

Shaopeng Liu and Guohui Tian

School of Control Science and Engineering, Shandong University, Jinan, 250061, China

Correspondence should be addressed to Guohui Tian; g.h.tian@sdu.edu.cn

Received 5 November 2018; Revised 4 April 2019; Accepted 15 April 2019; Published 24 April 2019

Academic Editor: Keigo Watanabe

Indoor scene classification plays a vital part in environment cognition of service robot. With the development of deep learning, fine-tuning CNN (Convolutional Neural Network) on target datasets has become a popular way to solve classification problems. However, this method cannot obtain satisfying indoor scene classification results because of overfitting when scene training datasets are insufficient. To solve this problem, an indoor scene classification method is proposed in this paper, which utilizes CNN feature of scene images to generate scene category features to classify scenes by a novel feature matching algorithm. The novel feature matching algorithm can further improve the speed of scene classification. In addition, overfitting is eliminated by our method even though the training data is limited. The presented method was evaluated on two benchmark scene datasets, Scene 15 dataset and MIT 67 dataset, acquiring 96.49% and 81.69% accuracy, respectively. The experiment results showed that our method was superior to other scene classification methods in terms of accuracy, speed, and robustness. To further evaluate our method, test experiments on unknown scene images from SUN 397 dataset had been done, and the models based on different training datasets obtained 94.34% and 79.80% test accuracy severally, which proved that the proposed method owned good performance in indoor scene classification.

1. Introduction

Scene cognition is a key point of service robot cognition. Scene information can help improve robot service level. Indoor scene classification is one of the most import missions of service robot, which can enable the robot to provide different services according to different scenes.

A great deal of researches on indoor scene classification had been done. Traditional scene classification method was usually based on manual vision features such as SIFT (scale-invariant feature transform) [1] and SURF (speeded up robust features) [2]. Reference [3] utilized an improved SIFT feature named RootSIFT to build a BoW (Bag of Word) model and combined selective attention for scene classification. In [4], SPM (spatial pyramid matching) model was proposed based on BoW to classify scenes. Reference [5] structured CLM (codebookless model) model by extracting SURF feature of scene image, which obtained better accuracy on indoor scene classification. However, the mentioned vision features are low-level features of images without rich semantic information. It is hard to get satisfying results on complicated scene classification.

In recent years, deep learning has already made huge progress on image classification [6, 7], object detection [8–10], and so on. Deep learning methods especially CNN have become popular solutions for scene classification. A scene classification model was presented in [11] based on deep CNN feature. In [12] accuracy of scene classification was improved by transferring learning. Transferring learning is to fine-tune a pretrained model on a new dataset and the pretrained model has been well trained on large scale dataset. By this way, better accuracy can be obtained in comparison with training from scratch. In [13] CNN feature transferring was employed to classify scenes and got good results. From the cited references we can find that the deep learning methods own more excellent performances in scene classification than manual vision features. However, there are still some problems as follows. (1) A deep CNN model with extremely large parameters needs enough training data even if trained by transferring learning. (2) Training CNN needs powerful

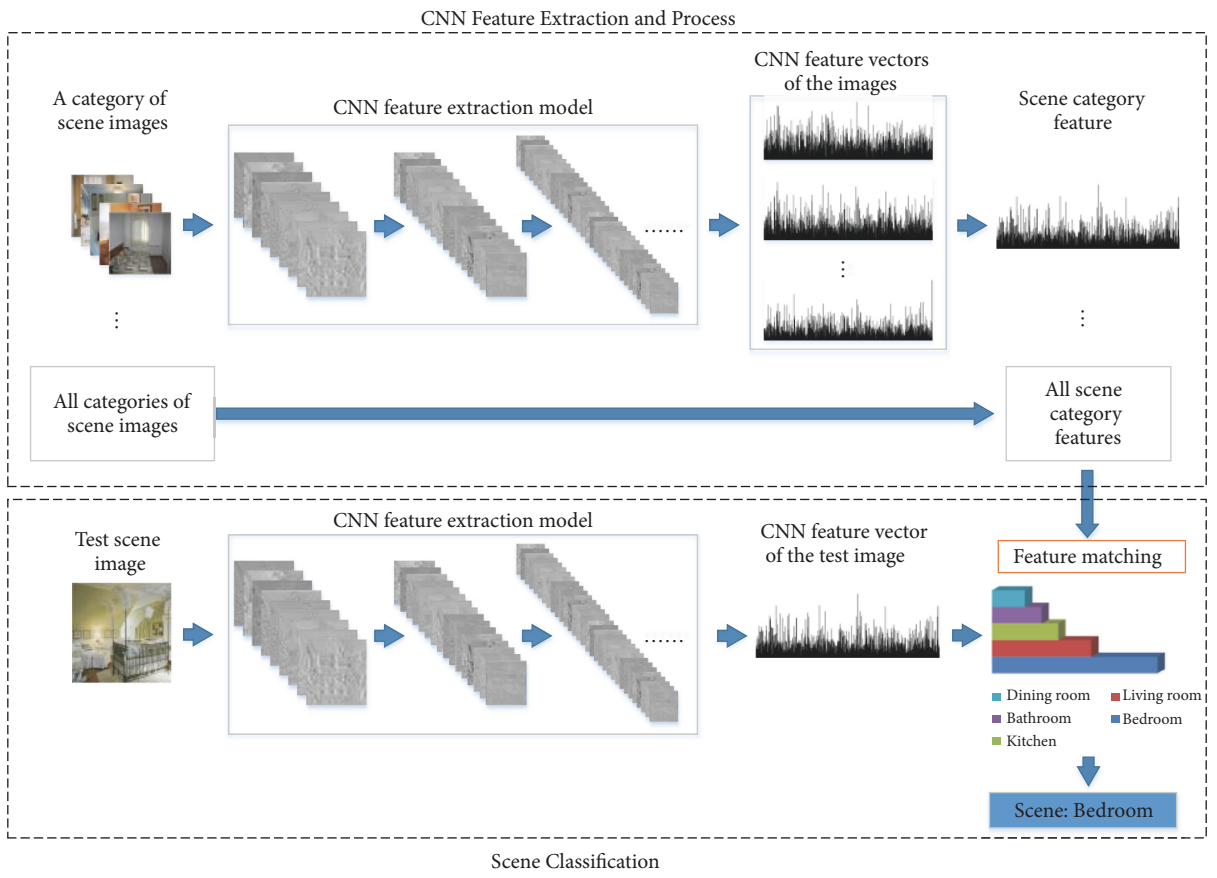


FIGURE 1: Overall framework of the proposed indoor scene classification method based on CNN feature. There are two parts in the framework. The first part is to generate scene category features by CNN feature extraction and process. The second step is to match CNN feature vector of the test scene image with the scene category features by a new feature matching algorithm to generate scores of different scenes. The largest score indicates the result of scene classification. For example, the category of the test scene image is bedroom with the highest score in Figure 1.

GPUs to speed up, which is expensive. (3) If the training dataset is insufficient, overfitting is around corner. Therefore, it is difficult to get satisfying results based on very limited indoor scene datasets by fine-tuning a pretrained CNN.

Aiming at the above problems, this paper proposes an indoor scene classification method for service robot based on CNN feature. Different from the general method of fine-tuning CNN, our method utilizes CNN feature of scene images to generate scene category features to classify indoor scenes by a new feature matching algorithm. The novel feature matching algorithm can further speed up the scene classification. Meanwhile overfitting can be eliminated by this method when training data is insufficient. The presented method was adequately estimated on two benchmark scene datasets, Scene 15 [4] dataset and MIT 67 [14] dataset, and tested on completely new scene images that are different from the training datasets.

2. Overall Framework

Essentially CNN is a kind of input to output mapping, which can learn a lot of mapping relationships between input and output and does not require any precise mathematical expression. CNN usually adopts alternating settings of convolution layer and sampling layer. The convolution layers are used to extract image features named CNN feature. CNN feature of network pretrained on large scale datasets includes abundant representation information. Therefore an indoor scene classification method for service robot based on CNN feature is proposed in this paper. The method contains two parts and overall framework is illustrated in Figure 1.

The first part is CNN feature extraction and process. A CNN feature extraction model is built by reconstructing a pretrained CNN model. The output of the CNN feature extraction model is one-dimensional feature vector with discriminative representation information. Then a category of scene images is processed by the model to create scene category feature that can generally represent this kind of scene. By this way, other scene category features can be obtained. This part is a learning process and the main purpose is to get the category features with high discrimination of various scenes for scene classification in the next part.

The second part is about scene classification. A test scene image is put into the same CNN feature extraction model to generate a CNN feature vector. Then the CNN feature vector is matched with the scene category features by a proposed feature matching algorithm, which calculates diverse scores

Table 1: Pretrained deep CNN models.

Model	Dataset	Network	Top-1 Accuracy
1	ImageNet11K+Places365	ResNet-50	0.3112
2	ImageNet11K+Places365	ResNet-152	0.3355
3	ImageNet11K	ResNet-152	0.4157

Table 2: Architectures of feature extraction models.

Layer name	Output size	Model 1	Model 2 and 3
		49-layer	151-layer
Conv1	112×112	$7 \times 7, 64$, stride 2	
Conv2_x	56×56	3×3 max pool, stride 2	
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
Conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
Conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	Average pool, flatten	

of each scene category. The scene category can be decided according to the maximum score.

3. Method Description

3.1. Obtaining Pretrained CNN Models. There are a lot of open source deep learning frameworks such as Theano [21], Caffe [22], and MXNet [23], which promote the development of deep learning. This paper is based on MXNet.

MXNet supports many kinds of programming languages and deep learning algorithms and provides diverse pretrained deep CNN models based on a variety of large scale datasets. Three types of pretrained model (shown in Table 1) are selected from MXNet in this paper. The selected deep CNN models are all ResNets [7] with different network layers. Model 1 and Model 2 were trained on a combined dataset including ImageNet11K [24] dataset and Places365 [25] dataset, achieving 0.3113 and 0.2255 top-1 accuracy, respectively. ImageNet11K dataset includes 11,221 category objects and 11,797,630 images totally. Places365 is dataset about scenes with 365 category scenes and 8,000,000 images. Model 3 was trained on ImageNet11K dataset. After training on these large scale datasets, the three models own powerful capacity to extract CNN feature.

3.2. Scene Category Feature. Firstly the CNN feature extraction model needs to be built based on the pretrained models by using flatten layer instead of softmax layer. Architectures of the feature extraction models are shown in Table 2 and ReLU activation functions are used in the models. Output of the CNN feature extraction model is a vector rich in semantic information. The dimensionality and length of the vector are 1 and 2048, respectively. The processes of generating scene category feature are listed as follows.

Suppose that the category number of a scene dataset is I and each category has j scene images.

Input. Scene images in training dataset.

Output. All scene category features.

Step 1. Put image j from scene category i into CNN feature extraction model to create a feature vector $V_{ij}^k = [v_{ij}^1, v_{ij}^2, v_{ij}^3, \ldots\ldots, v_{ij}^k]$ of the image. The $k = 2048$ is the length of the feature vector V_{ij}^k.

Step 2. Image CNN features $V_i = [v_{i1}^k, v_{i2}^k, v_{i3}^k, \ldots\ldots, v_{ij}^k]$ of all image in scene category i can be generated according to Step 1. Then the mean value of V_i is figured out by

$$\overline{V_I} = \frac{1}{j} \sum_{n=0}^{j} V_{in}^k \quad (1)$$

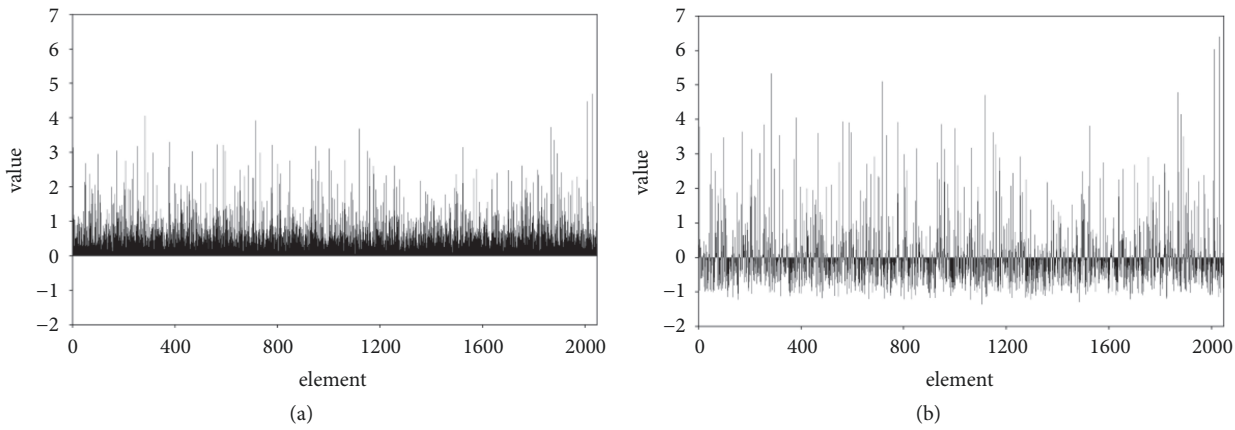


Figure 2: Visualization of scene category feature vector: (a) a raw vector, (b) a new vector normalized by Z-Score.

and $\overline{V_I}$ is the scene category feature vector of scene category i.

Step 3. Each scene category feature vector is generated by Step 2 and all scene category feature vectors are $V = [\overline{V_1}, \overline{V_2}, \overline{V_3} \ldots\ldots, \overline{V_I}]$.

Step 4. Scene category feature vectors V are normalized into $V^N = (\overline{V_1^N}, \overline{V_2^N}, \overline{V_3^N}, \ldots\ldots, \overline{V_I^N})$ by Z-Score

$$\overline{V_I^N} = \frac{\overline{V_I} - \mu}{\sigma} \tag{2}$$

where μ is the mean value of each scene category feature and σ is the standard deviation. If the original scene category feature vector is directly used for analysis, it will highlight the role of the vector with higher value in the comprehensive analysis and weaken the role of the vector with lower value. Therefore, in order to ensure the reliability of the results, the original vector needs to be standardized. And the improvement will be proved in subsequent experiments. Figure 2 shows element value changes of scene category feature vector after normalization.

After the aforesaid steps we can get all scene category feature vectors for scene classification. After that, put a test scene image into CNN feature extraction model and the CNN feature vector of the test scene image V_{test} is created, which also needs to be normalized into V_{test}^N by Z-Score.

3.3. Scene Category Feature Matching. Scene classification results are figured out by measuring the similarity between scene category feature V^N and CNN feature of test scene image V_{test}^N. Therefore feature matching is a key point. Service robot should possess the capacity of real-time scene classification, which requires the feature matching algorithm to be fast enough. Some common feature vector matching algorithms are compared and analysed as follows, and a new feature vector matching algorithm is proposed in this paper.

The difference between vector $X = (x_1, x_2, x_3, \ldots\ldots x_n)$ and vector $Y = (y, y_2, y_3, \ldots\ldots y_n)$ can be expressed by distance measure or similarity measure. Familiar distance measure is Euclidean distance.

$$ED_{dist}(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

Larger value of Euclidean distance means larger difference between the two vectors. The frequently used similarity measures are Pearson correlation coefficient $P(X, Y)$ and cosine similarity $\cos(X, Y)$.

$$\begin{aligned} P(X, Y) &= \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \\ &= \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sqrt{\sum_{i=1}^{n} (x_i - \mu_X)^2} \sqrt{\sum_{i=1}^{n} (y_i - \mu_Y)^2}} \end{aligned} \tag{4}$$

$$\cos(X, Y) = \frac{X \bullet Y}{|X|\,|Y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{5}$$

The output range of $P(X, Y)$ and $\cos(X, Y)$ are both $[-1, +1]$. The value close to 1 means that the two vectors are more similar. Suppose that vectors X and Y are normalized by Z-Score.

$$N_{Z-S}(X) = \frac{x_i - \mu_i}{\sigma_X} \tag{6}$$

Put (6) into (5) to get (7).

$$\begin{aligned} \cos(X, Y) &= \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \\ &= \frac{N_{Z-S}(X) \bullet N_{Z-S}(Y)}{\left|N_{Z-S}(X)\right| \left|N_{Z-S}(Y)\right|} \end{aligned} \tag{7}$$

Equation (4) can be simplified in (8) since the mean value of X and Y is 1 and the standard deviation is also 1.

$$P(X,Y) = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sqrt{\sum_{i=1}^{n}(x_i-\mu_X)^2}\sqrt{\sum_{i=1}^{n}(y_i-\mu_Y)^2}} = \frac{N_{Z-S}(X) \bullet N_{Z-S}(Y)}{\left|N_{Z-S}(X)\right|\left|N_{Z-S}(Y)\right|} \tag{8}$$

It can be found that Pearson correlation coefficient is equal to cosine similarity when the input vectors are normalized.

From variance formula (9) we can get (10).

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu_X)^2}{n-1}} \tag{9}$$

$$(n-1)\sigma_X^2 = \sum_{i=1}^{n}(x_i-\mu_X) \tag{10}$$

$\sum_{i=1}^{n} x_i^2$ can be simplified by formula (9) and (10) into (11).

$$\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n}(x_i-0)^2 = \sum_{i=1}^{n}(x_i-\mu_X)^2 = (n-1)\sigma_X^2 = n-1 \tag{11}$$

When n is big enough, $n-1$ can be seen as n, so $\sum_{i=1}^{n} x_i^2 = n$. In the same way, $\sum_{i=1}^{n} y_i^2 = n$. There is a direct linear relationship between the square of Euclidean distance and Pearson correlation coefficient [15]. With these conditions, the square of Euclidean distance can be unfolded as follows.

$$\begin{aligned} ED_{dist}(X,Y)^2 &= \sum_{i=1}^{n}(x_i-y_i)^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\ &= 2n - 2\sum_{i=1}^{n} x_i y_i \\ &= 2n\left(1 - \frac{1}{n}\sum_{i=1}^{n} x_i y_i\right) \\ &= 2n\left(1 - \frac{(1/n)\sum_{i=1}^{n}(x_i-0)(y_i-0)}{1 \bullet 1}\right) \\ &= 2n\left(1 - \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X\sigma_Y}\right) \\ &= 2n(1-P(X,Y)) \end{aligned} \tag{12}$$

In the same manner, Euclidean distance is equal to Pearson correlation coefficient as well as cosine similarity. Although the three feature matching algorithms have the same property, they own different computation speeds. They both need to extract a root which is time-consuming. Inspired by formula (12), a new feature matching algorithm $S(X,Y)$ is presented without extracting a root. The new algorithm can improve calculated speed which will be proved by experiments.

$$S(X,Y) = 1 - \frac{1}{2n}\sum_{i=1}^{n}(x_i-y_i)^2 \tag{13}$$

The CNN feature vector V of a test scene image will be matched with scene category feature vectors $V^N = (\overline{V_1^N}, \overline{V_2^N}, \overline{V_3^N}, \ldots\ldots, \overline{V_I^N})$ by the proposed feature matching algorithm $S(X,Y)$. The output score matrix is as follows.

$$score = \begin{bmatrix} S\left(V, \overline{V_1^N}\right) \\ S\left(V, \overline{V_2^N}\right) \\ S\left(V, \overline{V_3^N}\right) \\ \cdots \\ S\left(V, \overline{V_I^N}\right) \end{bmatrix} = \begin{bmatrix} score_1 \\ score_2 \\ score_3 \\ \cdots \\ score_I \end{bmatrix} \tag{14}$$

The largest element in the score matrix is the index of scene category. By this way we can get the category of the test scene image.

4. Experiments and Analysis

4.1. Datasets. Our method was evaluated on two benchmark scene datasets, Scene 15 dataset and MIT 67 dataset. Scene 15 dataset includes 15 categories of outdoor and indoor scenes. We selected 5 categories of scenes (living room, store, kitchen, PAR office, and bedroom) shown in Figure 3 because our study object service robot mainly worked indoors. There are 1,245 grayscale images in the 5 categories. MIT 67 dataset contains 67 types of indoor scenes and 15,620 RGB images totally. All of 67 categories of indoor scene image in MIT 67 are used to train the proposed models. Some examples of scene image in MIT 67 dataset are demonstrated in Figure 4. Experiment designs were specified as 5-fold cross-validation in order to make the experiment results convincible and repeatable. Each category scene images were divided into 5 parts; 1 part was used to created scene category features and the rest were tested. All images were resized to 224 × 224 pixels.

4.2. Experiment Results and Analysis. The proposed method was written in Python using MXNet deep learning framework and run on a PC. The PC operating system was Ubuntu 16.04.4 with Intel i5-6500 CPU, 32G memory, and 1 NVIDIA GTX 1080 graphics card. In order to fully test the performance of our method, some experiments were carried out as follows.

(1) The Impact of Normalization on Classification Results. This experiment was done to test if Z-Score normalization could improve scene classification accuracy. Two groups were set to make a comparison; one used Z-Score to process the CNN feature vector and the other did not. We utilized cosine similarity to match features since the three matching algorithms were not equivalent without Z-Score. The experiment results

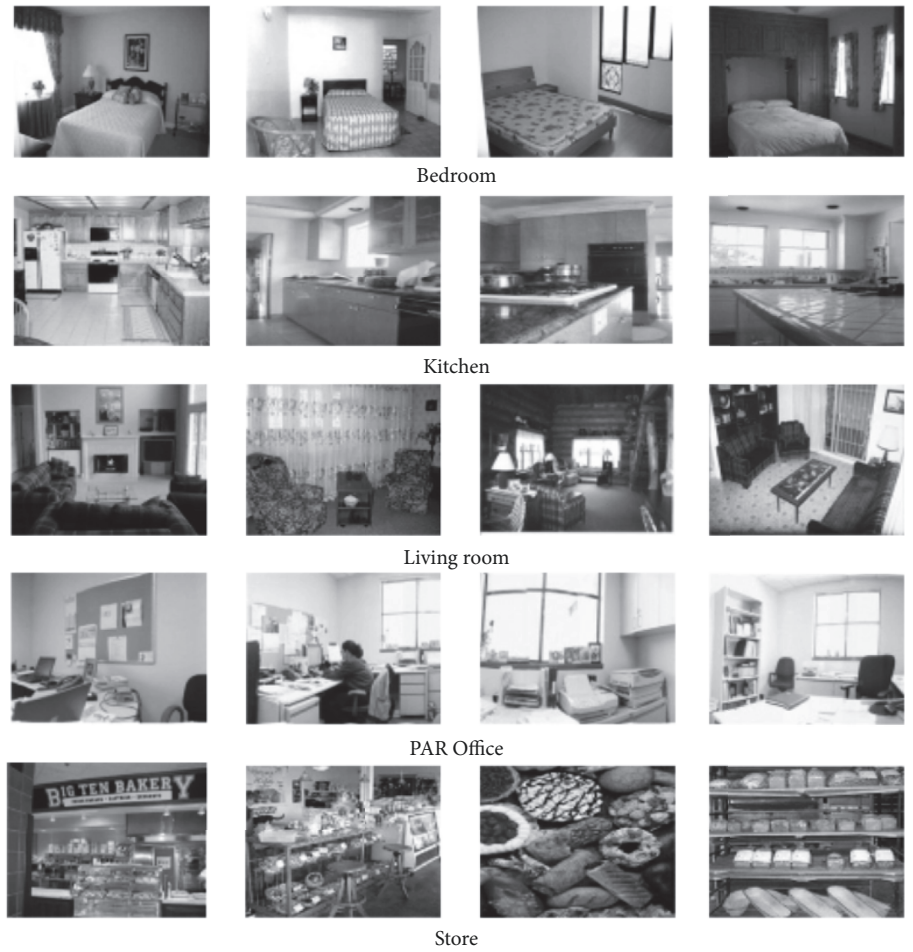


Figure 3: Some examples of indoor scene images in Scene 15 dataset.

Figure 4: Some examples of indoor scene image in MIT 67 dataset.

Table 3: Results of scene classification with or without Z-Score normalization.

Dataset	Pre-trained Model	Without Z-Score	With Z-Score
Scene 15	1	0.9411	0.9438
	2	0.9601	0.9649
	3	0.9522	0.9596
MIT 67	1	0.7630	0.7726
	2	0.7963	0.8169
	3	0.7762	0.7842

Table 4: Comparison between the proposed feature matching algorithm and correlative algorithms.

Dataset	Pre-trained Models	Speed (s/image)			
		ED	CS	PCC	Proposed algorithm
Scene 15	1	0.00705	0.0323	0.0362	0.00675
	2	0.0191	0.0431	0.0451	0.0151
	3	0.0189	0.0425	0.0451	0.0150
MIT 67	1	0.00807	0.359	0.396	0.00747
	2	0.0168	0.371	0.406	0.0158
	3	0.0171	0.367	0.405	0.0160

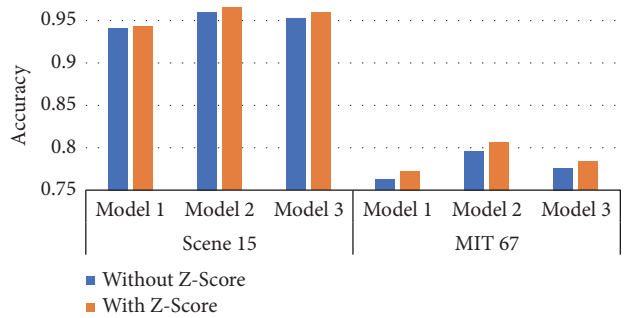


Figure 5: Scene classification accuracy without or with Z-Score normalization.

are listed in Table 2 and a histogram is provided in Figure 5 for comparison. From Table 3 and Figure 5 we can see that using Z-Score normalization can obtain better accuracy on both Scene 15 dataset and MIT 67 dataset with 3 different pretrained models.

(2) Computational Speed Comparison of Feature Matching Algorithms. Service robot should be able to recognize scenes in real time, so the speed of indoor scene classification is significant. In order to verify the advantage of the proposed feature matching algorithm on computing time, the following experiments were carried out in contrast with other algorithms. Euclidean distance (ED), Cosine similarity (CS), Pearson correlation coefficient (PCC), and the proposed feature matching algorithm were, respectively, tested on the two datasets based on the three different pretraining models. The scene classification results are shown in Table 4.

From the results in Table 4, we can see that CS and PCC have similar processing speeds, which are much slower than ED and our algorithm. Compared with ED, the proposed algorithm is faster. In addition, the scale of datasets and the layers of the pretrained models affect the scene classification speed. By contrast our feature matching algorithm is able to meet the service robot demand of real-time scene classification.

(3) Contrast with Transferring Learning on CNNs. A transfer learning strategy was used; fully connected layers of the pretrained CNN models were changed according to the number of scene categories and fine-tuned on the datasets. The training parameter sets were learning rate 0.0001, batch size 32, and epoch 100 (MIT 67) and 200 (Scene 15). Figure 6 shows the training curves of fine-tuning CNNs on Scene 15 dataset and MIT 67 dataset. Results of the comparison are listed in Table 5.

The training accuracy curves in Figure 6 indicate that the CNN models can perfectly fit in the training dataset and obtain about 100% training accuracy in Scene 15 dataset since the number of categories is small. However, it is hard to get the same high accuracy on test dataset and overfitting appears. In MIT 67 dataset, larger scene categories and deficient training dataset result in unstable accuracy curves and overfitting.

Table 5: Result comparison between fine-tuned CNN models and the proposed method.

Dataset	Pre-trained Model	Fine-tuned CNN Models	Our Method
Scene 15	1	0.8551	0.9438
	2	0.8674	0.9649
	3	0.8476	0.9596
MIT 67	1	0.7522	0.7726
	2	0.7653	0.8169
	3	0.7301	0.7842

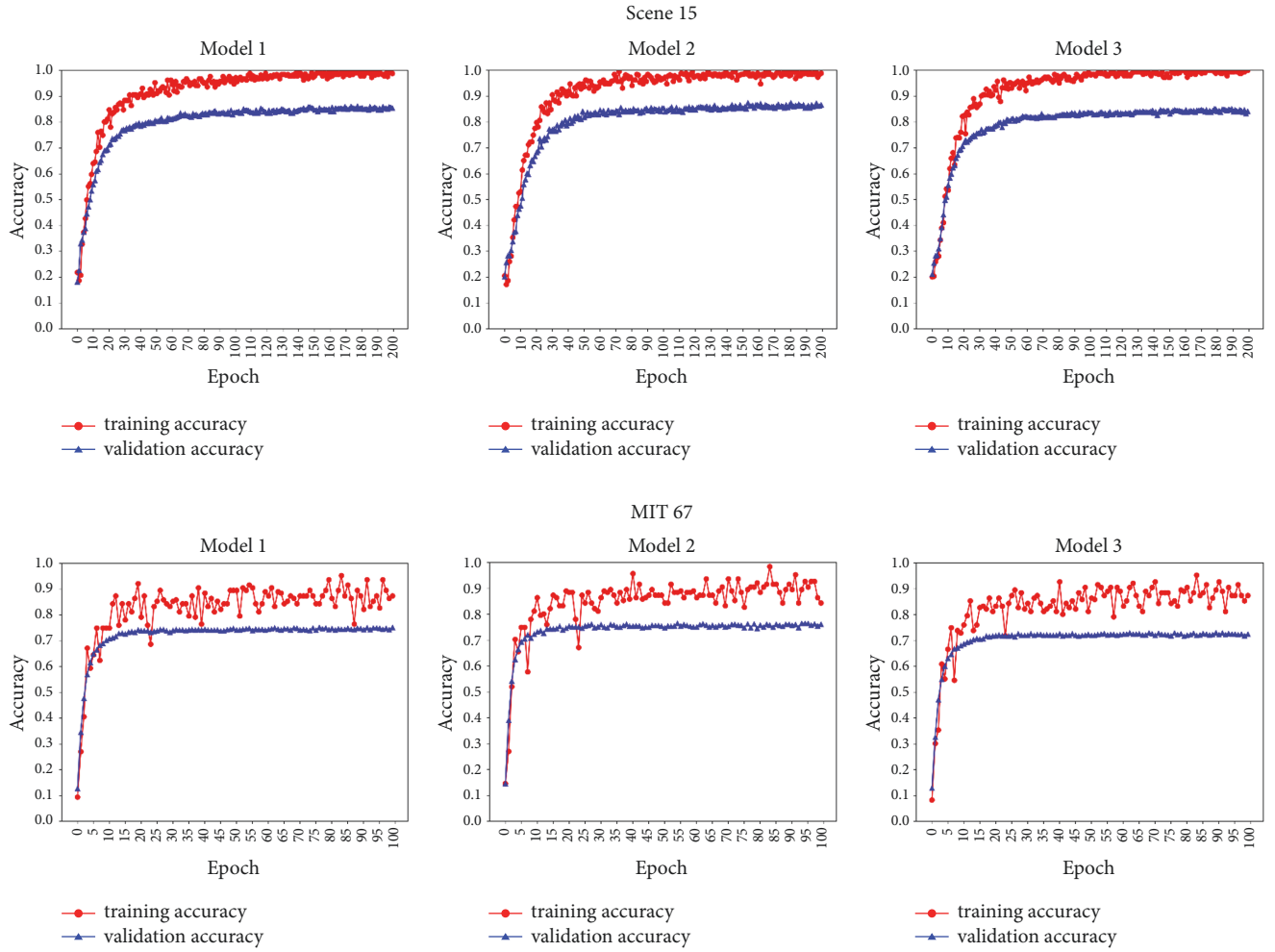


Figure 6: The training curves of fine-tuning CNNs on Scene 15 dataset and MIT 67 dataset with the pretrained models.

From Table 5 we can see that the proposed method can get better results without overfitting based on each pretrained model on each dataset.

(4) Contrast with Other Advanced Scene Classification Methods. In order to verify the performance of our method, other advanced indoor scene classification methods were used as references to carry out the following comparison tests. Table 6 demonstrates indoor scene classification performance of different methods. Confusion matrixes created by CLM+SVM [5] and our method are shown in Figure 7. Performance contrasts on MIT 67 dataset are listed in Table 7, and Figure 8 shows scene classification confusion matrixes of MIT 67 dataset on three pretrained models based on our method.

In Table 6 our method gets higher scene classification accuracy and better efficiency than other methods. Scene classification confusion matrixes in Figure 7 indicate that our method is more robust than CLM+SVM in [5]. For classification of a large number of scenes in MIT 67 dataset, it can be observed from Table 7 that our method with model 2 obtains better results than other methods. The classification confusion matrixes of MIT 67 dataset in Figure 8 demonstrate the robustness of our method.

Table 6: Comparison between the proposed method and correlative methods on Scene 15 dataset.

Method		Accuracy of Each Scene					Average Accuracy	Speed (s/image)
		Bedroom	Kitchen	Living room	PAR office	Store		
Method in [4]		0.65	0.60	0.70	0.65	0.95	0.71	0.36
Method in [15]		0.95	0.75	0.9833	0.7333	0.9333	0.87	0.51
Method in [5]		0.95	0.80	0.95	0.8667	0.9833	0.9104	0.36
Our method	Model 1	0.95	0.90	0.96	0.97	0.94	0.9438	0.00675
	Model 2	0.97	0.915	0.98	0.99	0.97	**0.9649**	0.0151
	Model 3	0.98	0.92	1.00	0.98	0.92	0.9596	0.0150

Table 7: Comparison between the proposed method and other advanced methods on MIT 67 dataset.

Method		Accuracy of Some Scenes					Average Accuracy
		Bedroom	Kitchen	Living room	PAR office	Store	
Method in [16]		0.413	0.289	0.521	0.505	0.185	0.389
Method in [17]		0.453	0.277	0.538	0.542	0.252	0.429
Method in [3]		0.47	0.276	0.542	0.587	0.235	0.443
Method in [18]		-	-	-	-	-	0.7286
Method in [19]		-	-	-	-	-	0.7963
Method in [20]		-	-	-	-	-	0.8075
Our method	Model 1	0.88	0.68	0.93	0.97	0.63	0.7726
	Model 2	0.93	0.75	0.95	0.97	0.70	**0.8169**
	Model 3	0.90	0.70	0.93	0.97	0.67	0.7842

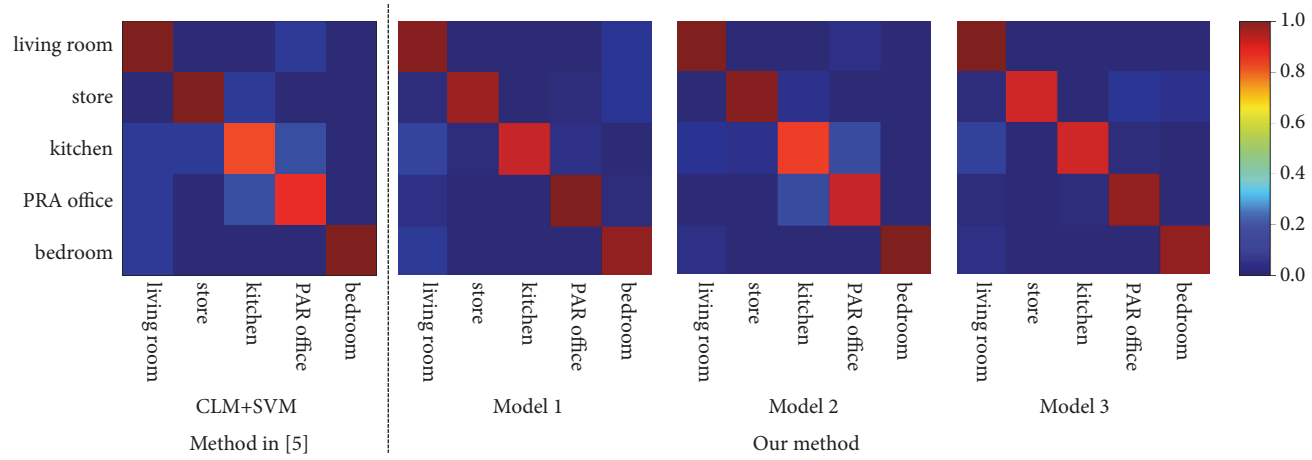


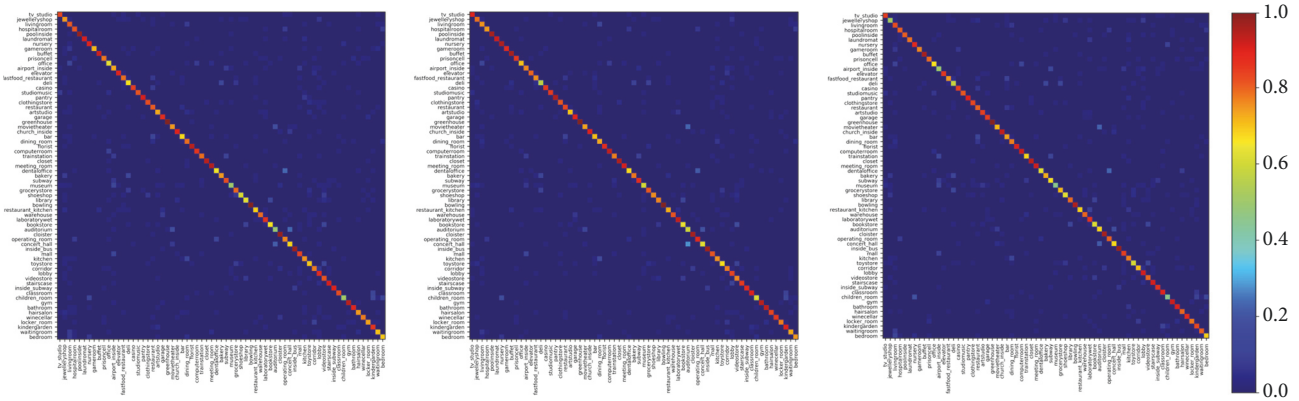
Figure 7: Comparison of scene classification confusion matrixes between the method in [5] and our method.

(5) Test Experiments on Different Scene Data. To further evaluate the indoor scene classification ability of our method, a completely new scene dataset, SUN 397 [26], which was different from the training data source, was used for test experiments. There are 397 well-sampled categories with 130,519 RGB scene images in SUN 397. All test images were resized to 224×224 pixels. To test the models trained on Scene 15 dataset, data of the same 5 categories with 1,336 indoor scene images were selected from SUN 397, and each scene had about 267 images. For the models trained on MIT 67 dataset, data of the same 67 categories with 15,937 indoor scene images were used to make the test experiments, and there were approximately 237 scene images in each category. Test results are list in Table 8. To further prove the performance of our method, scene classification test confusion matrixes are shown in Figure 9 (the models trained on Scene 15) and Figure 10 (the models trained on MIT 67).

From the test results in Table 8 a conclusion can be drawn that our method owns good ability to classify indoor scenes on a completely different source scene data, which proves that the proposed feature matching algorithm refers to the content distance between different indoor scenes. Although the models were trained on grayscale images from Scene 15 dataset, they can obtain more than 92% accuracy with different pretrained models, which demonstrates more evidence that our method is based on the content and semantics

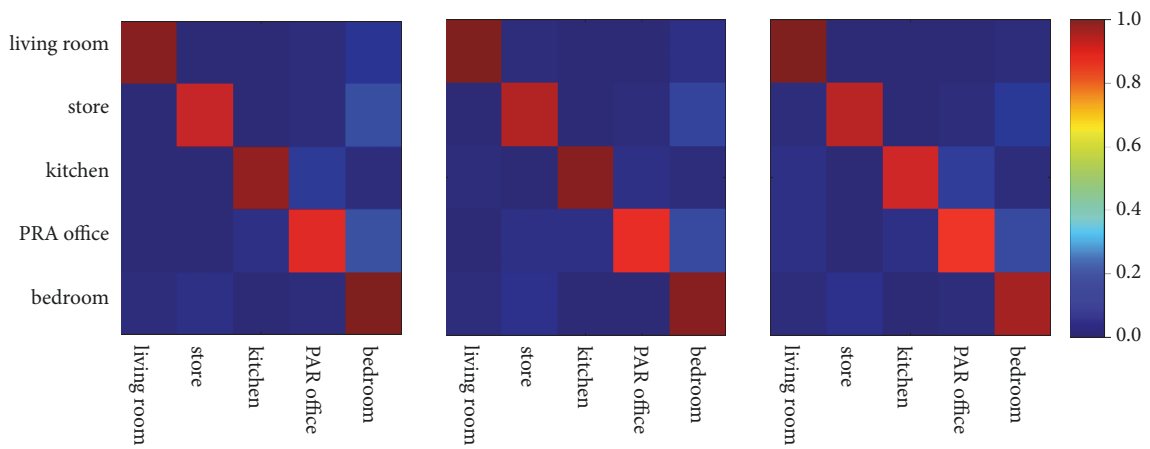
Table 8: Test results on SUN 397 dataset with diverse trained models.

Training Dataset	Categories of Test Scenes	Pre-trained Model	Test Accuracy
Scene 15	5	1	0.9219
		2	0.9434
		3	0.9353
MIT 67	67	1	0.7591
		2	0.7980
		3	0.7788



Model 1 Model 2 Model 3

Figure 8: Scene classification confusion matrixes of our method on MIT 67 dataset.



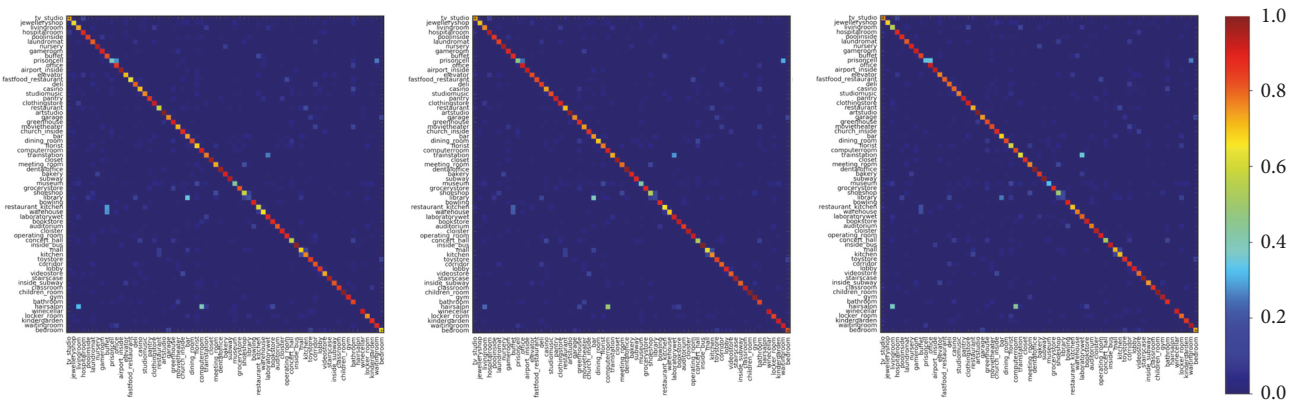
Model 1 Model 2 Model 3

Figure 9: Scene classification test confusion matrixes by the models trained on Scene 15 dataset.

of image to make scene classification instead of lower-end abstract features such as pixel, colour, and edge descriptor. In addition, Figures 9 and 10 show the test confusion matrixes of scene classification by various models trained on Scene 15 and MIT 67 datasets severally, which depicts the robustness of our method on test scene dataset.

5. Conclusions

In this paper an indoor scene classification method for service robot based on CNN feature is proposed. We utilize CNN feature of scene images to generate scene category features to classify indoor scenes by an improved feature matching algorithm. The novel feature matching algorithm can further speed up the scene classification. The presented method is adequately evaluated on two benchmark scene datasets, Scene 15 dataset and MIT 67 dataset. Compared with general method fine-tuning CNN on training dataset, this method can obtain satisfying accuracy without overfitting on a small amount of training dataset and does not need to be trained repeatedly. In contrast to other indoor scene classification methods, the scene classification results have



Model 1 Model 2 Model 3

Figure 10: Scene classification test confusion matrixes by the models trained on MIT 67 dataset.

been greatly improved in terms of accuracy, classification speed, and robustness by our method. The experiment results show that this method has good performance in indoor scene classification and can meet the task requirements of service robot indoor scene classification. Nowadays, with the continuous development of computer hardware and cloud robots, the computing capacity of service robots has been greatly improved. Our next step is to further improve scene cognition ability of service robots based on deep learning methods.

Data Availability

The data used to support the findings of this study are available from the public scene datasets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is Supported by National Natural Science Foundation of China (U1813215 and 61773239) and the Taishan Scholars Program of Shandong Province.

References

[1] M. Brown and S. Susstrunk, "Multi-spectral SIFT for scene category recognition," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 177–184, Colorado Springs, CO, USA, June 2011.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] J. Niu, X. Bu, K. Qian, and Z. Li, "An indoor scene recognition method combining global and saliency region features," *Jiqiren/Robot*, vol. 37, no. 1, pp. 122–128, 2015.

[4] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2169–2178, New York, USA, June 2006.

[5] P. Wu, Y. Li, F. Yang, L. Kong, and Z. Hou, "A CLM-Based Method of Indoor Affordance Areas Classification for Service Robots," *Jiqiren/Robot*, vol. 40, no. 2, pp. 188–194, 2018.

[6] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 770–778, Las Vegas, Nev, USA, June 2016.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 779–788, July 2016.

[9] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '17)*, pp. 6517–6525, Honolulu, Hawaii, USA, 2017.

[10] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9905, pp. 21–37, 2016.

[11] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 487–495, Montréal, Canada, December 2014.

[12] T. Akilan, Q. M. J. Wu, A. Safaei, and W. Jiang, "A late fusion approach for harnessing multi-CNN model high-level features," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 566–571, Windsor, Canada, October 2017.

[13] L. Zheng, Y. Zhao, S. Wang et al., "Good practice in CNN feature transfer," https://arxiv.org/abs/1604.00133.

[14] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops '09)*, pp. 413–420, Miami, USA, June 2009.

[15] Q. Wang, P. Li, L. Zhang, and W. Zuo, "Towards effective codebookless model for image classification," *Pattern Recognition*, vol. 59, pp. 63–71, 2016.

[16] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb, "Reconfigurable models for scene recognition," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2775–2782, Piscataway, USA, June 2012.

[17] L. Zhou, Z. Zhou, and D. Hu, "Scene classification using multi-resolution low-level feature combination," *Neurocomputing*, vol. 122, pp. 284–297, 2013.

[18] M. Dixit, . Si Chen, . Dashan Gao, N. Rasiwasia, and N. Vasconcelos, "Scene classification with semantic Fisher vectors," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2974–2983, Boston, MA, USA, June 2015.

[19] P. Tang, H. Wang, and S. Kwong, "G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition," *Neurocomputing*, vol. 225, pp. 188–197, 2017.

[20] S. Bai and H. Tang, "Categorizing scenes by exploring scene part information without constructing explicit models," *Neurocomputing*, vol. 160-168, no. 281, 2018.

[21] The Theano Development Team, R. AI-Rfou, G. Alain et al., "Theano: A Python framework for fast computation of mathematical expressions," https://arxiv.org/abs/1605.02688.

[22] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the ACM Conference on Multimedia (MM '14)*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.

[23] T. Chen, M. Li, Y. Li et al., "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems," https://arxiv.org/abs/1512.01274.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[25] B. Zhou, A. Lapedriza, J. Xiao et al., "Learning deep features for scene recognition using places database," in *International Conference on Neural Information Processing Systems, NIPS 2014*, pp. 487–495, Daegu, Korea, 2014.

[26] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: large-scale scene recognition from abbey to zoo," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3485–3492, San Francisco, Calif, USA, June 2010.