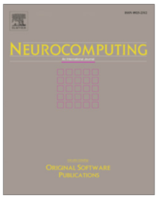
Neurocomputing 440 (2021) 36–44

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Visual affordance detection using an efficient attention convolutional neural network

Qipeng Gu [a,b], Jianhua Su [a,*], Lei Yuan [c]

[a] The Key Lab of Complex System and Intelligence Sciences, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] The School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[c] State key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, China

ARTICLE INFO

Article history:
Received 17 February 2020
Revised 25 September 2020
Accepted 2 January 2021
Available online 12 January 2021
Communicated by Zidong Wang

Keywords:
Affordance detection
Attention mechanism
Up-sampling layer

ABSTRACT

Visual affordance detection is an important issue in the field of robotics and computer vision. This paper proposes a novel and practical convolutional neural network architecture that adopts an encoder-decoder architecture for pixel-wise affordance detection. The encoder network comprises two modules: a dilated residual network that is the backbone for feature extraction, and an attention mechanism that is used for modeling long-range, multi-level dependency relations. The decoder network consists of a novel up-sampling layer that maps the low-resolution encoder feature to a high-resolution pixel-wise prediction map. Specifically, integrating an attention mechanism into our network reduces the loss of salient details and improves the feature representation performance of the model. The results of experiments conducted on the University of Maryland dataset (UMD) verify that the proposed network with the attention mechanism and up-sampling layer improved performance compared with classical methods. The proposed method lays the foundation for subsequent research on multi-task learning by physical robots.

1. Introduction

Affordances are properties or features of objects that present the possible set of actions afforded an agent by the environment. Gibson was the first to introduce the concept of affordances in 1979 [1]. Since then, the idea of affordances has been widely applied to the design of reliable robotic systems capable of translating perceptions into actions. For example, a hammer in a scene affords the opportunity for a human to grasp it or pound an object. Similarly, for an autonomous robot interacting with humans, it is of great significance to understand the affordances of objects for humanlike manipulation. If a robot is required to pound something with a hammer, its computer vision must give a high degree of perception that can be used to identify the hammer and locate it accurately. However, the robot needs to understand all the affordances of the hammer prior to choosing the correct actions in this task. This includes understanding, which part of the hammer can be grasped and which part can be used to pound.

Affordance detection has a wide range of applications, including robot manipulation, path planning, and autonomous driving [2–6]. Early works based on low-level visual cues have been surpassed by popular machine learning algorithms. In particular, remarkable progress has been made in affordance detection with deep neural network [2–8]. Existing work based on deep neural networks typically leverages mid-level visual cues [9] to label every pixel with an affordance type. Unlike the classical semantic segmentation application, affordance detection inherits two problems. First, the same affordance type can present different appearances. For example, a cup and a hammer both have the affordance type “grasp,” but they are quite different in appearance. Second, different affordance types may have similar appearances, such as the “contain” and “grasp” regarding a plate.

To address these problems, we propose a novel architecture to enhance the discriminative ability of feature representations for affordance detection. Primarily, our motivation to design such an architecture arises from affordance detection needing to model the shapes, orientations, appearances, and spatial relations of objects in the environment. In any scene, some affordance parts are large and some are small. Thus, the network requires retention of spatial acuity in the extracted image representation. Compared with previous works that used widely adopted convolutional neural networks to extract features (e.g., VGG [10] and Resnet [11]), our network uses a dilated residual network (DRN) [12] as a pre-trained model. The DRN can preserve spatial resolution in convolutional layers to produce high-resolution output maps.

[*] Corresponding author.
E-mail address: jianhua.su@ia.ac.cn (J. Su).

https://doi.org/10.1016/j.neucom.2021.01.018
0925-2312/© 2021 Elsevier B.V. All rights reserved.

Existing convolutional neural network (CNN) models rely on increasing depths to model long-range, multi-level relations for affordance detection, which is inefficient. In this work, we integrate the self-attention mechanism into the architecture. Previous works, such as [13–15], have shown that the attention mechanism has a positive influence in computer vision. It can not only determine where to focus, but can also improve the feature representation of interests. Thus, we adopt the self-attention mechanism to model relationship dependencies in the spatial and channel dimensions, because convolution operations extract features by blending channel and spatial information.

The up-sampling layer is an important part of affordance detection, because it learns an array of up-scaling filters to generate the pixel-wise prediction map. Most works on the up-sampling layer can be divided into three types: bilinear up-sampling [16], transposed convolution(TC) [17] [18,19] and un-pooling [20]. Many semantic segmentation tasks use bilinear up-sampling to generate output maps [21]. Transposed convolutions such as FCN [22] and GAN [23], have been widely used for pixel-wise prediction and image generation. Un-pooling is a method of upscaling low-resolution feature maps into an output map(e.g., SegNet [24] and DeconvNet [25]). However, in all of the above methods, fine details may be lost. Moreover, the bilinear up-sampling method is not learnable, because it has no parameters for up-sampling on a given policy. Therefore, we design an up-sampling layer that can easily generate a high-resolution affordance map. In this work, the proposed up-sampling layer can fit the network architecture by enabling end-to-end training to increase the accuracy of affordance prediction. In summary, the contributions of this work are as follows:

(1) To preserve spatial resolution in the convolutional layer, a DRN is employed as a pretained model to extract features. The fully connected layer of the DRN is removed, which reduces the number of parameters of the network and makes it easier to train.
(2) The attention mechanism is integrated into the architecture to model long-range, multi-level dependency relations for affordance detection.
(3) An up-sampling layer is designed to upscale the low-resolution feature map into a high-resolution affordance map.
(4) Evaluation of our model on the UMD [3], verifies that the proposed network achieves high detection accuracy.

2. Related work

2.1. Affordance detection

Affordance prediction has been extensively studied in the computer vision and robotics communities for several years. Early works on affordance detection sought to recognize 3D computer-assisted design objects (e.g., chairs) based on object functions. Stark and Bowyer [26] constructed generic recognition systems according to the functions of rigid 3D objects rather than shapes. They leveraged functional primitive chunks to address the recognition of multiple object categories. Aldoma et al. [27] proposed a visual–cue method to detect affordances in a scene, depending on the geometry and pose of the objects. Their method consists of two steps for affordance detection. First, a classifier is used to recognize the object in the scene. Then, the six-degrees-of-freedom poses of the object are estimated to map its affordances. Myers et al. [3] demonstrated that local shape and geometry primitives could be used to detect the pixel-wise affordances of tool parts. Their novel method was complicated, because the same affordance type could possess multiple appearances. Thus, different affordance types could have similar appearances. Two methods were subsequently introduced to train their models. First, they used superpixel-based hierarchical matching pursuit to extract geometric features (e.g., depth, normal, and curvature information). The features were classified using a linear support vector machine to produce the final affordance maps. The method achieved high accuracy with significant computational costs. In addition, the structured random forest was used to infer affordances based on decision trees, which provided less accurate predictions, but real-time performance.

Inspired by Mayers et al. [3], Nguyen et al. [28] used a CNN to extract geometric deep features instead of hand–engineered features. They proposed an encoder-decoder architecture to detect object affordances on multi-modal features. It was demonstrated that affordance detection with automatic feature learning achieved better performance. Another study [2] combined deep CNN and dense conditional random fields (CRF) to detect object affordances. They first used a deep CNN as an object detector to generate the object boxes. Then another network was used to produce feature maps from bounding boxes. Finally, the CRF, as a post-processing mechanism, improved the predictions. Similar to Nguyen et al. [2], Thanh-Toan et al. [8] proposed AffordanceNet to detect multiple objects and affordances in RGB images. This tool uses two branches: one for object detection and the other for affordance detection. It utilizes object detection to narrow the region of interest and to detect affordances via semantic segmentation.

The advantage of object detection is that it can generate bounding boxes, and separate the whole object from the complex background according to the object category, which make affordance detection easier. However, if the detector cannot detect the object, or if it misrecognizes the object regions, affordance detection will fail. Moreover, after object and affordance detection, the errors will increase cumulatively, and more feature details will be lost, Thus, affordance maps cannot be obtained efficiently. Chu et al. [29] and Lakani et al. [30] decouple the object class from the localized affordance labels of object parts for the learned affordance to generalize to unseen/novel categories. Our method starts from the global of the input image and predicts an affordance label for each pixel without knowing the object class.The predictions are shown in Figure 1. We use a self-attention mechanism to make our architecture focus on the regions of interest instead of using object detection. Doing so can, therefore, improve the affordance representation.

2.2. Attention mechanism

The attention mechanism of deep neural networks aims to model long-range and multi-level dependencies. Recently, attention mechanisms have been widely integrated into deep neural networks for many tasks (e.g., image generation [34], image classification [13] , image captioning [31] and image restoration [35]). Mnih et al. [13] were the first to present a visual attention mechanism based on a recurrent model, which was capable of extracting features from an image by selecting interesting regions. Xu et al. [31] introduced an attention-based model for image captioning, which used hard/soft pooling to select/average the most probable attentive regions or the spatial features with attentive weights. Chen et al. [32] proposed a spatial and channel-wise attention CNN for image captioning, which combined channel-wise and spatial attention into multiple layers. Hu et al. [33] focused on the channel relationships and proposed a squeeze-and-excitation-block to adaptively recalibrate channel-wise feature responses for image classification. Woo et al. [34] presented a convolutional block attention module (CBAM) for feed-forward CNNs. It was composed of a general, lightweight module and could be easily integrated into a CNN. The CBAM operated both channel-wise and spatial attention. Hu et al. [35] designed channel-wise and spatial attention residual blocks to dynamically adjust multi-level features so that they could capture more informative features and maintain longer-term information for image resolution.

Considering the advantages of attention mechanisms, we attempt to embed them into our architecture with the purpose of improving the feature representations of affordance detection. To capture more important information and adaptively model long-term dependency relations both globally and locally, we combine spatial and channel attention into our network.

3. Architecture

3.1. Overview

Following an encoder-decoder architecture, we propose a novel framework to detect object affordances. The architecture is illustrated in Fig. 2. The encoder network consists of a pre-trained model (DRN) and an attention module, where the DRN is used as a pre-trained model to extract features that can initialize the training process from weights trained on large datasets. The attention module is used to improve the feature representation performance. An up-sampling layer that uses the output of the attention module is designed to generate high–resolution output.

3.2. Pre-trained model

Pre-trained models have been widely used for deep neural network feature extraction, initializing weights without learning from scratch. We employ DRN as the backbone for this purpose. As is known, the classical CNN will lose salient detail information by continuously sampling the input image to a very small feature map. The DRN [12] is built upon the Resnet architecture presented by He [11], which can preserve more details without adding parameters. The original Resnet architecture consists of five groups of convolutional layers. The DRN differs from the original network structure in the last two groups. We denote each group as G_l, for $l = 1, \ldots, 5$. We denote the i^{th} layer in group $l$as G_l^i. $F_l^i$is the filter associated with layer G_l^i. Let I be the feature map in the layer, G_l^i. The output in the original model can thus be denoted as:

$$(G_l^i * F_l^i)(I) = \sum_{m+n=I} G_l^i(m)F_l^i(n) \tag{1}$$

The DRN introduces dilated convolutions to the two final groups. In the fourth group, two dilated convolutions are applied instead of the original convolutional operators.

$$(G_4^i * 2F_4^i)(I) = \sum_{m+2n=I} G_4^i(m)F_4^i(n) \tag{2}$$

When $i = 1$, the 5^{th} group also adopts Eq. (2). For all $i \geqslant 1$ in the 5^{th} group, their convolutions must be dilated by a factor of four:

$$(G_5^i * 4F_5^i)(I) = \sum_{m+4n=I} G_5^i(m)F_5^i(n) \tag{3}$$

The DRN aims to increase the receptive field of the convolution kernel as much as possible without reducing the feature map's resolution. We remove the fully connected layer and use it as a backbone to extract features, thus enlarging the size of the output to $1/8$of the input image.

3.3. Attention module

The purpose of the attention module for affordance detection is to focus on object-related features while eliminating irrelevant backgrounds. Inspired by Fu et al. [37], we integrate two attention mechanisms: a spatial attention module (SAM) and a channel attention module(CAM). SAM is used to model long-range spatial interdependencies in images, and CAM is designed to model channel interdependencies. With the output of these two attention

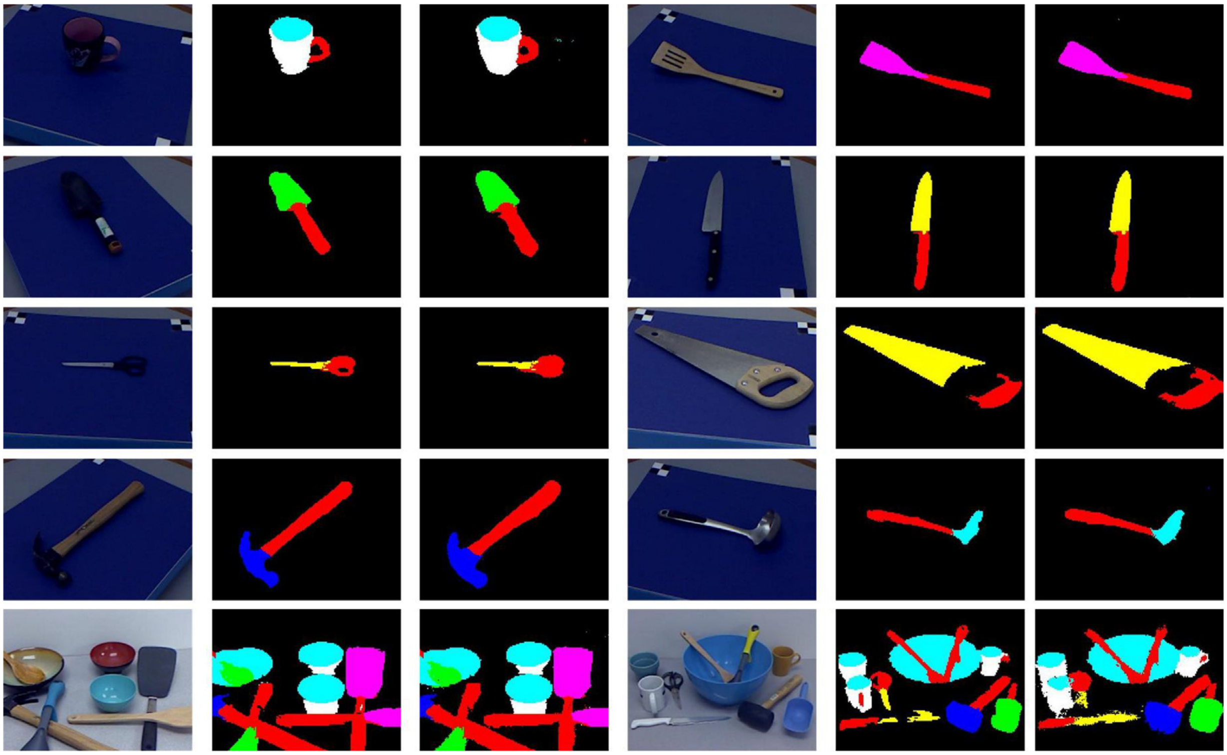


Fig. 1. Our architecture predictions on the UMD. **Left to right**: Input images, ground-truth affordance maps, predicted affordance maps.

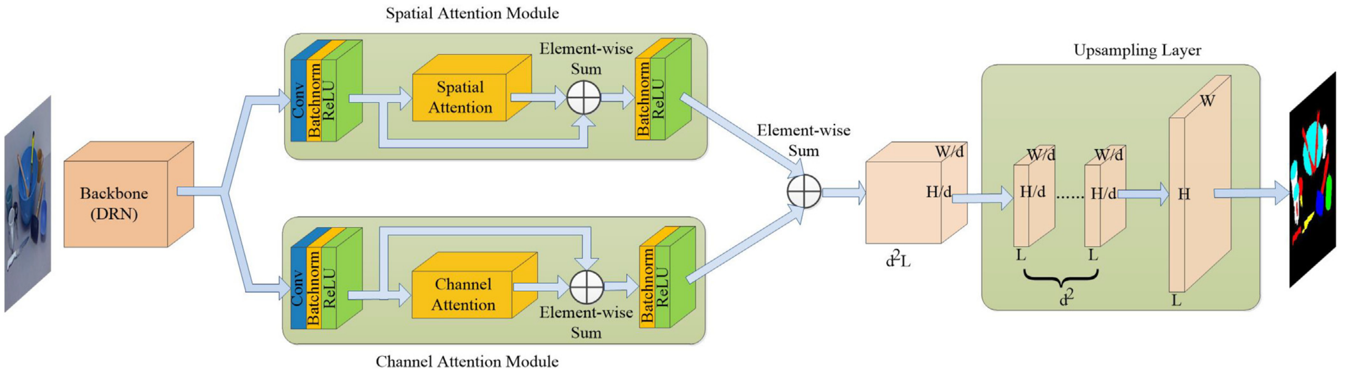


Fig. 2. Architectural overview of the proposed affordance detection method.

modules, we use an element sum to perform feature fusion, which improves feature representation. The specific configurations for SAM and CAM can be found in the -following subsections.

3.3.1. Spatial attention module

Assuming that the input feature map, $A \in \mathbb{R}^{C\times H\times W}$, comes from the previous layer, where $C$is the number of channels, and $H \times W$ the feature–map size, we design the SAM structure to model long-range dependency relations.

As shown in Fig. 3, we first feed A into a kernel size (1×1) convolution layer to generate three feature maps: $B, G \in \mathbb{R}^{C\times H\times W}$ and $D \in \mathbb{R}^{C\times H\times W}$. Then we use B and G to generate $\mathbb{E}^{(H+W-1)\times W\times H}$ via an affinity operation. We can obtain a vector $B_u \in \mathbb{R}^C$, at each position j in the spatial dimension of B. Meanwhile, we can obtain the set $\psi_j \in \mathbb{R}^{(H+W-1)\times C}$ from G which are in the same row or column as position u. $\psi_{i,j}$ is the $i-th$ element of ψ_j. The affinity operation can be denoted as:

$$d_{i,j} = B_u \psi_{i,j}^T \tag{4}$$

where $d_{i,j} \in \mathbb{R}^{(H+W-1)\times(W\times H)}$ is the degree of correlation between features B_j and $\psi_{i,j}, i = [1, \ldots, H+W-1]$. Additionally, we apply a matrix mutiplication between the transposed B and C. Subsequently, we perform a softmax operation to calculate the spatial attention map $E \in \mathbb{R}^{(H+W-1)\times(W\times H)}$:

$$e_{j,i} = \frac{exp(d_{ij})}{\sum_{i=1}^{H+W-1} d_{ij}} \tag{5}$$

At each position $j$in the spatial dimension of D, we can obtain a vector $D_j \in \mathbb{R}^C$ and a set $\varphi_j \in \mathbb{R}^{(H+W-1)\times C}$. The set φ_j is a collection of feature vectors in D that are in the same row or column as position j. We apply an aggregation operation to collect contextual information:

$$F_j = \alpha \sum_{i=1}^{H+W-1} e_{j,i}\varphi_{i,j} \tag{6}$$

where α is a hyperparameter and it is initialized as zero. The introduced α can gradually learn to assign more weight to crisscross evidence [36]. Finally, the input feature maps are added to get the output map $Y$of SAM:

$$Y_j = F_j + A_j \tag{7}$$

3.3.2. Channel attention module

The high-level channel maps in a trained CNN can be regarded as a category-selective classifier. In the affordance detection, we emphasize that the channel maps are also related, and some channels share similar semantic contexts. The motivation for using CAM is to model channel interdependencies. As illustrated in Fig. 4, we calculate channel attention map $Q \in \mathbb{R}^{C\times C}$ from input feature maps $A \in \mathbb{R}^{C\times H\times W}$:

$$q_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} s_{ij}}, where\ s_{ij} = A_i^T A_j \tag{8}$$

Simultaneously, we use reshaped input feature map $A$and the channel attention map to operate a matrix multiplication. Thus, the output $F \in \mathbb{R}^{C\times H\times W}$can be easily calculated:

$$F_j = \alpha \sum_{i=1}^{N} q_{j,i} A_i \tag{9}$$

where α can be learnable to assign weight. Finally, we can obtain the output map of CAM:

$$Y_j = F_j + A_j \tag{10}$$

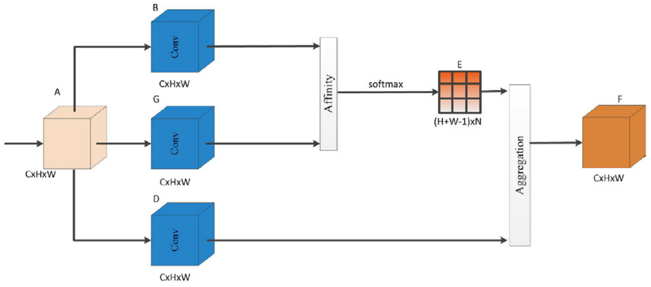


Fig. 3. Structure of the spatial attention module.

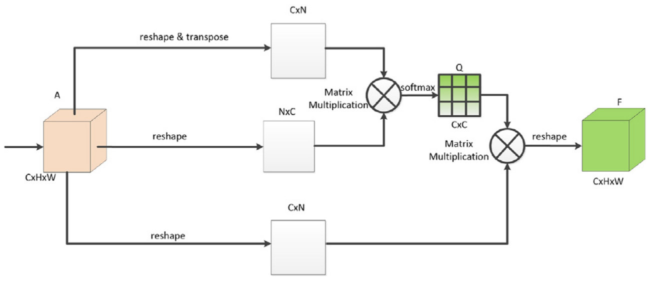


Fig. 4. Structure of the channel attention module.

3.4. Up-sampling layer

To get the high-resolution affordance maps from the encoder network, we design an up-sampling layer, as inspired by Hu et al. [35]. Fig. 2 depicts the architecture of our network with an up-sampling layer. Suppose that the size of an input RGB image is $H \times W$, the number of color channels is three, and the purpose of affordance detection is to label each pixel of the affordance map with a category label. After feeding the image into the encoder network, an $h \times w \times c$feature map will be obtained at the final layer, where $h = H/d, w = W/d$, and $c = d^2L$. d is the down-sampling factor, and Lrepresents the label class of affordance detection. Unlike other up-sampling methods, our up-sampling layer is learnable and uses a pixel shuffling operator to arrange an $h \times w \times c$ feature map into an $H \times W \times L$ output map. The operator uses convolution to divide the input $h \times w \times c$ feature map into d^2 subparts of $h \times w$. The last layer produces the output affordance maps with one up-scaling filter for each subpart without inserting extra values.

4. Experiments

To evaluate the proposed method, we conduct several experiments on the UMD [3]. The experimental results demonstrate that our method achieves good performance on the dataset. First, we introduce the datasets and implementation details. Then, we describe the different evaluation metrics of our experiment. We additionally report a set of ablation experiments conducted to validate the effectiveness of each model component. Finally, we perform more visualizations to illustrate our model.

4.1. Datasets and implementation details

The UMD [3] comprises 28,843 RGB-D images of a non-cluttered subset and 868 RGB-D images of a cluttered subset. This provides affordance labels for daily kitchen, workshop, and garden objects. The dataset contains 17 object categories and seven affordance classes which are summarized in Table 1. We randomly split the dataset into 70% training data and 30% validation data.

For training, we followed the procedure described in [20], using the PyTorch library [38] to train our model. The network was trained using the Adam optimizer with a 0.001 learning rate. The input images were center cropped to 240 × 320 from the original 480 × 640 to better fit the network. The widely-used cross-entropy loss function L was applied to train our network:

$$L(P,G) = -\frac{1}{N}\sum_{i=1}^{N}(g_i log(p_i) + (1-g_i)log(l-p_i)) \quad (11)$$

where $p_i \in P, g_i \in G$, P and G indicate the predicted map and ground-truth affordance map. N represents the total number of pixels. The model was trained on an NVIDIA 2080Ti GPU, which is trained from scratch until convergence with the loss no more reduction.

Table 1
Description of the affordance labels on UMD.

Affordance Label	Description	Example
Grasp	Indicates the location of manipulation of tools.	Hammer, cup
Cut	Indicates cutting for separating object.	Knife, scissors
Scoop	Indicates curved surface tools.	Trowel, spoon
Contain	Indicates the ability of an object to hold liquid.	Bowl, mug
Pound	Indicates striking tools.	Hammer, mallet
Support	Indicates flat parts.	Shovel, turner
Wrap-grasp	Indicates the location of manipulation of rounded tools.	Cup

4.2. Metrics

To evaluate the model, we adopted four metrics to present a comprehensive and insightful performance analysis.

Intersection over union (IoU) is a general metric that can be calculated for each category. Each kind is calculated, accumulated, and then averaged to obtain a mean IoU (mIoU), which reflects the global evaluation. Let P denote the predicted map and Gbe the ground-truth affordance map:

$$IoU = \frac{P \cap G}{P \cup G} \quad (12)$$

Mean absolute error (MAE) is evaluated between the ground-truth affordance map, G, and the predicted map, P. N is the total number of pixels. The MAE measures the conformity between the predicted map and the ground-truth map, and its value ranges from zero to one:

$$MAE = \frac{1}{N}|P - G| \quad (13)$$

Area under the curve (AUC) uses the area beneath the receiver operating characteristic (ROC) to consider the quality of the predicted map. AUC reflects the proportion of positive examples in front of negative ones in the model. We divide the affordance map, P, into 100 gray levels, and the threshold within the range 0 to 99. We calculate True Positive Rate (TPR) & False Positive Rate(FPR) for each threshold and set FPR as the x-axis, and TPR as the y-axis to form the ROC curve. The AUC is the area between the ROC curve and the x-axis.

Average precision (AP) is computed from the precision-recall (PR) curve. We divide the predicted map Pusing a fixed threshold that ranges from 0 to 255. For each threshold, recall & precision scores are computed and combined to form a PR curve to describe the model performance. Meanwhile, AP can be calculated from the PR Curve. It is the average value of the evenly spaced x-axis points from zero to one on the PR Curve.

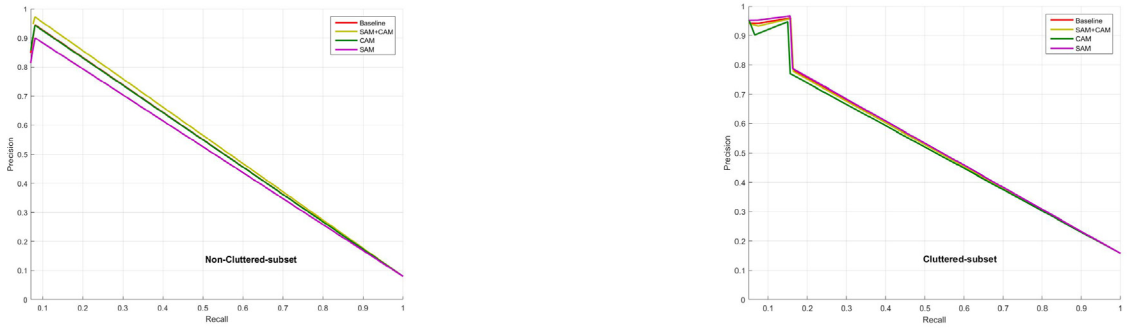
Weighted F_β^w proposed by Margolin et.al [39], is an extension of the F_β measure. It is calculated as follows:

$$F_\beta^w = (1+\beta^2)\frac{Precision^w Recall^w}{\beta^2 Precision^w + Recall^w} \quad (14)$$

where β^2 measures the importance of $Precision^w$ and $Recall^w$.

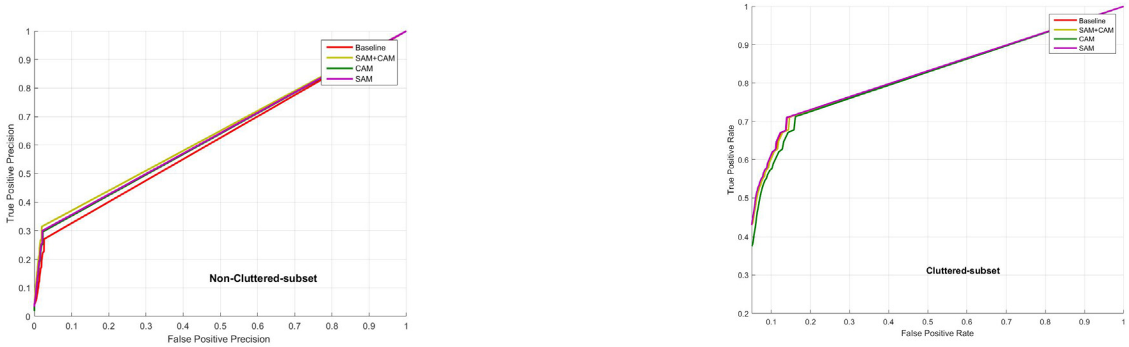
4.3. Ablation study

We employ an attention mechanism atop the DRN to improve the feature representation capability and use an up-sampling layer to generate the high-resolution affordance map. To verify the performance of the attention modules and the up-sampling layer in our network, we evaluated each module on the non-cluttered and cluttered subsets of the UMD. We chose DRN_D_22 with the up-sampling layer as the baseline. To evaluate the effects of the attention modules, we provided three variants: **1)** baseline+CAM; **2)** baseline+SAM; **3)** baseline+CAM+SAM. In this process, we only considered the relationship between the whole graph and the GT graph, rather than measuring each category.



(a) (b)

Fig. 5. PR curve on the non-cluttered subset and cluttered subset of UMD.



(a) (b)

Fig. 6. ROC curve on the non-cluttered subset and cluttered subset of UMD.

As shown in Figs. 5 and 6, AUC and AP cannot effectively evaluate the model. To analyze the effect of attention modules more accurately, we present Table 2. Compared with the baseline, using either SAM or CAM improves the results on the dataset. Employing SAM yields a 0.8816 mIoU in the non-cluttered subset and a 0.8881 mIoU in the cluttered subset, which gains 9.25% and 12.08% improvements, respectively. Meanwhile, employing the CAM individually outperforms the baseline by 7.41% and 12.04% in mIoU. The combination of SAM and CAM brings further mIoU improvements of 0.9387 on the non-cluttered subset and 0.8905 on the cluttered subset. By integrating CAM and SAM into the baseline, the other metrics also improve considerably.

We also evaluated the effect of the up-sampling layer on the model. In our experiment, we used bilinear up-sampling to build the pixel-wise prediction map, but the training process is not convergent, because the parameters of bilinear sampling [16] are not learnable, resulting in a gradient explosion. Un-pooling [20] also cannot be used in our network, because our DRN backbone does not use max pooling. Thus, we replaced the up-sampling layer of our network with a transpose convolution layer. We use mIoU as the metric, and present each class IoU in the result, as illustrated in Table 3. It can be seen that the results are significantly improved using our up-sampling layer compared with the TC method. In particular, we notice that a limitation of the TC method is that it only performs well with some categories having obvious features. For example, the TC method performs well with the $w-grasp, contain$, and *pound* classes ($IoU = 0.848, 0.901, 0.862$, respectively) on the non-cluttered subset. These are obvious classes having large areas. Conversely, it does not perform well with categories lacking obvious features, such as the $grasp, support$, and *scoop*classes ($IoU = 0.730, 0.762, 0.761$, respectively). This limitation does not appear in our method that includes that cluttered subset, which demonstrates that our method can guarantee a high-quality map of the low-resolution encoder feature maps to high-resolution affordance maps.

4.4. Comparison with state-of-the-art methods

We compared our method with extant methods on using the UMD. For a fair comparison, we did not use post-processing (e.g., CRF), but we used the same evaluation metric, F^w_β.

In practice, we first merged the prediction affordances having different masks into an image. We then converted the RGB image into a gray image, turning the affordance labels into gray levels. The F^w_β ultimately was used to calculate the accuracy of each affordance.

Our architecture clearly yielded overall good performance on the UMD. As shown in Table 4 and Table 5, our model obtains 0.941 accuracy on the non-cluttered subset of UMD, and 0.921 on the cluttered subset, which significantly outperforms existing methods. In particular, the HMF and SRF [3] methods are based on hand-designed features, and our model outperforms these by 38.4% and 48.1%, *respectively*, on the non-cluttered subset and, 67.1% and 75.6%, *respectively*, on the cluttered subset. We notice that deep CNN can learn deep features from the dataset, significantly improving its performance over methods based on hand-designed features. Notably, we emphasize the comparison between our method and AffordanceNet [8]. AffordanceNet first adopts object detection to narrow the region of interest, and then performs affordance detection. For fair comparison with AffordanceNet, we used VGG-16 as the backbone and obtained an improvement of 2.4%. Our method avoids the problems of missed detection and false detection caused by object detection, and improves the accuracy of predicting the affordance types of global pixels. Our model using DRN as the backbone achieves good performance on the non-cluttered subset, demonstrating that our model can retain more details to improve accuracy over methods using object detection.

4.5. Visualization

We conducted a set of attention visualizations on the final output maps of the baseline model and ours, as shown in Fig. 7. Notice that the output feature maps from the baseline model present little attentiveness. Our model introduces the SAM and CAM, which empowers the network to focus more on the region with tools. The existing CNN method for affordance detection usually utilizes object detection to narrow the region of interest. It then detects affordances using semantic segmentation. The accuracy of these methods is not only affected by affordance detection, but it is also affected by object detection. We integrate the attention modules into the detection architecture, making the model focus on interest by assigning weights instead of using bounding boxes.

5. Conclusion and future work

Deep CNN has been widely used in the fields of computer vision and robotics. However, factors such as memory and computation time during training and testing must be considered. These factors are related to the parameters with which the network needs to learn. The number of parameters in our model is 16M, including the pre-trained model and it is considerably smaller than other methods. In practical applications, a light-weight network is very important, because it can integrate a real-time system, according to the pictures collected by the camera, including real-time detection of the required information.

Affordance detection has been an researched extensively in recent years. Many works have been devoted to researching its related problems. The key challenge of affordance detection is classifying the same affordance type against different appearances and types having the same appearance. Existing CNN methods mainly adopt object detection to narrow the region of required objects with bounding boxes. Then, affordance detection is applied with semantic segmentation. However, it proved beneficial to integrate attention modules into our network. First, it does not require object detection, freeing-up a significant amount of computing resources. It can also model long-range dependency relations to boost performance in an end-to-end manner. We trust that our

Table 2
Performance on UMD, using different attention modules on non-cluttered subset and cluttered subset of UMD.

Models	Non-cluttered subset (single objects)					Cluttered subset (multiple objects)				
	mIoU	MAE	AUC	AP	F_β^w	mIoU	MAE	AUC	AP	F_β^w
Baseline	0.789	0.050	0.995	0.891	0.763	0.767	0.093	0.893	0.853	0.822
CAM	0.863	0.022	0.9970	0.925	0.845	0.887	0.095	0.962	0.811	0.833
SAM	0.882	0.026	0.997	0.911	0.911	0.888	0.083	0.988	0.896	0.875
CAM+SAM	0.938	0.012	0.998	0.972	0.941	0.891	0.089	0.993	0.859	0.921

Table 3
Comparison of our up-sampling layer and transpose convolution (TC).

Affordances	Non-cluttered subset (single objects)		Cluttered subset (multiple objects)	
	TC	Ours	TC	Ours
Background	0.987	0.995	0.963	0.965
Grasp	0.730	0.857	0.878	0.845
w-grasp	0.848	0.943	0.781	0.809
Cut	0.800	0.926	0.820	0.886
Contain	0.901	0.964	0.938	0.938
Support	0.762	0.938	0.805	0.910
Scoop	0.761	0.940	0.869	0.904
Pound	0.862	0.948	0.849	0.866
mIoU	0.831	0.939	0.863	0.891

Table 4
Performance on UMD, comparing state-of-the-art methods on the non-cluttered subset of UMD.

Affordances	Non-cluttered subset (single objects)						
	HMF	SRF	DeepLab	CNN	AffordanceNet	Ours(VGG16)	Ours(DRN)
Grasp	0.367	0.314	0.620	0.719	0.731	0.784	0.904
w-grasp	0.373	0.285	0.730	0.769	0.814	0.822	0.955
Cut	0.415	0.412	0.600	0.737	0.762	0.761	0.924
Contain	0.810	0.635	0.900	0.817	0.833	0.840	0.954
Support	0.643	0.429	0.600	0.780	0.821	0.844	0.963
Scoop	0.524	0.481	0.800	0.744	0.793	0.862	0.927
Pound	0.767	0.666	0.880	0.794	0.836	0.847	0.956
Average	0.557	0.460	0.733	0.766	0.799	0.823	0.941

Table 5
Performance on UMD, comparing state-of-the-art methods on the cluttered subset of UMD.

Affordances	Cluttered subset (multiple objects)			
	HMF	SRF	Ours (VGG16)	Ours (DRN)
Grasp	0.227	0.200	0.856	0.896
w-grasp	0.208	0.156	0.806	0.921
Cut	0.160	0.072	0.865	0.863
Contain	0.437	0.322	0.952	0.962
Support	0.297	0.098	0851	0.967
Scoop	0.165	0.114	0.912	0.893
Pound	0.257	0.072	0.886	0.944
Average	0.250	0.165	0.875	0.921

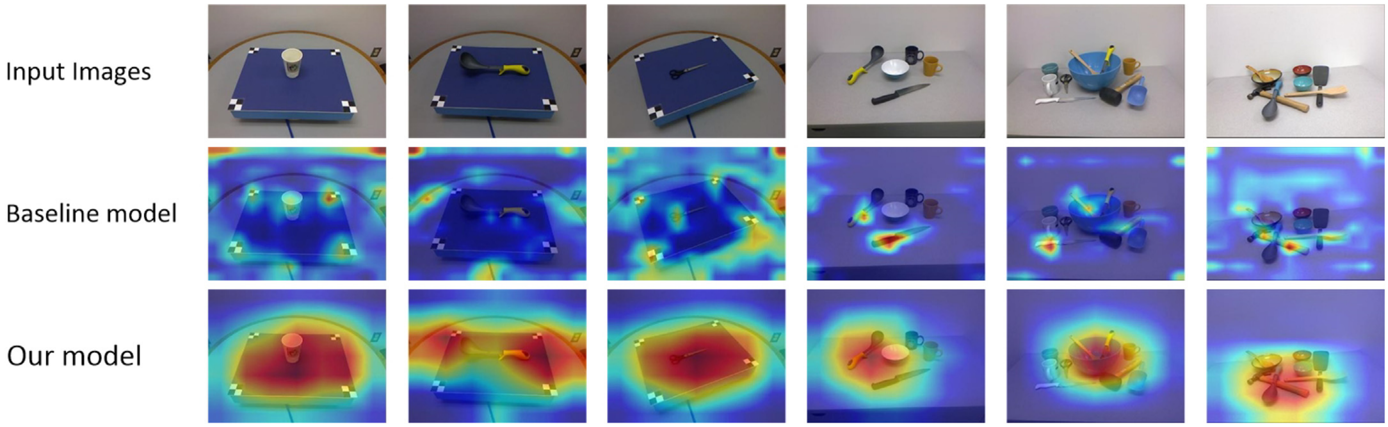


Fig. 7. Visualization results of attention modules on UMD.

experiments will encourage researchers to explore affordance detection in a multiplicity of directions.

In this paper, we proposed a novel deep CNN architecture for affordance detection. The main motivation was to design an efficient architecture to realize the labeling of each affordance type in an RGB image. We used a DRN instead of a VGG or Resnet as a backbone to extract features, because it retains more spatial information and improves the ability of feature representation. Moreover, we integrated the attention mechanism into our architecture to model long-range, multi-level dependency relations. We also verified that our attention modules are efficient and that they improve performance on the dataset via an ablation study. Moreover, we designed an up-sampling layer to map a low-resolution feature map from the encoder network output to a high-resolution affordance output. Different evaluation metrics have shown that our method is efficient and highly accurate. Our method on the UMD outperformed existing state-of-the-art methods.

For the future, we plan to exploit the 2D affordance detection framework to design 3D object affordance detection based on a deep neural network. We are also interested in the application of unsupervised methods of affordance detection.

CRediT authorship contribution statement

Qipeng Gu: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Jianhua Su:** Formal analysis, Investigation, Resources, Writing - review & editing, Funding acquisition. **Lei Yuan:** Project administration, Funding acquisition, Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by NSFC under Grant number 91848109, the Beijing Natural Science Foundation under Grant number 4182068, the Science and Technology on Space Intelligent Control Laboratory under HTKJ2019KL502013, and the State Key Laboratory of Rail Traffic Control and SafetyContract No. RS2018K009, Beijing Jiaotong University, and Major scientific and technological innovation projects in Shandong Province NO. 2019JZZY010430.

References

[1] J.J. Gibson, The theory of affordances, in: Perceiving, Acting, and Knowing: Toward and Ecological Psychology (1979) 62–82..
[2] Anh Nguyen, Dimitrios Kanoulas, Darwin G. Caldwell, Nikos G. Tsagarakis, Object-based affordances detection with convolutional neural networks and dense conditional random fields, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, https://doi.org/10.1016/j.neucom.2021.01.018.
[3] Austin Myers, Ching L. Teo, C. Fermller, Yiannis Aloimonos, Affordance detection of tool parts from geometric features, IEEE International Conference on Robotics and Automation (2015) 1374–1381, https://doi.org/10.1109/ICRA.2015.7139369.
[4] Hema S. Koppula, Ashutosh Saxena, Anticipating human activities using object affordances for reactive robotic response, IEEE Transactions on Pattern Analysis & Machine Intelligence 38 (1) (2015) 14–29, https://doi.org/10.1109/TPAMI.2015.2430335.
[5] David F. Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A. Efros, Ivan Laptev, Josef Sivic, People watching: human actions as a cue for single view geometry, International Journal of Computer Vision 110 (3) (2014) 259–274, https://doi.org/10.1007/s11263-014-0710-z.

[6] Chenyi Chen, Ari Seff, Alain Kornhauser, Jianxiong Xiao, DeepDriving: learning affordance for direct perception in autonomous driving, IEEE International Conference on Computer Vision (2015) 2380–7504, https://doi.org/10.1109/ICCV.2015.312.

[7] Roy, Anirban, Todorovic, Sinisa.10.1109/CVPR.2017.667 A multi-scale CNN for affordance segmentation in RGB images, European Conference on Computer Vision (2016). doi: 10.1007/978-3-319-46493-0_12.

[8] Do, Thanh Toan, Nguyen, Anh, Reid, Ian, AffordanceNet: an end-to-end deep learning approach for object affordance detection, IEEE International Conference on Robotics and Automation (2018) 2577-087X. doi: 10.1109/ICRA.2018.8460902.

[9] Marshall F. Tappen, William T. Freeman, Edward H. Adelson, Recovering intrinsic images from a single image, IEEE Transactions on Pattern Analysis & Machine Intelligence 27 (9) (2005) 1459–1472, https://doi.org/10.1109/TPAMI.2005.185.

[10] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science (2014).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Sun Jian, Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition (2016) 770–778, https://doi.org/10.1109/CVPR.2016.90 10.1109/CVPR.2017.667.

[12] Fisher Yu, Vladlen Koltun, Thomas Funkhouser, Dilated Residual Networks IEEE Conference on Computer Vision and Pattern Recognition (2017) 636–644, https://doi.org/10.1109/CVPR.2017.75.

[13] Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu, Recurrent models of visual attention, Advances in Neural Information Processing Systems (2014).

[14] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang, Residual attention network for image classification, IEEE Conference on Computer Vision and Pattern Recognition (2017) 6450–6458, https://doi.org/10.1109/CVPR.2017.683.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu, Spatial transformer networks, Advances in Neural Information Processing Systems (2015).

[16] Heechang Kim, Sangjun Park, Jin Wang, Yonghoon Kim, Jechang Jeong, Advanced bilinear image interpolation based on edge features, First International Conference on Advances in Multimedia (2009) 978–0–7695–3693-4. doi: 10.1109/MMEDIA.2009.14..

[17] Matthew D Zeiler, Rob Fergus, Visualizing and understanding convolutional networks, European Conference on Computer Vision (2014) 818–833, https://doi.org/10.1007/978-3-319-10590-1_53.

[18] Augustus Odena, Vincent Dumoulin, Chris Olah, Deconvolution and checkerboard artifacts, Distill (2016), https://doi.org/10.23915/distill.00003.

[19] Vincent Dumoulin, Francesco Visin, A guide to convolution arithmetic for deep learning, Arxiv abs/1603.07285. (2016).

[20] Volodymyr Turchenko, Eric Chalmers, Artur Luczak, A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe, ArXiv abs/1701.04949. (2017).

[21] Fu. Jun, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, Lu Hanqing, Dual attention network for scene segmentation, IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019), https://doi.org/10.1109/CVPR.2019.00326.

[22] Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis & Machine Intelligence 39 (4) (2014) 640–651, https://doi.org/10.1109/CVPR.2015.7298965.

[23] Alec Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, Computer ence (2015).

[24] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[25] Hyeonwoo Noh, Seunghoon Hong, Bohyung Han, Learning deconvolution network for semantic segmentation, IEEE International Conference on Computer Vision (2015) 2380–7504, https://doi.org/10.1109/ICCV.2015.178.

[26] L. Stark, K. Bowyer, Function-based generic recognition for multiple object categories, Cvgip Image Understanding 59 (1) (1994) 1–21, https://doi.org/10.1006/ciun.1994.1001.

[27] Aitor Aldoma, Federico Tombari, Markus Vincze, Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes, Proceedings - IEEE International Conference on Robotics and Automation (2012) 1732–1739, https://doi.org/10.1109/ICRA.2012.6224931.

[28] Anh Nguyen, Dimitrios Kanoulas, Darwin G. Caldwell, Nikos G. Tsagarakis, Detecting Object Affordances with Convolutional Neural Networks IEEE/RSJ International Conference on Intelligent Robots and Systems (2016) 2153–0866. doi: 10.1109/IROS.2016.7759429.

[29] Chu, Fu Jen, Ruinian Xu, Landan Seguin, Patricio A. Vela, Toward affordance detection and ranking on novel objects for real-world robotic manipulation, IEEE Robotics and Automation Letters 4 (4) (2019) 4070–4077, https://doi.org/10.1109/LRA.2019.2930364.

[30] Lakani, Safoura Rezapour, Rodrguez-Snchez, Antonio J., Piater, Justus, Towards affordance detection for robot manipulation using affordance for parts and parts for affordance, Autonomous Robots (2019) 43,1155–1172 doi: 10.1007/s10514-018-9787-5..

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, Show, attend and tell: neural image caption generation with visual attention, International Conference on Machine Learning (2015) 2048–2057.

[32] Chen Long, Hanwang Zhang, Jun Xiao, Liqiang Nie, Shao Jian, Liu Wei, Tat Seng Chua, SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning, IEEE Conference on Computer Vision and Pattern Recognition (2017) 6298–6306, https://doi.org/10.1109/CVPR.2017.667.

[33] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-excitation networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1) (2019), https://doi.org/10.1109/TPAMI.2019.2913372.

[34] Sanghyun Woo, Jongchan Park, Joon Young Lee, In So Kweon, CBAM: convolutional block attention module, European Conference on Computer Vision (2018) 3–19, https://doi.org/10.1007/978-3-030-01234-2_1.

[35] Yanting Hu, Jie Li, Yuanfei Huang, Xinbo Gao, Channel-wise and spatial feature modulation network for single image super-resolution, IEEE Transactions on Circuits and Systems for Video Technology (2019), https://doi.org/10.1109/TCSVT.2019.2915238.

[36] Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena, Self-attention generative adversarial networks, International Conference on Machine Learning (2019)..

[37] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, Hanqing Lu, Dual attention network for scene segmentation, The IEEE Conference on Computer Vision and Pattern Recognition (2019) 3146–3154, https://doi.org/10.1109/CVPR.2019.00326.

[38] Paszke, Adam, Gross, Sam and Massa, Francisco, Lerer et al., PyTorch: An imperative style, High-Performance Deep Learning Library Curran Associates Inc (2019) 8024–8035..

[39] Ran, Margolin, Zelnik-Manor, Lihi, Tal, Ayellet, How to evaluate foreground maps. IEEE Conference on Computer Vision and Pattern Recognition, (2014) 978–1–4799–5118–5 doi: 10.1109/CVPR.2014.39.

Qipeng Gu received the B.Eng. degree in automation engineering from China University of Petroleum(Beijing), Beijing, China, in 2018. Then, he is studying for a master's degree in the Institute of Automation, Chinese Academy of Sciences, Beijing,China. His background is in the fields of computer vision, robotics and deep learning. His current research interests include object detection, instance segmentation and robotic manipulation.

Jianhua Su received the B.Eng. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China, in 1999, the M.Eng. degree in Electronic and information engineering from the Beijing Jiaotong University, Beijing, in 2004, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009. He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His background is in the fields of control theory, robotics, automation, and manufacturing. His current research interests include intelligent robot system and train control system.

Lei Yuan received the M.S. degree in School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China in 2004 and then joined this university as a teacher in the same year. He is now an associate professor of the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing. His research areas are the design, test and simulation method for Train Control System of high-speed railway.