Neurocomputing 428 (2021) 206–217

Contents lists available at ScienceDirect

Neurocomputing

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

Cascaded hourglass feature fusing network for saliency detection ☆

Huiyuan Luo [a,b], Guangliang Han [a,*], Xiaotian Wu [a], Peixun Liu [a], Hang Yang [a], Xin Zhang [a,b]

[a] *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun 130033, China*
[b] *University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 11 March 2020
Revised 11 November 2020
Accepted 21 November 2020
Available online 05 December 2020

Keywords:
Saliency detection
Cascaded hourglass
Feature fusing
Pooling operations

ABSTRACT

Convolutional neural networks have been widely applied in saliency detection task because of its powerful feature extraction capability. Most of existing saliency detection models have achieved great progress by aggregating the strong multi-level features. However, it is still a challenging task to design the feature fusing strategy because of the various differences between multi-level features. In this paper, we explore the effect of cascaded pooling operations for saliency detection and propose a novel network to decode saliency cues from multi-level features progressively. We refer to the architecture as “cascaded hourglass” feature fusing network. The proposed network equips with three cascaded sub-modules to capture the multi-scale context and integrate multi-level features progressively. Specifically, we first propose a multi-scale context-aware feature extraction block with different dilated convolutional branches to obtain multi-scale context-aware saliency cues. Then, a hourglass feature fusing block with successive steps of pooling operations is applied to convert the features to multiple feature spaces. Furthermore, we stack a serial of the hourglass feature fusing blocks to purify the multi-level coarse features progressively. Finally, we combine the selective features with cascaded feature decoder to produce final saliency map. Extensive experiments demonstrate the proposed network compares favorably against state-of-the-art methods. Additionally, our model is efficient with the real-time speed of 28 FPS when processing a 400×300 image.

1. Introduction

Salient object detection aims to find the most discriminative objects or regions in images or videos. Primitively, it is an auxiliary way to explore the human vision and cognition mechanism. But recently, it has been developed as a pre-processing method in extensive computer vision tasks, such as object recognition [1], image or video segmentation [2,3], content-aware image editing [4], stereoscopic thumbnail generation [5] and object tracking [6].

Many saliency detection methods have been proposed in the past decade. Generally, the proposed methods can be simply divided into two categories: traditional methods, and deep-learning based methods. Traditional methods are mainly based on low-level cues and hand-crafted features, such as color features [7,8], spatial distribution [9,10], and so on [11–15]. However, these low-level features lack of semantic information, which may fail to detect the accurate salient regions in complex scenarios. Recently, convolutional neural networks (CNNs) have been widely used in various computer vision tasks [16,17] because of its powerful capability in visual feature representation. Similarly, benefit from the powerful multi-level features, especially the high-level semantics, most CNNs-based saliency detection methods have achieved a remarkable progress. Generally, most saliency detection methods adopt an encoder-decoder architecture: CNNs serve as the encoder to extract multi-level features, while their proposed structures take as the decoder to extract and purify saliency cues from multi-level features progressively. Many effective features decoding strategies have been designed, such as short connection [18], gate mechanism [19], attention model [20–23], residual learning [24,25], edge-aware model [26–28], and so on [29,30]. However, since different salient objects or regions possess extensive uncertain diversities in shape, size and features, it can still be a challenging task to decode the saliency cues from multi-level features.

In the architecture of CNNs, with the repeated stride and pooling operations, the extracted features gradually change from low-level representation to high-level representation. Specifically, low-level features contain more spatial details but full of noises, while high-level features represent rich semantics but lack of

☆ This work is supported by the National Science Foundation of China under Grant 61401425, 61602432.

* Corresponding author.
E-mail addresses: luohuiyuan@ciomp.ac.cn (H. Luo), hangl_ciomp@163.com (G. Han), wuxiaotian@ciomp.ac.cn (X. Wu), liupx@ciomp.ac.cn (P. Liu), yanghang@ciomp.ac.cn (H. Yang), zhangxin162@mails.ucas.ac.cn (X. Zhang).

https://doi.org/10.1016/j.neucom.2020.11.058
0925-2312/© 2020 Elsevier B.V. All rights reserved.

accurate location information. On the one hand, high-level and low-level features are complementary with each other, which is helpful for saliency detection. On the other hand, it also means that multi-level features are significantly different with each other, which will inevitably bring some noises to the result and degree the performance if we integrate multi-level features without discrimination. It has been proved by Wu et al. [31]that saliency detection is a low-level computer vision task which is more dependent on the high-level semantic information. How to capture the high-level semantic information is the crucial issue for saliency detection. In the architecture of CNNs, the successive convolutional operations can filter the features from coarse-level to refined-level, while the repeated pooling operations can transform the features from the low-level representation to the high-level representation. Therefore, the pooling operations can be an available method for saliency detection, which has been proved by Liu et al. [32] with the top-level performance for saliency detection.

In this paper, we further explore the intrinsic mechanism of pooling operations and propose a novel cascaded hourglass feature fusing network for saliency detection. Specifically, the proposed network contains three cascaded sub-modules: cascaded multi-scale context-aware feature extraction module (CMCFEM), cascaded hourglass feature fusing module (CHFFM), and cascaded feature decoder (CFD). Specifically, we first adopt a multi-scale context-aware feature extraction block (MCFEB) which contains a group of cascaded dilated convolutional operations to capture multi-receptive-filed features. Furthermore, we cascade a serial of MCFEBs together and form a top-down pathway as the CMCFEM to transmit the multi-receptive-filed features progressively. Second, we construct a hourglass feature fusing block (HFFB) with four cascaded encoder-decoder sub-branches to convert the features from one scale into multiple feature scales. Besides the first sub-branch, each sub-branch contains a pair of down-sample and up-sample operation: down-sample operation converts the feature to a new scale, while up-sample operation recovers the feature to the original size. In the middle of each sub-branch, two convolutional layers are embedded to capture more local context information and filter the coarse features. We cascade the four encoder-decoder sub-branches together and transmit the output of each sub-branch to next sub-branch, which weighs the coarse features from local receptive field to global receptive field progressively. Furthermore, to filter multi-level coarse features gradually, we hierarchically couple a serial of HFFBs as the cascaded hourglass feature fusing module (CHFFM). Since the structures of these cascaded blocks are seemed as a serial of cascaded hourglasses with different sizes, we denote our network as "cascaded hourglass" feature fusing network. Finally, four feature decoder blocks are equipped as cascaded feature decoder(CFD) to combine these refined features and make the final prediction. To overcome the ambiguities of the saliency edges and learn the hard pixels surrounding the boundaries, we adopt a weighted cross entropy loss to train the proposed model.

Extensive experiments demonstrate the performance of the proposed method can compare favorably against state-of-the-art methods. We conclude our main contributions as follows:

1. We propose a multi-scale context-aware feature extraction block (MCFEB) with a group of cascaded dilated convolutional operations to extract multi-receptive-filed features. Furthermore, we cascade a serial of MCFEBs together as CMCFEM to transmit the multi-receptive-filed features progressively.
2. We design a hourglass feature fusing block with cascaded pooling sub-branches to transform features from one-level to multiple scales. Moreover, a serial of hourglass features fusing blocks are stacked together to integrate multi-level features hierarchically.
3. The proposed model can achieve the state-of-the-art performance under different metrics. Furthermore, our method is not only comparably lightweight but also efficient. When processing a 400 × 300 image on an NVIDIA TITAN Xp GPU, the model based on ResNet-50 can run with the real-time speed of 28 FPS.

2. Related works

Many saliency detection methods have been proposed over the past decade. Early approaches produce the saliency map based on the low-level features. However, benefiting from the powerful feature extraction capability of CNNs, most CNNs-based methods can achieve comprehensive superiorities compared with the traditional methods. In this paper, we mainly focus on the deep-learning based saliency detection models in recent years.

Most of existing salient object detection networks are devoted to aggregate the multi-level features from CNNs to improve the performance. As mentioned in Section 1, many feature fusing strategies have been proposed, such as short connections, gate mechanism, attention model, residual learning, edge-aware saliency detection, and so on. For example, Hou et al. [18] present a skip-layer structures via dense short connections to fully integrate multi-level features together. Zhang et al. [19] construct a bi-directional gate structure between adjacent convolutional layers to bilaterally filter multi-level coarse features. Liu et al. [35] design an encoder-decoder structure to convert a coarse global prediction to refined saliency map hierarchically and progressively. Luo et al. [29] design a multi-relation grid structure to combine multi-scale local, deconvolution and global information together. Zhuge et al. [36] propose a novel approach that transforms prior information into an embedding space to select attentive features and filter out outliers for salient object detection. Zhang et al. [37] present a generic aggregating multi-level features framework to integrate multi-level feature maps into multiple resolutions. Wu et al. [31] abandon low-level features and only decode saliency cues from high-level features via a cascaded partial decoder framework. Deng et al. [24] design a recurrent residual refinement network equipped with residual refinement blocks to alternately extract different saliency cues from multi-level features. Mohammadi et al. [38] propose a feature guide network to distinguish the ambiguous salient and non-salient regions.

Attention mechanism is widely used in various vision task for its great ability to select features, such as fixation prediction [39,40], video saliency detection [41–43] and saliency detection [20,23,44,25,21]. For example, Zhang et al. [20] propose a novel attention guided network to selectively integrate multi-level features in a progressive manner. Zhao et al. [23] explore the different characteristics of multi-level features and purposefully utilize different attention module for the high-level and low-level features. Zhu et al. [44] present an output-guided attention module built with multi-scale outputs to overcome the problem of faulty information produced by traditional attention module. Chen et al. [25] embed the attention and residual refinement network together to guide feature learning in side out layers and then fuse the learned features for saliency detection. Chen et al. [21] utilize reverse attention to guide side-output residual learning in a top-down manner.

Recently, to capture the structural information of salient objects, more and more researchers construct their networks to extract the edge information or train their networks with structure information as auxiliary supervision. For example, Zhuge et al. [45] propose a boundary-guided aggregating feature fusion network for salient object detection. Zhou et al. [46] propose a siamese edge-enhancement network to preserve the edge structure for salient object detection. Su et al. [28] rethink saliency detection in terms of the selectivity or invariance of different features to construct the network. Zhao et al. [26] extract the salient edge information and salient object information simultaneously, and then guide the saliency detection with edge information. Liu et al. [32] introduce the pooling operation to expand the receptive fields of features and join edge information to train the whole network.

In a word, most of existing saliency detection methods are devoted to extract the saliency cues from multi-level features and integrate them with different strategies. In this paper, we further explore the effect of pooling operations in the saliency detection task and propose a cascaded hourglass feature fusing network for saliency detection.

3. The proposed network

The proposed network consists of three sub-modules: cascaded multi-scale context-aware feature extraction module (CMCFEM) to extract the multi-scale context features, cascaded hourglass feature fusing module (CHFFM) to progressively refine the combined multi-level features, and cascaded feature decoder (CFD) to decode the final saliency cues from the refined features. The whole structure of our proposed network is shown in Fig. 1.

3.1. Cascaded multi-scale context-aware feature extraction module

Multi-scale context information is beneficial to prevent the degradation of performance which caused by various diversities of salient objects. However, it is proved by zhang [47] that the empirical receptive field size of CNN is much smaller than the theoretical one especially on high-level layers. This will make many CNN-based saliency detection methods incorporate the momentous global scenery prior insufficiently. As above mentioned, saliency detection is a visual task which is more dependent on high-level semantics. Thus, it is necessary to expand the receptive field to facilitate the sufficient incorporation of global semantics. Atrous spatial pyramid pooling(ASPP) module [48] is a commonly used module in many saliency detection models [19,23]. It utilizes a group of dilated convolutional operations with different dilated rates to extract multi-receptive-fields contextual information. However, a convolutional operation with large dilation rate may also bring a gridding issue. Besides this, convolutional operations with large kernel sizes are also frequently used to capture the local contextual information [38]. But the large kernel size may increase the parameters of model and need more computer memories.

In this paper, we utilize a group of dilated convolutional operations and further cascade them together to excavate the multi-receptive-fields information progressively. As shown in Fig. 2, we first add a 3×3 convolutional layer to learn more local information and then split the input features into four sub-branches with a group of 1×1 convolutional operations. A 3×3 dilated convolutional layer is embedded in each branch to capture more local context. The dilation rates of the four dilated sub-branches are set to $\{1, 2, 4, 6\}$, respectively. Generally, the larger dilation rate can help the network to capture larger receptive-field context, but the larger dilation rate may also cause graver gridding effect and loss more spatial details. To counteract these drawback, we arrange the four dilated sub-branches from small dilation to large dilation and further introduce the short connections into the structure. Generally, the output features from the sub-branch with smaller dilation rate contains more spatial details, which can help the sub-branch with larger dilation rate remedy the gridding effect to some extent. Therefore, we transmit the output of one sub-branch to next sub-branch and add it to the original inputs of next sub-branch. Furthermore, inspired by ASPP, we adopt an image pooling branch with a global average pooling operation and a 1×1 convolutional layer to capture the image-level saliency cues. Finally, we concatenate all the outputs of the five sub-branches and employ a combination of 3×3 and 1×1 convolutional layer to integrate them together.

We stack four MCFEBs together as cascaded multi-scale context-aware feature extraction module (CMCFEM) to learn more local context from the multi-level side-outputs features. Given the VGG-Net version of FPN [49] as an example, we use the feature maps outputted by conv2, conv3, conv4, conv5 as the four side-output features. The four side-output feature maps have strides of $\{2, 4, 8, 16\}$ pixels with respect to the input image, and the channel numbers of four side-output features are set to $\{128, 256, 512, 512\}$, respectively. In this paper, to reduce the computation complexity, we reduce the channel numbers of all side-outputs features to 64 with 1×1 convolutional layer. Moreover, we transmit the output of previous block to the next and concatenate it with the side-output features as the mixed input for each MCFEB. Finally, we capture four corresponding side-output features from the four MCFEBs and deliver them to the subsequent sub-modules.

3.2. Cascaded hourglass feature fusing module

The multi-scale contextual information contained in the side-output features from CMCFEM is helpful for saliency detection. However, the multi-level features are only simply concatenated together in CMCFEM, which may bring some noises to the four side-output features. In this section, we design a hourglass feature fusing block (HFFB) and further stack them with connections. All of the connected HFFBs form a cascaded hourglass feature fusing module (CHFFM), which can dispel the noises and integrate the side-output features hierarchically.

As above mentioned, in the feature encoding process, the repeated stride and pooling operations can translate the features from low-level representation to the high-level representation. Similarly, in the feature decoding process, the pooling operations can be considered as a feasible method to eliminate the diversities between multi-level features. In this section, we propose a cascaded hourglass feature fusing module with a serial of pooling operations to integrate the multi-level features. The whole structure of block is shown in Fig. 3. We are not the first one to explore the effect of pooling operations for saliency detection. In PoolNet [32], Liu et al. have proved the huge potential of pooling operation for saliency detection with the top-level performance. Fig. 4 shows the basic pooling unit in PoolNet [32], which is called feature aggregation module (FAM). The pooling branches in FAM are organized by a parallel way, while these in the proposed module are stacked in a cascaded way. The number of sample operations of FAM is 6, while the number of the proposed module is 8. For the same scale input feature, the receptive filed of HFFB can reach to 68, which is larger than FAM with 40. Generally, our module can more sufficiently extract the feature in a larger receptive filed, while FAM can extract the feature in a smaller receptive filed but with relatively less parameters. We will provide more comparisons in the experiment section.

As shown in Fig. 3, we first add all input multi-level features together as the mixed input of this module, and then construct four sub-branches to progressively transform the features from coarse-level to refined-level. Specifically, we first adopt a sub-branch without any pooling operations to learn more local context from the mixed input features. Subsequently, three cascaded pooling sub-branches are embedded into the structure and the down-sampling rates of three pooling sub-branches are set to $\{2, 4, 8\}$ compared to the scale of the input feature maps, respectively. In each pooling sub-branch, a down-sample operation is used to transform the feature to the new scale feature space, and a

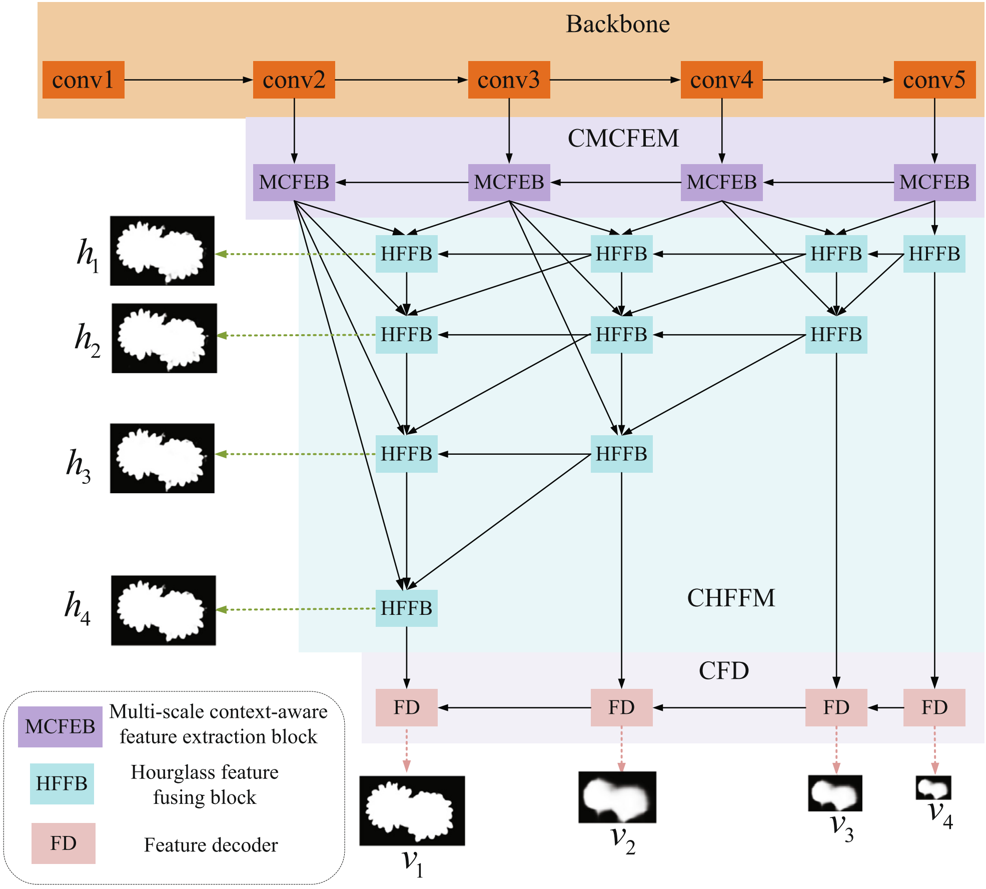


Fig. 1. The framework of our proposed network. CMCFEM represents the cascaded multi-scale context-aware feature extraction module. CHFFM is the cascaded hourglass feature fusing module. CFD is the cascaded feature decoder. $\{h_1, h_2, h_3, h_4\}$ are the predictions of the four horizontal pathways, and all the down-sampling rates in comparison of the input image are $1/2$ when VGGNet-19 [33] as backbone ($1/4$ when ResNet-50 [34] as backbone). While $\{v_1, v_2, v_3, v_4\}$ are the predictions of the four vertical pathways, and the down-sampling rates are $\{1/2, 1/4, 1/8, 1/16\}$ respectively when VGGNet-19 as backbone ($\{1/4, 1/8, 1/16, 1/32\}$ when ResNet-50 as backbone).

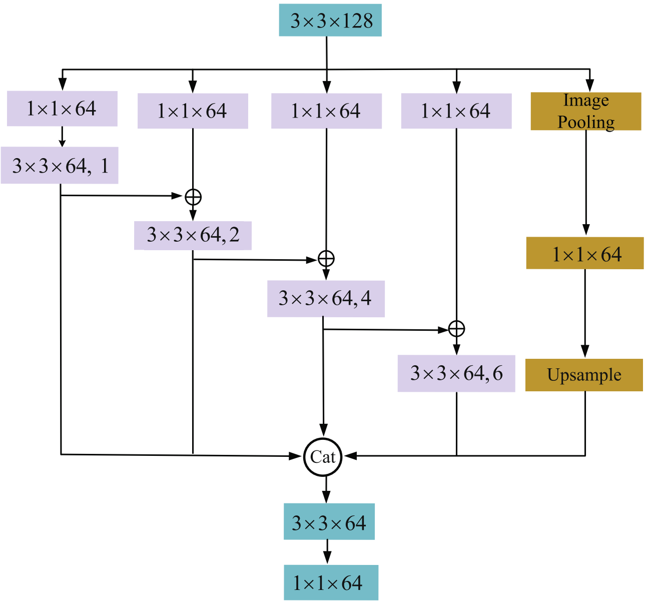


Fig. 2. The structure of multi-scale context-aware feature extraction block(MCFEB). "$3 \times 3 \times 128$" represents the "3×3" convolutional layer with the channel number 128. "$3 \times 3 \times 64, 1$" represents the "3×3" convolutional layer with the channel number 64 and the dilation rate 1. "Image pooling" represents a global average pooling operation. Other settings can be similarly learned.

corresponding up-sample operation is employed to recover the features to the original size. Furthermore, to avoid the excessive loss of spatial details produced by the repeated pooling operations, the inputs of the last two pooling sub-branches are composed with two items: the features down sampled from the last pooling sub-branch with the average pooling layer[1] (to avoid overstepping the boundary, we pad the feature image boundary with zero at each down-sample operation), and the down-sampled features from the original mixed input features with the bilinear interpolation function[2]. After each pooling operation (both down-sample and up-sample operation), a concomitant 3×3 convolutional operation is applied to reduce the aliasing effect produced by pooling operations. Furthermore, we embed a dilated 3×3 convolutional layer and a 3×3 convolutional layer in the middle of each sub-branch to further expand the receptive fields and learn more local context. The dilation of the four branches are set to $\{8, 4, 2, 1\}$, respectively. Finally, the output features of each sub-branch are up-sampled and further added to the outputs of next sub-branch. Different from the structure with four parallel pooling branches in [32], we cascade these four sub-branches together to transform the coarse-level features to the selective features progressively. Since the structure of this block is similar with two connected opposite hourglasses (as shown in Fig. 3(b)), we refer to the bottom-up (from high resolutions to low

[1] https://pytorch.org/docs/stable/nn.html#avgpool2d.

[2] https://pytorch.org/docs/stable/nn.functional.html?highlight=interpolate#interpolate

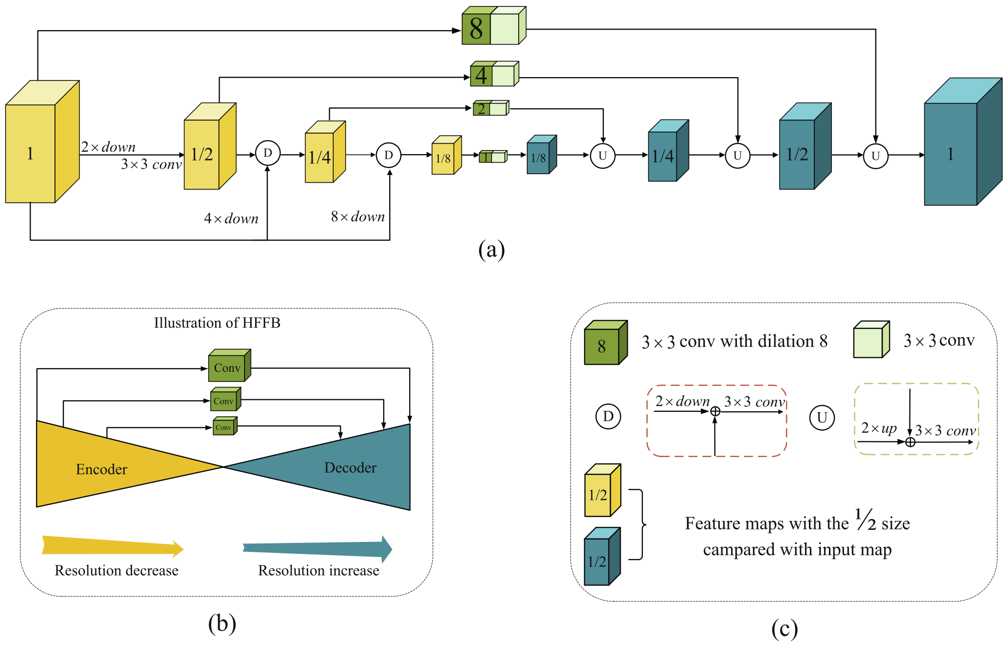


Fig. 3. The structure of hourglass feature fusing block(HFFB). It is the basic unit of CHFFM. (a) The detailed structure of HFFB. (b) The whole illustration of HFFB. (c) The legends of graph (a). For conciseness, we only explain the meanings of some items. The meanings of other items can be learned similarly.

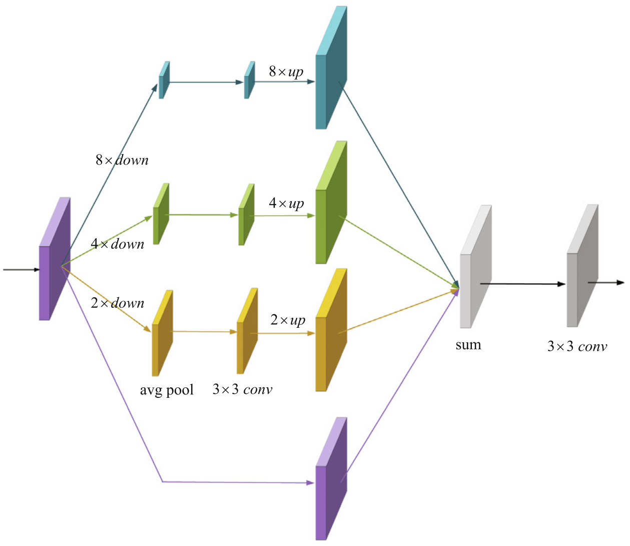


Fig. 4. The structure of basic pooling unit in PoolNet [32].

resolutions) and top-down (from low resolutions to high resolutions) encoder-decoder block as hourglass feature fusing block.

Generally, low-level features are more concrete and contain more spatial details but without discrimination, while high-level features are more abstract and more satisfied with the saliency detection task. Therefore, it is necessary to lay more emphases on low-level features when construct the network. In this paper, we stack a set of hourglass feature fusing blocks to focus on the low-level features and integrate the multi-level features progressively. As Fig. 1 shows, a serial of HFFBs are stacked to gradually integrate the side-output feature from CMCFEM. We denote the HFFBs used for same side-output feature from CMCFEM as a cascaded hourglass feature fusing layer (CHFFL) and the numbers of the four left–right CHFFLs are linearly decreasing from 4 to 1. According to the number of HFFBs, we denote the four left–right CHFFLs as CHFFL4, CHFFL3, CHFFL2, CHFFL1, respectively. It is worth to notice that the information of small objects cannot be well reconstructed if the pooling rate is larger than the size of small objects (e.g. if we adopt four average pooling operations, the objects under the size of 16 cannot be well reconstructed theoretically). Therefore, we only adopt two pooling sub-branches with down-sample rate $\{2,4\}$ in the HFFB of CHFFL1. Finally, we introduce the short dense connections to the structure and transform the features from high-level CHFFL to low-level CHFFL. The multi-level features transmitted to each HFFBs are first added together as a mixed input, and then the mixed input is delivered to the four sub-branches to dispel the intrinsic divergences. All of the connected HFFBs construct four horizontal and four vertical pathways. In macro view, these cascaded feature fusing blocks are seemed as several sets of cascaded hourglasses with different sizes. With these cascaded hourglasses, the side-output features from each MCFEBs are filtered from coarse-level to refined-level progressively and hierarchically.

3.3. Cascaded Feature Decoder

We design four corresponding decoder blocks to combine the refined features from the four CHFFLs. Similarly, the four decoder blocks are also cascaded in a top-down pathway. Besides the last decoder, each decoder takes the features from the same CHFFL and the output of the previous decoder as inputs. The structure is shown in Fig. 5. We first adapt two 3 × 3 convolutional layers to learn more local context from the features from the same CHFFL. In addition, we upsample the features from the previous decoder and couple a 3 × 3 convolutional layer to reduce the gridding effect. Subsequently, we concatenate them together and apply a combination of 3 × 3 and 1 × 1 convolutional layer to reduce the channel dimension of the mixed features to 64. As above mentioned, the high-level features are more consistent with saliency detection, therefore, the high-level feature can be severed as attentive information for low-level feature. Specifically, we first produce a high-level saliency map with a 1 × 1 convolutional layer. The produced saliency map which can be considered as the semantic attention map is then multiplied with the mixed features. Furthermore, we introduce the residual learning to this module and add the high-level-semantics guided mixed features into the original mixed features as the output of the feature decoder module. Finally, we couple a 3 × 3 and 1 × 1 convolutional layer to produce the final prediction (v_1 in this paper) from the output of last decoder module.

3.4. Loss Function

Binary cross entropy(BCE) is the most widely used loss function in salient object detection. However, BCE only calculates the pixel-level loss and ignores the structure of the salient object, which may obscure the edge of the salient region and degree the performance of model. Moreover, the pixels surrounding the edge are hard to distinguish and need to assign more weights. Recently, many saliency detection methods adopt the edge-aware [29,23] or IOU [27] loss function to capture the structure of salient object. In this paper, we employ a pixel position aware (PPA) loss function to learn the global structure of salient object and assign more weights to hard pixels, which has adopted by Wei et al. [50]. PPA loss is consisted of two weighted losses: a weighted binary cross entropy(wBCE) loss and a weighted IoU (wIoU) loss:

$$L_{ppa} = L_{wbce} + L_{wIou} \tag{1}$$

The weighted binary cross entropy(wBCE) loss function is formed as following:

$$L^s_{wbce} = -\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(1+\gamma\alpha_{ij})\sum_{l=0}^{1}\mathbf{1}(g^s_{ij}=l)\log\Pr(p^s_{ij}=l|\Psi)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\gamma\alpha_{ij}} \tag{2}$$

where $\mathbf{1}(.)$ is the indicator function, γ is the hyper-parameter set as 1. The notation $l \in \{0,1\}$ indicates two kinds of the labels. p^s_{ij} and g^s_{ij} are the prediction and ground truth of the pixel at location (i,j) in an image. Ψ represents all the parameters of the model and $\Pr(p^s_{ij}=l|\Psi)$ represents the predicted probability. α_{ij} is the weight to indicate the pixel importance, which is calculated according to the difference between the center pixel and its surroundings. It is formed as:

$$\alpha^s_{ij} = \left\|\frac{\sum_{m,n\in A_{ij}} gt^s_{mn}}{\sum_{m,n\in A_{ij}} 1} - gt^s_{ij}\right\| \tag{3}$$

where A^s_{ij} represents the area that surrounds the pixel(i,j). If α^s_{ij} is large, pixel at (i,j) is very different from its surroundings, which may represent an important pixel (e.g., edge) and deserves more attention. Similarly, the weighted Iou(wIoU) loss can be defined as:

$$L^s_{wiou} = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt^s_{ij} * p^s_{ij}) * (1+\gamma\alpha^s_{ij})}{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt^s_{ij} + p^s_{ij} - gt^s_{ij} * p^s_{ij}) * (1+\gamma\alpha^s_{ij})} \tag{4}$$

Furthermore, we apply multi-level deep supervision as an auxiliary loss to facilitate sufficient training vertically and horizontally. As Fig. 1 shows, we first obtain four horizontal saliency maps from the side-outputs of CHFFL4 with 1 × 1 convolutional layer. The four horizontal saliency maps denoted as $\{h_1, h_2, h_3, h_4\}$ represent the four horizontal pathways. Similarly, another four vertical saliency maps are produced from the outputs of the four cascaded feature decoder modules, and the four vertical saliency maps denoted as $\{v_1, v_2, v_3, v_4\}$ represent the four vertical pathways. Given the four horizontal and four vertical saliency maps, the total loss of our network can be defined as:

$$L = \sum_{h=1}^{4}\frac{L^h_{ppa}}{2^h} + \sum_{v=1}^{4}\frac{L^v_{ppa}}{2^v} \tag{5}$$

where L^h_{ppa} corresponds to the loss of the h-*th* horizontal pathways in our network and the pathways with heavier parameters are assigned with higher weights to ensure the abundant training (the loss of $\{h_1, h_2, h_3, h_4\}$ corresponds to the scale factor $\{1/2, 1/4, 1/8, 1/16\}$, respectively). L^v_{ppa} corresponds to the loss of the v-*th* vertical pathways in our network and the pathways of lower level features are assigned with higher weights to focus on the low-level features (the loss of $\{v_1, v_2, v_3, v_4\}$ corresponds to the scale factor $\{1/2, 1/4, 1/8, 1/16\}$, respectively).

4. Experiment

4.1. Experiment setup

Datasets. To evaluate the performance of our proposed framework, we conduct experiments on five commonly used benchmark datasets: ECSSD [51], DUTS [52], DUT-OMRON [53], HKU-IS [54], PASCAL-S [55]. ECSSD contains 1000 images which are semantically meaningful and structurally complex with pixel-wise ground truth. DUTS is a large-scale dataset containing two subsets: DUTS-TR and DUTS-TE. DUTS-TR contains 10,553 images designed for training and DUTS-TE has 5019 images for testing. DUT-OMRON has 5168 high quality images. Images of this dataset have one or more salient objects and relatively complex background. HKU-IS contains 4447 challenging images and most of them contain multiple disconnected salient objects. PASCAL-S includes 850 natural images selected from the PASCAL VOC 2010.

Evaluation metrics. To compare the performance of different methods, we adopt three widely-used metrics: precision and recall (PR) curve, F-measure, and mean absolute error (MAE). The precision and recall are computed by comparing the binarized saliency map against the ground truth mask. A pair of the precision and recall scores can be obtained with the threshold ranging from 0 to 255. Using the sequence of precision-recall pairs, the precision-recall (PR) curve can be plotted. F-measure is a harmonic mean of each pair of precision and recall, and defined as:

$$F_\beta = \frac{(1+\beta^2)\times \text{Precision}\times\text{Recall}}{\beta^2\times\text{Precision}+\text{Recall}} \tag{6}$$

where $\beta^2 = 0.3$ is used to emphasize the precision. For a fair comparison, we adopt maximum F-measure (max-F, larger is better) as the metric. We also use the MAE metric(smaller is better) to measure the average difference between the saliency prediction and the ground truth. It is computed as:

$$MAE = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\|P(i,i) - G(i,j)\| \tag{7}$$

where P is the predicted saliency map, and G is the corresponding ground truth.

Implementation details. We implement our network based on PyTorch repository[3] [56] and train it on the DUTS-TR dataset. In training process, the training images are randomly cropped, rotated, and horizontally flipped for data augmentation. We initialize the parameters of basic feature extractor with the well-pretrained backbone (VGGNet-19 [33] and ResNet-50 [34]), while other layers are randomly initialized. We use the stochastic gradient descent (SGD) algorithm to train the whole network with the momentum of 0.9, and weight decay 0.0005. During the training process, the initial learning rate is set as 0.005 for the network based on ResNet-50 (0.01 for VGGNet-19) and adjusted by the "poly" policy [57] with the power of 0.9. For the network based on ResNet-50, the training loss converges after 15 k iterations with the batch size of 32, while

[3] https://pytorch.org.

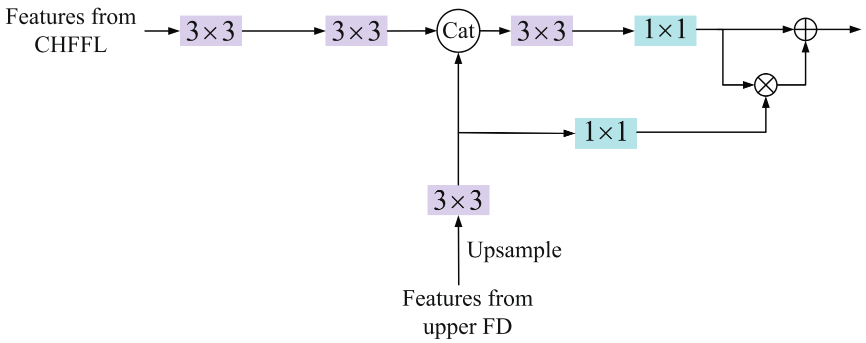


Fig. 5. The structure of feature decoder (FD). "Cat" is the concatenation operation.

25 k iterations with the batch size of 18 for the network based on VGGNet-19. We take the saliency map v_1 as the final prediction.

4.2. Ablation studies

In this section, we investigate the effect of different loss function first. Moreover, we compare the performance with or without scale factors in the loss function. Then we carry out a serial of experiments to learn the performance of the network with different sub-modules. To further verify the effect of our proposed sub-modules, we also test the effects of some other similar modules on model improvement. Specifically, we replace the MCFEB with the commonly used ASPP module [48] first, and then replace HFFB with the feature aggregation module (FAM) used in PoolNet [32]. We set the ResNet-50 version of FPN [49] as the baseline model. The quantitative and qualitative results of ablation studies are summarized in Table 1 and Fig. 6, respectively.

Evaluation of loss function: We first compare the performance of the model trained with BCE or PPA loss function. Although the proposed network trained with BCE can also achieve the well performance, PPA loss function can introduce the structure information into the network and achieve a better result. Actually, the introduction of structure information is instructive for saliency detection, which has been proved by many saliency detection methods [29,27,26,50,45,46,28]. As shown in Table 1, the PPA loss function mainly contributes to the MAE metric, which represents less errors in the difficult boundary areas.

Evaluation of scale factors: We compare the performance of the PPA loss function with or without scale factors. Scale factors in loss function are to make the training focus on the low-level features and the pathways with heavier parameters. As shown in Fig. 7, the scale factors can ease the optimization task with a faster convergence at early stages. Moreover, the scale factors pay more attentions to the shallow pathways with heavier parameters, which will train the network more fully. Seen from Table 2, in the two difficult datasets DUTS [52], DUT-OMRON [53], the network trained with scale factors can achieve the better results.

Comparisons of MCFEB and ASPP: ASPP module [48] is a commonly used module to capture the multi-receptive-fields context. ASPP module equips with a group of parallel dilated convolutional layers to capture more local context. However, the large dilation can cause serious gridding effect and lose beneficial spatial details. In this paper, to overcome these problems, we design a MCFEB and introduce the short connection into the structure to expand the receptive fields progressively. As shown in Fig. 6, compared with ASPP module, our proposed MCFEB is more deliberate to deal with the complex scene and can persist more saliency details. Moreover, as shown in Table 1, the quantitative results of MCFEB can comprehensively surpass the performances of ASPP module. Both results can prove the effect of the short connection between the dilated convolutional layers.

Comparisons of HFFB and FAM: Feature aggregation module (FAM) is a pooling module used in PoolNet [32] which comprises four parallel pooling sub-branches. It has been proved the huge potentiality of pooling operations for saliency detection with the top-level performance. We further explore the effect of pooling operations and design a novel HFFB. We carry out the quantitative comparisons in Table 1 first. As shown in Table 1, our proposed pooling module can achieve the comparative performance in terms of the listed quantitative metrics. Moreover, both pooling module can handle the complex scenes, but our proposed module is more accurate to suppress the noise and can reserve more vivid saliency details, which is shown in Fig. 6. Besides the parallel pooling structure in [32], our proposed cascaded pooling structure can also achieve a good result for saliency detection.

Evaluation of CFD: High-level features contains more high-level semantics, while low-level features contains more spatial details. $\{h_1, h_2, h_3, h_4\}$ and $\{v_1, v_2, v_3, v_4\}$ in Fig. 1 show the differences visually. Thus, we design the feature decoder and cascade them together to integrate these selective features. With CFD, the performance of our model can be further improved, which can be seen in Fig. 6 and Table 1.

4.3. Comparison with state-of-the-arts

We compare our proposed model with 14 previous state-of-the-art methods, including RFCN [58], DHS [35], Amulet [37], NLDF [29], DSS [18], BMPM [19], PAGRN [20], AFNet [22], PAGENet [59], MLMSNet [60], SRM [61], DGRL [30], PiCANet [62], CPD [31]. For fair comparison, all the saliency maps are provided by the authors or achieved by available codes.

Quantitative comparisons. We list all the quantitative results in Table 3 and Fig. 8. As shown in Table 3, the proposed network can outperform others on almost all the five datasets under different metrics. Although we did not realize the best performance on DUT-OMRON [53], our method demonstrates great competitiveness. Moreover, we display the PR curves in Fig. 8. These curves can provide a holistic evaluation of models. From these curves, we can observe that our model possesses a good capability to detect salient regions with a higher precision.

Visual comparisons. To evaluate the robustness and applicability of models, we choose some typical images from the public test dataset of saliency detection and exhibit all the images in Fig. 9. As shown in Fig. 9, the salient objects possess various characteristics, such as size, shape, numbers and so on. Specifically, the images in 1th to 4th rows represent the common salient objects but with relatively complex background. As we can see, our model can better suppress the background noise and make a more

Table 1
Quantitative results of the network based on the different modules. The best results are marked in red.

BCE	PPA	MCFEM	ASPP	CHFFM	FAM	CFD	ECSSD		HKU-IS		DUT-OMRON	
							max-F	MAE	max-F	MAE	max-F	MAE
√							0.8991	0.0647	0.8869	0.0562	0.7202	0.0963
√		√					0.9304	0.0453	0.9163	0.0417	0.7768	0.0735
√		√		√			0.9314	0.0413	0.9201	0.0369	0.7860	0.0658
√		√		√		√	0.9328	0.0410	0.9233	0.0367	0.7835	0.0668
	√						0.9071	0.0526	0.8964	0.0450	0.7386	0.0841
	√	√					0.9351	0.0389	0.9238	0.0347	0.7851	0.0690
	√		√				0.9308	0.0394	0.9175	0.0359	0.7824	0.0697
	√	√		√			0.9369	0.0373	0.9241	0.0339	0.7853	0.0670
	√	√			√		0.9298	0.0419	0.9190	0.0363	0.7841	0.0736
	√	√		√		√	0.9394	0.0364	0.9272	0.0338	0.7865	0.0680

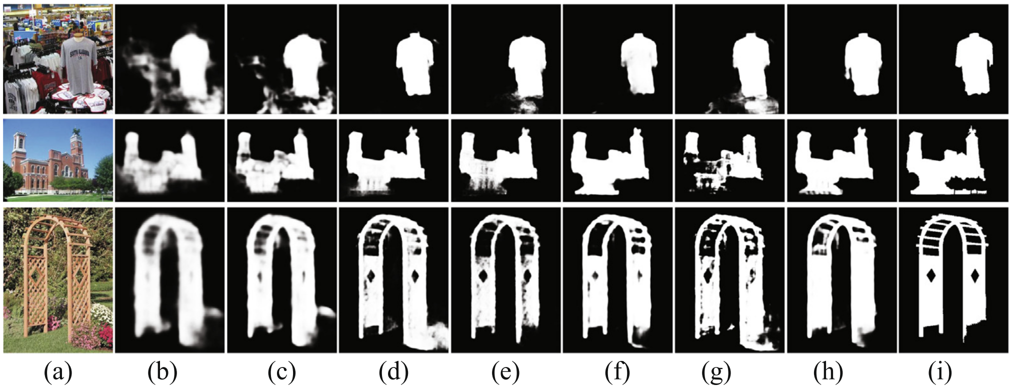


Fig. 6. Visual comparisons of our ablation studies. (a) Source Imges. (b) Results of baseline trained with BCE. (c) Results of baseline trained with PPA. (d) Results of baseline + CMCFEM. (e) Results of baseline + ASPP. (f) Results of baseline + CMCFEM + CHFFM. (g) Results of baseline + CMCFEM + FAM. (h) Results of baseline + CMCFEM + CHFFM + CFD. (i) Ground truth.

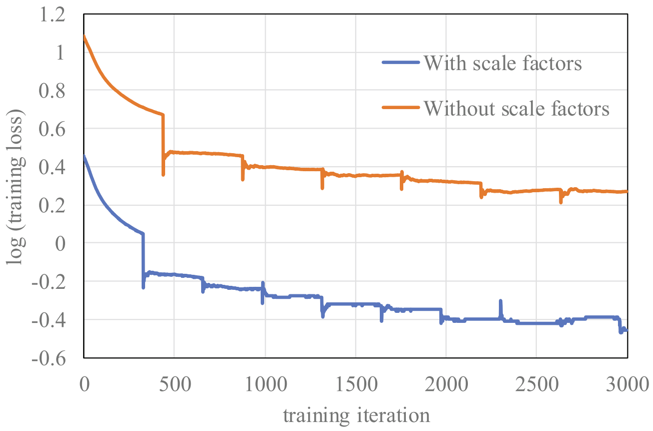


Fig. 7. The training loss of the PPA loss function with or without scale factors. To more clearly show the differences, we carry out a log operation for the training loss from 0 to 3000 training iteration.

accurate prediction. Moreover, our model can preserve more structural details, which can also be proved by the images in 5th row. Subsequently, the images in 6th to 7th row contain multiple salient objects. It is worth to notice that the salient objects in 6th row with various sizes are difficult to detect. However, our method can perfectly manage the scene and produce a more accurate prediction. The salient objects in the images of 8th to 9th row are low contrasted, which are hard to be distinguished. But our model can better pick out the salient objects. Finally, our method can also commendably predict the small salient objects, which can be seen in the images of 10th to 11th row. In a word, the proposed model can possess the good robustness and applicability to detect various salient objects.

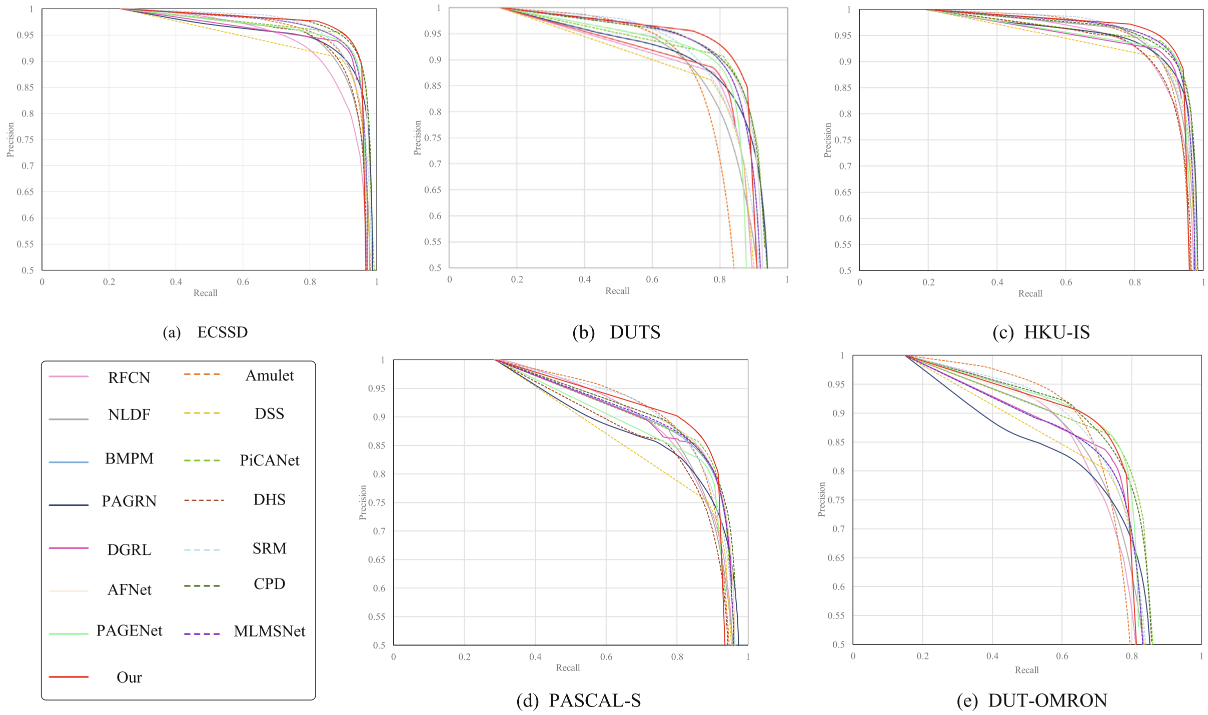
Memory comparisons. In general, a deeper neural network can make better performance, but it is also followed by a larger memory footprint and computation. So it will be difficult to apply the model to real-time detection and deploy the model in the mobile terminals. Thus it is necessary to make a balance between the effect and efficiency of the model, and the size of the saliency

Table 2
Quantitative results of the PPA loss function with/without scale factors.

With scale factors	Without scale factors	DUTS		DUT-OMRON	
		maxF	mae	maxF	mae
√		0.8719	0.0431	0.7865	0.0680
	√	0.8686	0.0435	0.7817	0.0695

Table 3
Quantitative comparisons of our model and the state-of-the-art models. "–" represents the model is trained on this dataset. The best three results are marked in red, green, blue.

Method	ECSSD		DUTS		HKU-IS		PASCAL-S		DUT-OMRON	
	maxF	mae	maxF	mae	maxF	mae	maxF	mae	maxF	mae
VGG backbone										
RFCN	0.8903	0.107	0.7845	0.0913	0.8926	0.0889	0.8546	0.1489	0.7423	0.1107
DHS	0.9065	0.059	0.8076	0.0675	0.8903	0.0525	0.8454	0.1159	-	-
Amulet	0.9147	0.0592	0.7778	0.0854	0.8953	0.0521	0.8617	0.1067	0.7427	0.0977
NLDF	0.905	0.0626	0.8126	0.0656	0.902	0.0477	0.8516	0.118	0.7532	0.0796
DSS	0.916	0.0526	0.8255	0.057	0.9099	0.0407	0.8585	0.1199	0.7715	0.0656
BMPM	0.9284	0.0446	0.8508	0.0493	0.9207	0.0387	0.8806	0.0943	0.774	0.0636
PAGRN	0.9268	0.061	0.8546	0.0561	0.9176	0.0475	0.8666	0.1204	0.7709	0.0709
AFNet	0.935	0.0418	0.8624	0.0461	0.9226	0.0358	0.8833	0.0899	0.7972	0.0573
PAGENet	0.9313	0.0424	0.8383	0.0523	0.9214	0.0313	0.873	0.0927	0.7915	0.0623
MLMSNet	0.9284	0.0445	0.8508	0.0493	0.9207	0.0387	0.8818	0.0922	0.774	0.0636
Our	0.936	0.0394	0.8673	0.0473	0.9291	0.0316	0.8784	0.0862	0.791	0.0705
ResNet backnone										
SRM	0.9173	0.0544	0.827	0.0592	0.9058	0.0459	0.8679	0.1029	0.7689	0.0693
DGRL	0.9223	0.0408	0.8289	0.05	0.9103	0.0356	0.8811	0.0888	0.7742	0.0618
PiCANet	0.9349	0.0464	0.8597	0.0509	0.9185	0.0433	0.883	0.0923	0.8028	0.0653
CPD	0.9393	0.0371	0.8655	0.0437	0.925	0.0342	0.885	0.0918	0.7966	0.056
Our	0.9394	0.0364	0.8719	0.0431	0.9272	0.0339	0.8858	0.084	0.7865	0.068



(a) ECSSD

(b) DUTS

(c) HKU-IS

(d) PASCAL-S

(e) DUT-OMRON

Fig. 8. PR curves of different methods on five datasets.

detection model can also be considered as the significant metrics. Fig. 10 displays the sizes and max F-measure of some models on ECSSD [51]. As shown in Fig. 10, some larger saliency detection models can not truly achieve the better performance. In contrast, our model contains less parameters but can achieve a better result. It is worth to notice that the performance of our model is slightly worse than the top-level network PoolNet [32] (0.939 vs 0.943), but our model is less than the half of the size of PoolNet [32] (123 MB vs 260 MB, both are based on ResNet-50 [34]). Furthermore, when dealing with an image with the size of 400 × 300 on an NVIDIA TITAN Xp GPU, our model based on ResNet-50 can run at the real-time speed of 28 FPS.

5. Conclusions

In this paper, we further explore the effect of pooling operations on saliency detection and design a novel feature fusing network based on the cascaded pooling operations. We first design a cascaded multi-scale context-aware feature extraction module

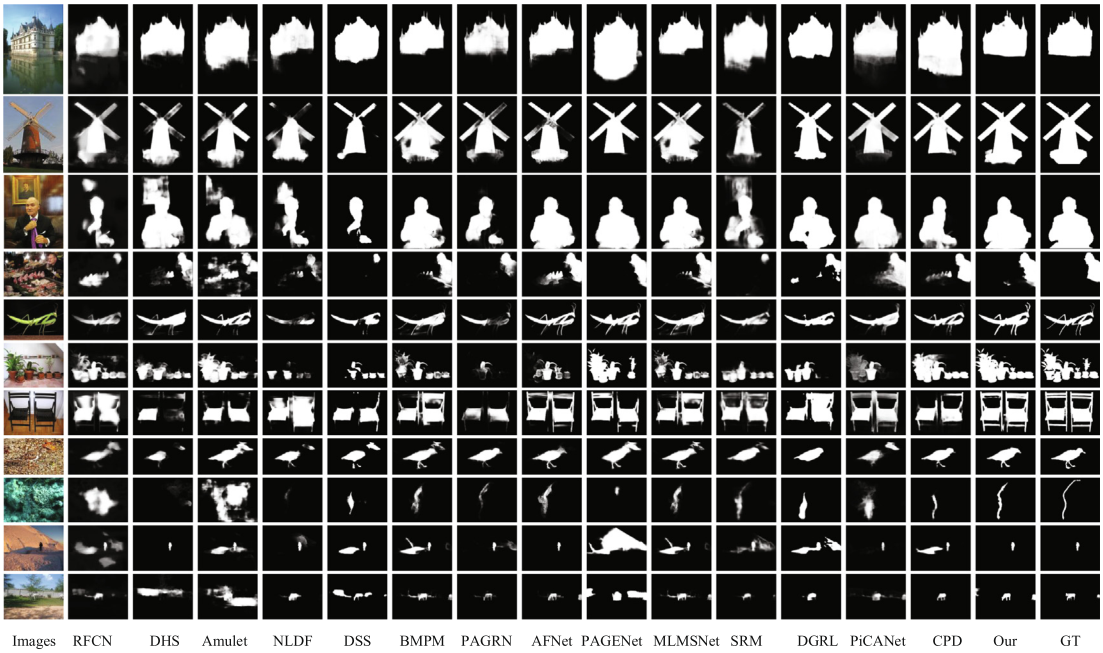


Fig. 9. The visual comparisons of our model and the state-of-the-art saliency methods.

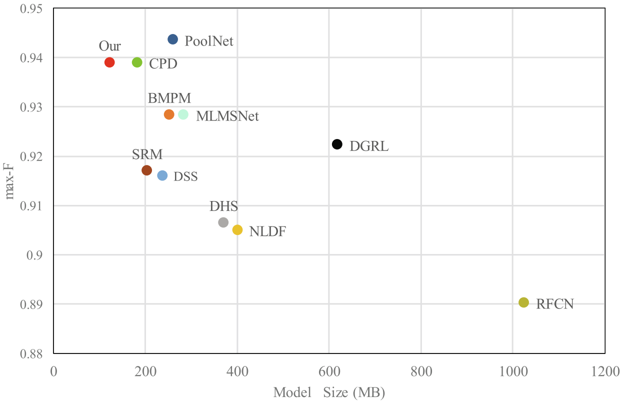


Fig. 10. Memory comparisons of some other saliency methods and our model, including RFCN [58], DHS [35], NLDF [29], DSS [18], BMPM [19], MLMSNet [60], SRM [61], DGRL [30], CPD [31], PoolNet [32].

(CMCFEM) to capture multi-receptive-fields features. Furthermore, a hourglass feature fusing block (HFFB) is proposed to convert the feature into multiple-scale feature spaces. With a serial of HFFBs, a cascaded hourglass feature fusing module (CHFFM) is constructed to further integrate multi-level features progressively. Finally, we adopt a cascaded feature decoder(CFD) to make the final prediction. Our model is not only lightweight but also efficient with the real-time speed. Extensive experiments demonstrate our network can achieve the state-of-the-art performance.

CRediT authorship contribution statement

Huiyuan Luo: Conceptualization, Methodology, Software, Writing - original draft. **Guangliang Han:** Investigation, Resources, Data curation. **Xiaotian Wu:** Software, Visualization, Formal analysis. **Peixun Liu:** Supervision, Funding acquisition, Project administration. **Hang Yang:** Software, Formal analysis, Validation. **Xin Zhang:** Writing - review & editing.

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.

References

[1] U. Rutishauser, D. Walther, C. Koch, P. Perona, Is bottom-up attention useful for object recognition?, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, Vol. 2, IEEE, 2004, pp. II–II. doi:10.1109/cvpr.2004.1315142.

[2] K.-Y. Chang, T.-L. Liu, S.-H. Lai, From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model, in, CVPR 2011, IEEE (2011) 2129–2136, https://doi.org/10.1109/cvpr.2011.5995415.

[3] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (1) (2018) 20–33, https://doi.org/10.1109/TPAMI.2017.2662005.

[4] M.-M. Cheng, F.-L. Zhang, N.J. Mitra, X. Huang, S.-M. Hu, Repfinder: finding approximately repeated scene elements for image editing, ACM Transactions on Graphics (TOG) 29 (4) (2010) 1–8, https://doi.org/10.1145/1833349.1778820.

[5] W. Wang, J. Shen, Y. Yu, K.L. Ma, Stereoscopic thumbnail creation via efficient stereo saliency detection, IEEE Transactions on Visualization and Computer Graphics (2017), https://doi.org/10.1109/TVCG.2016.2600594, 1–1.

[6] S. Hong, T. You, S. Kwak, B. Han, Online tracking by learning discriminative saliency map with convolutional neural network, in: in: International conference on machine learning, 2015, pp. 597–606, https://doi.org/10.1109/TIP.2015.2510583.

[7] M.-M. Cheng, N.J. Mitra, X. Huang, P.H. Torr, S.-M. Hu, Global contrast based salient region detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (3) (2014) 569–582, https://doi.org/10.1109/CVPR.2011.5995344.

[8] F. Perazzi, P. Krähenbühl, Y. Pritch, A. Hornung, Saliency filters: Contrast based filtering for salient region detection, in, IEEE conference on computer vision and pattern recognition, IEEE 2012 (2012) 733–740, https://doi.org/10.1109/CVPR.2012.6247743.

[9] L. Zhou, Z. Yang, Z. Zhou, D. Hu, Salient region detection using diffusion process on a two-layer sparse graph, IEEE Transactions on Image Processing 26 (12) (2017) 5882–5894, https://doi.org/10.1109/tip.2017.2738839.

[10] H. Luo, G. Han, P. Liu, Y. Wu, Salient region detection using diffusion process with nonlocal connections, Applied Sciences 8 (12) (2018) 2526, https://doi.org/10.3390/app8122526.

[11] Y. Qin, H. Lu, Y. Xu, H. Wang, Saliency detection via cellular automata, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 110–119, https://doi.org/10.1109/CVPR.2015.7298606.

[12] W. Zhu, S. Liang, Y. Wei, J. Sun, Saliency optimization from robust background detection, in, in: Proceedings of the IEEE conference on computer vision and

pattern recognition, 2014, pp. 2814–2821, https://doi.org/10.1109/CVPR.2014.360.
[13] W. Wang, J. Shen, L. Shao, F. Porikli, Correspondence driven saliency transfer, IEEE Transactions on Image Processing 25 (11) (2016) 5025–5034, https://doi.org/10.1109/TIP.2016.2601784.
[14] S. Zhao, Z. Lei, M. Sun, A. Ma, J. Shen, Diffusion-based saliency detection with optimal seed selection scheme, Neurocomputing 239 (MAY24) (2017) 94–101, https://doi.org/10.1016/j.neucom.2017.02.007.
[15] Guo Fang, Wang Wenguan, Shen Jianbing, Shao Ling, Yang Jian, Tao Dacheng, Video saliency detection using object proposals, IEEE Transactions on Cybernetics, doi: 10.1109/TCYB.2017.2761361.
[16] Y. Rivenson, Y. Wu, A. Ozcan, Deep learning in holography and coherent imaging, Light: Science & Applications 8 (1) (2019) 1–8, https://doi.org/10.1038/s41377-019-0196-0.
[17] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354–377, https://doi.org/10.1016/j.patcog.2017.10.013.
[18] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P.H. Torr, Deeply supervised salient object detection with short connections, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3203–3212, https://doi.org/10.1109/cvpr.2017.563.
[19] L. Zhang, J. Dai, H. Lu, Y. He, G. Wang, A bi-directional message passing model for salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1741–1750, https://doi.org/10.1109/cvpr.2018.00187.
[20] X. Zhang, T. Wang, J. Qi, H. Lu, G. Wang, Progressive attention guided recurrent network for salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 714–722, https://doi.org/10.1109/cvpr.2018.00081.
[21] S. Chen, X. Tan, B. Wang, X. Hu, Reverse attention for salient object detection, in, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 234–250, https://doi.org/10.1007/978-3-030-01240-3_15.
[22] M. Feng, H. Lu, E. Ding, Attentive feedback network for boundary-aware salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1623–1632, https://doi.org/10.1109/cvpr.2019.00172.
[23] T. Zhao, X. Wu, Pyramid feature attention network for saliency detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3085–3094, https://doi.org/10.1109/CVPR.2019.00320.
[24] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, P.-A. Heng, in: R3net: Recurrent residual refinement network for saliency detection, in Proceedings of the 27th International Joint Conference on Artificial Intelligence, AAAI Press, 2018, pp. 684–690, https://doi.org/10.24963/ijcai.2018/95.
[25] S. Chen, B. Wang, X. Tan, X. Hu, Embedding attention and residual network for accurate salient object detection, IEEE Transactions on Cybernetics (2018) 1–13, https://doi.org/10.1109/TCYB.2018.2879859.
[26] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, M.-M. Cheng, Egnet: Edge guidance network for salient object detection, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8779–8788, https://doi.org/10.1109/iccv.2019.00887.
[27] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, Basnet: Boundary-aware salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7479–7489, https://doi.org/10.1109/cvpr.2019.00766.
[28] J. Su, J. Li, Y. Zhang, C. Xia, Y. Tian, Selectivity or invariance: Boundary-aware salient object detection, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3799–3808, https://doi.org/10.1109/ICCV.2019.00390.
[29] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, P.-M. Jodoin, Non-local deep features for salient object detection, in, in: Proceedings of the IEEE Conference on computer vision and pattern recognition, 2017, pp. 6609–6617, https://doi.org/10.1109/cvpr.2017.698.
[30] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, A. Borji, Detect globally, refine locally: A novel approach to saliency detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3127–3135, https://doi.org/10.1109/cvpr.2018.00330.
[31] Z. Wu, L. Su, Q. Huang, Cascaded partial decoder for fast and accurate salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3907–3916, https://doi.org/10.1109/cvpr.2019.00403.
[32] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, A simple pooling-based design for real-time salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3917–3926, https://doi.org/10.1109/CVPR.2019.00404.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[35] N. Liu, J. Han, Dhsnet: Deep hierarchical saliency network for salient object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 678–686, https://doi.org/10.1109/cvpr.2016.80.
[36] Y. Zhuge, Y. Zeng, H. Lu, Deep embedding features for salient object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 9340–9347. doi:10.1609/aaai.v33i01.33019340.
[37] P. Zhang, D. Wang, H. Lu, H. Wang, X. Ruan, Amulet: Aggregating multi-level convolutional features for salient object detection, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 202–211, https://doi.org/10.1109/iccv.2017.31.
[38] S. Mohammadi, M. Noori, A. Bahri, S.G. Majelan, M. Havaei, Cagnet: Content-aware guidance for salient object detection, Pattern Recognition (2020), https://doi.org/10.1016/j.patcog.2020.107303 107303.
[39] W. Wang, J. Shen, X. Dong, A. Borji, R. Yang, Inferring salient objects from human fixations, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (8) (2020) 1913–1927, https://doi.org/10.1109/TPAMI.2019.2905607.
[40] W. Wang, J. Shen, X. Dong, A. Borji, Salient object detection driven by fixation prediction, in, IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018 (2018) 1711–1720, https://doi.org/10.1109/CVPR.2018.00184.
[41] W. Wang, J. Shen, Deep visual attention prediction, IEEE Transactions on Image Processing 27 (5) (2018) 2368–2378, https://doi.org/10.1109/TIP.2017.2787612.
[42] W. Wang, J. Shen, J. Xie, M.M. Cheng, A. Borji, Revisiting video saliency prediction in the deep learning era, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2019) 1–1. doi:10.1109/TPAMI.2019.2924417.
[43] Q. Lai, W. Wang, H. Sun, J. Shen, Video saliency prediction using spatiotemporal residual attentive networks, IEEE Transactions on Image Processing PP (99) (2019) 1–1. doi:10.1109/TIP.2019.2936112.
[44] S. Zhu, L. Zhu, Ognet: Salient object detection with output-guided attention module, arXiv preprint arXiv:1907.07449.
[45] Y. Zhuge, G. Yang, P. Zhang, H. Lu, Boundary-guided feature aggregation network for salient object detection, IEEE Signal Processing Letters 25 (12) (2018) 1800–1804, https://doi.org/10.1109/LSP.2018.2875586.
[46] S. Zhou, J. Wang, F. Wang, D. Huang, Se2net: Siamese edge-enhancement network for salient object detection, arXiv preprint arXiv:1904.00048.
[47] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene cnns, arXiv preprint arXiv:1412.6856.
[48] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587.
[49] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125, https://doi.org/10.1109/CVPR.2017.106.
[50] Q.H. Jun Wei, Shuhui Wang, F3net: Fusion, feedback and focus for salient object detection, in, in: AAAI Conference on Artificial Intelligence (AAAI), 2020.
[51] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 1155–1162, https://doi.org/10.1109/CVPR.2013.153.
[52] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, X. Ruan, Learning to detect salient objects with image-level supervision, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 136–145, https://doi.org/10.1109/CVPR.2017.404.
[53] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3166–3173, https://doi.org/10.1109/CVPR.2013.407.
[54] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5455–5463, https://doi.org/10.1109/CVPR.2015.7299184.
[55] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 280–287, https://doi.org/10.1109/CVPR.2014.43.
[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in, Advances in Neural Information Processing Systems (2019) 8024–8035.
[57] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: Looking wider to see better, arXiv preprint arXiv:1506.04579.
[58] L. Wang, L. Wang, H. Lu, P. Zhang, X. Ruan, Saliency detection with recurrent fully convolutional networks, in: European conference on computer vision, Springer, 2016, pp. 825–841, https://doi.org/10.1007/978-3-319-46493-0_50.
[59] W. Wang, S. Zhao, J. Shen, S.C. Hoi, A. Borji, Salient object detection with pyramid attention and salient edges, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1448–1457, https://doi.org/10.1109/cvpr.2019.00154.
[60] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, E. Ding, A mutual learning method for salient object detection with intertwined multi-supervision, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8150–8159, https://doi.org/10.1109/CVPR.2019.00834.
[61] T. Wang, A. Borji, L. Zhang, P. Zhang, H. Lu, A stagewise refinement model for detecting salient objects in images, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4019–4028, https://doi.org/10.1109/ICCV.2017.433.
[62] N. Liu, J. Han, M.-H. Yang, Picanet: Learning pixel-wise contextual attention for saliency detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3089–3098, https://doi.org/10.1109/CVPR.2018.00326.

Huiyuan Luo received the B.S degree from Harbin Institute of Technology, Weihai in 2016. He is currently studying toward his Ph.D. degree at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His current research interests are mainly focused on saliency detection and deep learning.

Guangliang Han received the M.S. and Ph.D. degrees at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, in 2000 and 2003, respectively. He is currently the research fellow in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His current research interests are mainly focused on computer vision, image processing, and object tracking.

Xiaotian Wu received the B.Eng. degree from Jilin University in 2009, and the M.S. degree from Xiamen University in 2012. He is currently the assistant research fellow in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His current research interests are mainly focused on embedded system design, image processing, and object tracking.

Peixun Liu received his Ph.D. degree from Jilin University in 2015. He is currently an associate research fellow in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His research interests include image processing, object detection, and robot automation.

Hang Yang received his B.S. and Ph.D. degrees from Jilin University in 2007 and 2012, respectively. He is currently an associate research fellow in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His research interests include image restoration, object tracking.

Xin Zhang received her B.Eng. degree from Northeastern University at Qinhuangdao in 2016. She is currently studying toward her Ph.D. degree at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. Her current research interests are mainly focused on deep learning, object classification of remote sensing.