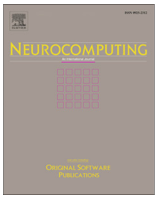
Neurocomputing 449 (2021) 24–37

Contents lists available at ScienceDirect

Neurocomputing

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



LF^{3}Net: Leader-follower feature fusing network for fast 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 Sciences, Changchun 130033, China*
[b] *University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 18 November 2020
Revised 12 March 2021
Accepted 22 March 2021
Available online 29 March 2021
Communicated by Zidong Wang

Keywords:
Saliency detection
Stackbelberg game
Leader-follower model
Feature fusing

ABSTRACT

Recently, convolutional neural networks (CNNs) have been widely used for saliency detection. Most of existing saliency detection methods produce saliency maps from the complementary multi-level convolutional features. However, it is still a challenging task to accurately integrate multi-level features for saliency detection. In this paper, we explore the intrinsic relationships between multi-level features and introduce the Stackelberg game theory as a new strategy to fuse multi-level features for saliency detection. Based on the theory, we propose a leader-follower feature fusing network (LF^{3}Net) to obtain saliency maps. We first apply a multi-scale context-aware leader-follower attention module (MCLAM) to select multi-scale spatial and semantic information. Then, we propose a leader-follower feature fusing module (LF^{3}M) to integrate the multi-level features. Extensive experiments on five datasets show that the proposed method outperforms the state-of-the-art approaches under different evaluation metrics. In addition, our network can run fast at the real-time speed of 75 FPS.

1. Introduction

The task of saliency detection is to locate the most visually attractive objects in an image, which is inspired by the mechanism of the human attention. Recently, as a helpful pre-processing method, it attracts lots of interests and has been effectively applied in many vision tasks, such as image segmentation [1], content-aware image editing [2], and object tracking [3].

Recently, convolutional neural networks (CNNs) have been widely used in various computer vision tasks because of its powerful capability in visual feature representation [4]. Similarly, benefit from the powerful multi-level features, most CNNs-based saliency detection methods have achieved a remarkable progress compared with traditional methods [5–8]. Furthermore, it has been proved by [9] that saliency detection is a low-level vision task which is more dependent on high-level semantic information. 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. Generally, the global semantics are mainly contained in high-level features. Therefore, compared with low-level features, high-level features have a natural superiority in the saliency detection task. It is crucial to make full use of the superiority of high-level features in the saliency detection task.

Recently, to take advantage of the prior, more and more researchers transmit the high-level features to low-level features and further integrate them with different strategies. As shown in Fig. 1, in terms of the coupled mode of feature, we roughly synopsize these methods into three sets: progressive mode, skip-layer mode, and aggregated direct-connection mode. Most of existing saliency detection methods adopt the progressive mode [10–14]. They transmit the features between neighboring convolutional layers and integrate them progressively. However, this mode performs the integrations indirectly among multi-level features, which may be deficient because of the long-term dependency problem. Some other existing methods introduce the skip-layer strategy to transmit the high-level features to low-level features directly, such as DSS [15], CAGNet [16] and PFPN [17]. Besides this, some other methods utilize the aggregated direct-connection mode, which assembles the high-level and low-level features as two aggregated sets first, such as R^3 Net [18] and PFA [19]. The skip-layer and aggregated direct-connection structures, transmitted the high-level semantics to the low-level features directly, may be more sufficient to fuse the multi-level features. However, the high-level and low-level features are significantly different with each other. If we directly combine them without any discrim-

☆ 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).

https://doi.org/10.1016/j.neucom.2021.03.080
0925-2312/© 2021 Elsevier B.V. All rights reserved.

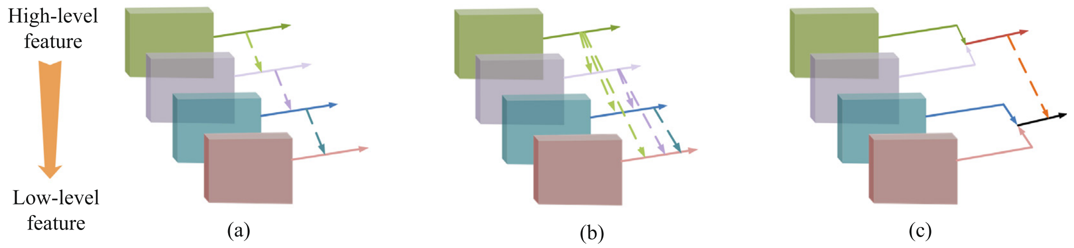


Fig. 1. Illustration of different feature coupled modes. (a) Progressive mode. (b) Skip-layer mode. (c) Aggregated direct-connection mode.

ination, it will inevitably bring the noises to result and decrease the performance. Thus it is crucial to design the features fusing strategy deliberately.

In this paper, we further explore the intrinsic relationship between multi-level features and design a novel direct-connection structure for saliency detection. Furthermore, we introduce the Stackelberg game theory [20] into saliency detection as a new feature fusing strategy. Stackelberg game theory describes a dynamic competition process between two participants, which can be simply considered as a leader-follower model. We will elaborate it in the following section. Inspired by Stackelberg game theory, we propose a leader-follower feature fusing network (LF^{3}Net) to fuse different level features for saliency detection. As shown in Fig. 2, the high-level and low-level features are considered as two participants to compete with each other. At first, we design a multi-scale context-aware leader-follower attention module (MCLAM) to select multi-scale saliency cues. MCLAM equips with three sub-modules: the multi-scale context-aware feature extraction unit (MCFEU) to capture contextual information, the leader and follower attention units to purify the coarse leader and follower features, respectively. Then, we propose a leader-follower feature fusing module (LF^{3}M) to fuse the selective features from the attention modules. LF^{3}M equips with two features fusing units, one is the leader feature fusing unit (LF^{2}U), and the other is the follower feature fusing unit (F^{3}U). In the LF^{3}M, the leader and follower features are arranged in a principal and subordinate way. We take the leader saliency map from the last feature fusing module as the final output. The whole network is trained in an end-to-end manner. Our contributions are summarized as follows:

1. We further explore the intrinsic relationships between multi-level features and introduce the Stackelberg game theory as a new feature fusing strategy for saliency detection.
2. Based on the Stackelberg game theory, we design a leader-follower feature fusing network equipped with two modules to select and fuse different level features for saliency detection.
3. Extensive experiments on five public datasets show that the proposed method outperforms the state-of-the-art approaches under different evaluation metrics. Furthermore, when processing the five test datasets on an NVIDIA TITAN Xp GPU, the model based on ResNet-50 can run with the real-time speed of 75 FPS.

2. Related works

Most of existing salient object detection networks are devoted to aggregate the multi-level features from CNNs to improve the performance [8,7]. Many effective feature fusing strategies have been designed, such as short connection [15], gate mechanism [11], attention model [21,22,19,23,24], residual learning [18], edge-aware model [12,25], and so on [13,26]. For example, Liu et al. [10] design an encoder-decoder structure to convert a coarse global prediction to refined saliency map hierarchically and progressively. Zhang et al. [11] construct a bi-directional gate structure between adjacent convolutional layers to bilaterally filter multi-level cluttered features. Wu et al. [9] abandon low-level features and only decode saliency cues from high-level features via a cascaded partial decoder framework. Wei et al. [27] capture the

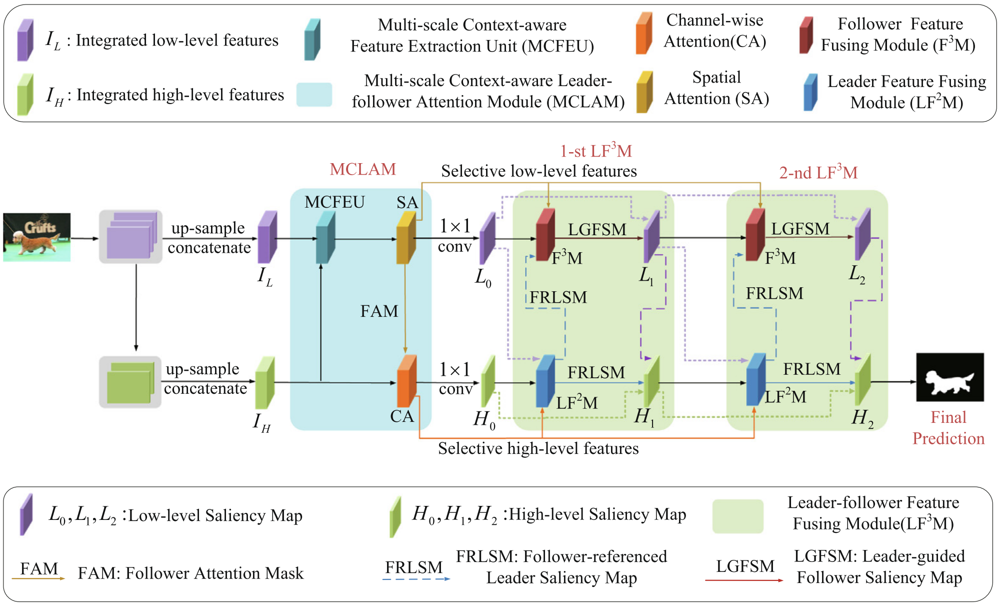

I_L : Integrated low-level features
Multi-scale Context-aware Feature Extraction Unit (MCFEU)
Channel-wise Attention(CA)
Follower Feature Fusing Module (F^{3}M)
I_H : Integrated high-level features
Multi-scale Context-aware Leader-follower Attention Module (MCLAM)
Spatial Attention (SA)
Leader Feature Fusing Module (LF^{2}M)
MCLAM
1-st LF^{3}M
Selective low-level features
2-nd LF^{3}M
MCFEU
SA
up-sample
concatenate
I_L
1×1 conv
L_0
LGFSM
F^{3}M
L_1
FRLSM
LGFSM
F^{3}M
L_2
FRLSM
FAM
up-sample
concatenate
I_H
CA
1×1 conv
H_0
FRLSM
LF^{2}M
H_1
FRLSM
LF^{2}M
H_2
Final Prediction
Selective high-level features
L_0, L_1, L_2 :Low-level Saliency Map
H_0, H_1, H_2 :High-level Saliency Map
Leader-follower Feature Fusing Module(LF^{3}M)
FAM
FAM: Follower Attention Mask
FRLSM
FRLSM: Follower-referenced Leader Saliency Map
LGFSM
LGFSM: Leader-guided Follower Saliency Map

Fig. 2. The overall framework of our proposed network.

saliency cues in a top-down pathway and then feed them back progressively.

Attention mechanism is widely used in saliency detection task for its great ability to select features. For example, Zhang et al. [21] propose a novel attention guided network to selectively integrate multi-level features in a progressive manner. Chen et al. [22] utilize reverse attention to guide side-output residual learning in a top-down manner. Zhang et al. [23] propose an augmenting feed-forward neural networks with the pyramid pooling and channel attention module to aggregate global contexts. Attention mechanism has also been applied in various related tasks, such as saliency detection in video [24], optical images [28], and RGB-D images [29]. For example, Li et al. [24] introduce a series of novel motion guided attention modules to guide the saliency detection in videos. Chen et al. [29] design a gated multi-modality attention module to capture cross-modal long-range dependencies for RGB-D saliency detection.

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, Su et al. [25] rethink saliency detection in terms of the selectivity or invariance of different features to construct the network. Zhao et al. [12] extract the regions and edges of salient objects simultaneously, and further guide the saliency detection with edge information. Liu et al. [30] introduce the pooling operation to expand the receptive fields of features and join edge information to train the whole network.

3. The proposed network

In this paper, we propose a novel saliency detection method equipped with two modules. At first, we design a multi-scale context-aware leader-follower attention module (MCLAM) to select multi-level contextual features. Then, a leader-follower feature fusing module (LF^{3}M) is introduced to integrate the multi-level features.

3.1. Stackelberg game for saliency detection

Algorithm 1: Forward process of Stackelberg game

```
Input: $I_A$,$I_B$: input of A and B;
Output: $A_{Nash}$, $B_{Nash}$: output of A and B under Nash
    equilibrium
$f_1$,$f_2$,$f_3$ are the functions of step 1,2,3 respectively.
repeat
    step1 A decides the output based on $I_A$ and $I_B$:
      $A_{init} = f_1(I_A, I_B)$;
    step2 B makes its output according to $A_{init}$ and $I_B$:
      $B_{out} = f_2(A_{init}, I_B)$;
    step3 A produces the finial output in terms of $A_{init}$ and
      $B_{out}$: $A_{out} = f_3(A_{init}, B_{out})$;
untilA and B achieve Nash equilibrium
return$A_{Nash}$, $B_{Nash}$
```

In the Stackelberg competition [20], there are two participants to compete with each other for its respective maximizing profits. We generalize the competition process in Fig. 3. As shown in Fig. 3 (a), one participant (denoted as participant “A” for conveniently elaborating) first gives a preliminary estimated output in terms of the conditions of itself and the other, which is represented as step1 in Fig. 3 (a). And then, the other (denoted as participant “B”) can get the output from participant “A” as a reference, shown as step2 in Fig. 3(a), participant “B” will adjust its response accounting for the given output of participant “A”. The response function can be known by participant “A”. Subsequently, to make a Nash equilibrium in Stackelberg competition, participant “A” will perform step3 to decide its output under the given results of participant “B”. Algorithm. 1 summarizes this forward inference process. In this competition, participant “B” is guided or affected by participant “A” and first gives its final decision, while participant “A” is aware of the result of participant “B”, and later outputs its result. Contrast to participant “B”, the participant “A” has a first-moving but last-arriving advantage, which makes it be aware of more accurate information. Just because the superiority of knowing full information, participant “A” can gain the upper hand in this competition. Therefore, we consider the participant “A” as a leader in the competition, while the participant “B” as a follower.

The process of solving the Stackelberg equilibrium is the reverse recursion process [20], which is much similar to the back propagation. During the reverse solving process, given the result of participant “A” (initial output of “A” in Fig. 3 (b)), the participant “B” will first decide its preliminary prediction. Then the participant “A” is aware of the initial prediction of “B” and make its last decision. Subsequently, participant “B” will make its final result in terms of the result of participant “A”. In each step, the participant will adjust its parameters to make its max profits until they achieve the Nash equilibrium. The process to make a Nash equilibrium is consistent with the process to make a parameter convergence of the proposed network. In the reverse solving process, the participant “B” will take as the leader, while the participant “A” takes as the leader in the forward process. This intrinsical leadership exchanging mechanism will help two participants learn from each other sufficiently.

Saliency detection is a low-level vision task which is more dependent on high-level semantic information [9]. Compared with low-level features, high-level features have a natural superiority in the saliency detection task. How to make full use of the superiority is crucial in the saliency detection task. As above mentioned, Stackelberg game is a leader-follower dynamic competition process, which is perfectly consistent with the intrinsic relationships between multi-level features in the saliency detection task. Therefore, in this paper, we introduce this competition theory to fuse the different level features for saliency detection. We consider high-level and low-level features as two participants to take part in this Stackelberg competition, the max profits of both in this competition are to try their best to reduce the residuals between their predictions and the ground truths. During the process to make a Nash equilibrium, high-level and low-level features can learn the beneficial saliency cues from each other and dispel the differences instructively. Specifically, when high-level features are the leader, a direct connection will first transmit low-level features to high-level flow. This connection can not only make high-level features capture the spatial details directly, but also furthest retain the accurate spatial details. Then the high-level features will first make an estimated prediction in terms of low-level features and themselves. The high-level estimated prediction, contained both semantics and spatial information, can supply low-level features and avoid the ambiguities of structural details. Subsequently, the follower low-level features will produce more accurate saliency prediction with the guidance of above high-level estimated prediction. Furthermore, the more accurate spatial details reserved in low-level features will be fed back to the high-level features, which makes the high-level features be selectively and precisely aware of the spatial details. Similarly, the process when low-level features take as the leader can be learned. This kind of bi-directional feedback mechanism between multi-level features is perfectly consistent with the Stackelberg game theory. Different

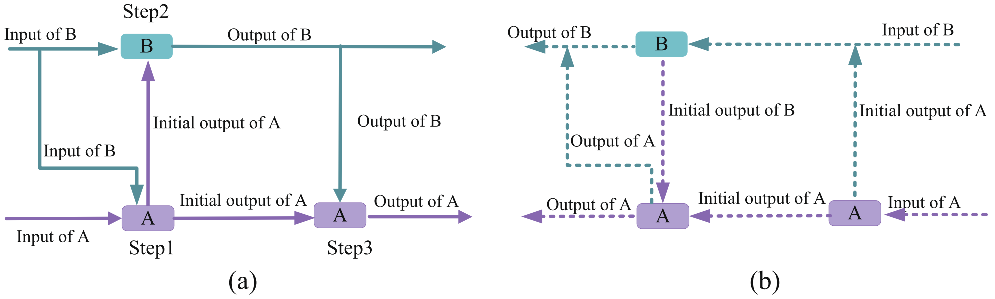


Fig. 3. The diagram of the Stackelberg game. (a) Forward process. (b) Reverse solving process.

from the common top-down or down-top feature fusing mechanism [11,31], which considers the contributions of low-level and high-level features equally, Stackelberg competition can perfectly match the intrinsic precedence relation between multi-level features. Based on this motivation, we propose the leader-follower feature fusing network (LF^{3}Net).

To reduce the redundant saliency information, we first divide the multi-level side-output features into two feature sets: high-level and low-level feature sets. Given the VGGNet version of FPN [32] as an example, we use the feature maps outputted by conv1, conv2, conv3, conv4, conv5 as the five side-output features. We first up-sample the feature maps from the first three layers to the size of conv1, and then combine them with a cross-channel concatenation and two convolutional layers with the kernel sizes of 3×3 and 1×1, respectively. We define the integrated low-level features I_L as:

$$I_L = \varphi(f_1, f_2, f_3) \tag{1}$$

where f_1, f_2, f_3 represent the side outputs of the first three layers, φrepresents the above operations to integrate these features, the channel dimension of I_L is set as 256. Similarly, the integrated high-level features can be expressed as:

$$I_H = \phi(f_4, f_5) \tag{2}$$

where f_4, f_5 represent the side outputs of the last two layers, ϕ represents the above features integrated operations similar to I_L. The size of I_H is same with I_L.

As mentioned in Section 1, the high-level features possess a natural superiority in the saliency detection task. Based on the empirical prior, we naturally take the high-level features I_H as the leader in our leader-follower structure. Furthermore, we will provide more experimental evidences to prove this prior in the section of experiment.

3.2. Multi-scale context-aware leader-follower attention module

It is common to know that the high-level and low-level features extracted from FCN are complementary to each other. However, due to the large randomness in scale, shape, and position of the salient objects, it may bring some redundant and harmful information to saliency detection if we directly integrate the different level features indiscriminately. In addition, the context is also quite important for saliency detection.

In terms of these, we design a multi-scale context-aware leader-follower attention module (MCLAM) equipped with three units, the multi-scale context-aware feature extraction unit (MCFEU), the leader attention unit (LAU), and the follower attention unit (FAU). As above mentioned, we first integrate the side output features from different layers as integrated low-level and high-level features, respectively. The integrated high-level features I_H are taken as the leader features L_f, while low-level features I_L are considered as the follower features F_f. At first, we concatenate the leader features and follower features as the inputs of MCFEU. As shown in Fig. 4 (b), 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. Furthermore, to learn the context progressively and reduce the griding effect produced by dilated convolutional operations, we introduce the short connections into the structure and transmit the output of one sub-

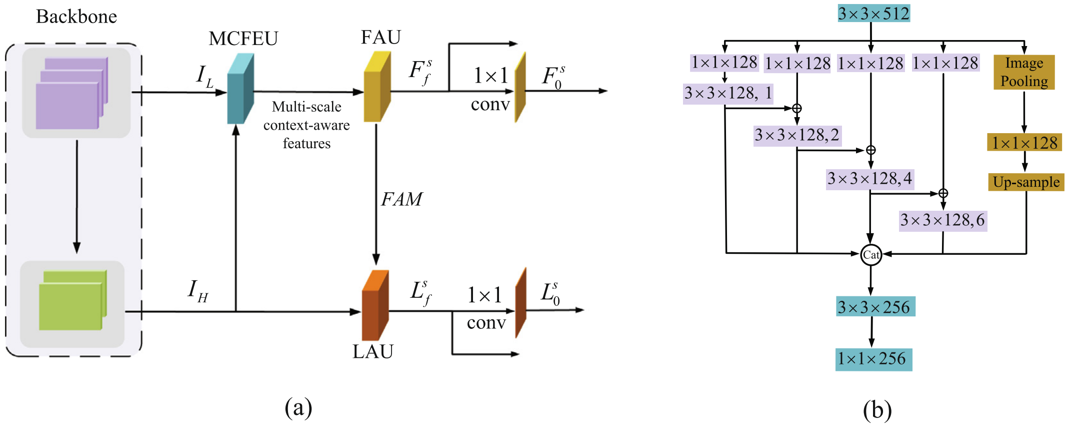


Fig. 4. Illustration of our attention module. (a) The overall structure of the multi-scale context-aware leader-follower attention module (MCLAM). We consider the change from low-level to high-level features in backbone as the first transformation in the competition. (b) The structure of multi-scale context-aware feature extraction unit (MCFEU). “$3 \times 3 \times 128, 1$“ represents the 3×3 convolutional kernel with the 128 channels and dilation 1.

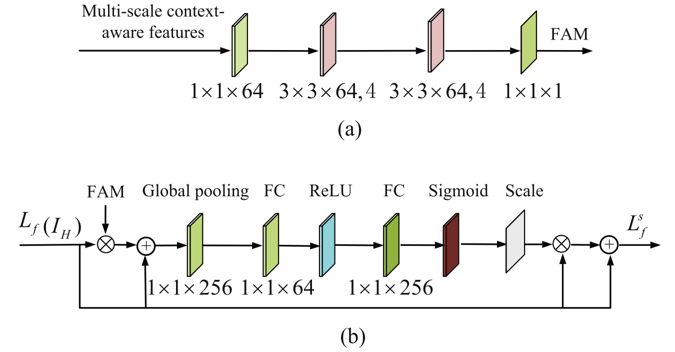


Fig. 5. Details of attention modules. (a) Spatial attention module. "$3 \times 3 \times 64, 4$" represents the 3×3 convolutional kernel with 64 channels and dilation 4. (b) Channel-wise attention module. FC is the fully connected layers.

branch to next sub-branch. We add the delivered input to the original input together as a new mixed input for next sub-branch. Inspired by ASPP [33], 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.

Subsequently, the outputs of MCFEU are passed to the follower attention unit (FAU) to select follower features and generate a follower attention mask (FAM). Similar to [19], in terms of the different characteristics of multi-level features, it is necessary to design the different attention modules for different level features. Generally, the channel-wise attention unit is utilized to select the high-level semantics, while the spatial attention unit is used to focus on the spatial details. The structures of FAU and LAU only depend on the choice of leader features. When high-level features are taken as the leader, the leader attention unit (LAU) will naturally represent the channel-wise attention unit, while FAU represents the spatial attention unit and FAM is a spatial-attention mask. Relatively, the situation when low-level features as leader can be similarly inferred. In this paper, high-level features are taken as the leader, therefore, LAU is the channel-wise attention unit, while FAU is the spatial attention unit. The structures of the spatial attention and channel-wise attention units are as same as [34,35], which are shown in Fig. 5 (a) and (b) respectively. We element-wise multiply FAM and the follower channel features together to get the multi-scale context-aware follower features. And then, with a residual learning strategy, the final selective follower features can be defined as:

$$F_f^s = F_f \times FAM + F_f \tag{3}$$

Subsequently, we transmit the FAM to the leader attention unit (LAU) and integrate it with the leader channels features as the inputs of LAU:

$$L_{input} = L_f \times FAM + L_f \tag{4}$$

Similarly, the final selective leader features after LAU can be formulated as:

$$L_f^s = \psi(L_{input}) \times L_f + L_f \tag{5}$$

ψ represents the convolutional operations in the leader attention unit. Finally, we predict two initial saliency maps from F_f^s and L_f^s with two 1×1 convolutional layers, and name them as F_0^s (low-level saliency map L_0) and L_0^s (high-level saliency map H_0), respectively. We transmit both the saliency maps and selective features to the subsequent feature fusing module.

3.3. Leader-follower feature fusing module

We design a leader-follower feature fusing module (LF^{3}M) to integrate the multi-level selective features for saliency detection. LF^{3}M contains two units, which are denoted as leader feature fusing unit (LF^{2}U) and follower feature fusing unit (F^{3}U), respectively. We elaborate the structure of this module with the n-th LF^{3}M as an example for conveniences.

As shown in Fig. 6, we first concatenate the selective leader features L_f^s and the previous follower saliency map F_{n-1}^s as the inputs of the leader feature fusing unit (LF^{2}U). We squeeze the channels with two groups of 1×1 filters, and then, two corresponding groups of 3×3 convolutional filters with different dilation rates

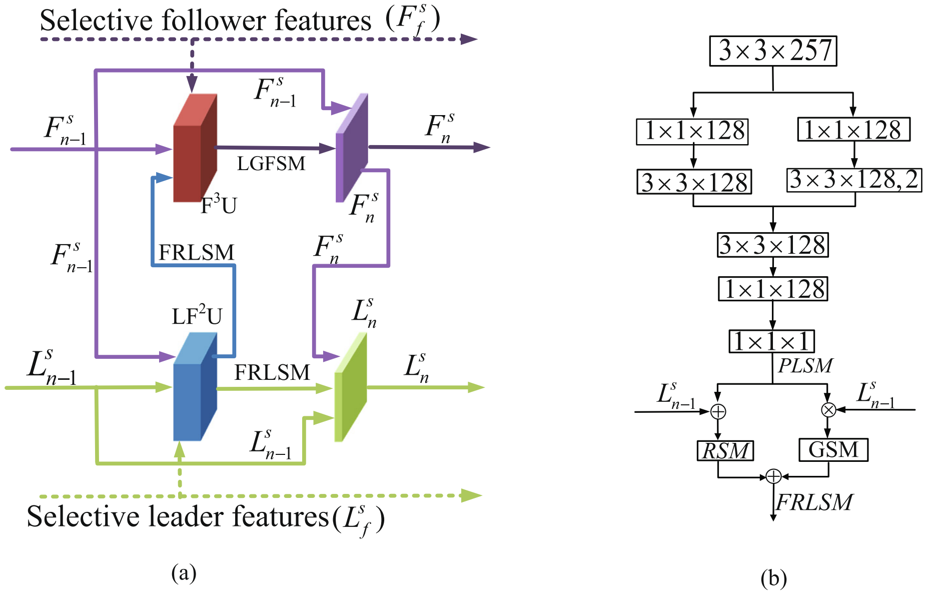


Fig. 6. The structure of the n-th feature fusing module and its constituent unit. (a) The overall structure of LF^{3}M. (b) The structure of leader feature fusing unit (LF^{2}U). The structure of F^{3}U is as same as LF^{2}U, but F^{3}U first produces a primary follower saliency map (PFSM), and then, PFSM is multiplied and added with the follower saliency map F_{n-1}^s, respectively. We consider the leader-guided follower saliency map (LGFSM) as the output of F^{3}U.

are applied to extract more local saliency cues. Next, we combine 3×3 and 1×1 convolutional layers to integrate these cross-channel saliency cues and produce a primary leader saliency map (PLSM). In my view, PLSM can be considered as a residual or gate for the leader saliency map L^s_{n-1}. Therefore, we generate a residual saliency map (RSM) and a gated saliency map (GSM) with the element-wise addition and multiplication, respectively. Finally, we add RSM and GSM together to acquire the final output leader saliency map of LF^{2}U. Since the output leader saliency map of LF^{2}U is produced with the follower saliency map as a reference, we name it as the follower-referenced leader saliency map (FRLSM). Then, we concatenate FRLSM and the selective follower features F^s_f as the inputs of follower feature fusing unit (F^{3}U). For a convenience, the structure of F^{3}U is as same as LF^{2}U. Similarly, since the feature-fused follower saliency map in F^{3}U is guided by leader features, we name the final output of F^{3}U as the leader-guided follower saliency map (LGFSM). With a residual learning, the final following saliency map F^s_n in this paper can be defined as:

$$F^s_n = F^s_{n-1} + LGFSM \tag{6}$$

In fact, LGFSM can also be considered as a residual which contains both the leader and follower saliency cues. The follower first gives a final output in competition. Subsequently, to make a Nash equilibrium for saliency detection, we transmit F^s_n to the leader features flow, and the final leader saliency map L^s_n in this module can be expressed as:

$$L^s_n = F^s_n + FRLSM + L^s_{n-1} \tag{7}$$

In this paper, the high-level features take as the leader, thus the leader saliency map L^s_n is equal to the n-th high-level saliency map (H_n), while the follower saliency map F^s_n is equal to the n-th low-level saliency map (L_n). As mentioned in Section 3.1, in Stackelberg competition, the leader features are aware of full selective saliency cues and have a superiority in this competition. We integrate multi-level features in the competition process but place extra emphasis on leader features to obtain more selective saliency cues.

We stack a serial of LF^{3}Ms in the network to refine the coarse features progressively. The number of LF^{3}M is set as 2, we will provide more experimental evidences to evaluate this setting in the section of ablation study.

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 edges of the salient region and reduce the performance of model. Moreover, the pixels surrounding the edge are hard to distinguish and need to assign more weights. In this paper, we employ a pixel position aware (PPA) loss function to learn the global structure of salient objects and assign more weights to hard pixels, which has been adopted by Wei et al. [27]. PPA loss is consisted of two weighted losses: a weighted binary cross entropy (wBCE) loss and a weighted IoU (wIoU) loss:

$$L^s_{ppa} = L^s_{wbce} + L^s_{wIou} \tag{8}$$

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

$$L^s_{wbce} = -\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(1+\gamma\alpha_{ij})\sum_{l=0}^{1}\mathbf{1}(g^s_{ij}=l)\log\Pr(p^s_{ij}=l|\Psi)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\gamma\alpha_{ij}} \tag{9}$$

where γ is the hyper-parameter set as 1. p^s_{ij} and g^s_{ij} are the prediction and ground truth of the pixel (i,j). 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:

$$\alpha^s_{ij} = \left\| \frac{\sum_{m,n\in A_{ij}} gt^s_{mn}}{\sum_{m,n\in A_{ij}} 1} - gt^s_{ij} \right\| \tag{10}$$

where A^s_{ij} represents the area surrounding 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:

$$L^s_{wIou} = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt^s_{ij} * p^s_{ij}) * (1+\gamma\alpha^s_{ij})}{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt^s_{ij} + p^s_{ij} - gt^s_{ij} * p^s_{ij}) * (1+\gamma\alpha^s_{ij})} \tag{11}$$

Furthermore, we apply multi-level deep supervision as an auxiliary loss to facilitate training sufficiently. The total loss of our network can be explained as:

$$L_{total} = \sum_{i=0}^{N}(L^i_L + L^i_F) \tag{12}$$

where N is the number of LF^{3}M, L^i_L, L^i_F represent the PPA losses of i-th leader and follower saliency map, 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 [36], DUTS [37], DUT-OMRON [38], HKU-IS [39], PASCAL-S [40]. 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 10553 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:

$$F_\beta = \frac{(1+\beta^2)\times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{13}$$

where $\beta^2 = 0.3$ is used to emphasize the precision. For a fair comparison, we adopt maximum F-measure (maxF, larger is better), average F-measure (avgF, larger is better) as the metrics. We also use the MAE metric (smaller is better) to measure the average dif-

Table 1
Quantitative results of the network based on the different leader features. LF^{3}Net_H_H represents LF^{3}Net with one MCLAM and one LF^{3}M, and both take high-level features as leader. LF^{3}Net_L_L represents LF^{3}Net with low-level features as leader in both modules. LF^{3}Net_H_HH denotes LF^{3}Net with one MCLAM and two LF^{3}M. Other settings can be learned similarly. The best two results are marked in red, blue.

Settings	ECSSD			DUTS			HKU-IS			PASCAL-S			DUT-OMRON		
	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
LF^{3}Net_H_H	0.934	0.911	0.038	0.868	0.839	0.043	0.922	0.899	0.035	0.889	0.866	0.082	0.783	0.755	0.068
LF^{3}Net_H_L	0.931	0.911	0.039	0.863	0.838	0.044	0.921	0.902	0.034	0.881	0.862	0.086	0.773	0.750	0.068
LF^{3}Net_L_H	0.926	0.907	0.041	0.851	0.826	0.047	0.912	0.892	0.037	0.881	0.862	0.088	0.766	0.745	0.069
LF^{3}Net_L_L	0.926	0.905	0.043	0.854	0.826	0.047	0.912	0.889	0.038	0.880	0.860	0.089	0.768	0.743	0.070
LF^{3}Net_H_HH	0.937	0.919	0.037	0.870	0.848	0.042	0.924	0.906	0.033	0.889	0.870	0.083	0.785	0.763	0.064
LF^{3}Net_H_HL	0.935	0.918	0.039	0.868	0.846	0.043	0.922	0.905	0.034	0.882	0.867	0.085	0.784	0.764	0.064
LF^{3}Net_H_LH	0.929	0.895	0.044	0.857	0.801	0.050	0.914	0.872	0.041	0.883	0.846	0.090	0.772	0.721	0.074
LF^{3}Net_H_LL	0.922	0.897	0.045	0.850	0.817	0.049	0.908	0.879	0.041	0.877	0.854	0.088	0.767	0.739	0.072
LF^{3}Net_L_HH	0.920	0.902	0.045	0.847	0.826	0.048	0.910	0.894	0.038	0.879	0.859	0.090	0.768	0.746	0.071
LF^{3}Net_L_HL	0.910	0.879	0.057	0.825	0.787	0.059	0.900	0.869	0.047	0.859	0.830	0.104	0.746	0.710	0.081
LF^{3}Net_L_LH	0.922	0.902	0.043	0.849	0.822	0.048	0.908	0.886	0.039	0.875	0.856	0.090	0.766	0.742	0.069
LF^{3}Net_L_LL	0.918	0.885	0.048	0.843	0.799	0.052	0.903	0.864	0.044	0.872	0.842	0.091	0.764	0.724	0.074

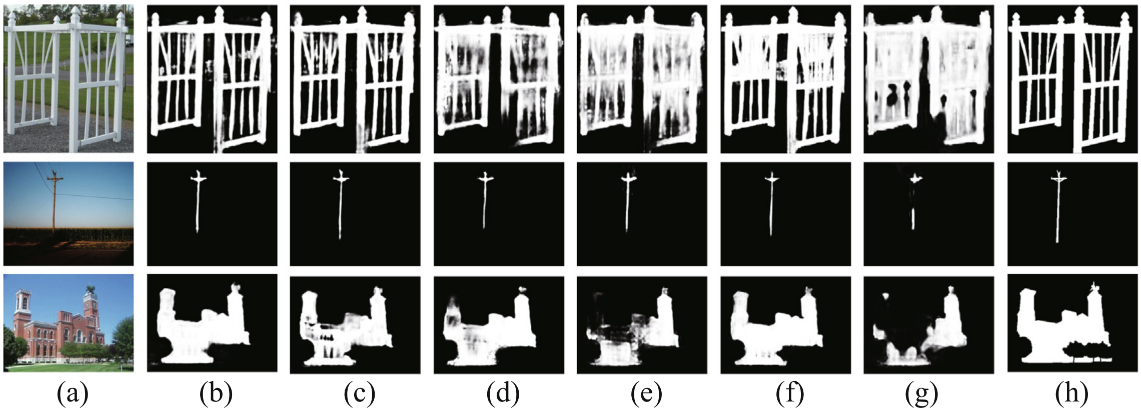


Fig. 7. The visual comparisons of the network with different leader features combinations. (a) Source images. (b) Results of LF^{3}Net_H_H. (c) Results of LF^{3}Net_H_L. (d) Results of LF^{3}Net_L_H. (e) Results of LF^{3}Net_L_L. (f) Results of LF^{3}Net_H_HH. (g) Results of LF^{3}Net_L_LL. (h) Ground truth.

Table 2
The MAE of the networks equipped with different MCLAM on DUTS and DUT-OMRON datasets.

	DUTS	DUT-OMRON
NoMCFEU	0.0692	0.0933
ASPP	0.0681	0.0909
DenseASPP	0.0698	0.0947
Our	0.0668	0.0905

ference between the saliency prediction and the ground truth. It is computed as the average pixel-wise absolute difference between the ground truth and the predicted saliency map:

$$MAE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\|P(i,j) - G(i,j)\| \tag{14}$$

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

Implementation Details. We implement our network based on PyTorch repository[1] 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 [41] or ResNet-50 [42]), 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 of 0.0005. During the training process, the initial learning rate is set as 0.001 and adjusted by the "poly" policy [43]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 24, while 20 k iterations with the batch size of 8 for the network based on VGGNet-19. We take the saliency map from the last leader feature fusing module (H_2) as the final prediction.

4.2. Ablation studies

The high-level features possess a superiority in the saliency detection task, to prove it, we carry out a serial of experiments to compare the performance of different leader feature combinations. Then, we conduct the ablation study to evaluate the importance of the sub-module PPA, MCLAM and LF^{3}M in our network. Finally, we construct the network based on different number of LF^{3}Ms to find

[1] https://pytorch.org

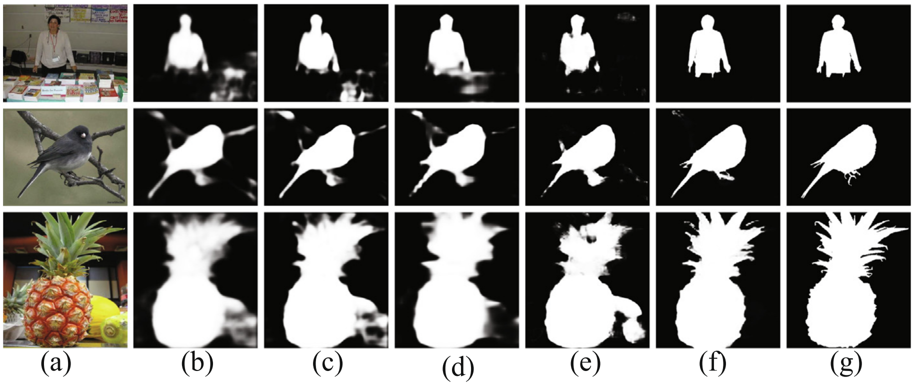


Fig. 8. The visual performance of ablation studies. (a) Source Images. (b) Results of baseline trained with BCE. (c) Results of baseline trained with PPA. (d) Results of PPA + MCLAM. (e) Results of PPA + LF^{3}M. (f) Results of PPA + MCLAM + LF^{3}M. (g) Ground truth.

Table 3
Quantitative results of the network based on the different modules. The best results are marked in red.

BCE	PPA	MCLAM	LF^{3}M	DUTS			PASCAL-S			DUT-OMRON		
				maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
√				0.804	0.745	0.071	0.852	0.813	0.108	0.720	0.666	0.096
√		√	√	0.858	0.827	0.049	0.887	0.861	0.088	0.782	0.754	0.069
	√			0.821	0.782	0.059	0.858	0.835	0.098	0.739	0.703	0.084
	√	√		0.829	0.797	0.058	0.861	0.841	0.090	0.747	0.721	0.082
	√		√	0.830	0.801	0.058	0.863	0.843	0.092	0.740	0.722	0.082
	√	√	√	0.870	0.848	0.042	0.889	0.870	0.083	0.785	0.763	0.064

Table 4
Quantitative results of the network with different number of LF^{3}Ms. The best results are marked in red.

Setting	FPS	ECSSD			DUTS			HKU-IS			PASCAL-S			DUT-OMRON		
		maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
LF^{3}M_1	88	0.934	0.911	0.039	0.869	0.839	0.043	0.923	0.899	0.035	0.889	0.866	0.082	0.783	0.755	0.068
LF^{3}M_2	75	0.937	0.919	0.037	0.870	0.848	0.042	0.924	0.906	0.033	0.889	0.870	0.083	0.785	0.763	0.064
LF^{3}M_3	64	0.937	0.919	0.037	0.871	0.848	0.042	0.925	0.907	0.033	0.886	0.867	0.083	0.788	0.766	0.065
LF^{3}M_4	57	0.935	0.919	0.038	0.870	0.847	0.043	0.924	0.906	0.034	0.886	0.867	0.086	0.790	0.767	0.064

the optimal network structure. We set the ResNet-50 version of FPN [32] as the baseline model.

Evaluation of different leader features: We perform a series of experiments to evaluate the contributions of different level features in our network. As shown in Table 1, the leader feature of MCLAM is key for the performance, the results of LF^{3}Net_H_H and LF^{3}Net_H_L are evidently higher than LF^{3}Net_L_H and LF^3-Net_L_L. With the same leader feature of MCLAM, the choose of LF^{3}M may improve less for the performance, but the results when high-level features take as the leader are roughly better than low-level features. These quantitative results in Table 1 demonstrate that the combinations with high-level features as leader features can evidently outperform low-level features. Moreover, with a high-level features leader, the network can perfectly suppress the noise and make a more accurate prediction, which can be intuitively seen in Fig. 7. Conclusively, these evidences can strongly prove the prior that high-level features may possess an implicit superiority in saliency detection task.

Evaluation of different modules: We first compare the performance of networks equipped with different MCLAMs. We carry out three ablation studies: MCLAM without MCFEU; MCLAM with ASPP(replaces MCFEU); MCLAM with DenseASPP [44](replaces MCFEU). As shown in the Table 2, MCLAM with MCFEU can make a clear improvement compared with other similar modules ASPP, DenseASPP. Then, we compare the performance of baseline model trained with BCE or PPA loss function. Compared with BCE, PPA

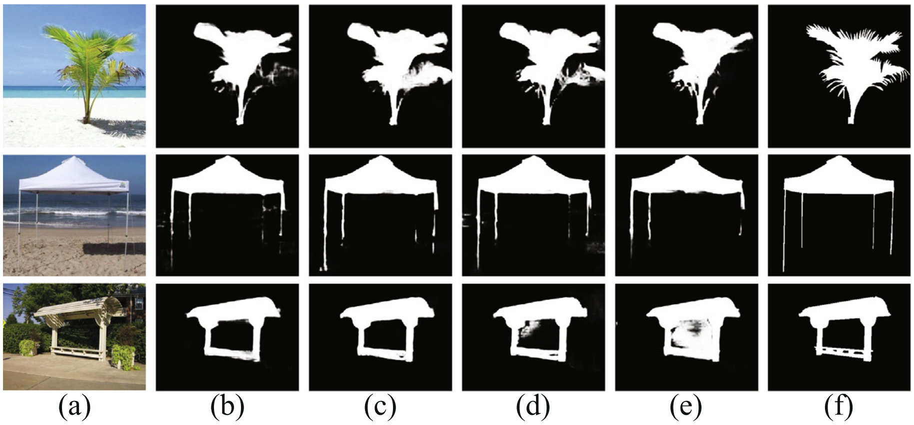


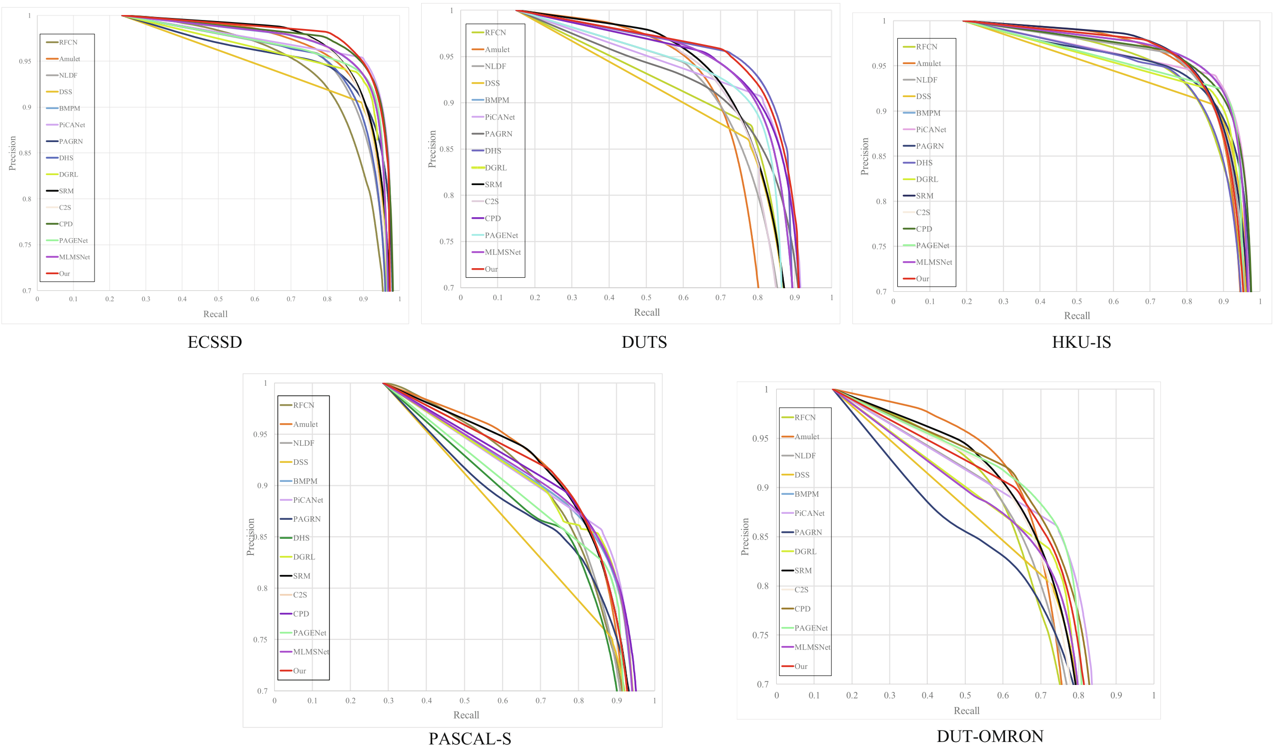
Fig. 9. The visual performance of the network with different numbers of LF^{3}M. (a) Source Images. (b) Results of the network with one LF^{3}M. (c) Results of the network with two LF^{3}Ms. (d) Results of the network with three LF^{3}Ms. (e) Results of the network with four LF^{3}Ms. (f) Ground truth.

Table 5
Quantitative comparisons of different methods. The best two results are marked in red, blue. The subscript of methods represent the publication year.

Method	ECSSD			DUTS			HKU-IS			PASCAL-S			DUT-OMRON		
	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
VGG-based															
RFCN$_{16}$	0.890	0.811	0.107	0.785	0.729	0.091	0.893	0.805	0.089	0.855	0.786	0.149	0.742	0.656	0.111
DHS$_{16}$	0.907	0.884	0.059	0.808	0.776	0.068	0.890	0.867	0.053	0.845	0.828	0.116	-	-	-
Amulet$_{17}$	0.915	0.882	0.059	0.778	0.729	0.085	0.895	0.856	0.052	0.862	0.830	0.107	0.743	0.692	0.098
NLDF$_{17}$	0.905	0.892	0.063	0.813	0.791	0.066	0.902	0.887	0.048	0.852	0.841	0.118	0.753	0.736	0.080
DSS$_{17}$	0.916	0.911	0.053	0.826	0.817	0.057	0.910	0.904	0.041	0.859	0.854	0.120	0.772	0.764	0.066
C2S$_{18}$	0.911	0.888	0.053	0.811	0.777	0.062	0.899	0.868	0.046	0.868	0.846	0.099	0.759	0.728	0.072
BMPM$_{18}$	0.928	0.900	0.045	0.851	0.816	0.049	0.921	0.888	0.039	0.881	0.856	0.094	0.774	0.745	0.064
RAS$_{18}$	0.921	0.900	0.056	0.831	0.804	0.059	-	-	-	0.850	0.836	0.129	0.786	0.761	0.061
PAGRN$_{18}$	0.927	0.902	0.061	0.855	0.823	0.056	0.918	0.888	0.048	0.867	0.844	0.120	0.771	0.748	0.071
PAGENet$_{19}$	0.931	0.912	0.042	0.838	0.817	0.052	0.921	0.916	0.031	0.873	0.865	0.093	0.792	0.772	0.062
JDF$_{19}$	0.927	0.900	0.049	0.832	0.801	0.059	0.920	0.892	0.039	0.877	0.847	0.105	0.801	0.779	0.057
MLMSNet$_{19}$	0.928	0.900	0.045	0.851	0.816	0.049	0.921	0.888	0.039	0.882	0.857	0.092	0.774	0.775	0.064
GFLN$_{19}$	0.920	0.909	0.054	0.843	0.829	0.052	0.912	0.900	0.041	0.860	0.852	0.116	0.762	0.749	0.066
ASNet$_{19}$	0.932	0.899	0.047	0.835	0.793	0.061	0.922	0.884	0.041	0.883	0.857	0.095	-	-	-
CANet$_{20}$	0.920	0.895	0.049	0.820	0.793	0.056	0.910	0.882	0.040	0.872	0.852	0.096	0.758	0.734	0.071
Our	0.929	0.912	0.043	0.862	0.839	0.047	0.922	0.907	0.035	0.877	0.857	0.089	0.800	0.770	0.068
ResNet-based															
SRM$_{17}$	0.917	0.897	0.054	0.827	0.798	0.059	0.906	0.881	0.046	0.868	0.848	0.103	0.769	0.744	0.069
DGRL$_{18}$	0.922	0.913	0.041	0.829	0.821	0.050	0.910	0.900	0.036	0.881	0.865	0.089	0.774	0.766	0.062
PiCANet$_{18}$	0.935	0.901	0.046	0.860	0.816	0.051	0.919	0.880	0.043	0.883	0.854	0.092	0.803	0.762	0.065
CapSal$_{19}$	0.862	0.852	0.074	0.824	0.802	0.062	0.881	0.864	0.057	0.876	0.867	0.097	0.639	0.631	0.095
CPD$_{19}$	0.939	0.924	0.037	0.866	0.845	0.044	0.925	0.904	0.034	0.885	0.871	0.092	0.797	0.780	0.056
DFNet$_{20}$	0.936	0.854	0.057	0.869	0.794	0.050	0.926	0.844	0.046	0.883	0.797	0.113	0.800	0.733	0.061
Our	0.937	0.919	0.037	0.870	0.848	0.042	0.924	0.906	0.033	0.889	0.870	0.083	0.785	0.763	0.064

loss function can introduce the structure information into the network and achieve a better result. Actually, it has seemed as a prior that the structure information is instructive for saliency detection [13,12,27]. Second, the multi-scale context information extracted by MCLAM can help the network weigh the saliency more accurately, which can be proved by the performance in Fig. 8 and Table 3. Finally, we evaluate the effect of LF^{3}M with a serial of experiments. The performance can perfectly prove the effect of our feature fusing strategy. And it is worth to notice that the combination of MCLAM and LF^{3}M can achieve a better result, which further proves the necessities of both modules.

Evaluation of different number of LF^{3}Ms: We construct the network equipped with different number of LF^{3}Ms. to find the optimal network structure. The results are summarized in Table 4 and



ECSSD DUTS HKU-IS

PASCAL-S DUT-OMRON

Fig. 10. The PR curves of different saliency detection methods.

Table 6
Running times and model sizes of some saliency methods.

Model	Our-R	Our-V	DHS	NLDF	Amulet	BMPM	PiCANet	CPD	CANet	DFNet	GCPANet	PFPN
Time(s)	0.013	0.008	0.05	2.21	0.05	0.06	0.097	0.023	0.091	0.043	0.02	0.025
Model size(MB)	156	121	376	400	126	252	189	183	74.5	439.5	256	765

Fig. 9. As shown in Table 4, the network equipped with two LF^{3}Ms can generally achieve the best performance, which also means that the network with two LF^{3}Ms can make a well Nash equilibrium. When the network reach the Nash equilibrium, the additional LF^{3}M can not improve the performance clearly, even the additional LF^{3}M will break the Nash equilibrium and decline the performance, which can be seen from the performance of the network with four LF^{3}Ms. Remarkably, when we test the proposed LF^{3}Net on the five datasets with an NVIDIA TITAN Xp GPU, the LF^{3}Net based on ResNet-50 and VGGNet-19 can run fast at the real-time speed of 75 and 120 FPS, respectively. Our method is much faster than most FCN-based saliency detection methods. We will provide more evidences to prove it in the next section.

4.3. Comparison with state-of-the-arts

We compare our proposed LF^{3}Net with 21 previous state-of-the-art methods, including RFCN [45], DHS [10], Amulet [46], NLDF [13], DSS [15], BMPM [11], RAS [22], PAGRN [21], C2S [47], PAGENet [48], JDF [49],MLMSNet [50],GFLN [51], SRM [52], DGRL [26], PiCANet [53], CapSal [54], CPD [9],ASNet [55], CANet [56], DFNet [57]. For fair comparison, all the saliency maps are provided by the authors or achieved by available codes or software.

Quantitative Comparisons. The quantitative results are shown in Table 5 and Fig. 10. Table 5 illustrates the performance of different methods in terms of three metrics: maxF, avgF, and MAE. Our proposed network LF^{3}Net can consistently outperform others on most datasets under different metrics, which demonstrates the effectiveness of our new feature fusing strategy. Although we do not realize the best performance on DUT-OMRON [38], our method demonstrates strong competitiveness. We also plot the PR curves of different methods on five datasets as a holistic evaluation metric. As shown in Fig. 10, the PR curves of our method perform better on most datasets than other methods.

Table 6 summarizes the running times and model sizes of some state-of-the-art methods. We provide the performance of ResNet-based and VGGNet-based methods, which are denoted as Our-R and Our-V respectively. As it can be seen, both methods can possess a higher efficiency with less parameters among all other compared methods. In a word, our method can strike a well balance between effect and efficiency, which is highly conducive to the further deployment.

Visual Comparisons. We also perform some qualitative comparisons in Fig. 11. We pick the images with various salient objects as examples. These salient objects vary from the number, size, contrast, texture, and so on. Compared to others, our method can achieve the better performance. For example, our network can well suppress the noise of background and make a holonomic and homogeneous prediction, which can be proved by the images in 1*th* to 4*th* rows. Moreover, our network can perfectly deal with the scene of small object, such as the images in 5*th* row. Generally, it is hard to distinguish the salient object in the low-contrast scenes, such as the images in 6*th* to 7*th* rows. However, our model can

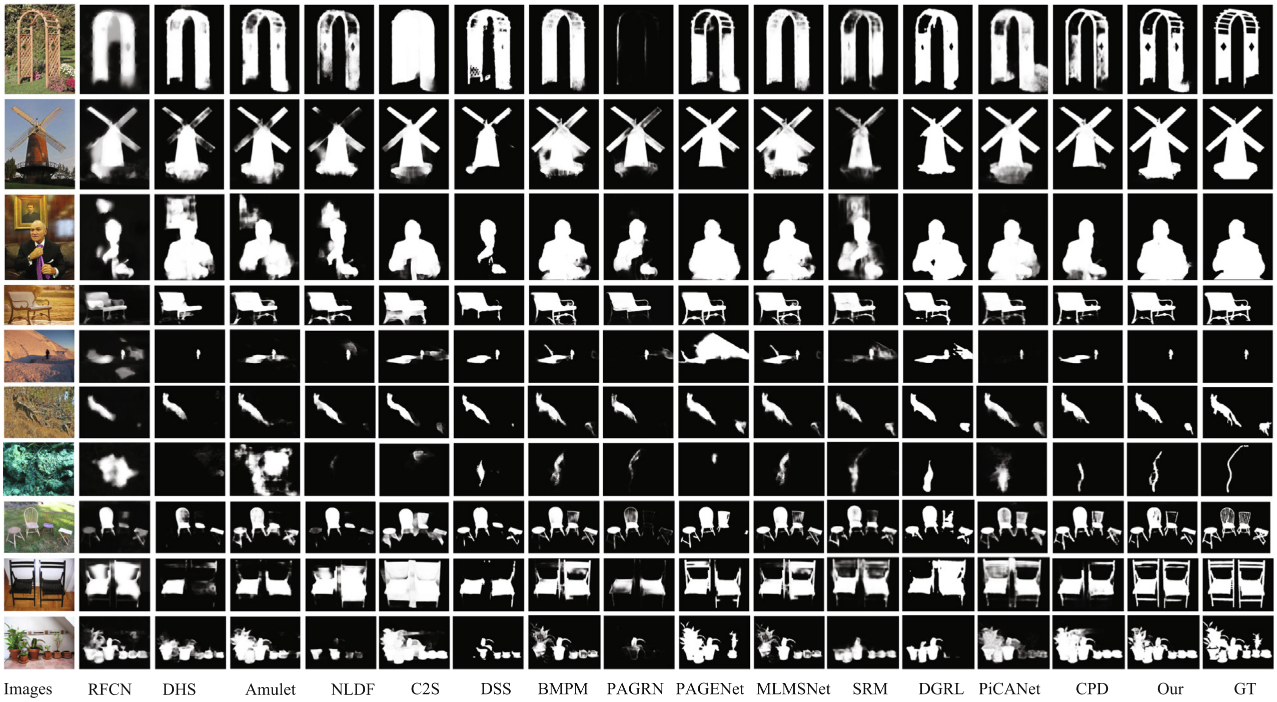


Fig. 11. The visual comparisons of different saliency detection methods.

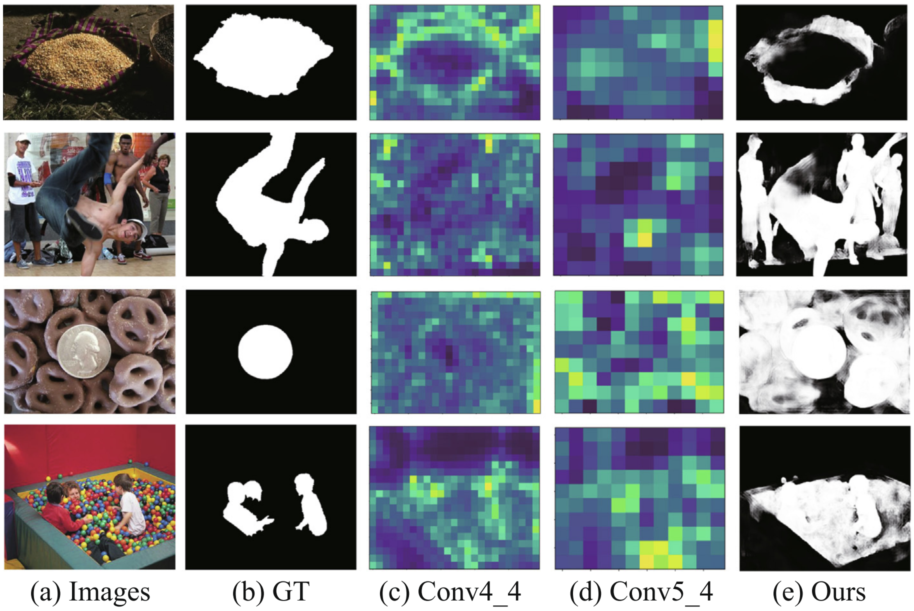


Fig. 12. Some failure cases of the proposed method. Conv4_4 and Conv5_4 represent the side-output feature maps of VGGNet19.

commendably distinguish these salient objects and make the more accurate predictions. Finally, our model shows the good capability of managing the scenes with multiple salient objects (the images of 8*th* to 10*th* rows). In a word, these visual results strongly demonstrate the good robustness and applicability of our method for these various salient objects.

4.4. Failure cases and further work

Fig. 12 shows some failure cases of the proposed method. As above mentioned, the high-level features play a major role in the saliency detection task. To make full use of the prior, we proposed the leader-follower features fusing network, thereby the high-level features are crucial for the performance. However, not all the high-level features are accurate and beneficial for saliency detection, which can be seen from the Fig. 12(c) and (d). Some salient areas are not activated in the high-level features, in some cases, the activated areas are distributed throughout the whole scenes, including the background areas. These inaccurately activated high-level features can easily lead to the failure cases.

To solve the problem, we need to obtain the more accurate high-level features with the deeper and wider network. Moreover, the proposed model only introduces the Stackelberg game to decode the features. In fact, the Stackelberg game theory can also be extended into the feature encoder and make the produced features fit better with the saliency detection task. We will explore it in the further work.

Essentially, the proposed network describes an unequal information competition model between different level features, which is organized in a cooperative as well as confrontational way. The model can be extended into other visual tasks. Heuristically, the game relationship may be a new direction to design the network.

5. Conclusions

In this paper, we introduce the Stackelberg game theory as a new feature fusing strategy, and based on the theory, we propose a novel leader-follower feature fusing network for saliency detection. The network equips with two sub-modules: one is the multi-scale context-aware leader-follower attention module (MCLAM) to capture the selective multi-scale features, and the other is the leader-follower feature fusing module (LF^{3}M) to fuse multi-level features. Extensive experimental results on five datasets prove that our method can outperform most of the state-of-the-art saliency detection approaches with a higher effect and efficiency. Furthermore, Stackelberg game theory can be used in all computer visual tasks which the high-level features possess the dominated position. We will further explore its effects and introduce it into other visual tasks.

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.

Acknowledgement

aaa

References

[1] 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, pp. 2129–2136.https://doi.org/10.1109/cvpr.2011.5995415..

[2] 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.

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

[4] Y. Rivenson, Y. Wu, A. Ozcan, Deep learning in holography and coherent imaging, Light: Sci. Appl. 8 (1) (2019) 1–8, https://doi.org/10.1038/s41377-019-0196-0.

[5] W. Wang, J. Shen, L. Shao, F. Porikli, Correspondence driven saliency transfer, IEEE Trans. Image Process. 25 (11) (2016) 5025–5034, https://doi.org/10.1109/TIP.2016.2601784.

[6] R. Cong, J. Lei, H. Fu, J. Hou, Q. Huang, S. Kwong, Going from RGB to RGBD saliency: a depth-guided transformation model, IEEE Trans. Cybern. 50 (8) (2020) 3627–3639, https://doi.org/10.1109/TCYB.2019.2932005.

[7] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: an in-depth survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021), https://doi.org/10.1109/TPAMI.2021.3051099, 1-1.

[8] R. Cong, J. Lei, H. Fu, M. Cheng, W. Lin, Q. Huang, Review of visual saliency detection with comprehensive information, IEEE Trans. Circuits Syst. Video Technol. 29 (10) (2019) 2941–2959, https://doi.org/10.1109/TCSVT.2018.2870832.

[9] Z. Wu, L. Su, Q. Huang, Cascaded partial decoder for fast and accurate salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3907–3916, https://doi.org/10.1109/cvpr.2019.00403.

[10] N. Liu, J. Han, DHSNet: deep hierarchical saliency network for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 678–686, https://doi.org/10.1109/cvpr.2016.80.

[11] L. Zhang, J. Dai, H. Lu, Y. He, G. Wang, A bi-directional message passing model for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1741–1750, https://doi.org/10.1109/cvpr.2018.00187.

[12] 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: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8779–8788, https://doi.org/10.1109/iccv.2019.00887.

[13] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, P.-M. Jodoin, Non-local deep features for salient object detection, in: Proceedings of the IEEE Conference on computer vision and pattern recognition, 2017, pp. 6609–6617, https://doi.org/10.1109/cvpr.2017.698.

[14] Z. Chen, Q. Xu, R. Cong, Q. Huang, Global context-aware progressive aggregation network for salient object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence 34(07), 2020, pp. 10599–10606, https://doi.org/10.1609/aaai.v34i07.6633.

[15] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P.H. Torr, Deeply supervised salient object detection with short connections, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3203–3212, https://doi.org/10.1109/cvpr.2017.563.

[16] S. Mohammadi, M. Noori, A. Bahri, S.G. Majelan, M. Havaei, CAGNet: Content-aware guidance for salient object detection, Pattern Recogn. (2020), https://doi.org/10.1016/j.patcog.2020.107303 107303.

[17] B. Wang, Q. Chen, M. Zhou, Z. Zhang, X. Jin, K. Gai, Progressive feature polishing network for salient object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence 34(07), 2020, pp. 12128–12135, https://doi.org/10.1609/aaai.v34i07.6892.

[18] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, P.-A. Heng, R^3 Net: 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.

[19] T. Zhao, X. Wu, Pyramid feature attention network for saliency detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3085–3094, https://doi.org/10.1109/CVPR.2019.00320.

[20] D. Fudenberg, J. Tirole, Game Theory (1992), https://doi.org/10.2307/2554596.

[21] X. Zhang, T. Wang, J. Qi, H. Lu, G. Wang, Progressive attention guided recurrent network for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 714–722, https://doi.org/10.1109/cvpr.2018.00081.

[22] 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.

[23] L. Zhang, J. Wu, T. Wang, A. Borji, H. Lu, A multistage refinement network for salient object detection, IEEE Trans. Image Process. 29 (2020), https://doi.org/10.1109/TIP.2019.2962688, 1-1.

[24] H. Li, G. Chen, G. Li, Y. Yu, Motion guided attention for video salient object detection, IEEE/CVF International Conference on Computer Vision (ICCV) 2019 (2019) 7273–7282, https://doi.org/10.1109/ICCV.2019.00737.

[25] J. Su, J. Li, Y. Zhang, C. Xia, Y. Tian, Selectivity or invariance: Boundary-aware salient object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3799–3808, https://doi.org/10.1109/ICCV.2019.00390.

[26] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3127–3135, https://doi.org/10.1109/cvpr.2018.00330.

[27] Q.H. Jun Wei, Shuhui Wang, F3Net: Fusion, feedback and focus for salient object detection, in: AAAI Conference on Artificial Intelligence (AAAI), Vol. 34, 2020, pp. 12321–12328..

[28] Q. Zhang, R. Cong, C. Li, M.M. Cheng, Y. Fang, X. Cao, Y. Zhao, S. Kwong, Dense attention fluid network for salient object detection in optical remote sensing images, IEEE Trans. Image Process. 30 (2021) 1305–1317, https://doi.org/10.1109/TIP.2020.3042084.

[29] Z. Chen, R. Cong, Q. Xu, Q. Huang, DPANet: Depth potentiality-aware gated attention network for rgb-d salient object detection, IEEE Trans. Image Process. (2020), https://doi.org/10.1109/TIP.2020.3028289, 1-1.

[30] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, A simple pooling-based design for real-time salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3917–3926, https://doi.org/10.1109/CVPR.2019.00404.

[31] W. Wang, J. Shen, M. Cheng, L. Shao, An iterative and cooperative top-down and bottom-up inference network for salient object detection, IEEE/CVF

Conference on Computer Vision and Pattern Recognition (CVPR) 2019 (2019) 5961–5970, https://doi.org/10.1109/CVPR.2019.00612.

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125, https://doi.org/10.1109/CVPR.2017.106.

[33] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587..

[34] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[35] J. Park, S. Woo, J.-Y. Lee, I.S. Kweon, BAM: Bottleneck attention module, arXiv preprint arXiv:1807.06514..

[36] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 1155–1162, https://doi.org/10.1109/CVPR.2013.153.

[37] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, X. Ruan, Learning to detect salient objects with image-level supervision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 136–145, https://doi.org/10.1109/CVPR.2017.404.

[38] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3166–3173, https://doi.org/10.1109/CVPR.2013.407.

[39] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5455–5463, https://doi.org/10.1109/CVPR.2015.7299184.

[40] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 280–287, https://doi.org/10.1109/CVPR.2014.43.

[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556..

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[43] W. Liu, A. Rabinovich, A.C. Berg, ParseNet: Looking wider to see better, arXiv preprint arXiv:1506.04579..

[44] M. Yang, K. Yu, C. Zhang, Z. Li, K. Yang, DenseASPP for semantic segmentation in street scenes, IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018 (2018) 3684–3692, https://doi.org/10.1109/CVPR.2018.00388.

[45] 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.

[46] P. Zhang, D. Wang, H. Lu, H. Wang, X. Ruan, Amulet: Aggregating multi-level convolutional features for salient object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 202–211, https://doi.org/10.1109/iccv.2017.31.

[47] X. Li, F. Yang, H. Cheng, W. Liu, D. Shen, Contour knowledge transfer for salient object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 355–370, https://doi.org/10.1007/978-3-030-01267-0_22.

[48] W. Wang, S. Zhao, J. Shen, S.C. Hoi, A. Borji, Salient object detection with pyramid attention and salient edges, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1448–1457, https://doi.org/10.1109/cvpr.2019.00154.

[49] Y. Xu, D. Xu, X. Hong, W. Ouyang, R. Ji, M. Xu, G. Zhao, Structured modeling of joint deep feature and prediction refinement for salient object detection, IEEE/CVF International Conference on Computer Vision (ICCV) 2019 (2019) 3788–3797, https://doi.org/10.1109/ICCV.2019.00389.

[50] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8150–8159, https://doi.org/10.1109/CVPR.2019.00834.

[51] Y. Zeng, P. Zhang, Z. Lin, J. Zhang, H. Lu, Towards high-resolution salient object detection, IEEE/CVF International Conference on Computer Vision (ICCV) 2019 (2019) 7233–7242, https://doi.org/10.1109/ICCV.2019.00733.

[52] T. Wang, A. Borji, L. Zhang, P. Zhang, H. Lu, A stagewise refinement model for detecting salient objects in images, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4019–4028, https://doi.org/10.1109/ICCV.2017.433.

[53] N. Liu, J. Han, M.-H. Yang, PiCANet: Learning pixel-wise contextual attention for saliency detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3089–3098, https://doi.org/10.1109/CVPR.2018.00326.

[54] L. Zhang, J. Zhang, Z. Lin, H. Lu, Y. He, Capsal: Leveraging captioning to boost semantics for salient object detection, IEEE Conference on Computer Vision and Pattern Recognition 2019 (2019) 6017–6026, https://doi.org/10.1109/CVPR.2019.00618.

[55] W. Wang, J. Shen, X. Dong, A. Borji, R. Yang, Inferring salient objects from human fixations, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 1913–1927, https://doi.org/10.1109/TPAMI.2019.2905607.

[56] J. Li, Z. Pan, Q. Liu, Y. Cui, Y. Sun, Complementarity-aware attention network for salient object detection, IEEE Trans. Cybern. (2020) 1–14, https://doi.org/10.1109/TCYB.2020.2988093.

[57] M. Noori, S. Mohammadi, S.G. Majelan, A. Bahri, M. Havaei, DFNet: discriminative feature extraction and integration network for salient object detection, Eng. Appl. Artif. Intell. 89 (2020), https://doi.org/10.1016/j.engappai.2019.103419 103419.

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.