



A single-stream adaptive scene layout modeling method for scene recognition

Qun Wang^{1,2,3,4} · Feng Zhu^{1,2,3} · Zhiyuan Lin^{1,2,3,4} · Jianyu Wang^{1,2,3,5} · Xiang Li⁶ · Pengfei Zhao^{1,2,3,4}

Received: 7 March 2023 / Accepted: 25 March 2024 / Published online: 27 April 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Scene recognition has been the foundation of research in computer vision fields. Because scene images typically are composed of specific regions distributed in some layout, so modeling layouts of various scenes is a key clue for scene recognition. Existing methods usually require an additional stream to detect regions for subsequent modeling, which accumulate errors and may miss important information. Meanwhile, they use manual features to model relations between regions, which weakens the representation ability of layouts. In this paper, we propose a single-stream adaptive scene layout modeling approach based on a layout modeling module (LMM), which constructs layouts without additional detection streams and adaptively captures the relations to take advantage of graph attention network. LMM is directly concatenated to a convolutional neural network, where each pixel of the activation maps of the last convolutional layer is defined as a region that is the initial input node of the LMM. LMM first models the layout of each region, and then uses all regions with layout information to model the entire scene. Layout relations are encoded as edges, which are automatically analyzed according to region co-occurrence and relative position. Our work can be understood as optimizing features of the activation maps from a scene layout modeling perspective for scene recognition. Experimental results on MIT67, SUN397, and Places365 show that our single-stream model achieves competitive performance.

Keywords Scene recognition · Scene layout · Convolutional neural network · Graph attention network

1 Introduction

Scene recognition has become one of the most challenging problems in computer vision and can be applied to many fields, such as AI cameras [1] and robot navigation [2]. Unlike object-level images, scene-level images typically

consist of a variety of regions with different distributions. Researchers have been committed to exploring the scene layout for scene recognition. The existing scene layout modeling method is a two-step process, as shown in Fig. 1a, first using the region detection stream to detect the region and then modeling the layout between regions to

✉ Feng Zhu
fzhu@sia.cn

Qun Wang
wangqun@sia.cn

Zhiyuan Lin
linzhiyuan@sia.cn

Jianyu Wang
wangjianyu@sia.cn

Xiang Li
1810587@stu.neu.edu.cn

Pengfei Zhao
zhaopengfei@sia.cn

¹ Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang 110016, China

² Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

³ Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China

⁴ University of Chinese Academy of Sciences, Beijing 100049, China

⁵ Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110016, China

⁶ Faculty of Computer Science and Engineering, Northeastern University, Shenyang 110016, China

represent the scene. In the first stage, some [3–5] use off-the-shelf object detection networks or semantic segmentation models. Because scene-level images are not annotated in detail, only the models trained on other datasets can be used. Once the detection results are unsatisfactory, the accumulated error will affect the final accuracy. Others [6, 7] utilize the clustering or region proposal methods to extract discriminating regions directly on feature maps. Since they need to filter candidate boxes to ease computational pressure when modeling, it's easy to miss other potentially important information. In the second stage, some methods based on recurrent neural networks, such as RNN-based [5, 8, 9] and LSTM-based [4, 7], learn the spatial dependencies between image regions. But their spatial layouts transmit context from a specified direction, lacking global information interaction. For this purpose, Some approaches [3, 6] take advantage of the graph propagation mechanism to analyze the global layout. However, they all need to manually design relation features (e.g., geometric and morphological relations), which makes it difficult to represent diverse spatial layouts robustly.

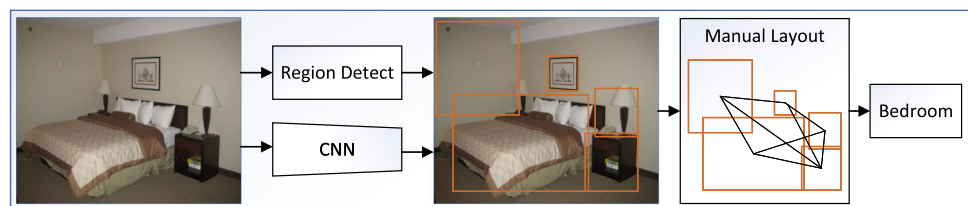
To mitigate the drawback mentioned above, a single-stream adaptive scene layout modeling method is proposed, which does not add any other streams to detect regions and adaptively explores relations between regions to assist the implementation of scene layout modeling. In detail, we forcibly define each pixel of the last convolutional layer as the initial graph node. The reason for using this fixed approach to obtain regions is mainly inspired by the ability of pre-trained Places-CNNs [10] that can capture the significant semantics of each pixel [11]. However, the pre-trained Places-CNNs only perform global average pooling (GAP) [12] to get final scene features. GAP can be understood as a non-discriminatory analysis of region co-

occurrence without their layouts. Therefore, following this idea, we design a layout modeling module (LMM) that adaptively analyzes the importance of each region in the scene according to the layout. LMM mainly builds a graph attention model, which first models the context of each initial node, and then uses the optimized nodes to build the entire scene layout. Note that unlike methods [3, 6] their node relations (i.e., edges of the graph) are adaptively obtained using their semantics and position. Our method is straightforward but works surprisingly well. We evaluate our single-stream model on three benchmark databases, MIT67 [13], SUN397 [14], and Places365 [10]. Our single-stream method outperforms the detector-based scene layout methods and achieves highly competitive results compared to other multi-stream models, with accuracy of 88.58% on MIT67 and 74.32% on SUN397. Moreover, our model achieves 56.53% Top-1 accuracy when extended to Places365, one of the largest datasets.

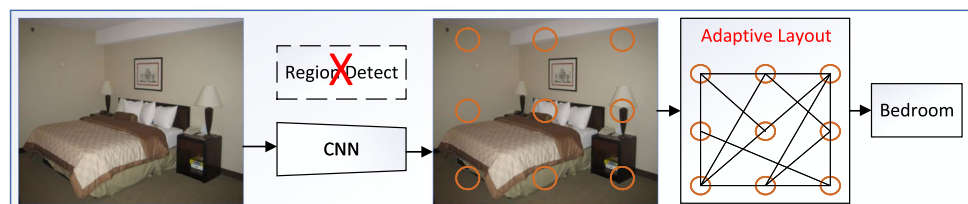
Our main contributions are summarized as follows:

- A single-stream adaptive scene layout modeling approach is proposed to construct the layout directly on the activation maps without additional object detection streams.
- Based on graph attention networks, a layout modeling module (LMM) is introduced to model each region's layout and the entire scene adaptively without needing manual relational features.
- Extensive experimental results on three datasets with different difficulties demonstrate the superiority and generalization of our method. Our model is intelligible in structure and impressive in results.

Fig. 1 A comparison of **a** double-stream manual scene layout modeling pipeline and **b** our single-stream adaptive scene layout modeling pipeline. Our method removes the additional region detection process and operates directly on the activation maps. At the same time, adaptive layout modeling is carried out on each region, and important regions are selected to represent the scene



(a) Double-stream manual scene layout modeling pipeline



(b) Our single-stream adaptive scene layout modeling pipeline

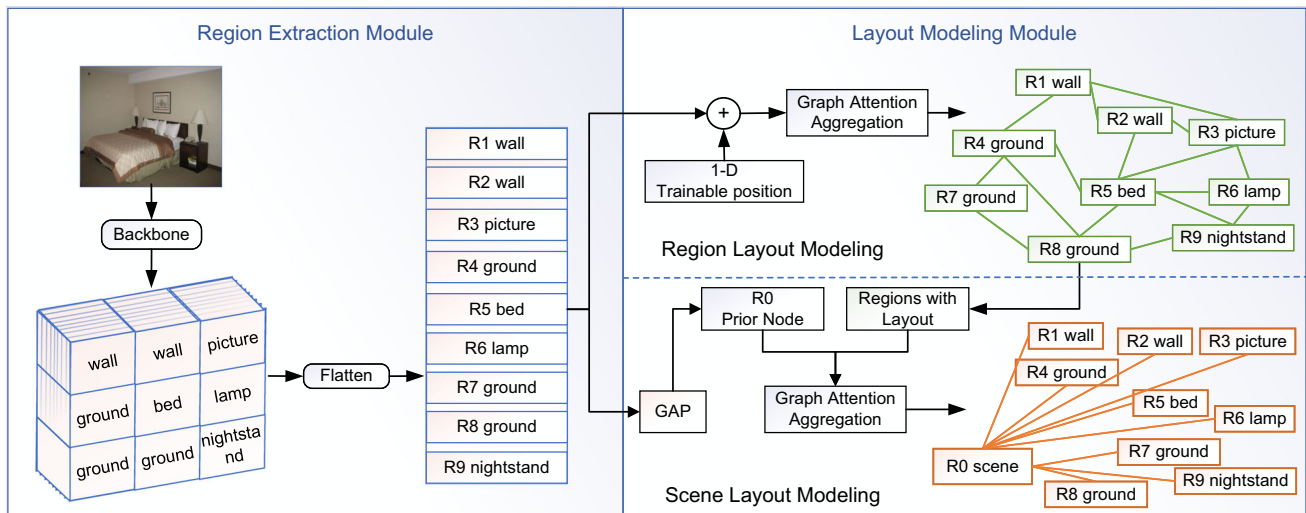


Fig. 2 The architecture of our approach consists of two modules. All region features are extracted using a pre-trained CNN model in the region extraction module. Then, let them adaptively model the layouts of the region and the scene in the layout modeling module

2 Related works

Scene recognition is an important research topic in the field of computer vision. In recent years, the powerful representation ability of convolutional neural networks (CNNs) [15–19] has dramatically improved the accuracy of scene recognition. But due to complex layouts such as multi-object, multi-scale, and multi-position information in scene images, these models [17–19] that are initially applied to natural image classification are not comfortable when directly processing this task. Therefore, the researchers [20–37] are keener to use CNNs as feature extractors, then encode the features to represent complex scenes. For example, some researchers combine CNNs with VLAD [21, 26] or Fisher Vector (FV) [27, 28] to generate scene features. Yee et al. [32] use spatial pyramid pooling to address the challenge of objects at different scene scales. Some studies [20, 22–25, 35, 36] consider that features from a single model cannot adequately represent the scene. Wang et al. [24] use two CNNs and input images of different resolutions to capture information at different coarse and fine scales. In [23], the important features extracted from the object-centric CNNs and scene-centric CNNs are selected based on a correlative context gating module. Sun et al. [25] propose a comprehensive representation by fusing information on object semantics, global appearance, and contextual appearance from three CNNs. Meanwhile, Xie et al. [20] make full use of the advantages of ViT [38] and CNNs to explore discriminative features. Despite their high performance, they mainly stack as many features as possible based on multiple scales or multiple models and do not explore the essence of recognizing the scene, that is, understanding the distribution of objects in the scene.

To improve performance, some methods [39–42] propose to represent the scene in terms of the co-occurrence of objects. Zhou et al. [40] generate a Bayesian object relations matrix to model the scene structure based on the object information obtained by the scene parsing algorithm [43]. Pereira et al. [41] utilizes YOLOv3 [44] to detect objects in the scene, encoding their categories and numbers as the scene layout. Although they take a step towards exploring the scene's layout, they only explore the co-occurrence relations between objects and lack their position modeling. Some earlier works [4, 5, 7–9] attempted to model the spatial layout of a scene using recurrent neural networks. To represent the scene, Zuo et al. [8] use RNN to obtain spatial modeling information in four directions (left, right, top, bottom). In [9], four directions (i.e., top-left to bottom-right, bottom-right to top-left, bottom-left to top-right, and top-right to bottom-left) are updated. Although they designed multiple directions for information flows, they could not interact with information globally. In order to compensate for this shortcoming, approaches [3, 6] represent the scene as a graph model, of which nodes as regions and edges represent the relations between regions. In [3], the segmentation network (i.e., DeeplabV2 [45] pre-trained on the COCO-stuff [46]) is used to obtain amorphous regions firstly. Then the context relations between regions are explored from geometric and morphological aspects. Chen et al. [6] chooses a clustering way to find the most representative regions in the candidate regions detected by an adaptive threshold on the feature maps. In addition to manual geometric relations, semantic relations have also been added. At present, if only relying on the features of a single model, the graph model-based methods

outperform the state-of-the-art methods, which also shows that scene layout is very important for scene recognition.

Therefore, instead of using additional detection streams, our proposed method directly performs layout modeling on the feature maps, which alleviates the information omission and error accumulation caused by the detection streams. At the same time, the regions defined on the feature maps adaptively model the layout without manual features based on the graph attention network [47].

3 Our approach

Our single-stream adaptive scene layout modeling approach consists of a region extraction module and a graph modeling module. The framework of our model is shown in Fig. 2.

3.1 Region extraction module

Each scene often contains multiple regions, so it is necessary to extract the regions in the scene before modeling the layout. Unlike previous works that use region detectors [3] or clustering on the activation maps [6], our approach returns to simplicity, that is, directly defining the activation maps as a region maps. This operation is inspired by the work of Zhou et al. [11] which demonstrates that the CNN pre-trained on [48] can perform both scene recognition and

region localization in a single forward-pass, without ever having been explicitly taught the notion of regions.

In practice, given an image, we feed it into a pre-trained CNN to extract the activation maps $\mathcal{A}(A \in \mathbb{R}^{H \times W \times C})$ from the last convolutional layer. Based on the same assumption of [11], we define each pixel on the activation maps as a region and flatten the activation maps by the channel dimension. Now, we obtain the region set $\mathcal{M} = \{m_1, m_2, \dots, m_n\}$, where m_i is the i -th pixel which represents the i -th region, and $n = H \times W = |\mathcal{M}|$ is the number of the region. We denote $\mathcal{X} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$ as the feature matrix of region \mathcal{M} where x_i is the feature vector of m_i , and $d = C$ is the dimension of region feature.

3.2 Graph modeling module

To explore the scene layout, we propose to propagate context by a graph model. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph where node set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ and \mathcal{E} is edge set between \mathcal{V} . We define $\mathcal{H} = [h_1, h_2, \dots, h_n]^T \in \mathbb{R}^{n \times d}$ is the feature matrix with d -dimension node feature where h_i is defined as the representation of node v_i . Typically, modern graph neural networks (GNNs) follow a learning schema that iteratively updates the representation of a node by aggregating representations of its first or higher-order neighbors. For basic GNN layers, the general “message-passing” architecture is employed for the information aggregation:

$$\mathcal{H}^{(l+1)} = F(\mathcal{H}^{(l)}, A), \tag{1}$$

where $\mathcal{H}^{(l)} = [h_1^{(l)}, h_2^{(l)}, \dots, h_n^{(l)}]^T$ denotes the feature matrix \mathcal{H} at the l -th step in the GNN, $\mathcal{H}^{(l+1)}$ is the updated feature matrix, A represents the adjacency matrix (i.e. edge relations) that captures the *importance* between nodes, and F is defined as “message-passing” function using A . In this paper, we obtain the adjacency matrix A based on graph attention mechanism using semantics and position. The following will use the l -th step as an example to illustrate how to obtain the adjacency matrix A .

3.2.1 Node representation

Objects with the same semantics may be distributed in multiple positions in a scene. In order to distinguish them, we need to add position information. Effortlessly, the coding of each region position in our method is much simpler than [3, 6], which uses the bounding boxes of each detected regions to get the position, we directly take advantage of the inherent position information of image convolution features. We perform standard learnable 1D

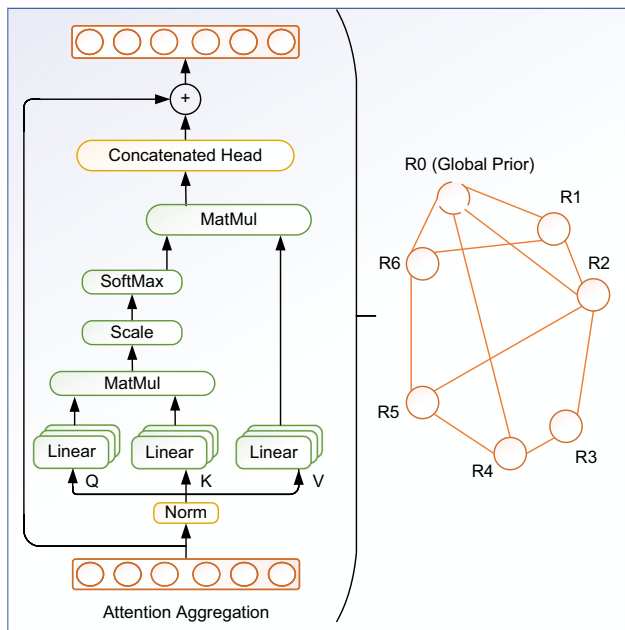


Fig. 3 Flowchart of Attention aggregation. The attention aggregation implements the *importance* analysis between regions in the region layout modeling and the discriminative feature aggregation in the scene layout modeling

position embeddings for their positions. Let $\mathcal{P}_E = [p_1, p_2, \dots, p_n]^T \in \mathbb{R}^{n \times d}$ is the embedding feature matrix, where $p_i \in \mathbb{R}^d$ is the embedding features of i -th position in the activation maps.

We combine semantics and position to capture the *importance* between regions by the graph. First, we define region set \mathcal{M} as node set \mathcal{V} . Then, the position features are embedded into semantic node features to generate the initial node feature matrix $\mathcal{H}^{(0)}$ and it can be formulated as:

$$\mathcal{H}^{(0)} = \mathcal{X} + \mathcal{P}_E, \tag{2}$$

3.2.2 Region layout modeling

Now, we have represented the region nodes. We perform attention aggregation for each region node to model their layouts between regions and adaptively obtain relations with other regions, details are shown in Fig. 3. Specifically, to obtain sufficient expressive power to transform input features, one learnable linear transformation is required. Note that before this operation we perform a layer normalization to constrain the features to an approximate distribution range:

$$\mathcal{H}'^{(l)} = \text{LayerNorm}(\mathcal{H}^{(l)}). \tag{3}$$

The features of input node \mathcal{V} is projected by three matrices $W_Q^{(l)} \in \mathbb{R}^{d \times d_K}$, $W_K^{(l)} \in \mathbb{R}^{d \times d_K}$ and $W_V^{(l)} \in \mathbb{R}^{d \times d_V}$. For simplicity of illustration, we consider the single-head self-attention and let $d_K = d_V = d$. The extension to the multi-head attention is standard and straightforward. Then, we sequentially calculate the *importance* of each node $j \in \mathcal{V}$ to node i :

$$e_{ij}^{(l)} = \frac{(h_i'^{(l)} W_Q^{(l)}) (h_j'^{(l)} W_K^{(l)})^T}{\sqrt{d_K}}. \tag{4}$$

To make *importance* $e_{ij}^{(l)}$ easily comparable across different nodes, we normalize them across all of j using the softmax function:

$$\alpha_{ij}^{(l)} = \text{softmax}(e_{ij}^{(l)}) = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{V}} \exp(e_{ik}^{(l)})}, \tag{5}$$

where $\alpha_{ij}^{(l)}$ is (i, j) -element of adjacency matrix A at the l -th step in the GNN.

Once obtained, the normalized attention weights $\alpha_{ij}^{(l)}$ are used to aggregate the corresponding node to the output node bias $h_i'^{(l+1)}$ by a linear combination of their features. This process can be formalized as:

$$h_i'^{(l+1)} = \left(\sum_{j \in \mathcal{V}} \alpha_{ij}^{(l)} (h_j'^{(l)} W_V^{(l)}) \right). \tag{6}$$

The above is a process of single-head attention. To enhance the representational power, we find extending our mechanism to employ multi-head attention to be beneficial [49]. In detail, N independent attention mechanisms execute the transformation of Eq. (6), and then their features are concatenated, resulting in the following bias feature representation:

$$h_i'^{(l+1)} = \parallel_{n=1}^N \left(\sum_{j \in \mathcal{V}} (\alpha_{ij}^n)^{(l)} (h_j'^{(l)} (W_V^n)^{(l)}) \right). \tag{7}$$

Because feature nodes need to be updated iteratively, we keep the input and output feature dimensions consistent. That is to say, the number of heads N depends on the projection dimension of the projection matrix, i.e., $d_v = \frac{d}{N}$. Finally, add the residual structure [18] to get the output node features of the l -th step:

$$h^{(l+1)} = h^{(l)} + h_j'^{(l+1)}. \tag{8}$$

3.2.3 Scene layout modeling

So far, we have obtained region nodes with layout information $\mathcal{H}^{(L)} = [h_1^{(L)}, h_2^{(L)}, \dots, h_n^{(L)}]^T$, where L is the last step of region layout modeling. Since we encode all the information in the scene into nodes, there may be unimportant or even negative nodes. Therefore, our method adaptively selects important regions in the scene layout modeling. Specifically, we add a global prior node v_0 representing the whole image to aggregate important nodes and weaken others. To generate the global prior node, we perform global average pooling (GAP) on the activation maps from the last convolutional layer and define its feature as $h_0 \in \mathbb{R}^d$. As a result, the node set \mathcal{V} and feature matrix $\mathcal{H}^{(L)}$ are updated to $\{v_0, v_1, v_2, \dots, v_n\}$ and $[h_0, h_1^{(L)}, h_2^{(L)}, \dots, h_n^{(L)}]^T \in \mathbb{R}^{(n+1) \times d}$, respectively. Then, attention aggregation is utilized again on the global prior node to select important node features. Finally, we perform layer normalization on the aggregated global prior nodes to accelerate convergence and feed them into one layer fully connected network to predict the scene category. To demonstrate the effectiveness of the global prior node, we apply average pooling on $\mathcal{H}^{(L)}$ in ablation experiments to generate the scene representation as a d -dimensions vector.

Table 1 Comparison with state-of-the-art methods on MIT67 and SUN397 (%)

Methods	# of models	# of Scales	Resolutions	MIT67	SUN397
MFA-FS [28]	2	3	(96, 128, 160)	87.23	71.06
MFAFVNet [27]	2	3	(96, 128, 160)	87.97	72.0
MFAFSNet [33]	2	3	(96, 128, 160)	88.05	72.43
EMFS [34]	3	4	224	86.90	72.60
SDO [37]	2	9	–	86.76	73.21
ChAM [22]	2	1	224	87.1	74.04
Attribute-CNN [50]	1	2	(224, 448)	88.06	74.12
DL-CNN [51]	1	2	–	86.43	70.13
Places365-VGG [10]	1	1	224	76.53	63.24
Places205-VGG [48]	1	1	224	79.76	61.99
Places365-ResNet [10]	1	1	224	82.84	68.89
LSO-VLADNet [21]	1	1	448	81.70	61.60
ARG [3]	1	1	448	88.13	73.58
M2M BiLSTM [7]	1	1	–	88.25	71.81
LGN [6]	1	1	448	88.06	74.06
Places365-VGG [10]	1	1	512	79.70	64.12
Places205-VGG [48]	1	1	512	79.78	62.57
Places365-ResNet [10]	1	1	512	84.63	69.84
Ours	1	1	224	84.78	73.11
Ours	1	1	384	87.74	74.32
Ours	1	1	448	88.36	74.14
Ours	1	1	512	88.58	73.96

The best results are in bold

Table 2 Comparison with state-of-the-art methods on Place365 (%)

Methods	Resolutions	Top-1	Top-5
Essence [52]	224	55.21	80.42
Places365-VGG [10]	224	55.24	84.91
Places365-ResNet [10]	224	54.74	85.08
ChAM [22]	224	56.51	86.00
LGN [6]	224	56.50	86.24
Our	224	56.53	86.39

The best results are in bold

4 Experiments

In this section, we evaluate the effectiveness of our method on three well-known and publicly available datasets, MIT67 [13], SUN397 [14], and Places365 [10].

4.1 Datasets

MIT67 Dataset [13]. It contains 67 classes of a wide variety of indoor environments and 15,620 images. There are at least 100 images per category. According to the standard evaluation protocol, we set 80 images for training and 20 images for evaluation.

Table 3 Effect of different image resolutions on MIT67

Resolutions	Factors	Baseline	Ours	Acc. Gains
224 × 224	#params	23.65M	48.82M	1.94
	GFLOPs	4.13	5.17	
	Acc. (%)	82.84	84.78	
384 × 384	#params	23.65M	48.82M	3.78
	GFLOPs	12.14	15.17	
	Acc. (%)	83.96	87.74	
448 × 448	#params	23.65M	48.82M	2.61
	GFLOPs	16.53	20.65	
	Acc. (%)	85.75	88.30	
512 × 512	#params	23.65M	48.82M	3.95
	GFLOPs	21.59	26.97	
	Acc. (%)	84.63	88.59	

We evaluate the classification accuracy, and report the model size and the computational complexity

SUN397 Dataset [14]. It is a larger scene dataset, which comprises around 108,754 images from 397 scene categories. Each category has at least 100 different numbers of images. 50 training images and 50 validation images per class are used to evaluate the competing methods following the commonly used evaluation protocol.

Places365 Dataset [10]. It is explicitly designed for scene recognition, which has two training subsets, Places365-Standard and Places365-challenge. In this paper, we choose Places365-Standard as the training set, which consists of around 1.8 million training images and 365 scene categories. The validation set of Places365-Standard contains 100 images per category. Both top-1 and top-5 accuracy are reported as the evaluation metric.

4.2 Implementation details

In our comparative experiment, we use ResNet50 [18] pre-trained on places365 [10] as the backbone. Since image resolution also affects accuracy, we select multiple common resolutions $\{224 \times 224, 384 \times 384, 448 \times 448, 512 \times 512\}$ for a fair comparison. We randomly crop and resize to the corresponding size and random horizontal flipping during the training. In the testing, we first resize the image to $\{256 \times 256, 416 \times 416, 480 \times 480, 544 \times 544\}$ and then crop the center to $\{224 \times 224, 384 \times 384, 448 \times 448, 512 \times 512\}$. Note that this is the 1-crop test method on MIT67 [13] and SUN397 [14]. The standard 10-crops test method is agreed upon on Places365 [10] and an evaluation measurement is the average classification accuracy of 10-crops. The batch sizes of $\{224 \times 224, 384 \times 384, 448 \times 448, 512 \times 512\}$ are $\{128, 60, 32, 32\}$. The initial learning rates are set to 10^{-3} , 10^{-3} and 10^{-4} for MIT67, SUN397, and Places365, respectively. The minimum learning rate is 10^{-5} . We train our models end-to-end for 40 epochs using the SGD optimizer with CosineAnnealingLR to adjust the learning rate. To avoid overfitting, one GNN layer is used in the region layout modeling. The number of attention heads is 8.

Table 4 Complexity and robustness analysis of our model

Datasets	Factors	Baseline	Ours	Acc. Gains
MIT67 [13]	#params	23.65M	48.82M	1.94
	GFLOPs	4.13	5.17	
	Acc (%)	82.84	84.78	
SUN3977 [14]	#params	24.32M	49.50M	4.05
	GFLOPs	4.13	5.17	
	Acc (%)	69.06	73.11	
Places365 [10]	#params	24.26M	49.43M	1.79
	GFLOPs	4.13	5.17	
	Acc (%)	54.74	56.53	

The resolution is set to 224×224 . The accuracy, the model size and the computational complexity on MIT67, SUN39, and Places365 are reported

4.3 Comparison with state-of-the-art methods

In this subsection, we conduct extensive experiments on three datasets to compare the performance with state-of-the-art methods.

We compare the results with advanced methods on MIT67 and SUN397 in Table 1. From Table 1, we see that the methods [3, 6, 7] based on scene layout modeling achieve higher accuracy than other methods. This also proves that scene recognition from the perspective of scene layout is feasible and effective. At the same time, because our method reduces the loss of information during region extraction and improves the ability to express relations, our method obtains better accuracy than other layout methods [3, 6, 7]. Various multi-model, multi-scale combination methods are also reported in Table 1, but our single-stream model still outperforms them. Because the region information at a 224×224 resolution may disappear with multiple downsampling, which results in a lack of critical region information when subsequently modeling the layout, the accuracy of our method is reduced at a resolution of 224×224 . When the resolution is increased, the advantages of layout modeling become apparent. For a fair comparison, we also implement some typical methods (Places205-VGG [48], Places365-VGG [10], and Places365-ResNet [10]) at 512×512 resolution. It can be clearly seen that the accuracy of our method is still higher than the typical methods, which proves the effectiveness and competitiveness of our method.

We also demonstrated the effectiveness of our approach at Places365. For a fair comparison, we only execute models at resolutions of 224×224 , and the results are shown in Table 2. As seen from Table 2, our single-stream adaptive modeling method can achieve a Top-1 accuracy of 56.53%. Compared to the baseline Places365-ResNet [10], our model can gain a 1.79% improvement of Top1 accuracy, proving our proposed method's effectiveness.

4.4 Ablation study

In this subsection, we conduct ablation studies on MIT67 to better understand our method's effects. Unless specified, the resolution is set to 512×512 .

4.4.1 Analysis of different resolutions

We know that the number of region nodes depends on the size of the activation maps. We perform some detailed experiments to explore the impact of image resolution on our model performance. We take the resolutions of $\{224 \times 224, 384 \times 384, 448 \times 448, 512 \times 512\}$ for experiments, and results illustrated in Table 3. The results show that the

Table 5 Effect of different modules on MIT67

Backbone	Region layout modeling		Scene layout modeling		#params	GFLOPs	Acc. (%)
	Semantics	Position	Prior node	Average			
✓	–	–	–	✓	23.65M	21.59	84.63
✓	–	–	✓	–	36.24M	23.74	87.46
✓	✓	–	✓	–	48.82M	26.97	88.28
✓	–	✓	✓	–	48.82M	26.97	87.84
✓	✓	✓	✓	–	48.82M	26.97	88.58
✓	✓	✓	–	✓	36.24M	24.81	87.69

We evaluate the classification accuracy, and report the model size and the computational complexity

Table 6 Impact of the region detection performance on MIT67

Region Detectors	AP	#params	GFLOPs	Acc. (%)
Faster-RCNN [53]	37.4	48.82M	26.97	78.21
Mask-RCNN [54]	38.2	48.82M	26.97	78.51
DETR [55]	39.9	48.82M	26.97	79.03
RetinaNet [56]	39.5	48.82M	26.97	79.10
De-DETR [57]	47.0	48.82M	26.97	80.89
DINO [58]	49.0	48.82M	26.97	81.10

We evaluate the classification accuracy, and report the model size and the computational complexity

accuracy of our method gradually improves as the resolution increases. Moreover, our gain also increases, indicating that as the information in the scene increases, our advantages of layout modeling gradually become apparent. As seen from Table 3, when the resolution is 512×512 , the accuracy of the baseline decreases, indicating that although more information is obtained, the interference information will increase accordingly. In contrast, our adaptive layout modeling method can eliminate some noisy information.

4.4.2 Complexity and robustness analysis of our model

To analyze the complexity and robustness of our model, we compare our model with the baseline (ResNet50 pre-trained on Places365 [10]) in terms of the accuracy, the model size and the computational complexity on three benchmark datasets, as shown in Table 4. We set the resolution to 224×224 . The results in Table 4 show that our method performs well on MIT67 [13], SUN397 [14], and Places365 [10] datasets and brings 1.94%, 4.05% and 1.79% gains, respectively. The improved classification accuracies on all three datasets with different styles and different number of images demonstrate the excellent

Table 7 Effect of the number of heads on MIT67

Number of heads	#params	GFLOPs	Accuracy (%)
1	48.82M	26.97	88.13
2	48.82M	26.97	88.21
8	48.82M	26.97	88.58
16	48.82M	26.97	88.28

The accuracy, the model size and the computational complexity are reported

Table 8 Effect of the number of GNN layers on MIT67

Number of layers	#params	GFLOPs	Accuracy (%)
1	48.82M	26.97	88.58
2	61.41M	30.20	87.83
3	74.00M	33.42	87.59

The accuracy, the model size and the computational complexity are reported

generalization and robustness of our method. In addition, our method increases about 1.04 GFLOPs (at a resolution of 224×224) and 25 M parameters to the baseline, which is acceptable compared to multi-model and multi-scale methods. We also show the model size and computational complexity at different resolutions in Table 3. Our method only increases about 5 GFLOPs when the resolution is 512×512 .

4.4.3 Impact of the region detection performance

Scene images usually consist of many regions in some layout. Accurately detecting regions is required to model the final scene. In our approach, each pixel on the activation maps obtained from the visual backbone is treated as a

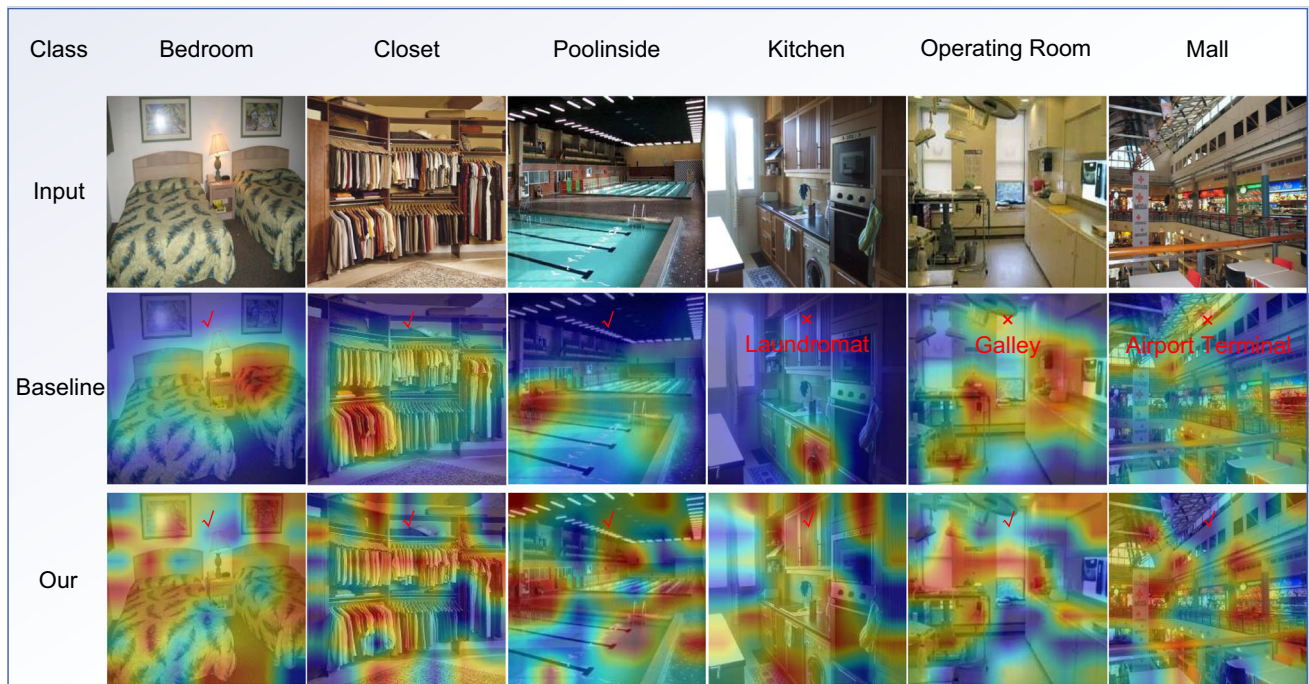


Fig. 4 Visualization of method concerns. The ground truth about the scene class of the image is on top of the image. The first row shows the input image. The second row shows the class activation map

(CAM). The third row shows the attention regions of the global prior node in our method (the redder the color, the more attention)

region, and the regions are implicitly modeled for final scene modeling. Our final scene modeling is most relevant to visual backbone detection performance. To analyze the impact of differences in region detection performance on the final scene modeling, we use the visual backbone (ResNet50 [18]) as our region extractor, fine-tuned in different object detectors. The impact of different detection performance of visual backbones for final scene modeling are shown in Table 6. As the performance of the region detectors improves from 37.4 AP to 49.0 AP, the classification accuracy also increases from 78.21 to 81.10%. This suggests that different region detection performance largely influences the final scene modeling. Thus, it is critical that the visual backbone accurately detects the regions for the final scene modeling.

4.4.4 Effect of different modules

We analyze the contribution of each module and their different implementations to the model performance, and Table 5 shows the results. Compared with the baseline (Backbone + GAP), each proposed modules contributes to the accuracy improvement. As you can see from Table 5, the performance improves after adding region layout modeling. Meanwhile, position information brings a gain of 0.3%, indicating that the position is also a clue to identify the scene. However, the gain decreases when the relations of regions are adapted only by position, which

also proves that the scene layout needs the combination of semantics and position. In scene layout modeling, the method based on the global prior node adaptively selects discriminative regions, which can reduce interference information (Table 6).

4.4.5 Effect of the multi-head attention

The multi-head attention mechanism in Transformer brings performance improvement to NLP tasks. We also conduct experimental analysis, and Table 7 presents the results. The experimental results show that increasing the number of attention head can bring gain, but it will reach saturation when it rises to a certain threshold. In our method, the optimal number of attention head is 8. In addition, the multi-head attention can improve accuracy without increasing the parameters and computational complexity.

4.4.6 Effect of the multi-layer GNN

When performing GNN aggregation, the number of layers is generally set to 2 or 3. Because too many layers will cause each node to aggregate its neighbors multiple times, the features of all nodes will become similar. The individual characteristics of each node cannot be distinguished. To this end, we evaluate models with different numbers of layers, and the results are shown in Table 8. When the number of GNN layers increases, the parameters and

computational complexity also increase, but the accuracy does not improve. In fact, the number of nodes that need to be operated in our model is small, and only one GNN layer is required to complete the adaptive aggregation.

4.4.7 Visualization of methods

We visualized the respective concerns of our method and baseline Places365-ResNet50 [10] in Fig. 4. The focus of the baseline is demonstrated by the class activation map (CAM). Our approach is to average the attention weights of the global prior node. Obviously, the baseline's focus is local, while our method's focus is more global and primarily distributed in semantic regions related to scene categories. Fortunately, we comprehensively consider the global information, correcting the shortcoming of the baseline that only focuses on local information, such as the mistake of recognizing "kitchen" as "laundromat" due to only focusing on "washing machine." For example, when recognizing "operating room" and "shopping mall," we paid more attention to "surgical instruments" and "shopping mall billboards," respectively.

5 Conclusion

In this paper, we propose a single-stream adaptive scene layout modeling method for scene recognition. Our method does not require additional streams to detect regions and can directly process the activation maps as regions. Based on the graph attention network, the scene layout is built, where the attention mechanism is used to adaptively capture the relations between regions. This mechanism automatically analyzes the importance of regions based on semantics and position, which can improve the ability to express relations. Comprehensive experiments on MIT67, SUN397, and Places365 have demonstrated the effectiveness and generalization of our method for scene recognition. We also hope that our method will be helpful to other scholars.

In the future, we will consider combining multi-scale information from different convolutional layers. From the experimental results, the performance is low when the resolution is poor, because the effective information is easily filtered out by multiple pooling. While high-resolution performance is excellent, the computational cost is still a lot of pressure at current levels. Therefore, we can explore how to discover more region information at low resolutions to help build more accurate scene layouts.

Data availability MIT67 dataset [13] is available at <http://web.mit.edu/torralba/www/indoor.html>, SUN397 dataset [14] is available at

<https://vision.princeton.edu/projects/2010/SUN/>, and Places365 dataset [10] is available at <http://places2.csail.mit.edu/index.html>.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

- Liu S, Tian G, Zhang Y, Duan P (2022) Scene recognition mechanism for service robot adapting various families: a CNN-based approach using multi-type cameras. *IEEE Trans Multimedia* 24:2392–2406. <https://doi.org/10.1109/TMM.2021.3080076>
- Gao C, Chen J, Liu S, Wang L, Zhang Q, Wu Q (2021) Room-and-object aware knowledge reasoning for remote embodied referring expression. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 3063–3072. <https://doi.org/10.1109/CVPR46437.2021.00308>
- Zeng H, Song X, Chen G, Jiang S (2022) Amorphous region context modeling for scene recognition. *IEEE Trans Multimedia* 24:141–151. <https://doi.org/10.1109/TMM.2020.3046877>
- Javed SA, Nelakanti AK (2017) Object-level context modeling for scene classification with context-CNN. arXiv preprint [arXiv:1705.04358](https://arxiv.org/abs/1705.04358)
- Song X, Jiang S, Wang B, Chen C, Chen G (2020) Image representations with spatial object-to-object relations for RGB-D scene recognition. *IEEE Trans Image Process* 29:525–537. <https://doi.org/10.1109/TIP.2019.2933728>
- Chen G, Song X, Zeng H, Jiang S (2020) Scene recognition with prototype-agnostic scene layout. *IEEE Trans Image Process* 29:5877–5888. <https://doi.org/10.1109/TIP.2020.2986599>
- Laranjeira C, Lacerda A, Nascimento ER (2019) On modeling context from objects with a long short-term memory for indoor scene recognition. In: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp 249–256. <https://doi.org/10.1109/SIBGRAPI.2019.00041>
- Zuo Z, Shuai B, Wang G, Liu X, Wang X, Wang B, Chen Y (2015) Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In: 2015 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 18–26. <https://doi.org/10.1109/CVPRW.2015.7301268>
- Zuo Z, Shuai B, Wang G, Liu X, Wang X, Wang B, Chen Y (2016) Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Trans Image Process* 25(7):2983–2996. <https://doi.org/10.1109/TIP.2016.2548241>
- Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2018) Places: a 10 million image database for scene recognition. *IEEE Trans Pattern Anal Mach Intell* 40(6):1452–1464. <https://doi.org/10.1109/TPAMI.2017.2723009>
- Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2014) Object detectors emerge in deep scene CNNs. arXiv preprint [arXiv:1412.6856](https://arxiv.org/abs/1412.6856)
- Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint [arXiv:1312.4400](https://arxiv.org/abs/1312.4400)
- Quattoni A, Torralba A (2009) Recognizing indoor scenes. In: 2009 IEEE conference on computer vision and pattern recognition, pp 413–420. <https://doi.org/10.1109/CVPR.2009.5206537>
- Xiao J, Hays J, Ehinger KA, Oliva A, Torralba A (2010) Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and

- pattern recognition, pp 3485–3492 . <https://doi.org/10.1109/CVPR.2010.5539970>
15. Liu K, Moon S (2021) Dynamic parallel pyramid networks for scene recognition. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2021.3129227>
 16. Qiao Z, Yuan X, Zhuang C, Meyarian A (2021) Attention pyramid module for scene recognition. In: 2020 25th international conference on pattern recognition (ICPR), pp 7521–7528 . <https://doi.org/10.1109/ICPR48806.2021.9412235>
 17. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 7132–7141. <https://doi.org/10.1109/CVPR.2018.00745>
 18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
 19. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp. 5987–5995. <https://doi.org/10.1109/CVPR.2017.634>
 20. Xie Y, Yan J, Kang L, Guo Y, Zhang J, Luan, X (2022) FCT: fusing CNN and transformer for scene classification. *Int J Multimedia Inf Retrieval* 1–8
 21. Chen B, Li J, Wei G, Ma B (2018) A novel localized and second order feature coding network for image recognition. *Lect Notes Comput Sci* 76:339–348
 22. López-Cifuentes A, Escudero-Viñolo M, Bescós J, García-Martín Á (2020) Semantic-aware scene recognition. *Lect Notes Comput Sci* 102:107256
 23. Seong H, Hyun J, Kim E (2020) FOSNet: an end-to-end trainable deep neural network for scene recognition. *IEEE Access* 8:82066–82077. <https://doi.org/10.1109/ACCESS.2020.2989863>
 24. Wang L, Guo S, Huang W, Xiong Y, Qiao Y (2017) Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs. *IEEE Trans Image Process* 26(4):2055–2068. <https://doi.org/10.1109/TIP.2017.2675339>
 25. Sun N, Li W, Liu J, Han G, Wu C (2019) Fusing object semantics and deep appearance features for scene recognition. *IEEE Trans Circuits Syst Video Technol* 29(6):1715–1728. <https://doi.org/10.1109/TCSVT.2018.2848543>
 26. Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J (2018) Netvlad: CNN architecture for weakly supervised place recognition. *IEEE Trans Pattern Anal Mach Intell* 40(6):1437–1451. <https://doi.org/10.1109/TPAMI.2017.2711011>
 27. Li Y, Dixit M, Vasconcelos N (2017) Deep scene image classification with the MFAFVNet. In: 2017 IEEE international conference on computer vision (ICCV), pp 5757–5765. <https://doi.org/10.1109/ICCV.2017.613>
 28. Dixit MD, Vasconcelos N (2016) Object based scene representations using fisher scores of local subspace projections. *Adv Neur Inf* 29
 29. Tang P, Wang H, Kwong S (2017) G-ms2f: Googlenet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing* 225:188–197
 30. Liu S, Tian G, Xu Y (2019) A novel scene classification model combining resnet based transfer learning and data augmentation with a filter. *Neurocomputing* 338:191–206
 31. Yang S, Ramanan D (2015) Multi-scale recognition with DAG-CNNs. In: 2015 IEEE international conference on computer vision (ICCV), pp 1215–1223. <https://doi.org/10.1109/ICCV.2015.144>
 32. Yee PS, Lim KM, Lee CP (2022) Deepscene: Scene classification via convolutional neural network with spatial pyramid pooling. *Expert Syst Appl* 193:116382
 33. Dixit M, Li Y, Vasconcelos N (2020) Semantic fisher scores for task transfer: using objects to classify scenes. *IEEE Trans Pattern Anal Mach Intell* 42(12):3102–3118. <https://doi.org/10.1109/TPAMI.2019.2921960>
 34. Song X, Jiang S, Herranz L (2017) Multi-scale multi-feature context modeling for scene recognition in the semantic manifold. *IEEE Trans Image Process* 26(6):2721–2735. <https://doi.org/10.1109/TIP.2017.2686017>
 35. Guo S, Huang W, Wang L, Qiao Y (2017) Locally supervised deep hybrid model for scene recognition. *IEEE Trans Image Process* 26(2):808–820. <https://doi.org/10.1109/TIP.2016.2629443>
 36. Wang Z, Wang L, Wang Y, Zhang B, Qiao Y (2017) Weakly supervised patchnets: describing and aggregating local patches for scene recognition. *IEEE Trans Image Process* 26(4):2028–2041. <https://doi.org/10.1109/TIP.2017.2666739>
 37. Cheng X, Lu J, Feng J, Yuan B, Zhou J (2018) Scene recognition with objectness. *Lect Notes Comput Sci* 74:474–487
 38. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*
 39. Miao B, Zhou L, Mian AS, Lam TL, Xu Y (2021) Object-to-scene: learning to transfer object knowledge to indoor scene recognition. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 2069–2075 . <https://doi.org/10.1109/IROS51168.2021.9636700>
 40. Zhou L, Cen J, Wang X, Sun Z, Lam TL, Xu Y (2021) Borm: Bayesian object relation model for indoor scene recognition. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 39–46. <https://doi.org/10.1109/IROS51168.2021.9636024>
 41. Pereira R, Gonçalves N, Garrote L, Barros T, Lopes A, Nunes UJ (2020) Deep-learning based global and semantic feature fusion for indoor scene classification. In: 2020 IEEE international conference on autonomous robot systems and competitions (ICARSC), pp. 67–73. <https://doi.org/10.1109/ICARSC49921.2020.9096068>
 42. Yeo W-H, Heo Y-J, Choi Y-J, Kim B-G (2020) Place classification algorithm based on semantic segmented objects. *Appl Sci* 10(24):9069
 43. Zhou B, Zhao H, Puig X, Xiao T, Fidler S, Barriuso A, Torralba A (2019) Semantic understanding of scenes through the ade20k dataset. *Int J Comput Vis* 127:302–321
 44. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. *arXiv preprint arXiv:1804.02767*
 45. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>
 46. Caesar H, Uijlings J, Ferrari V (2018) Coco-stuff: Thing and stuff classes in context. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 1209–1218 . <https://doi.org/10.1109/CVPR.2018.00132>
 47. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. *Stat* 1050:20
 48. Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: *Advances in neural information processing systems*, 27
 49. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *Advances in neural information processing systems*, 30
 50. Zeng H, Song X, Chen G, Jiang S (2020) Learning scene attribute for scene recognition. *IEEE Trans Multimedia* 22(6):1519–1530. <https://doi.org/10.1109/TMM.2019.2944241>

51. Liu Y, Chen Q, Chen W, Wassell I (2018) Dictionary learning inspired deep network for scene recognition. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
52. Qiu J, Yang Y, Wang X, Tao D (2021) Scene essence. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8318–8329. <https://doi.org/10.1109/CVPR46437.2021.00822>
53. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28
54. He K, Gkioxari G, Dollár P, Girshick R (2020) Mask R-CNN. *IEEE Trans Pattern Anal Mach Intell* 42(2):386–397. <https://doi.org/10.1109/TPAMI.2018.2844175>
55. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European conference on computer vision. Springer, Berlin, pp 213–229
56. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
57. Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable detr: deformable transformers for end-to-end object detection. arXiv preprint [arXiv:2010.04159](https://arxiv.org/abs/2010.04159)
58. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum H-Y (2022) Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint [arXiv:2203.03605](https://arxiv.org/abs/2203.03605)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.