A2SPPNet: Attentive Atrous Spatial Pyramid Pooling Network for Salient Object Detection

Yu Qiu, Yun Liu✉, Yanan Chen, Jianwen Zhang, Jinchao Zhu, Jing Xu✉

***Abstract*—Recent progress in salient object detection (SOD) mainly depends on the Atrous Spatial Pyramid Pooling (ASPP) module for multi-scale learning. Intuitively, different input images, different pixels, and different network layers may have different preferences for various feature scales. However, ASPP treats all feature scales as equally important by a simple sum operation. To this end, we propose Attentive Atrous Spatial Pyramid Pooling (A2SPP) by adding a new Cubic Information-Embedding Attention (CIEA) module at each branch of ASPP. In this way, each position in the 3D feature map can automatically learn the feature scales it prefers. Specifically, CIEA consists of Spatial-Embedding Channel Attention (SECA) and Channel-Embedding Spatial Attention (CESA). Instead of the previous direct squeeze and ignoring of one dimension when computing the attention for the other dimension, SECA/CESA attempts to embed spatial/channel information into channel/spatial attention, respectively. In addition, CIEA learns SECA and CESA for each 3D position simultaneously rather than previous separate computation of channel and spatial attention for each 2D position. Incorporating A2SPP and CIEA, the proposed A2SPPNet performs favorably against previous state-of-the-art SOD methods.**

***Index Terms*—Salient object detection, saliency detection, ASPP, A2SPP, attention mechanism.**

I. Introduction

SALIENT object detection (SOD), also known as saliency detection, aims at detecting the most conspicuous objects/regions in an image [1]–[3]. SOD is a crucial preprocessing step for many computer vision tasks such as visual tracking [4], image retrieval [5], video object segmentation [6], content-aware image editing [7], image thumbnailing [8], object recognition [9], and weakly-supervised learning [10], [11]. Recent progress on SOD mainly relies on convolutional neural networks (CNNs), especially fully convolutional networks (FCNs) [12], which can extract pixel-wise deep features from raw images and then make image-to-image prediction. Although numerous methods have been proposed to significantly improve SOD [13]–[30], it still remains a challenge to predict accurate saliency maps for natural images, especially for images with complicated scenarios.

Multi-scale learning, the effectiveness of which has been widely proven in the computer vision community [31]–[36], plays an essential role in CNN-based SOD for the recognition of objects with i) various scales in different natural images and ii) various aspect ratios of different object-parts in the same image. It is widely known that CNNs naturally achieve multi-scale learning by representing high-level semantic information at the top sides and low-level fine details at the bottom sides. Since the encoder-decoder architecture has the ability to take complementary use of multi-scale features from multiple CNN levels, encoder-decoder networks have dominated this field [22]–[30], [37]–[49]. The encoder is usually the existing pre-trained image classification models, *e.g.,* VGG [50] and ResNet [51], while most efforts are put on the design of the decoder by exploring various effective connections for combining multi-scale features [23], [40], [52]–[54].

Despite the great success brought about by encoder-decoder networks [28]–[30], [37]–[48], the natural multi-scale learning of CNNs is limited, because only several CNN sides are used for feature decoding while the scales and shapes of real-world objects are uncertain. To enrich multi-scale features, it is a good choice to connect atrous convolutions [55] with various dilation rates. A typical module using atrous convolutions is the Atrous Spatial Pyramid Pooling (ASPP) module [32] that is originally proposed for semantic segmentation. ASPP aggregates the features obtained from multiple atrous convolution branches with various dilation rates so that it can enlarge the receptive field to incorporate multi-scale contextual information without sacrificing spatial resolution. Current state-of-the-art saliency methods still rely on the ASPP module [32] for better multi-scale learning [17], [40], [56]–[59].

However, ASPP [32] suffers from an obvious limitation, which may affect its ability for feature representation. Specifically, ASPP directly adds up convolutional features of multiple scales by viewing all scales as equally important. The equal importance may be an improper assumption, because i) different input images and different positions may have different preferences for multi-scale features due to the scale varieties in object/object-parts, and ii) different network layers may have different preferences for multi-scale features due to the intrinsic CNN properties. Therefore, directly adding up multi-scale features without selection may lead to suboptimal representation and introduce unnecessary noises.

To address the above problem, we note that the attention mechanism, which is inspired by the human visual system [60], [61], can be used to enhance the necessary activation and suppress the noisy activation of CNN feature maps. However, there are some obvious limitations in existing attention mechanisms [62]–[65]. First, they usually squeeze spatial/channel information directly through pooling operations when deriving channel/spatial attention, causing serious information loss. Second, they usually compute channel

This work is supported by the Tianjin Research Innovation Project for Postgraduate Students (No.2020YJSZXB04). *(Corresponding author: Yun Liu and Jing Xu.)*

Y. Qiu, Y. Chen, J. Zhang, J. Zhu, and J. Xu are with College of Artificial Intelligence, Nankai University.

Y. Liu is with Computer Vision Lab, ETH Zurich.

and spatial attention separately, ignoring the interrelationship between two kinds of attention. Third, they usually enhance the feature map using sequential channel and spatial attention rather than modeling 3D attention directly. To this end, we propose a **Cubic Information-Embedding Attention (CIEA)** module. CIEA first introduces two attention sub-modules, *i.e.,* **Spatial-Embedding Channel Attention (SECA)** and **Channel-Embedding Spatial Attention (CESA)**, which can not only learn channel and spatial attention but also embed the information from one dimension into the attention map of the other dimension. In this way, the learned channel/spatial attention can encode the whole input feature map without information loss. The complementary spatial/channel information can make channel/spatial attention have a global view for the input feature map, respectively. Then, CIEA combines channel and spatial attention into a 3D attention map to learn a weight for each position in the 3D feature map, unlike the previous separate usage of channel and spatial attention.

With CIEA incorporated, we propose the **Attentive Atrous Spatial Pyramid Pooling (A2SPP)** module. By adding a CIEA module to each branch of the original ASPP, the resulting A2SPP can automatically enhance essential scales and suppress noisy scales for each 3D position. Such a way of feature scale aggregation improves the capability of multi-scale representation. Moreover, A2SPP adds an attentive residual branch to enrich multi-scale features. Using A2SPP and CIEA as basic modules, we build a simple encoder-decoder network, *i.e.,* **A2SPPNet**, for SOD. Specifically, we place A2SPP in the partial top sides of the decoder to learn multi-scale high-level semantic features. For other bottom sides, we use CIEA to further refine the local information because it is unnecessary to perform multi-scale learning for low-level fine-grained features. We also introduce a **Semantic Guidance Learning (SGL)** technique to utilize high-level semantic information for guiding the learning of the decoder.

Extensive experiments on six challenging datasets suggest that the proposed A2SPP is more effective than the traditional ASPP, demonstrating that the attention mechanism CIEA is effective in improving multi-scale representation learning. In addition, A2SPPNet favorably outperforms existing state-of-the-art SOD methods in terms of popular evaluation metrics. By predicting better saliency maps, A2SPPNet would be useful for many real-world applications. For example, it can be directly used to improve mobile phone photography [27] and human-robot interaction [66], [67]. Moreover, A2SPPNet can also be viewed as a pre-processing technique and thus has the potential to improve many other real-world applications such as image/video compression [68], [69], content-based image retrieval and image collection browsing [70]–[73], photo collage/media re-targeting/cropping/thumbnailing [74]–[76], as well as the applications mentioned at the beginning of this introduction section.

The main contributions of this paper include the following:

- We propose the A2SPP module that adopts a CIEA module at each branch of ASPP [32] to achieve an automatic selection of feature scales for better multi-scale learning.
- We design the CIEA module to learn the 3D attention map, which consists of SECA and CESA sub-modules, which introduce spatial and channel information dependencies to channel and spatial attention calculations, respectively.
- We build a simple A2SPPNet using the proposed A2SPP and CIEA modules, which achieves state-of-the-art performance for SOD.

II. Related Work

A. Salient Object Detection

Early SOD methods heavily rely on hand-crafted features [1]–[3] and heuristic priors such as color contrast [1], center prior [2], and background prior [3]. Nevertheless, hand-craft features and priors can hardly capture high-level semantic information which is important to locate salient objects especially in complicated scenes.

Recently, due to the powerful representation capability of CNNs, CNN-based saliency detectors have dominated this field and the accuracy has been remarkably boosted [13]–[30], [37]–[44], [77]–[88]. It is well accepted that the high-level semantic information extracted at the top CNN layers can better locate the coarse positions of salient objects, while the low-level information extracted at the bottom layers can refine the details (*e.g.,* object boundaries) of salient objects. Both the high-level and low-level information is important to accurately segment salient objects [27], [52]. Therefore, most existing CNN-based methods focus on aggregating multi-level and multi-scale features by designing effective decoders [25]–[30], [37]–[49], [89], [90]. For example, Zhang *et al.* [40] designed a bi-directional structure to pass messages between multi-level features controlled by a gate function. Wang *et al.* [91] proposed a pyramid attention structure that can focus more on salient regions while exploiting multi-scale information. Liu *et al.* [89] explored the pooling operation for SOD guided by a designed global guidance module. They also designed a feature aggregation module to make the high-level semantic information well fused with low-level features. Instead of exploring the fusion of multi-level features extracted from the encoder, we aim at how to enrich the multi-scale learning of SOD networks, because SOD heavily depends on multi-scale learning for locating the position and segmenting the details of salient objects with various scales.

Pang *et al.* [92] proposed MINet to enrich multi-scale learning for SOD. They first used an aggregate interaction module to integrate the features from adjacent network sides. Then, they applied a self-interaction module at each side to learn multi-scale features, where the input feature is downsampled by a factor of 2 and then processed, communicating with the original input feature. We believe that such a self-interaction module is limited in multi-scale learning, as only two scales (*i.e.,* 1 and 1/2) are explored. In this paper, we propose A2SPP to explore general multi-scale learning under multiple scales in an intuitive manner.

B. Atrous Spatial Pyramid Pooling

Atrous Spatial Pyramid Pooling (ASPP) [32] is originally proposed for semantic segmentation. ASPP adopts four par-

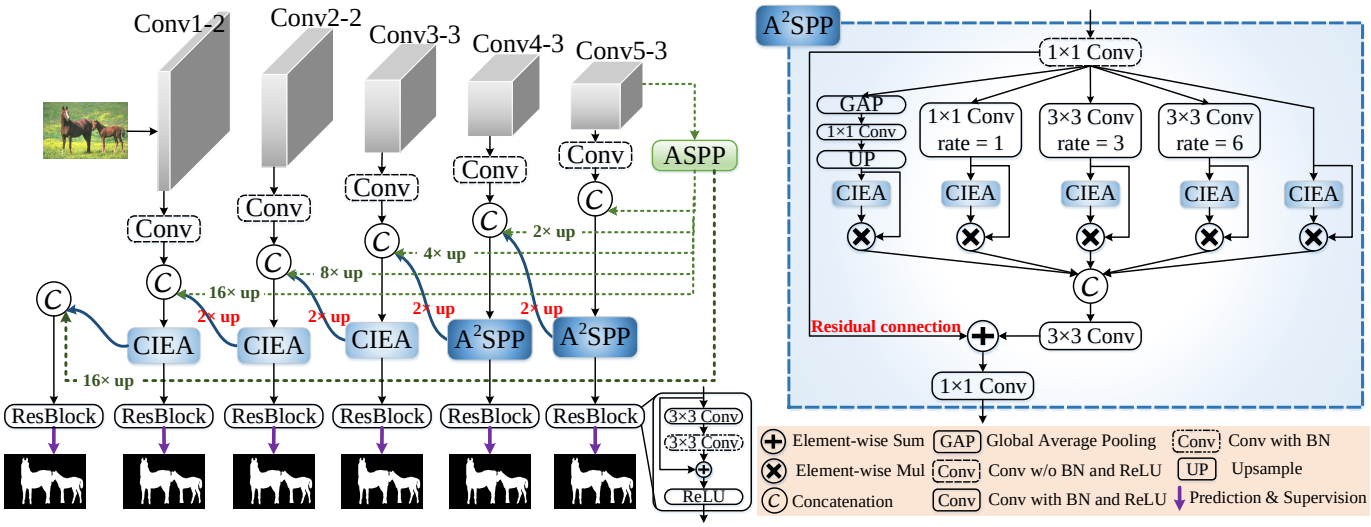


Fig. 1. Left: network architecture of the proposed A2SPPNet with the VGG16 backbone [50]. Right: structure of the proposed A2SPP module.

allel atrous 3×3 convolutions [55] with different dilation rates (*i.e.,* $rates = (6, 12, 18, 24)$) on top of the encoder for capturing multi-scale information. Then, Chen *et al.* [93] found that when the dilation rate gets larger, the 3×3 convolution filter degenerates to a 1×1 filter as only the center weight is effective. Hence they replaced the $(3 \times 3, rates = 24)$ convolution with a 1×1 convolution. They also added a global average pooling branch to extract image-level features.

For SOD, many studies [40], [56]–[59], [89] use ASPP or its variants for extracting multi-scale features without sacrificing spatial resolution. For example, Liu *et al.* [89] proposed a feature aggregation module (FAM) to replace the atrous convolutions [55] in ASPP with successive pooling and vanilla convolutions. Zhao *et al.* [59] and Zhang *et al.* [40] proposed CPFE and MCFE modules by tuning the dilation rates of ASPP. Zhao *et al.* [59] used the channel attention to enhance the features extracted by their CPFE module, but this is orthogonal to our goal that aims at automatically learning each point's preference to different feature scales. To this end, we adopt a CIEA module at each branch of the A2SPP module for an automatic scale selection.

C. Attention Mechanisms

The attention mechanism is first found in human visual system [60], [61]. It enhances essential information and filters out noisy information. Similar ideas are applied to neural networks and have achieved successes in many tasks such as scene recognition [94] and and image captioning [95]. The recent development of SOD has also benefited from attention mechanisms [30], [37], [40], [59], [91]. In general, attention mechanisms can be categorized into two classes: *squeeze-and-excitation* (SE) style and *non-local* style.

SE style recalibrates the channel- or spatial-wise feature responses by rescaling different channels or spatial positions. He *et al.* [62] proposed *squeeze-and-excitation* networks, which first squeezes global spatial information into a channel descriptor and then maps the channel descriptor to a set of channel weights for recalibrating the importance of different channels. Inspired by this, Woo *et al.* [63] introduced both channel and spatial attention mechanisms to process features along the channel and spatial dimensions, respectively. However, all these SE-style methods obtain channel and spatial attention by squeezing the feature map along spatial or channel dimension, which is not effective enough for complete context modeling. Moreover, their channel and spatial attention is computed separately, ignoring the interactions between channel and spatial dimensions.

Non-local style actually learns query-independent attention maps for each query position to model pixel-level pairwise relations. a weighted sum of the features at all positions for capturing long-range position interactions. Based on NLNet [25], many works [96], [97] focus on decreasing the computation and GPU memory consumption brought by the matrix multiplications of the standard non-local module. Overall, non-local-style methods aim at modeling pixel-level pairwise relations via self-attention mechanisms. However, they all share an essential problem, *i.e.,* the prohibitive computational cost and vast GPU memory occupation hinder its usage in applications.

III. Methodology

A. Overall Framework

As shown in Fig. 1, the proposed salient object detector, A2SPPNet, is an encoder-decoder network. For the encoder, A2SPPNet uses the typical VGG16 [50] or ResNet50 [51] as its encoder. Here, we take VGG16 as an example to describe our A2SPPNet, while the ResNet50-based A2SPPNet can be easily derived. We make two modifications to VGG16: (i) we remove the final fully-connected layers of VGG16 to serve as an FCN [12] for image-to-image translation; (ii) we remove the last pooling layer, *i.e.,* only remaining four pooling layers. We focus on designing an effective decoder which can be viewed as a top-down generation path, and each block originates from the side output of the corresponding encoder

side and the preceding decoder side. The key of the decoder is multi-scale learning for extracting and aggregating multi-scale features in an effective way. To achieve this goal, we design a novel module, **Attentive Atrous Spatial Pyramid Pooling (A2SPP)**, and place it in the partial top sides of the decoder. As displayed in Fig. 1 and Section III-B, the A2SPP module is an improved version of ASPP [32], [93]. A2SPP aims at solving the ASPP's improper assumption that all feature scales are of equal importance, by learning attention for the feature scale selection using the **Cubic Information-Embedding Attention (CIEA)** module. The structure of the CIEA module is illustrated in Fig. 2, which will be introduced in Section III-C.

As we know, the top layers of CNNs learn high-level semantic abstraction of the input image, which is essential for locating salient objects. In contrast, the bottom layers contain low-level fine-grained details, which are useful for refining object details [25]–[30], [37]–[49]. A2SPP inherits the natural property of ASPP to enhance the high-level semantic features through multi-scale learning [32], [93], [98]. Hence, we place an A2SPP module at each of the top two decoder sides. For the bottom sides, on one hand, it is unnecessary to adopt A2SPP because the bottom layers aim at learning low-level fine-grained details. On the other hand, it is meaningless to apply A2SPP at the bottom sides, because ASPP is originally designed to process small feature maps at the top sides and thus unsuitable to process large feature maps at the bottom sides [32], [93], [98]. Hence, we just place a CIEA module at each of the bottom three decoder sides for feature enhancement. Our experiments in Section IV-C1e demonstrate the advantage of such a two-part decoder.

We also follow the idea in PoolNet [89] to use high-level semantic features for guiding the learning of each decoder side, namely **Semantic Guidance Learning (SGL)**. Specifically, we connect an ASPP module on top of the encoder. The output is transformed into feature maps with 32, 16, 8, 4, and 2 channels by 1×1 convolutions, which are fed into each decoder side (before A2SPP or CIEA) from top to bottom, respectively. This output is also transformed into an 8-channel feature map that is concatenated with the feature map at the bottom decoder side for final saliency prediction, as shown in Fig. 1. The above ASPP module consists of four branches: a 1×1 convolution, two atrous 3×3 convolutions with dilation rates of 3 and 6, and the global average pooling. Each branch generates a 256-channel feature map, and we concatenate the outputs of four branches to produce multi-scale features. Note that we adopt traditional ASPP rather than our A2SPP in SGL. As we know, different network sides prefer different feature scales [99], [100]. A2SPP adds feature scale selection to ASPP, where it is difficult for a specific selection to satisfy all network sides. In other words, the feature scale selection in A2SPP may be unnecessary because the preferable scales of different sides are uncertain. In Section IV-C4c, we empirically verify this hypothesis.

The feature map obtained from each A2SPP or CIEA module is fed into a residual block containing two sequential 3×3 convolution layers with batch normalization and nonlinearization. The output channels from top to bottom are 128, 128, 64, 16, and 16. Then, a 1×1 convolution with a single output channel is applied to the output feature map of each decoder side. The *sigmoid* activation function is followed to predict the saliency probability map whose values range from 0 to 1. The ground truth is imposed to supervise these intermediate saliency predictions for deep supervision which has been proved to be effective for SOD [23], [28], [37], [39], [42], [44], [54], [89], [91], [101]. Similarly, we derive the final saliency map from the bottom decoder side, as shown in Fig. 1. This final saliency map is also supervised by the ground truth in the training phase and serves as the output of A2SPPNet in the test phase.

The proposed A2SPPNet is trained end-to-end using the standard *binary cross entropy loss* (BCE). The total loss can be calculated as

$$\mathcal{L} = \mathrm{BCE}(\mathbf{P}, \mathbf{G}) + \lambda \sum_{i=1}^{5} \mathrm{BCE}(\mathbf{P}_i, \mathbf{G}), \tag{1}$$

where $\mathbf{G}$ denotes the ground-truth saliency map. $\mathbf{P}$ denotes the final predicted saliency map, and $\mathbf{P}_i$ is the predicted saliency map at the i-th decoder side. λ represents the weighting scalar for loss balance. In this paper, we empirically set λ to 0.4 as suggested by [33], [102].

B. Attentive Atrous Spatial Pyramid Pooling

The traditional ASPP module [32], [93] uses several atrous convolutions to process deep features in parallel. Since these atrous convolutions have different dilation rates, ASPP can extract multi-scale features. ASPP assumes that all parallel branches are of equal importance, and ASPP sums the features from all branches directly without selection. However, *different input images, different positions, and different network layers* would have different preferences for feature scales because of the various scales of objects/parts and the diversity of multi-scale deep features. Each position in each feature map should have the ability to learn feature scales that it prefers, through which necessary feature scales should be emphasized and redundant feature scales should be suppressed. Therefore, it is improper for ASPP to view all feature scales as equally important.

To address this problem, we improve ASPP to A2SPP. The structure of A2SPP is illustrated in Fig. 1. We add a novel CIEA module after the atrous convolution of each A2SPP branch. With a 3D atrous-convolved feature map as input, the CIEA module learns a 3D attention map with the same size as the input. For each branch, the attention value at each position represents the importance of the corresponding feature scale for this position. We multiply the attention map and the atrous-convolved feature map in an element-wise way. The resulting feature maps of all branches are concatenated. In this way, each position in the 3D feature space automatically learns its specific linear combination of all feature scales. With this automatic scale selection, the output feature map learns a better multi-scale feature representation.

Specifically, we adopt a 1×1 convolution and a varying number of atrous 3×3 convolutions with different dilation rates according to the spatial resolutions of feature maps from

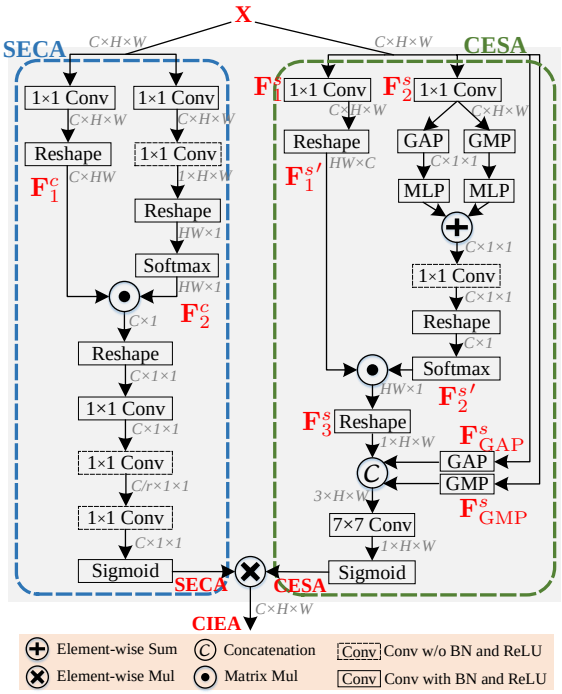


Fig. 2. Network structure of the proposed CIEA mechanism.

different layers. For example, the output stride is 16 for the fifth convolution layer after four pooling layers, we use two atrous 3×3 convolutions with dilation rates of $(3, 6)$. For the fourth convolutional layer, we use three atrous 3×3 convolutions with dilation rates of $(3, 6, 9)$. The reason for this design is that when a excessively large dilation rate is applied to a small feature map, the number of valid convolution filter weights that do not act upon the padded zeros becomes small [93], to make the dilation rates be in accordance with the size of the feature map. Moreover, since global features have been proved to be essential for SOD [1], [25], [28], A2SPP also has a parallel branch that contains a global average pooling layer and a CIEA module, which is designed to extract image-level features. In addition, inspired by the effectiveness of residual connections [51], [94], we add an attentive residual shortcut branch to A2SPP by introducing another CIEA module to process the input feature map.

We concatenate the feature maps from all parallel branches. The number of channels for all A2SPP branches at both the fifth and fourth blocks is 128. After concatenation, we change the fused feature maps to 128 channels again using a 3×3 convolution. Since we employ several attention modules in parallel, multiplying by factors that are within the range of $(0, 1)$ can weaken the activation, leading to vanishing gradients. Hence, we add a residual connection [51] at the end of A2SPP to facilitate gradient propagation, followed by a 1×1 convolution layer to obtain the final feature map.

C. Cubic Information-Embedding Attention

As discussed in Section II-C, there are three obvious limitations in existing attention mechanisms [30], [37], [57], [59], [91]: i) they squeeze the spatial/channel information directly for the channel/spatial attention learning, and as a result, there is much information loss for complete context modeling; ii) they learn channel and spatial attention separately, ignoring the interactions between the channel and spatial dimensions; iii) previous channel/spatial attention mechanisms cannot learn the attention value for each point of the 3D feature map.

To overcome the above limitations, we propose the CIEA module to generate a 3D attention map with the same size as the input feature map (problem iii)). When computing the attention for one dimension, CIEA does not squeeze the other dimension and thus avoids the information loss in previous methods (problem i)). In addition, CIEA can directly model the channel- and spatial-wise dependency simultaneously rather than modeling the channel and spatial attention separately, as before (problem ii)). The detailed architecture of the CIEA module is illustrated in Fig. 2. Given an input feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, CIEA computes a cubic attention map $\mathrm{CIEA}(\mathbf{X}) \in \mathbb{R}^{C \times H \times W}$, where C, H, and W represent the channel, height, and width of the input feature map, respectively. The enhanced feature $\mathbf{Y}$ is computed as

$$\mathbf{Y} = \mathbf{X} \otimes \mathrm{CIEA}(\mathbf{X}) + \mathbf{X}, \tag{2}$$

where $\otimes$ denotes element-wise multiplication. A residual connection [51] is adopted to facilitate gradient flow.

CIEA consists of two information embedding attention mechanisms, *i.e.,* **Spatial-Embedding Channel Attention (SECA)** and **Channel-Embedding Spatial Attention (CESA)**. The SECA branch not only learns the dependencies among feature channels but also embeds the spatial information to generate channel attention $\mathrm{SECA}(\mathbf{X}) \in \mathbb{R}^{C \times 1 \times 1}$. Similarly, the CESA branch attempts to encode the channel information into spatial dimensions to learn spatial attention $\mathrm{CESA}(\mathbf{X}) \in \mathbb{R}^{1 \times H \times W}$. We aggregate these two branches to obtain the cubic attention map:

$$\mathrm{CIEA}(\mathbf{X}) = \mathrm{SECA}(\mathbf{X}) \otimes \mathrm{CESA}(\mathbf{X}), \tag{3}$$

where $\mathrm{SECA}(\mathbf{X})$ and $\mathrm{CESA}(\mathbf{X})$ are replicated into the input size of $\mathbb{R}^{C \times H \times W}$ before multiplication. For combining $\mathrm{SECA}(\mathbf{X})$ and $\mathrm{CESA}(\mathbf{X})$, the element-wise multiplication in Eq. (3) can be replaced with summation. Both combining methods achieve similar performance, and the multiplication performs slightly better, as shown in Section IV-C4b. Hence, we empirically choose element-wise multiplication.

1) Spatial-Embedding Channel Attention: The channel attention mechanism exploits the inter-channel relationship for learning "what" to focus on or suppress. As mentioned above, traditional channel attention mechanisms usually obtain channel vectors by squeezing the spatial information directly using the *global average pooling* (GAP) or *global max pooling* (GMP) on the input feature map. Intuitively, a single average or max value cannot characterize the whole spatial dimension exactly. Hence, such crude pooling would lead to much information loss, making the resulting channel attention suboptimal in modeling the global contextual information. Therefore, the proposed SECA absorbs spatial information into channel attention in a non-squeeze way for global context modeling.

As shown in Fig. 2, we first feed the input feature map $\mathbf{X}$ into two 1×1 convolution layers (with nonlinearization)

to generate two new feature maps, both of which have the size of $\mathbb{R}^{C\times H\times W}$. Then, one of them is reshaped to $\mathbf{F}_1^c \in \mathbb{R}^{C\times HW}$. The other is first changed to a 1-channel feature map by a 1×1 convolution (without nonlinearization) and then, reshaped to $\mathbf{F}_2^c \in \mathbb{R}^{HW\times 1}$. Next, we apply *softmax* to $\mathbf{F}_2^c$ for normalization, *i.e.,* $\mathbf{F}_2^c = \mathrm{Softmax}(\mathbf{F}_2^c)$. After that, we perform a matrix multiplication for $\mathbf{F}_1^c$ and $\mathbf{F}_2^c$ to obtain a spatial-embedding matrix $\mathbf{F}^c$, which can be summarized in the following formulas:

$$\begin{aligned} \mathbf{F}^c &= \mathbf{F}_1^c \otimes \mathbf{F}_2^c, \\ \mathbf{F}^c \in \mathbb{R}^{C\times 1}, &\mathbf{F}_1^c \in \mathbb{R}^{C\times HW}, \mathbf{F}_2^c \in \mathbb{R}^{HW\times 1}. \end{aligned} \tag{4}$$

$\mathbf{F}^c$ is further reshaped to $\mathbf{F}^c \in \mathbb{R}^{C\times 1\times 1}$. Then, we connect three 1×1 convolution layers to process $\mathbf{F}^c$. To reduce the number of parameters, we set the number of output channels of the first 1×1 convolution as $C/4$, which is returned to C using the third 1×1 convolution. Finally, the *sigmoid* function is applied in SECA. In summary, we can formulate the computation of SECA as

$$\begin{aligned} \mathrm{SECA}(\mathbf{X}) = \sigma(\mathrm{Conv}_{1\times 1}(\mathrm{Conv}_{1\times 1}(\mathrm{Conv}_{1\times 1}(\mathbf{F}^c)))), \\ \mathrm{SECA}(\mathbf{X}) \in \mathbb{R}^{C\times 1\times 1}, \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ is the *sigmoid* function.

Next, let us discuss how SECA embeds the spatial information into channel attention. As shown in Eq. (4), $\mathbf{F}^c$ is generated by the matrix multiplication of $\mathbf{F}_1^c$ and $\mathbf{F}_2^c$. In this way, the response at each position of $\mathbf{F}^c$ is computed by the sum of the product of all position pairs in $\mathbf{F}_1^c$ and $\mathbf{F}_2^c$. In contrast to previous methods that adopt GAP or GMP to squeeze the spatial information directly, SECA abstracts the spatial information in a learning way. Therefore, spatial information can contribute to channel attention through learning rather than through a predefined average or maximum operation.

2) Channel-Embedding Spatial Attention: The spatial attention mechanism aims at exploiting the relationships among spatial positions for learning "where" to focus on or suppress. Given an input feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, traditional spatial attention mechanisms squeeze the channel information directly by using a convolution for reducing the number of channels to 1. We believe that such a direct squeeze is insufficient to make good use of channel information. More importantly, such a direct squeeze can capture only local information, while global context modeling has been proved to be essential for SOD [1], [19], [25], [28], [41], [77]. To address this aim, we propose CESA to embed channel information into spatial attention for global context modeling.

The pipeline of CESA is shown in Fig. 2. We first feed the input $\mathbf{X}$ into two 1×1 convolution layers with nonlinearization, respectively. The generated feature maps are $\mathbf{F}_1^s \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{F}_2^s \in \mathbb{R}^{C\times H\times W}$. Then, we reshape $\mathbf{F}_1^s$ to $\mathbf{F}_1^{s\prime} \in \mathbb{R}^{C\times HW}$. For $\mathbf{F}_2^s$, we apply GAP and GMP for capturing global contextual information. A multi-layer perceptron (MLP) with two layers is adopted to process the results of GAP and GMP, respectively. The MLP's hidden layer has $C/4$ output neurons for reducing the number of parameters, and the output layer has C neurons. Then, we integrate these two feature vectors using the element-wise sum and a 1×1 convolution. This can be formulated as

$$\mathbf{F}_2^{s\prime} = \mathrm{Conv}_{1\times 1}(\mathrm{MLP}(\mathrm{GAP}(\mathbf{F}_2^s)) + \mathrm{MLP}(\mathrm{GMP}(\mathbf{F}_2^s))), \tag{6}$$

where we have $\mathbf{F}_2^{s\prime} \in \mathbb{R}^{C\times 1\times 1}$. $\mathbf{F}_2^{s\prime}$ can be further reshaped to $\mathbb{R}^{C\times 1}$. The *softmax* function is used to normalize $\mathbf{F}_2^{s\prime}$ into the value range of $[0, 1]$, *i.e.,* $\mathbf{F}_2^{s\prime} = \sigma(\mathbf{F}_2^{s\prime})$.

We continue by multiplying $\mathbf{F}_1^{s\prime}$ and $\mathbf{F}_2^{s\prime}$ to allow the channel information to be embedded into spatial attention. This can be written as

$$\begin{aligned} \mathbf{F}_3^s &= \mathbf{F}_1^{s\prime} \otimes \mathbf{F}_2^{s\prime}, \\ \mathbf{F}_3^s \in \mathbb{R}^{HW\times 1}, &\mathbf{F}_1^{s\prime} \in \mathbb{R}^{HW\times C}, \mathbf{F}_2^{s\prime} \in \mathbb{R}^{C\times 1}. \end{aligned} \tag{7}$$

Then, $\mathbf{F}_3^s$ is reshaped to $\mathbf{F}_3^s \in \mathbb{R}^{1\times H\times W}$. Moreover, to enhance the spatial information, we also apply the GAP and GMP along the channel dimension of the input $\mathbf{X}$ to squeeze its number of channels to 1. Hence, we obtain two 1-channel feature maps $\mathbf{F}_{\mathrm{GAP}}^s \in \mathbb{R}^{1\times H\times W}$ and $\mathbf{F}_{\mathrm{GMP}}^s \in \mathbb{R}^{1\times H\times W}$. $\mathbf{F}_3^s$, $\mathbf{F}_{\mathrm{GAP}}^s$, and $\mathbf{F}_{\mathrm{GMP}}^s$ are concatenated, followed by a 7×7 convolution with 1-channel output for feature fusion. Finally, the *sigmoid* function is applied to obtain the final spatial attention. In summary, we can formulate these operations as

$$\mathrm{CESA}(\mathbf{X}) = \sigma(\mathrm{Conv}_{7\times 7}(\mathrm{Concat}(\mathbf{F}_3^s, \mathbf{F}_{\mathrm{GAP}}^s, \mathbf{F}_{\mathrm{GMP}}^s))), \tag{8}$$

where we have $\mathrm{CESA}(\mathbf{X}) \in \mathbb{R}^{1\times H\times W}$. Note that we do not design similar operations (denoted as $\mathbf{F}_{\mathrm{GAP}}^c \in \mathbb{R}^{C\times 1\times 1}$ and $\mathbf{F}_{\mathrm{GMP}}^c \in \mathbb{R}^{C\times 1\times 1}$) for SECA. SECA adopts convolutions to squeeze the channel dimension for deriving $\mathbf{F}_2^c$, which means that the dimension squeeze is accomplished in a learnable manner. In contrast, CESA can only adopt global pooling to squeeze the spatial dimension for deriving $\mathbf{F}_2^{s\prime}$, where the dimension squeeze is accomplished in a fixed (non-learnable) manner. The fixed dimension squeeze would make CESA sacrifice a substantial amount of spatial information, and thus, we add $\mathbf{F}_{\mathrm{GAP}}^s$ and $\mathbf{F}_{\mathrm{GMP}}^s$ to supplement $\mathbf{F}_3^s$ by more spatial information. However, SECA does not need similar operations due to its learnable dimension squeeze. Our experiments in Section IV-C4a also demonstrate our hypothesis: $\mathbf{F}_{\mathrm{GAP}}^s$ and $\mathbf{F}_{\mathrm{GMP}}^s$ can bring about improvement to the performance, while $\mathbf{F}_{\mathrm{GAP}}^c$ and $\mathbf{F}_{\mathrm{GMP}}^c$ cannot.

Next, we discuss how CESA embeds channel information into spatial attention. Through Eq. (6), we extract a channel global context vector. Through Eq. (7), the response at each position of $\mathbf{F}_3^s$ is computed by the sum of the product of all channel pairs in $\mathbf{F}_1^{s\prime}$ and $\mathbf{F}_2^{s\prime}$. In this way, each channel feature vector is absorbed using a dense correlation with a global view. Therefore, the channel information is embedded into spatial attention rather than the previous simple reduction of the number of channels through convolutions.

IV. Experiments

A. Experimental Setup

1) Implementation Details: The proposed method is implemented using the PyTorch framework. The backbone network, *i.e.,* VGG16 [50], is initialized using the ImageNet-pretrained model. We adopt the Adam optimizer to optimize the network. The learning rate policy is *poly*, in which

TABLE I

COMPARISON BETWEEN THE PROPOSED A2SPPNET AND 32 STATE-OF-THE-ART METHODS IN TERMS OF F_β (↑), MAE (↓), F_β^w (↑), AND S_m (↑) ON SIX DATASETS. THE BEST RESULT OF EACH COLUMN IS HIGHLIGHTED IN **BOLD**.

	#	Methods	SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
			F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
VGG16	1	MDF [78]	0.764	0.182	-	-	0.807	0.138	0.680	0.115	0.669	0.128	0.707	0.114
	2	LEGS [19]	0.733	0.194	0.766	0.119	0.830	0.118	0.668	0.134	0.663	0.126	0.652	0.137
	3	ELD [20]	0.758	0.154	0.837	0.074	0.866	0.081	0.700	0.092	0.726	0.095	0.727	0.092
	4	RFCN [21]	0.802	0.161	0.892	0.080	0.896	0.097	0.738	0.095	0.754	0.100	0.782	0.089
	5	DCL [22]	0.831	0.131	0.892	0.063	0.895	0.080	0.733	0.095	0.747	0.096	0.785	0.082
	6	DHS [44]	0.822	0.128	0.889	0.053	0.903	0.062	-	-	0.752	0.082	0.807	0.066
	7	NLDF [25]	0.837	0.123	0.902	0.048	0.902	0.066	0.753	0.080	0.762	0.080	0.806	0.065
	8	Amulet [23]	0.795	0.144	0.897	0.051	0.913	0.061	0.743	0.098	0.755	0.094	0.778	0.085
	9	UCF [24]	0.805	0.148	0.888	0.062	0.901	0.071	0.730	0.120	0.758	0.112	0.772	0.112
	10	SRM [26]	0.840	0.126	0.906	0.046	0.914	0.056	0.769	0.069	0.778	0.077	0.826	0.059
	11	PiCA [28]	0.836	0.102	0.916	0.042	0.923	0.049	0.766	0.068	0.783	0.083	0.837	0.054
	12	BRN [41]	0.843	0.103	0.910	0.036	0.919	0.043	0.774	0.062	0.769	0.076	0.827	0.050
	13	C2S [29]	0.819	0.122	0.898	0.046	0.907	0.057	0.759	0.072	0.775	0.083	0.811	0.062
	14	RAS [30]	0.847	0.123	0.913	0.045	0.916	0.058	0.785	0.063	0.772	0.075	0.831	0.059
	15	DSS [27]	0.842	0.122	0.913	0.041	0.915	0.056	0.774	0.066	0.770	0.074	0.827	0.056
	16	PAGE-Net [91]	0.837	0.110	0.918	0.037	0.927	0.046	0.791	0.062	0.766	0.080	0.838	0.052
	17	AFNet [101]	0.848	0.108	0.921	0.036	0.930	0.045	0.784	0.057	0.791	0.072	0.857	0.046
	18	DUCRF [79]	0.836	0.121	0.920	0.040	0.924	0.052	0.802	0.057	0.762	0.080	0.833	0.059
	19	CPD [53]	0.848	0.113	0.924	0.033	0.930	0.044	**0.794**	0.057	0.795	0.068	0.861	0.043
	20	PoolNet [89]	0.863	0.111	0.925	0.037	0.939	0.045	0.791	0.060	0.800	0.068	0.866	0.043
	21	HRSOD [56]	0.819	0.138	0.912	0.042	0.916	0.058	0.752	0.066	0.784	0.068	0.836	0.051
		A2SPPNet	**0.865**	**0.100**	**0.940**	**0.027**	**0.946**	**0.033**	0.792	**0.055**	**0.808**	**0.066**	**0.882**	**0.038**
ResNet50	22	BASNet [80]	0.849	0.112	0.928	0.032	0.938	0.040	0.805	0.056	0.783	0.073	0.859	0.048
	23	EGNet [81]	0.859	0.110	0.928	0.034	0.938	0.044	0.794	0.056	0.800	0.070	0.870	0.044
	24	F^3Net [82]	0.857	0.104	0.929	0.032	0.940	0.040	0.802	0.059	0.795	0.069	0.859	0.044
	25	GCPANet [77]	0.842	0.100	0.935	0.032	0.942	0.037	0.796	0.057	0.803	0.066	0.872	0.038
	26	LDF [83]	0.863	0.101	0.935	0.028	0.939	0.041	0.803	0.057	**0.815**	**0.064**	**0.886**	0.039
	27	ITSD [84]	0.867	0.098	0.926	0.035	0.939	0.040	0.802	0.063	0.806	0.068	0.875	0.042
	28	MINet [92]	0.842	0.099	0.929	0.032	0.937	0.040	0.780	0.057	0.808	0.066	0.870	0.040
	29	GateNet [85]	0.851	0.108	0.927	0.036	0.933	0.045	0.784	0.061	0.808	0.068	0.866	0.045
	30	PA-KRN [86]	0.870	0.095	0.938	0.028	0.943	0.038	**0.806**	0.054	0.813	**0.064**	0.885	0.037
	31	TSPOANet [87]	0.848	0.116	0.918	0.039	0.931	0.047	0.778	0.061	-	-	0.847	0.049
	32	FCSOD [88]	0.825	0.119	0.897	0.039	0.910	0.047	0.708	0.067	0.773	0.071	0.819	0.045
		A2SPPNet	**0.872**	**0.094**	**0.941**	**0.025**	**0.954**	**0.028**	0.798	**0.053**	0.808	**0.064**	**0.886**	**0.034**

	#	Methods	SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
			F_β^w	S_m	F_β^w	S_m	F_β^w	S_m	F_β^w	S_m	F_β^w	S_m	F_β^w	S_m
VGG16	1	MDF [78]	0.528	0.655	-	-	0.619	0.764	0.494	0.714	0.508	0.718	0.507	0.729
	2	LEGS [19]	0.550	0.662	0.616	0.743	0.692	0.788	0.523	0.713	0.538	0.721	0.510	0.697
	3	ELD [20]	0.634	0.707	0.743	0.820	0.783	0.841	0.593	0.750	0.621	0.764	0.607	0.733
	4	RFCN [21]	0.591	0.722	0.707	0.858	0.725	0.860	0.562	0.774	0.592	0.793	0.586	0.793
	5	DCL [22]	0.669	0.763	0.770	0.871	0.782	0.873	0.584	0.762	0.624	0.794	0.632	0.803
	6	DHS [44]	0.685	0.752	0.816	0.870	0.837	0.884	-	-	0.666	0.802	0.705	0.820
	7	NLDF [25]	0.708	0.759	0.838	0.879	0.835	0.875	0.634	0.770	0.676	0.801	0.710	0.817
	8	Amulet [23]	0.674	0.755	0.817	0.886	0.839	0.894	0.626	0.781	0.650	0.796	0.657	0.804
	9	UCF [24]	0.673	0.763	0.779	0.875	0.805	0.884	0.574	0.760	0.613	0.785	0.595	0.782
	10	SRM [26]	0.670	0.739	0.835	0.887	0.849	0.894	0.658	0.798	0.684	0.818	0.721	0.836
	11	PiCA [28]	0.721	0.787	0.847	0.905	0.862	0.914	0.691	0.826	0.688	0.823	0.745	0.860
	12	BRN [41]	0.670	0.768	0.835	0.895	0.849	0.902	0.658	0.806	0.684	0.813	0.721	0.842
	13	C2S [29]	0.700	0.757	0.835	0.889	0.849	0.896	0.663	0.799	0.685	0.812	0.717	0.832
	14	RAS [30]	0.718	0.761	0.850	0.889	0.855	0.894	0.695	0.812	0.691	0.813	0.739	0.839
	15	DSS [27]	0.711	0.747	0.862	0.881	0.864	0.884	0.688	0.790	0.702	0.805	0.752	0.826
	16	PAGE-Net [91]	0.721	0.769	0.865	0.903	0.879	0.912	0.722	0.825	0.698	0.815	0.768	0.854
	17	AFNet [101]	0.726	0.773	0.869	0.905	0.880	0.913	0.717	0.826	0.719	0.829	0.784	0.867
	18	DUCRF [79]	0.697	0.760	0.855	0.903	0.858	0.907	0.706	0.821	0.663	0.801	0.723	0.836
	19	CPD [53]	0.718	0.765	0.879	0.904	0.888	0.910	0.715	0.818	0.730	0.831	0.799	0.866
	20	PoolNet [89]	0.731	0.781	0.865	0.908	0.880	0.915	0.710	0.829	0.724	0.839	0.783	0.875
	21	HRSOD [56]	0.622	0.702	0.851	0.882	0.853	0.883	0.645	0.772	0.713	0.820	0.746	0.830
		A2SPPNet	**0.753**	**0.792**	**0.911**	**0.922**	**0.915**	**0.932**	**0.742**	**0.834**	**0.758**	**0.842**	**0.835**	**0.888**
ResNet50	22	BASNet [80]	0.728	0.766	0.889	0.909	0.898	0.916	0.741	0.836	0.721	0.823	0.802	0.865
	23	EGNet [81]	0.736	0.782	0.876	0.912	0.886	0.919	0.727	0.836	0.727	0.836	0.796	0.878
	24	F^3Net [82]	0.742	0.785	0.883	0.915	0.900	0.924	0.710	0.823	0.726	0.835	0.790	0.872
	25	GCPANet [77]	0.731	0.776	0.889	0.918	0.900	0.922	0.734	0.830	0.732	0.839	0.817	0.884
	26	LDF [83]	0.754	0.787	0.891	0.918	0.890	0.915	0.715	0.826	0.727	0.835	0.807	0.879
	27	ITSD [84]	0.764	0.791	0.881	0.906	0.897	0.914	0.734	0.829	0.739	0.836	0.813	0.887
	28	MINet [92]	0.740	0.781	0.889	0.912	0.899	0.919	0.719	0.822	0.742	0.838	0.812	0.874
	29	GateNet [85]	0.729	0.774	0.872	0.910	0.881	0.917	0.703	0.821	0.733	0.838	0.785	0.870
	30	PA-KRN [86]	**0.765**	0.785	0.903	0.920	0.901	0.916	**0.752**	0.830	0.752	0.839	0.827	0.884
	31	TSPOANet [87]	0.718	0.774	0.862	0.907	0.876	0.913	0.697	0.823	-	-	0.767	0.865
	32	FCSOD [88]	0.668	0.721	0.856	0.872	0.868	0.879	0.618	0.745	0.719	0.809	0.756	0.822
		A2SPPNet	0.758	**0.798**	**0.914**	**0.924**	**0.929**	**0.933**	**0.752**	**0.838**	**0.763**	**0.844**	**0.846**	**0.891**

TABLE II
COMPARISON BETWEEN THE PROPOSED A2SPPNET AND 32 STATE-OF-THE-ART METHODS IN TERMS OF E_m (↑) ON SIX DATASETS.

	Methods	SOD	HKU-IS	ECSSD	DUT-OMRON	THUR15K	DUTS-test
VGG16	MDF [78]	0.607	-	0.535	0.442	0.470	0.433
	LEGS [19]	0.724	0.776	0.814	0.652	0.686	0.670
	ELD [20]	0.742	0.754	0.792	0.641	0.661	0.650
	RFCN [21]	0.267	0.216	0.241	0.224	0.228	0.218
	DCL [22]	0.527	0.448	0.464	0.461	0.429	0.408
	DHS [44]	0.771	0.802	0.808	-	0.672	0.700
	NLDF [25]	0.820	0.885	0.881	0.735	0.740	0.766
	Amulet [23]	0.608	0.710	0.719	0.542	0.594	0.558
	UCF [24]	0.565	0.577	0.613	0.416	0.420	0.367
	SRM [26]	0.797	0.801	0.812	0.677	0.661	0.691
	PiCA [28]	0.696	0.668	0.688	0.580	0.566	0.589
	BRN [41]	0.824	0.938	0.936	0.843	0.823	0.886
	C2S [29]	0.814	0.843	0.860	0.730	0.721	0.752
	RAS [30]	0.803	0.851	0.858	0.757	0.727	0.756
	DSS [27]	0.832	0.938	0.928	0.836	0.815	0.872
	PAGE-Net [91]	0.800	0.851	0.866	0.777	0.739	0.791
	AFNet [101]	0.814	0.839	0.849	0.760	0.735	0.767
	DUCRF [79]	0.751	0.754	0.761	0.803	0.646	0.658
	CPD [53]	0.843	0.915	0.921	0.830	0.794	0.860
	PoolNet [89]	0.795	0.821	0.823	0.717	0.703	0.739
	HRSOD [56]	0.778	0.801	0.826	0.701	0.683	0.717
	A2SPPNet	**0.851**	**0.954**	**0.956**	**0.855**	**0.846**	**0.914**
ResNet50	BASNet [80]	0.832	0.936	0.938	**0.857**	0.815	0.883
	EGNet [81]	0.806	0.843	0.843	0.736	0.721	0.754
	F^3Net [82]	**0.851**	0.895	0.902	0.783	0.769	0.820
	GCPANet [77]	0.850	0.915	0.918	0.815	0.799	0.863
	LDF [83]	0.827	0.831	0.816	0.712	0.710	0.751
	ITSD [84]	0.870	0.927	0.929	0.823	0.809	0.878
	MINet [92]	0.844	0.901	0.906	0.790	0.783	0.838
	GateNet [85]	0.834	0.856	0.863	0.749	0.748	0.782
	PA-KRN [86]	0.850	0.907	0.903	0.772	0.758	0.832
	TSPOANet [87]	0.819	0.859	0.869	0.753	-	0.777
	FCSOD [88]	0.790	0.936	0.932	0.795	0.836	0.886
	A2SPPNet	0.847	**0.955**	**0.958**	0.855	**0.854**	**0.920**

the current learning rate equals the base rate multiplied by $(1 - curr\_iter/max\_iter)^{power}$. We set the initial learning rate to 1e-4 and the $power$ to 0.9. The weight decay is set to 1e-4. We train our network for 50 epochs in total with a batch size of 16. All experiments are performed on a TITAN Xp GPU.

2) Datasets: Following recent studies [26], [28], [38], [41], [52]–[54], [56], [59], [89], [101], we utilize the DUTS training set [103] to train A2SPPNet. The DUTS training set consists of 10553 images with corresponding pixel-wise saliency annotations. For the performance evaluation, we use the DUTS test set and five other widely used datasets, including SOD [104], HKU-IS [78], ECSSD [105], DUT-OMRON [106], and THUR15K [107]. These six test datasets contain 5019, 300, 4447, 1000, 5168, and 6232 natural complicated images, respectively, with high-quality human labels.

3) Evaluation Criteria: In this paper, we evaluate the performance of SOD models using four widely used evaluation metrics, including the max F-measure score F_β, mean absolute error (MAE), weighted F-measure score F_β^ω, structure-measure S_m, and enhanced alignment-measure E_m.

For the computation of max F-measure, we first convert the predicted saliency maps into binary maps under varying thresholds in the range of $[0, 1]$. Then, we compare these binary maps with the ground truth to compute a series of precision-recall value pairs. Based on these precision-recall value pairs, we provide an overall performance evaluation metric, *i.e.,* F-measure score F_β, which is the weighted harmonic mean of the precision and recall. The formula of the F-measure score is

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \tag{9}$$

where β^2 is typically set to 0.3, as in previous work [23], [26]–[28], [37], [38], [40], [41], [52]–[54], [56], [59], [89], [101], to emphasize more the precision than the recall. We calculate F_β scores under all thresholds and report the best score based on an optimal threshold.

The MAE metric is used to measure the absolute error between a saliency map and the corresponding ground truth. MAE can be computed as

$$\text{MAE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |\mathbf{P}(i,j) - \mathbf{G}(i,j)|, \tag{10}$$

where $\mathbf{P}$ and $\mathbf{G}$ represent the predicted and ground-truth saliency maps which are normalized to [0, 1], respectively. H and W are the image height and width, respectively. $\mathbf{P}(i,j)$ / $\mathbf{G}(i,j)$ denotes the saliency score at location (i,j).

Moreover, Margolin *et al.* [108] proposed the weighted F-measure score F_β^ω, which is defined as

$$F_\beta^\omega = \frac{(1+\beta^2) \times \text{Precision}^\omega \times \text{Recall}^\omega}{\beta^2 \times \text{Precision}^\omega + \text{Recall}^\omega}, \tag{11}$$

in which the weighted precision Precision^ω and weighted recall Recall^ω are defined in [108]. The meaning of β^2 is the same as that in Eq. (9). In addition to the above metrics, which are based on pixel-wise errors, we also report structure-measure S_m [109] to simultaneously evaluate region-aware and object-aware structural similarities. S_m is calculated as

$$S_m = \alpha S_o + (1-\alpha) S_r, \tag{12}$$

where S_o and S_r are object-aware and region-aware structural similarities, respectively. The balance parameter α is set to 0.5 by default. Besides, we also use enhanced alignment-measure E_m, which is designed in the binary map evaluation field to jointly capture image-level statistics and local pixel matching information [110].

Other than numerical results, we also perform non-numerical evaluation to compare our method to baselines, including F-measure *vs.* threshold curves (FT curves) and the precision *vs.* recall curves (PR curves). FT curves show the F-measure scores at various thresholds and can thus suggest the quality of the predicted saliency maps clearly. The PR curves can summarize the compromise between precision and recall.

B. Performance Comparison

We compare the proposed A2SPPNet with 32 previous state-of-the-art methods, including MDF [78], LEGS [19], ELD [20], RFCN [21], DCL [22], DHS [44], NLDF [25], Amulet [23], UCF [24], SRM [26], PiCA [28], BRN [41], C2S [29], RAS [30], DSS [27], PAGE-Net [91], AFNet [101], DUCRF [79], HRSOD [56], CPD [53], BASNet [80], PoolNet [89], EGNet [81], F^3Net [82], GCPANet [77], LDF [83], ITSD [84], MINet [92], GateNet [85], PA-KRN [86], TSPOANet [87], and FCSOD [88]. For fair comparisons, the predicted saliency maps of all of these methods are provided by the

TABLE III
COMPARISON OF A2SPPNET TO 10 RECENT COMPETITIVE METHODS IN TERMS OF PARAMETERS, FLOPS, AND RUNTIME. A2SPPNET† AND A2SPPNET‡ REPRESENT THE PROPOSED A2SPPNET WITH VGG16 AND RESNET50 BACKBONES, RESPECTIVELY.

Methods	Publication	#Param (M)	FLOPs (G)	Time (s)
BASNet [80]	CVPR'2019	85.02	508.99	0.091
PoolNet [89]	CVPR'2019	66.66	194.27	0.044
EGNet [81]	ICCV'2019	105.54	627.75	0.127
F^3Net [82]	AAAI'2020	24.94	34.68	0.023
GCPANet [77]	AAAI'2020	63.53	138.96	0.030
LDF [83]	CVPR'2020	49.12	65.60	0.086
ITSD [84]	CVPR'2020	16.68	181.65	0.052
MINet [92]	CVPR'2020	46.45	374.55	0.077
GateNet [85]	ECCV'2020	125.62	288.07	0.068
PA-KRN [86]	AAAI'2021	96.46	453.35	0.101
A2SPPNet†	-	22.19	107.05	0.091
A2SPPNet‡	-	38.70	31.27	0.074

authors online or generated by their released code with default settings. Moreover, we do not report the performance of MDF [78] on the HKU-IS [78] dataset, because MDF adopts HKU-IS for training.

1) Quantitative Evaluation: The numeric comparisons with respect to F_β, MAE, F_β^ω, and S_m on six datasets are summarized in Table I. The comparison results of E_m on six datasets are shown in Table II. We report the results of A2SPPNet with VGG16 [50] and ResNet50 [51] backbones. From Table I and Table II, it can be clearly seen that our A2SPPNet significantly outperforms other competitors in almost all cases. Among all methods, the ResNet50 version of A2SPPNet achieves the best F_β values of 87.2%, 94.1%, 95.4%, and 88.6% on SOD, HKU-IS, ECSSD, and DUTS-test datasets, respectively. For DUT-OMRON and THUR15K datasets, A2SPPNet achieves comparable results to the best methods. In terms of the metrics MAE and S_m, both the VGG16 and ResNet50 versions of A2SPPNet perform best on all six datasets. In terms of the metric F_β^ω, both the VGG16 and ResNet50 versions of A2SPPNet attain the best performance on six datasets except for the ResNet50 version on the SOD dataset. In terms of the metric E_m, both the VGG16 and ResNet50 versions of A2SPPNet achieve the best performance in all cases except for the ResNet50 version on the SOD and DUT-OMRON datasets. Note that even for those cases where A2SPPNet does not perform best, A2SPPNet only achieves slightly worse performance than the best one. The reason why A2SPPNet does not improve the performance on the SOD dataset significantly may be that the SOD dataset contains many challenging scenarios which require more complicated models to resolve. Another reason may be that the SOD dataset only has 300 images and the performance has been saturated. Therefore, we can come to the conclusion that A2SPPNet sets a new state-of-the-art .

2) FT curves and PR curves: We display the FT curves and PR curves of A2SPPNet and other state-of-the-art methods on six datasets in Fig. 3. The higher the curve is, the better the performance. As can be seen, the proposed A2SPPNet favorably outperforms the other counterparts.

3) Complexity Analysis: Table III summarizes the comparison of A2SPPNet to 10 recent competitive methods, in terms of the number of parameters, the number of FLOPs, and the runtime. It can be observed that A2SPPNet has a relatively smaller number of parameters compared with the other counterparts. In addition, the ResNet50-based A2SPPNet has the smallest number of FLOPs, *i.e.,* the lowest computational complexity. At the same time, the speed of A2SPPNet is comparable to that of the others.

4) Qualitative Evaluation: To further explicitly show the effectiveness of the proposed A2SPPNet, we show the qualitative comparison between A2SPPNet and 12 state-of-the-art methods in Fig. 4. We select some representative images from the above datasets to incorporate a variety of difficult circumstances, including complicated scenes, large objects, salient objects with thin structures, multiple objects with various sizes, low contrast between foreground and background, and confusing background, from top to bottom. Generally, it can be seen that our method can successfully segment the objects with fine details, leading to better saliency predictions in various scenarios.

C. Ablation Studies

In this part, we conduct a series of ablation experiments to verify the effectiveness of the components of the proposed A2SPPNet. All variants use the VGG16 [50] backbone.

1) Effect of Component Modules: We start with the simple U-shaped encoder-decoder structure with skip connections, which uses two 3×3 convolution layers with nonlinearization to connect adjacent sides of the decoder (the 1^{st} line of Table IV). This configuration is viewed as the baseline, and the other settings remain the default.

a) Effect of multi-scale learning: On top of the baseline, to prove the effectiveness of multi-scale learning, we replace the convolution blocks of the top two decoder sides with typical ASPP modules. The results are shown in the 2^{nd} line of Table IV. It can be seen that the typical ASPP module leads to a significant performance advancement, which shows the importance of multi-scale learning. Intuitively, there exist various scales in different SOD images and various aspect ratios of different object-parts in the same image, and thus, multi-scale learning is needed for better SOD.

b) Effect of the A2SPP module: To prove that the proposed A2SPP module is better than the typical ASPP module, we continue by replacing ASPP with the A2SPP module. The results are shown in the 3^{rd} line of Table IV. It is clear that A2SPP can significantly improve the performance, demonstrating that A2SPP is more effective than ASPP and the ordinary convolution. Hence, as mentioned above, A2SPP can improve the capability of multi-scale learning by resolving the improper assumption of ASPP that all feature scales are of equal importance. A2SPP is designed to automatically enhance the essential information and suppress the noisy information of various scales. In this way, A2SPP significantly improves SOD, which heavily relies on multi-scale learning.

c) Effect of the CIEA module: Based on the model in Section IV-C1b, we replace the ordinary convolutions at three bottom decoder sides with CIEA. From the results in the 4^{th} line of Table IV, it can be seen that introducing CIEA can consistently improve SOD performance in almost

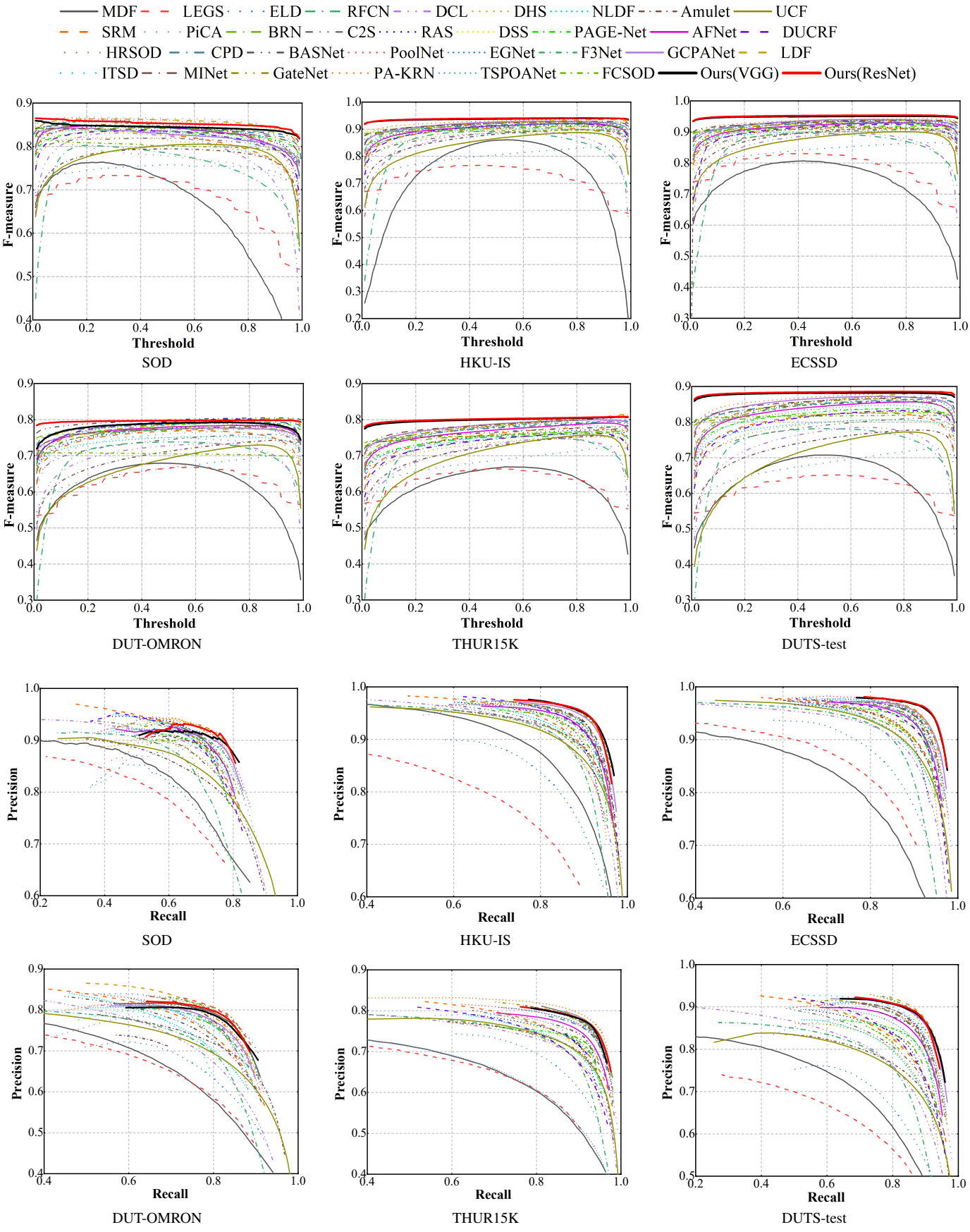


Fig. 3. FT curves (top two lines) and PR curves (bottom two lines) of A2SPPNet and 32 state-of-the-art methods on six datasets. A2SPPNet performs favorably against all other competitors. Note that the F-measure in the FT curves refers to F_β.

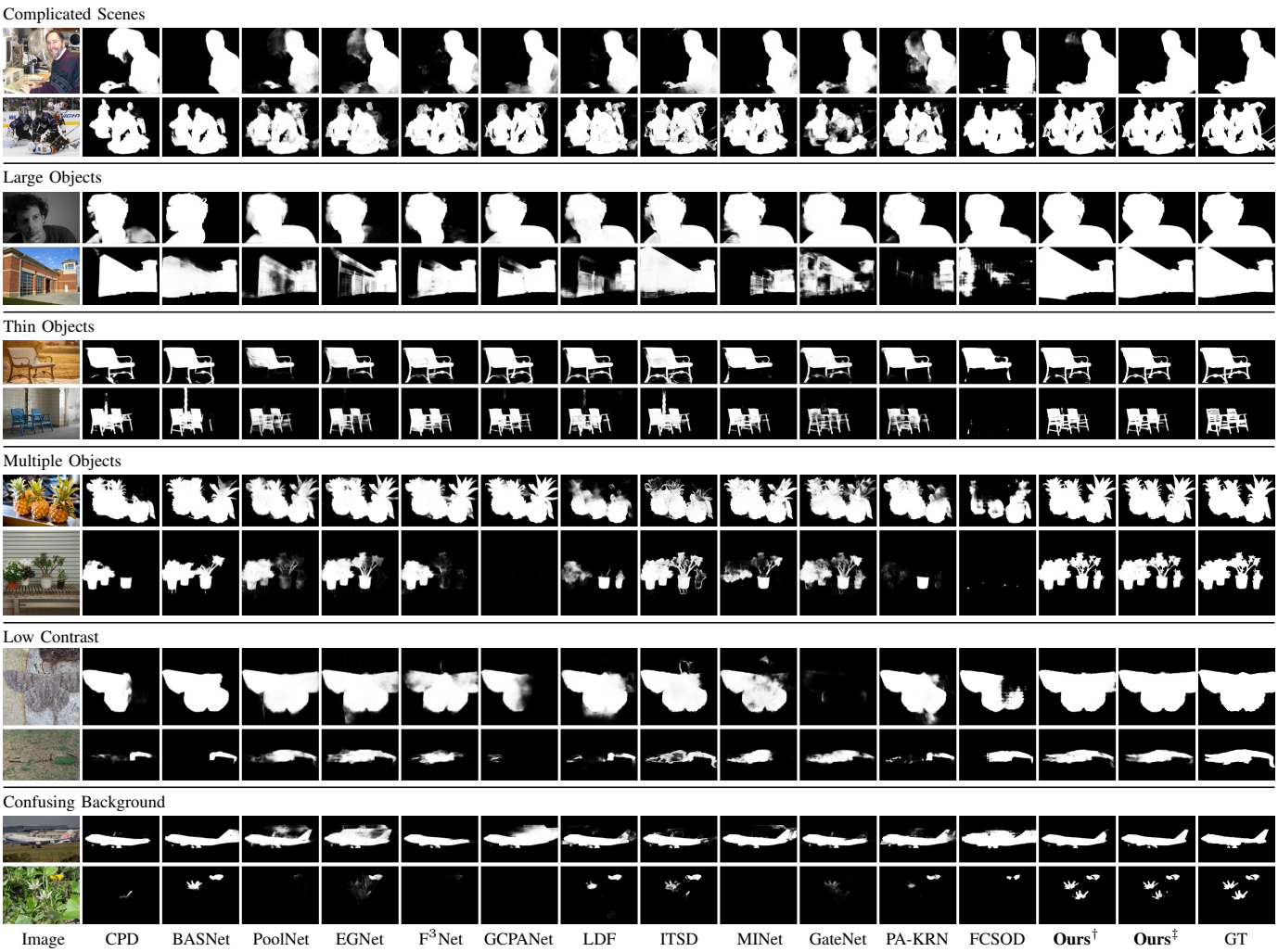


Fig. 4. Qualitative comparison between A2SPPNet and 12 recent state-of-the-art methods. "Ours†" and "Ours‡" represent the proposed A2SPPNet with VGG16 and ResNet50 backbones, respectively. GT: Ground truth.

TABLE IV
EFFECT OF THE MAIN COMPONENTS OF A2SPPNET.

#	Conv	ASPP	A2SPP	CIEA	SGL	All CIEA	All A2SPP	SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
								F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
1	✔							0.810	0.125	0.905	0.041	0.910	0.057	0.713	0.079	0.774	0.075	0.813	0.055
2		✔						0.842	0.114	0.923	0.037	0.934	0.047	0.765	0.065	0.793	0.073	0.850	0.048
3			✔					0.851	0.107	0.930	0.034	0.938	0.043	0.771	0.062	0.794	0.068	0.864	0.044
4			✔	✔				0.860	0.103	0.936	0.028	0.944	0.035	**0.792**	0.056	0.801	0.067	0.880	0.040
5			✔	✔	✔			**0.865**	**0.100**	**0.940**	**0.027**	**0.946**	**0.033**	**0.792**	**0.055**	**0.808**	**0.066**	**0.882**	**0.038**
6			✔	✔	✔	✔		0.860	0.105	0.930	0.032	0.939	0.041	0.780	0.060	0.802	0.068	0.871	0.042
7			✔	✔	✔		✔	0.845	0.104	0.927	0.032	0.939	0.038	0.756	0.058	0.804	0.067	0.858	0.040

all cases, which suggests the effectiveness of adding attention to the bottom decoder sides. CNN feature maps, especially those from bottom decoder sides, contain a substantial amount of noisy activation. It has been proved that the attention mechanism can effectively suppress the noisy activation and enhance the necessary activation, thus improving the SOD performance.

d) Effect of the SGL technique: This paper follows PoolNet [89] to deliver the top global information to each decoder side for informing each decoder side about where salient objects are and what salient objects look like. Such an SGL technique can thus guide the learning of the decoder. To evaluate the effectiveness of SGL, we add the SGL technique in the 5^{th} line of Table IV. The comparison between the models with SGL and without SGL demonstrates the superiority of the SGL technique for performance improvement.

e) Effect of the two-part decoder: In this paper, we place A2SPP at the partial top decoder sides, and for the bottom sides, we place the CIEA modules. To test the effectiveness of this two-part decoder design, we conduct two ablation studies: (i) we place CIEA at all decoder sides; (ii) we place A2SPP at all decoder sides. The evaluation results of these two models

TABLE V
ABLATION STUDIES FOR THE HYPER-PARAMETERS OF A2SPPNET.

Configurations		SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
		F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
Default Configurations		0.865	**0.100**	**0.940**	**0.027**	**0.946**	**0.033**	**0.792**	**0.055**	**0.808**	**0.066**	**0.882**	0.038
#Channels of decoder	(256, 128, 64, 32, 16)	0.862	0.102	0.935	0.030	0.939	0.035	0.787	0.058	0.805	0.072	0.874	0.040
	(256, 256, 128, 64, 16)	0.863	0.109	0.934	0.029	0.942	0.037	0.777	0.057	0.805	0.068	0.872	0.038
	(128, 128, 64, 32, 16)	0.854	0.110	0.935	0.028	0.944	0.034	0.781	0.056	0.807	0.068	0.874	0.039
	(128, 64, 32, 16, 8)	0.856	0.108	0.935	0.029	0.944	0.034	0.781	0.056	0.803	0.069	0.875	**0.037**
Dilation rates of A2SPP	(1, 2, 4), (1, 2, 4, 8)	0.857	0.104	0.933	0.029	0.942	0.036	0.774	0.059	0.805	0.067	0.871	0.039
	(1, 4, 8), (1, 4, 8, 12)	0.861	0.105	0.936	0.029	0.945	0.035	0.787	0.056	**0.808**	**0.066**	0.875	0.039
	(1, 3, 6), (1, 3, 6)	0.853	0.111	0.937	0.028	0.944	0.036	0.784	0.057	0.803	0.070	0.875	0.039
	(1, 3, 6, 12), (1, 3, 6, 12)	0.857	0.107	0.933	0.031	0.945	0.035	0.777	0.059	0.807	0.070	0.874	0.039
#Channels of SGL	((128, 64, 32, 16, 8), 8)	0.856	0.107	0.936	0.028	0.945	0.034	0.777	0.058	0.807	0.070	0.877	**0.037**
	((32, 16, 8, 4, 2), 8)	0.857	0.112	0.926	0.033	0.941	0.038	0.764	0.061	0.804	0.069	0.873	0.041
	((64, 32, 16, 8, 4), 16)	0.853	0.108	0.934	0.029	0.944	0.034	0.773	0.058	0.801	0.070	0.871	0.039
	((64, 32, 16, 8, 4), 4)	**0.866**	0.103	0.937	0.029	0.945	0.034	0.789	0.058	0.806	0.070	0.875	0.039

* "#Channels of decoder" means the number of channels of the decoder. Note that for the bottom sides, these numbers refer to the number of channels of the CIEA modules; otherwise, for A2SPP modules. "Dilation rates of A2SPP" represents the dilation rates of atrous convolutions in the A2SPP modules at top two decoder sides. "#Channels of SGL" refers to the number of SGL channels for each decoder side and the final saliency prediction.

are displayed in the 6^{th} and 7^{th} lines of Table IV. It can be observed that the default A2SPPNet performs better than the other versions, which suggests the effectiveness of the design of the two-part decoder. The reason for this phenomenon is that A2SPP inherits the natural property of ASPP in learning multi-scale high-level semantic representations that exist in top CNN layers for locating salient objects coarsely [32], [93], [98]. In contrast, the bottom layers aim at learning low-level fine-grained features for refining object details, and thus, A2SPP appears to be meaningless here, especially considering that A2SPP is unsuitable for processing large feature maps at the bottom sides, as discussed in Section III-A.

f) Visualization of ablation designs: Fig. 5 displays some feature visualization figures and the corresponding saliency maps of various ablation designs to show how features evolve in A2SPP. The baseline is the simple U-shaped encoder-decoder network with skip connections (the 1^{st} line of Table IV). As shown in Fig. 5(b), the baseline method can obtain only the rough locations of salient objects. From Fig. 5(c), it can be seen that ASPP can markedly refine the shapes of salient objects, which suggests the importance of multi-scale learning in SOD. We continue by replacing all CIEA modules in A2SPPNet with SECA and CESA, respectively. As depicted in Fig. 5(d) and Fig. 5(e), both SECA and CESA can improve the quality of saliency maps, which demonstrates the effectiveness of these two attention modules. Fig. 5(f) illustrates the features and saliency maps produced by the default A2SPPNet with CIEA, where better results are observed than with the separate usage of SECA or CESA. Hence, A2SPPNet with CIEA can discover salient objects and refine object details better.

2) Impact of Hyper-parameters: In this part, we study the impact of various hyper-parameters of A2SPPNet and explain why the default hyper-parameters are set.

a) Numbers of channels of the decoder: Table V displays the results when choosing different numbers of channels for the A2SPP and CIEA modules. A2SPPNet appears to be robust to the changes in the numbers of decoder channels. The default setting achieves a bit better performance. Thus, we set the default numbers of channels to (128, 128, 64, 16, 16) from top to bottom in the decoder.

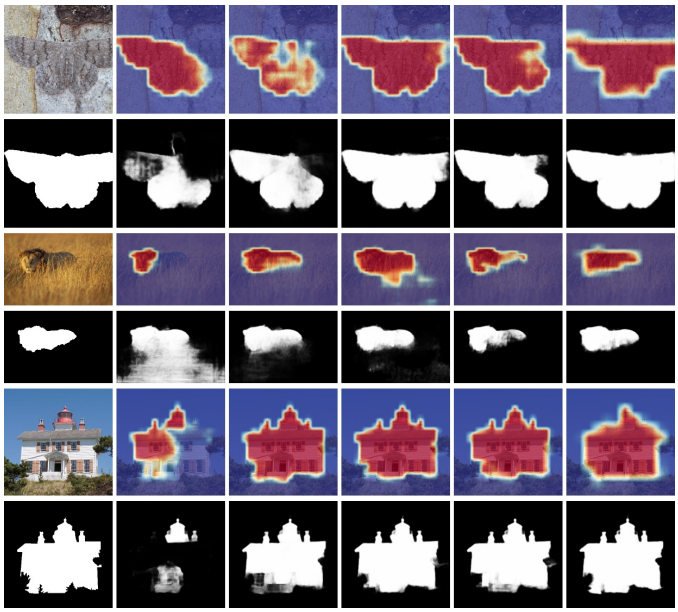


Fig. 5. Feature visualization maps and saliency maps of various ablation designs. (a) Image and ground truth; (b) Baseline; (c) Baseline + ASPP; (d) A2SPPNet w/ SECA; (e) A2SPPNet w/ CESA; (f) A2SPPNet w/ CIEA.

b) Dilation rates of the A2SPP module: The dilation rates of atrous convolutions can determine the receptive field sizes of A2SPP. We summarize the results when using different dilation rates for A2SPP in Table V. The results still appear to be robust, and the default setting, *i.e.,* (1, 3, 6) and (1, 3, 6, 9), performs best.

c) Numbers of SGL channels: The proposed SGL technique plays an essential role in guiding the learning of the decoder, and thus, we conduct an ablation study on the numbers of output channels of SGL modules. The evaluation results are displayed in Table V. The default numbers of SGL channels in this paper are $((64, 32, 16, 8, 4), 8)$. It can be observed that the default configuration of A2SPPNet outperforms other baselines slightly.

TABLE VI
ABLATION STUDIES FOR A2SPPNET WITH DIFFERENT ATTENTION MECHANISMS.

Configurations	SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
A2SPPNet	**0.865**	**0.100**	**0.940**	**0.027**	**0.946**	**0.033**	0.792	**0.055**	**0.808**	**0.066**	**0.882**	**0.038**
A2SPPNet$_{\mathrm{SE}}$	0.854	0.103	0.932	0.031	0.935	0.039	0.785	0.056	0.804	0.067	0.871	0.041
A2SPPNet$_{\mathrm{BAM}}$	0.855	0.105	0.934	0.030	0.936	0.038	0.792	0.058	0.805	0.068	0.872	0.040
A2SPPNet$_{\mathrm{CBAM}}$	0.848	0.107	0.931	0.031	0.938	0.038	0.783	0.059	0.803	0.069	0.868	0.041
A2SPPNet$_{\mathrm{GC}}$	0.864	0.104	0.935	0.031	0.942	0.037	**0.801**	0.060	0.804	0.068	0.878	0.041

TABLE VII
ABLATION STUDIES FOR SOME DESIGN CHOICES OF A2SPPNET.

Configurations	SOD		HKU-IS		ECSSD		DUT-OMRON		THUR15K		DUTS-test	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
A2SPPNet	0.865	0.100	0.940	0.027	0.946	0.033	0.792	0.055	0.808	0.066	0.882	0.038
A2SPPNet$_{\mathrm{SECA+}}$	0.862	0.102	0.936	0.030	0.946	0.035	0.795	0.060	0.807	0.067	0.877	0.041
A2SPPNet$_{\mathrm{CESA-}}$	0.863	0.097	0.934	0.031	0.943	0.035	0.790	0.056	0.804	0.067	0.881	0.043
A2SPPNet$_{\mathrm{Add}}$	0.865	0.099	0.936	0.030	0.946	0.034	0.789	0.059	0.803	0.068	0.876	0.040
A2SPPNet$_{\mathrm{SGL}}$	0.864	0.098	0.934	0.031	0.946	0.035	0.792	0.058	0.811	0.067	0.878	0.042

3) Comparison to Existing Attention Modules: The proposed CIEA expands the traditional attention mechanisms by introducing spatial and channel information dependencies in the channel and spatial attention calculations, respectively. One could wonder about the superiority of CIEA when compared to existing attention mechanisms. To accomplish this goal, we replace all CIEA modules in A2SPPNet with existing SE [62], BAM [63], CBAM [64], and GC [65] attention modules. The comparison results are displayed in Table VI. It can be observed that CIEA consistently outperforms existing attention modules, which implies the superiority of CIEA in SOD.

4) Validation of Some Design Choices: Here, we validate some design choices of A2SPPNet.

a) $\mathbf{F}^s_{\mathrm{GAP}}$ *and* $\mathbf{F}^s_{\mathrm{GMP}}$ *in CESA:* In Eq. (8), we add $\mathbf{F}^s_{\mathrm{GAP}}$ and $\mathbf{F}^s_{\mathrm{GMP}}$ to CESA to enhance the spatial information. One can ask why we do not add similar $\mathbf{F}^c_{\mathrm{GAP}}$ and $\mathbf{F}^c_{\mathrm{GMP}}$ to SECA. As discussed in Section III-C2, the reason is that CESA can only use global pooling (*i.e.,* GAP and GMP) to squeeze the spatial dimension, leading to the loss of spatial information. In contrast, SECA uses 1×1 convolutions to squeeze the channel dimension, *i.e.,* in a learnable manner. Hence, $\mathbf{F}^c_{\mathrm{GAP}}$ and $\mathbf{F}^c_{\mathrm{GMP}}$ appear to be unnecessary for SECA, because the fixed dimension squeeze of GAP and GMP would not further improve the learnable squeeze. To verify this hypothesis, we conduct two ablation studies: i) adding $\mathbf{F}^c_{\mathrm{GAP}}$ and $\mathbf{F}^c_{\mathrm{GMP}}$ to SECA; ii) removing $\mathbf{F}^s_{\mathrm{GAP}}$ and $\mathbf{F}^s_{\mathrm{GMP}}$ from CESA. The results are provided in Table VII. It can be seen that A2SPPNet achieves almost the same results with or without adding $\mathbf{F}^c_{\mathrm{GAP}}$ and $\mathbf{F}^c_{\mathrm{GMP}}$ to SECA. A2SPPNet consistently performs slightly worse when removing $\mathbf{F}^s_{\mathrm{GAP}}$ and $\mathbf{F}^s_{\mathrm{GMP}}$ from CESA.

b) Aggregation strategy of SECA and CESA: For the aggregation of SECA and CESA in Eq. (3), there are two natural combining methods, *i.e.,* element-wise multiplication and summation. A2SPPNet utilizes element-wise multiplication as the default in Eq. (3). Here, we evaluate the element-wise summation, as shown in Table VII. It can be seen that these two choices attain similar performance, and the multiplication performs slightly better. Hence, we empirically use multiplication.

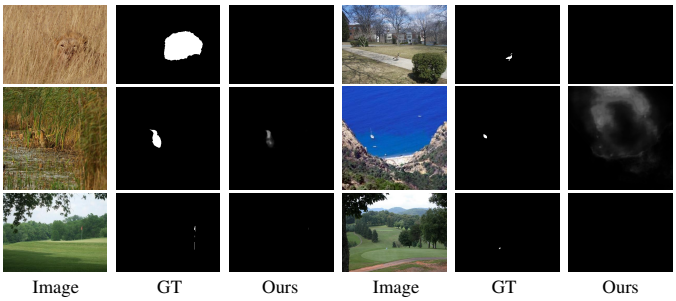


Fig. 6. Some failure cases of the proposed method. GT: Ground truth.

c) ASPP in the SGL technique: For the SGL technique, we apply the ASPP module rather than the proposed A2SPP. Compared to ASPP, A2SPP adds an automatic selection of feature scales. As discussed in Section III-A, this addition would be unnecessary for SGL, because different network sides prefer different feature scales [99], [100] and a specific scale selection of A2SPP cannot satisfy all sides. Here, we replace the ASPP module in SGL with our A2SPP. As shown in Table VII, the experimental results with A2SPP are slightly worse, which validates the above hypothesis.

d) Information-embedding attention: The CIEA module embeds spatial and channel information dependencies in the channel and spatial attention calculations, respectively. To validate this design, we remove these dependencies from CIEA, and CIEA will degenerate into BAM [63], which also combines channel and spatial attention for producing a 3D attention map. Then, we replace all CIEA modules in A2SPPNet with the BAM module [63]. The results have been shown in Table VI. The consistent performance degradation in all cases confirms that it is essential to introduce the information dependencies of one dimension when computing the attention of the other dimension.

5) Failure Case Analysis: As our method is not oracle, it also has some failure examples. We show some failure predictions of our method in Fig. 6. As can be seen, our

method may fail for inconspicuous salient objects and tiny objects. We argue that these scenarios are also very challenging for other SOD methods. Hence, there is still a long way towards the ideal SOD solution.

V. Conclusion

In this paper, we propose the Attentive Atrous Spatial Pyramid Pooling (A2SPP) module for better multi-scale learning by adding a novel Cubic Information-Embedding Attention (CIEA) module at each branch of ASPP [32], [93]. CIEA can model the channel- and spatial-wise dependency of the 3D feature map and generate a 3D attention map with the same size as the input feature map. In this way, A2SPP can automatically learn a combination of features from various scales, for each point in the 3D feature map. Since each point learns its preference for multi-scale features, A2SPP outperforms ASPP for multi-scale learning, as demonstrated by our ablation studies. Considering the different characteristics of high-level and low-level features, we add A2SPP at the top two decoder sides and place the CIEA at the bottom three sides to build our salient object detector, namely, A2SPPNet. Extensive experiments demonstrate that A2SPP is more effective than traditional ASPP, and that A2SPPNet can significantly improve the SOD performance when compared to previous state-of-the-art methods.

References

[1] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 37, no. 3, pp. 569–582, 2015.

[2] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 2083–2090.

[3] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014, pp. 2814–2821.

[4] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2009, pp. 1007–1013.

[5] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process. (TIP)*, vol. 22, no. 1, pp. 363–376, 2013.

[6] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 40, no. 1, pp. 20–33, 2017.

[7] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: finding approximately repeated scene elements for image editing," *ACM Trans. Graph. (TOG)*, vol. 29, no. 4, pp. 83:1–83:8, 2010.

[8] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 2232–2239.

[9] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)*, vol. 24, no. 5, pp. 769–779, 2013.

[10] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7268–7277.

[11] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, "Leveraging instance-, image- and dataset-level information for weakly supervised instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2020.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 3431–3440.

[13] Q. Ren, S. Lu, J. Zhang, and R. Hu, "Salient object detection by fusing local and global contexts," *IEEE Trans. Multimedia (TMM)*, vol. 23, pp. 1442–1453, 2020.

[14] Y. Tang and X. Wu, "Salient object detection using cascaded convolutional neural networks and adversarial learning," *IEEE Trans. Multimedia (TMM)*, vol. 21, no. 9, pp. 2237–2247, 2019.

[15] G. Ma, C. Chen, S. Li, C. Peng, A. Hao, and H. Qin, "Salient object detection via multiple instance joint re-learning," *IEEE Trans. Multimedia (TMM)*, vol. 22, no. 2, pp. 324–336, 2019.

[16] K. Fu, Q. Zhao, and I. Y.-H. Gu, "Refinet: A deep segmentation assisted refinement network for salient object detection," *IEEE Trans. Multimedia (TMM)*, vol. 21, no. 2, pp. 457–469, 2018.

[17] J. Li, Z. Pan, Q. Liu, and Z. Wang, "Stacked U-shape network with channel-wise attention for salient object detection," *IEEE Trans. Multimedia (TMM)*, vol. 23, pp. 1397–1409, 2020.

[18] M. Nawaz and H. Yan, "Saliency detection using deep features and affinity-based robust background subtraction," *IEEE Trans. Multimedia (TMM)*, vol. 23, pp. 2902–2916, 2020.

[19] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 3183–3192.

[20] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 660–668.

[21] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 825–841.

[22] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 478–487.

[23] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 202–211.

[24] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 212–221.

[25] Z. Luo, A. K. Mishra, A. Achkar, J. A. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 6609–6617.

[26] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 4019–4028.

[27] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 41, no. 4, pp. 815–828, 2019.

[28] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 3089–3098.

[29] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, "Contour knowledge transfer for salient object detection," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 355–370.

[30] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 234–250.

[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 2117–2125.

[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2017.

[33] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 2881–2890.

[34] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang, "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 41, no. 8, pp. 1939–1946, 2019.

[35] Y. Liu, P.-T. Jiang, V. Petrosyan, S.-J. Li, J. Bian, L. Zhang, and M.-M. Cheng, "DEL: Deep embedding learning for efficient image segmentation." in *Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 864–870.

[36] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J. Bian, and D. Tao, "Semantic edge detection with diverse deep supervision," *Int. J. Comput. Vis. (IJCV)*, vol. 130, pp. 179–198, 2022.

[37] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 714–722.

[38] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji, "Learning to promote saliency detectors," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 1644–1653.

[39] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 1711–1720.

[40] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 1741–1750.

[41] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 3127–3135.

[42] M. A. Islam, M. Kalash, and N. D. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7142–7150.

[43] X. Chen, A. Zheng, J. Li, and F. Lu, "Look, perceive and segment: Finding the salient objects in images via two-stream fixation-semantic CNNs," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1050–1058.

[44] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 678–686.

[45] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, "Lightweight salient object detection via hierarchical visual perception learning," *IEEE Trans. Cybernetics*, vol. 51, no. 9, pp. 4439–4449, 2021.

[46] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "SAM-Net: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process. (TIP)*, vol. 30, pp. 3804–3814, 2021.

[47] Y.-H. Wu, Y. Liu, L. Zhang, W. Gao, and M.-M. Cheng, "Regularized densely-connected pyramid network for salient instance segmentation," *IEEE Trans. Image Process. (TIP)*, vol. 30, pp. 3897–3907, 2021.

[48] Y. Liu, J. Han, Q. Zhang, and C. Shan, "Deep salient object detection with contextual information guidance," *IEEE Trans. Image Process. (TIP)*, vol. 29, pp. 360–374, 2019.

[49] Q. Zhang, Z. Huo, Y. Liu, Y. Pan, C. Shan, and J. Han, "Salient object detection employing a local tree-structured low-rank representation and foreground consistency," *Pattern Recogn.*, vol. 92, pp. 119–134, 2019.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 770–778.

[52] Y. Qiu, Y. Liu, X. Ma, L. Liu, H. Gao, and J. Xu, "Revisiting multi-level feature fusion: A simple yet effective network for salient object detection," in *Int. Conf. Image Process. (ICIP)*, 2019, pp. 4010–4014.

[53] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 3907–3916.

[54] Y. Liu, M.-M. Cheng, X.-Y. Zhang, G.-Y. Nie, and M. Wang, "DNA: Deeply supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybernetics*, 2021.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[56] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7234–7243.

[57] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7254–7263.

[58] Y. Liu, Q. Zhang, D. Zhang, and J. Han, "Employing deep part-object relationships for salient object detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1232–1241.

[59] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 3085–3094.

[60] R. A. Rensink, "The dynamic representation of scenes," *Visual Cognition*, vol. 7, no. 1-3, pp. 17–42, 2000.

[61] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews Neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.

[62] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7132–7141.

[63] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–14.

[64] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[65] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Int. Conf. Comput. Vis. Worksh. (ICCVW)*, 2019, pp. 0–0.

[66] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[67] Y. Sugano, Y. Matsushita, and Y. Sato, "Calibration-free gaze sensing using saliency maps," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2010, pp. 2667–2674.

[68] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process. (TIP)*, vol. 19, no. 1, pp. 185–198, 2009.

[69] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process. (TIP)*, vol. 13, no. 10, pp. 1304–1318, 2004.

[70] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2photo: Internet image montage," *ACM Trans. Graph. (TOG)*, vol. 28, no. 5, pp. 1–10, 2009.

[71] S. Feng, D. Xu, and X. Yang, "Attention-driven salient edge(s) and region(s) extraction with application to CBIR," *Signal Processing*, vol. 90, no. 1, pp. 1–15, 2010.

[72] L. Li, S. Jiang, Z.-J. Zha, Z. Wu, and Q. Huang, "Partial-duplicate image retrieval via saliency-guided visual matching," *IEEE MultiMedia*, vol. 20, no. 3, pp. 13–23, 2013.

[73] J. Sun, J. Xie, J. Liu, and T. Sikora, "Image adaptation and dynamic browsing based on two-layer saliency combination," *IEEE Trans. on Broadcasting*, vol. 59, no. 4, pp. 602–613, 2013.

[74] S. Goferman, A. Tal, and L. Zelnik-Manor, "Puzzle-like collage," *Comput. Graph. Forum (CGF)*, vol. 29, no. 2, pp. 459–468, 2010.

[75] H. Huang, L. Zhang, and H.-C. Zhang, "Arcimboldo-like collage using internet images," in *ACM SIGGRAPH Conf. Asia (SIGGRAPH Asia)*, 2011, pp. 1–8.

[76] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum, "Picture collage," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, vol. 1, 2006, pp. 347–354.

[77] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," *AAAI Conf. Artif. Intell. (AAAI)*, pp. 10 599–10 606, 2020.

[78] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 5455–5463.

[79] Y. Xu, D. Xu, X. Hong, W. Ouyang, R. Ji, M. Xu, and G. Zhao, "Structured modeling of joint deep feature and prediction refinement for salient object detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 3789–3798.

[80] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 7479–7489.

[81] J.-X. Zhao, J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 8779–8788.

[82] J. Wei, S. Wang, and Q. Huang, "F^3Net: Fusion, feedback and focus for salient object detection," in *AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 12 321–12 328.

[83] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020, pp. 13 025–13 034.

[84] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020, pp. 9141–9150.

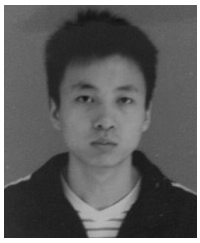
[85] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 35–51.

[86] B. Xu, H. Liang, R. Liang, and P. Chen, "Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection," in *AAAI Conf. Artif. Intell. (AAAI)*, vol. 35, no. 4, 2021, pp. 3004–3012.

[87] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2021.

[88] D. Zhang, H. Tian, and J. Han, "Few-cost salient object detection with adversarial-paced learning," in *Annu. Conf. Neur. Inform. Process. Syst. (NeurIPS)*, 2020, pp. 12 236–12 247.
[89] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 3917–3926.
[90] Y. Qiu, Y. Liu, S. Li, and J. Xu, "MiniSeg: An extremely minimum network for efficient COVID-19 segmentation," in *AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 4846–4854.
[91] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 1448–1457.
[92] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020, pp. 9413–9422.
[93] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
[94] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 3156–3164.
[95] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 2048–2057.
[96] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 593–602.
[97] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 603–612.
[98] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
[99] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–13.
[100] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 472–480.
[101] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 1623–1632.
[102] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7151–7160.
[103] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 136–145.
[104] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2010, pp. 49–56.
[105] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 1155–1162.
[106] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 3166–3173.
[107] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "SalientShape: Group saliency in image collections," *The Vis. Comput. (TVCJ)*, vol. 30, no. 4, pp. 443–453, 2014.
[108] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014, pp. 248–255.
[109] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 4548–4557.
[110] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 698–704.

Yu Qiu is a PhD candidate at the College of Artificial Intelligence, Nankai University. She received her bachelor's degree from Northwest A&F University in 2017. Her research interests include computer vision and machine learning.



Yun Liu received his bachelor's and doctoral degrees from Nankai University in 2016 and 2020, respectively. Currently, he works with Prof. Luc Van Gool as a postdoctoral scholar at Computer Vision Group, ETH Zurich. His research interests include computer vision and machine learning.

Yanan Chen is studying at the College of Artificial Intelligence, Nankai University. She graduated from Hebei University of Technology with a bachelor's degree in 2019. Her main research direction is machine learning.

Jianwen Zhang is a master's degree student at the College of Artificial Intelligence, Nankai University. She received her bachelor's degree from Hebei University of Technology in 2019. Her research interest is deep learning.

Jinchao Zhu is currently pursuing the PhD degree with the College of Artificial Intelligence, Nankai University. His research focuses on salient object detection, camouflaged object detection, under-water computer vision, and sonar image segmentation.

Jing Xu is a professor at the College of Artificial Intelligence, Nankai University. She received her doctoral degree from Nankai University in 2003. She has published more than 100 papers in software engineering, software security, and big data analytics. She won the second prize of Tianjin Science and Technology Progress Award twice, in 2017 and 2018.