Disentangled Capsule Routing for Fast Part-Object Relational Saliency

Liu, Yi ; Zhang, Dingwen; Liu, Nian; Xu, Shoukun; Han, Jungong

Published in:
IEEE Transactions on Image Processing

Publication date:
2022

Citation for published version (APA):
Liu, Y., Zhang, D., Liu, N., Xu, S., & Han, J. (Accepted/In press). Disentangled Capsule Routing for Fast Part-Object Relational Saliency. *IEEE Transactions on Image Processing*.

Disentangled Capsule Routing for Fast Part-Object Relational Saliency

Yi Liu*, Dingwen Zhang*, Nian Liu, Shoukun Xu†, and Jungong Han†

***Abstract*—Recently, the Part-Object Relational (POR) saliency underpinned by the Capsule Network (CapsNet) has been demonstrated to be an effective modeling mechanism to improve the saliency detection accuracy. However, it is widely known that the current capsule routing operations have huge computational complexity, which seriously limited the usability of the POR saliency models in real-time applications. To this end, this paper takes an early step towards a fast POR saliency inference by proposing a novel disentangled part-object relational network. Concretely, we disentangle horizontal routing and vertical routing from the original omnidirectional capsule routing, thus generating Disentangled Capsule Routing (DCR). This mechanism enjoys two advantages. On one hand, DCR that disentangles orthogonal 1D (*i.e.*, vertical and horizontal) routing greatly reduces parameters and routing complexity, resulting in much faster inference than omnidirectional 2D routing adopted by existing CapsNets. On the other hand, thanks to the light POR cues explored by DCR, we could conveniently integrate the part-object routing process to different feature layers in CNN, rather than just applying it to the small-scaled one as in previous works. This helps to increase saliency inference accuracy. Compared to previous POR saliency detectors, DPORTNet infers visual saliency $(5 \sim 9) \times$ faster, and is more accurate. DPORTNet is available under the open-source license at https://github.com/liuyi1989/DCR.**

***Index Terms*—Salient object detection, part-object relationship, capsule network, disentangled capsule routing, multi-level information integration**

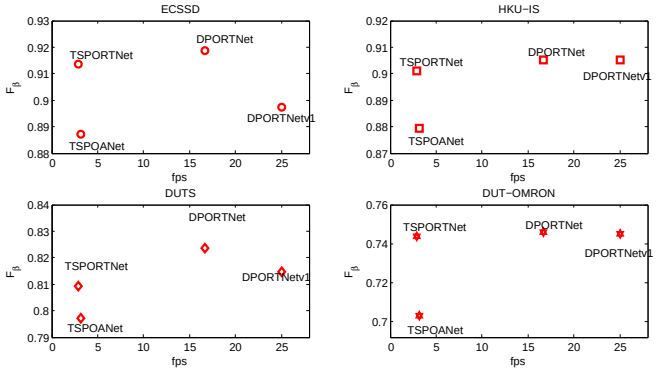


Fig. 1: Performance and speed for different POR saliency methods on four benchmarks. The input image of TSPOANet [9], TSPORTNet [10], and proposed "DPORTNet" is cropped into 352×352. "DPORTNetv1" is a modified version of DPORTNet by cropping the input image into 176×176.

I. Introduction

THE task of salient object detection is committed to imitating the human innate ability to identify the most attractive regions or objects from an image scene. Due to its potential to localize the visually meaningful regions in a scene, it can serve as a preprocessing step to improve the computational efficiency for a wide range of vision tasks, including segmentation [1], [2], image fusion [3], image retrieval [4], object recognition [5], *etc*.

The research of salient object detection stems from Liu's work [6], where visual saliency detection was considered a binary segmentation problem. Since then, a wide range of works [7] have been proposed to solve this problem based on hand-crafted features, *e.g.*, color, texture, *etc.* These methods, however, encountered a performance bottleneck due to the limited representation ability of hand-crafted features. Thanks to the emergence of deep learning, especially Convolutional Neural Networks (CNNs), the performance of salient object detection approaches has been improved substantially [8] in the past few years. Concretely, CNN-based approaches attempt to learn rich distinguishable features to highlight those high-contrast regions in an image, which are assembled to make up the entire saliency map. However, the CNN-based methods may often end up with incomplete segmentation of the salient object because of an underlying mechanism that the saliency of each image region is computed separately. To solve this problem, [9] and [10] proposed the idea of Part-Object Relational (POR) visual saliency by imposing the POR property to the task of salient object detection, which was implemented by the Capsule Network (CapsNet) [11].

Nonetheless, the preliminary attempts of POR saliency [9], [10] build POR cues exploration upon omnidirectional 2D routing, *i.e.*, each capsule must be routed into all other capsules across the image scale, which has two limitations. First, this omnidirectional routing comes at the cost of having a large number of network parameters and heavy routing complexity, both slowing down the saliency inference dramatically. As shown in Fig. 1, TSPOANet [9] and TSPORTNet [10] appear to have a speed of 3fps, which is inapplicable to real-time

Yi Liu and Shoukun Xu are with School of Computer Science and Artificial Intelligence, Aliyun School of Big Data, and School of Software, Changzhou University, Changzhou, Jiangsu, 213000, China. Email: liuyi0089@gmail.com, jpuxsk@163.com.

Dingwen Zhang is with the Hefei Comprehensive National Science Center, Institute of Artificial Intelligence, Hefei 230026, China, and School of Automation, Northwestern Polytechnical University, Xi'an, Shannxi, 710129, China. Email: zhangdingwen2006yyy@gmail.com.

Nian Liu is with Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. Email: liunian228@gmail.com.

Jungong Han is with Department of Computer Science, Aberystwyth University, U.K. Email: jungonghan77@gmail.com.

*: Equal contribution.

†: Equally corresponding authors.

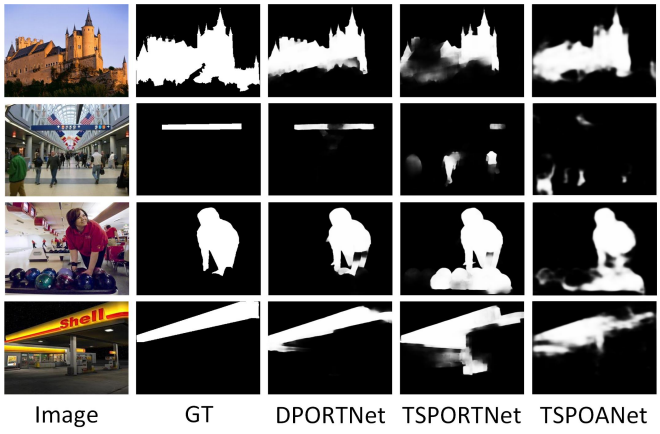


Fig. 2: Visual illustration for different POR saliency detectors. Our method can detect the accurate salient object, compared with TSPOANet [9] and TSPORTNet [10].

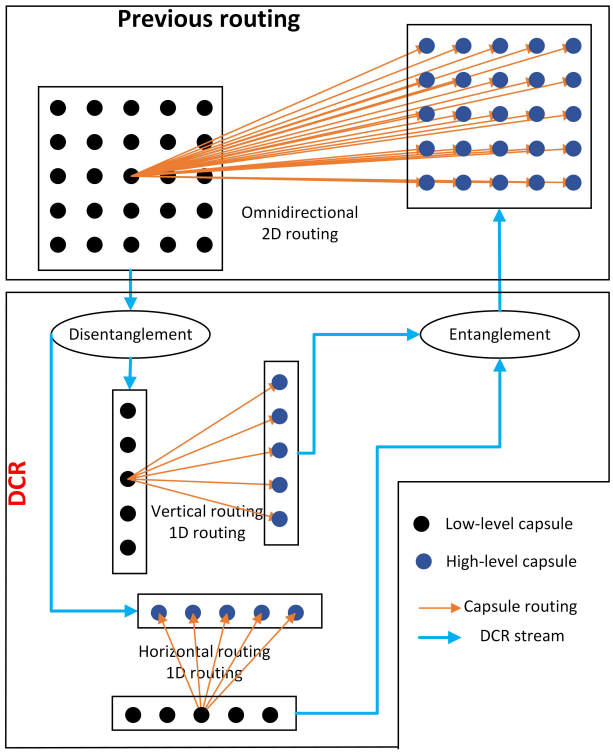


Fig. 3: Illustration for DCR. The disentanglement produces orthogonal 1D (vertical and horizontal) capsule routing from the omnidirectional 2D capsule routing. On top of that, two 1D capsule routing results are entangled to the 2D capsules.

scenarios[1]. Secondly, the complex omnidirectional routing limits the POR cues exploration to the small-scaled feature layer, thus leading to inaccurate saliency prediction in complex scenes. As seen in Fig. 1, TSPOANet [9] and TSPORTNet [10] do not achieve satisfactory F_β values due to the omnidirectional 2D routing. Apparently, these two limitations arise from omnidirectional 2D routing that has to compromise speed, accuracy, and simplicity. Visually in Fig. 2, the previous POR saliency methods, *i.e.*, TSPOANet [9] and TSPORTNet [10], sometimes add noise to the saliency maps in complex scenes. Particularly, the second row of Fig. 2 shows that the complicated scene fools TSPOANet [9] and TSPORTNet [10].

In this paper, we streamline the omnidirectional capsule routing for state-of-the-art CapNet-based saliency detectors and propose a Disentangled Part-Object Relational Network (DPORTNet) for **fast** POR saliency inference. Our main innovation lies in the proposed Disentangled Capsule Routing (DCR) towards fast POR cues exploration. Specifically, we disentangle vertical and horizontal primary capsules from the original full-resolution capsule maps for capsule routing. This way allows a vertical 1D routing and a horizontal 1D routing to replace the original omnidirectional 2D routing to explore the part-object relationships of the capsule nodes. On top of that, the obtained orthogonal (vertical and horizontal) capsules are entangled by matrix multiplication to restore the full-resolution capsule matrix. This mechanism brings two advantages. First, as shown in Fig. 3, DCR enables orthogonal routing, instead of omnidirectional routing adopted in existing CapsNets, which greatly reduces parameters and routing complexity. In doing so, we can significantly speed up saliency inference. It can be seen in Fig. 1, our model, *i.e.*, DPORTNet/DPORTNetv1 achieves much faster fps compared with TSPOANet [9] and TSPORTNet [10]. Secondly, because of the lightweight POR cues explored by DCR, we can conveniently apply the part-object routing process to multiple feature layers in CNN, rather than just small-scaled feature layers as in previous works [9], [10], which leads to better saliency prediction. This can be verified in Fig. 1, where our model, *i.e.*, DPORTNet/DPORTNetv1 surpasses TSPOANet [9] and TSPORTNet in terms of F_β. Besides, it can be seen from Fig. 2 that our method can detect the accurate salient object, compared with TSPOANet [9] and TSPORTNet [10]. Also, experiments on four benchmarks show that the proposed POR saliency method is superior to the state-of-the-art methods.

To sum up, the contributions of this paper are as follows:

(1) We design a fast capsule routing algorithm, *i.e.*, DCR, by involving disentangled representation for CapsNet towards fast POR cues exploration. To the best of our knowledge, this is the first attempt to adopt disentangled representation to CapsNet.

(2) On top of DCR, we design a POR saliency network, *i.e.*, DPORTNet, which utilizes DCR in multiple layers to learn multi-level POR cues for saliency prediction. In other words, the proposed simple DCR routing algorithm enables multi-level POR cues exploration, which is absent in the existing CapsNet-based POR saliency detection methods because of their heavy routing algorithms.

This paper is organized as follows. Sec. II reviews the related references to our work. Sec. III describes the details of the proposed DCR algorithm. Sec. IV designs a fast part-object relational saliency network using DCR. Sec. V carries out abundant experiments and analyses to understand our method. Sec. VI concludes the paper.

II. Related Work

In this section, we will review references related to our work, including salient object detection, CapsNet, and disen-

[1]Usually the real-time requirement is 24 fps.

tangled representation.

A. CNNs for Salient Object Detection

To date, a large number of works have been proposed for the task of salient object detection. Hand-crafted features dominate early salient object detectors, for which a comprehensive review can be found in [7]. The emergence of deep learning, especially CNNs, has improved the performance substantially [12]–[15]. Here, we focus on the CNN-based salient object detectors that are most related to our method.

The preliminary study simply adopts CNNs for salient object detection. For example, Li *et al.* [12] learned multi-scale features via CNNs for salient object detection. Gupta *et al.* [16] extracted adjacent-layer features at one resolution for saliency prediction. Wang *et al.* [17] designed a salient object detection architecture via local estimation and global search. These works were mostly implemented using the fully connected networks and thereby demanded many resources. Later, this problem was settled by adopting the fully convolutional network [18] for salient object detection. For example, Liu *et al.* [19] involved global prediction and hierarchical refinement to detect the salient object. In view of different semantics captured by different stages of CNN features, many researchers attempted to integrate multi-level features for saliency prediction [20], [21]. For instance, multi-level features were integrated into multiple scales for salient object detection [20]. Ma *et al.* [22] aggregated adjacent features layer by layer to fuse important details and semantics and discard interference information. Besides, context plays a vital role in deep understanding of saliency detection [23], [24]. For example, Liu *et al.* [23] proposed a contextual information guidance strategy for multi-level information integration towards salient object detection. Gupta *et al.* [25] proposed a gate-based context extraction module to emphasize invariance features for different scales of visual patterns. Siris *et al.* [26] exploited the semantic scene contexts to learn the salient objects from the scene. Zhao *et al.* [27] designed three complementary branches for saliency detection, including semantic path, spatial path, and boundary path. Li *et al.* [28] utilized a purificatory mechanism to find the salient objects using a structural similarity loss to model the region-level relationships for saliency calibration. Yang *et al.* [29] proposed a progressive self-guided loss function to train the salient object detection network. More salient object techniques can be found in [30]. Xu *et al.* [31] simulated the human biological mechanism of globally located and locally segmenting salient objects. Tang *et al.* [32] solved the problem of high-quality salient object detection by designing a low-resolution saliency classification network and a high-resolution refinement network.

B. CapsNets for Part-Object Relational Salient Object Detection

The concept of capsule was developed in [33]. A capsule contains a group of neurons to represent the instantiation parameters of the entity, *e.g.*, pose, deformation, texture, *etc.* Sabour *et al.* [34] implemented a vector CapsNet via representing a capsule as a vector and designing a dynamic routing algorithm. Hinton *et al.* [11] improved the idea via a matrix CapsNet, which was achieved by encapsulating a capsule as a pose matrix and an activation value, and designing a robust Expectation-Maximization (EM) routing algorithm. The pavement of CapsNet continued with the development of a stacked capsule autoencoder in an unsupervised manner [35]. Besides, many variants have been proposed to enhance CapsNet [36]–[39].

In view of the advances of CapsNet, it has been applied to many computer vision tasks, *e.g.*, video object segmentation [40], multi-label classification [41], object segmentation [42], *etc.* CapsNet has also been well studied for salient object detection [9], [10], [43]. Liu *et al.* [9] introduced the POR property implemented by CapsNet for salient object detection. Concretely, a two-stream strategy was developed in [9] to implement CapsNet, which could reduce the computational cost and parameters, and also noisy capsule assignments to some extent. In their extended version [10], a correlation-aware routing algorithm was proposed to speed up the training procedure and increasing the accuracy of part-object relationships, which resulted in a further performance enhancement.

The difference between our work and CapsNet can be explained as follows. Due to the disentangled representation, our DCR implements orthogonal 1D routing, instead of omni-directional 2D routing adopted by the existing CapsNets. This implementation greatly reduces the network parameters and routing complexity, resulting in faster POR cues exploration for efficient saliency inference, as can be verified in Fig. 1.

Besides, the difference between our method and the existing POR saliency methods [9], [10] lies in two folds. First, our orthogonal 1D routing greatly speeds up the POR cues exploration, compared to omnidirectional 2D routing in [9], [10], resulting in faster saliency inference. Secondly, the existing POR saliency methods [9], [10] explore single-scale (*i.e.*, 44×44) POR cues for saliency prediction, while our method explores multi-scale (*i.e.*, 88×88, 44×44, and 22×22) POR cues, which help capture richer POR cues for better saliency prediction.

C. Disentangled Representation

The goal of disentangled representation is to extract explanatory factors from diverse data variation for generating a meaningful representation, which has been studied for various tasks. For example, Chio *et al.* [44] disentangled 1D-discriminative and 1D-excluded factors from visible-thermal images. The former was used for cross-modality matching. Yin *et al.* [45] disentangled semantics to fulfill the high-level semantic consistency and low-level semantic diversity requirements for text-to-image generation. For pose exstimation, Li *et al.* [46] disentangled the pose to predict rotation and translation separately. Liu *et al.* [47] disentangled shape features from 2D images during 3D face shapes reconstruction for face recognition. Gilbert *et al.* [48] disentangled image structure and style during patch search and selection for style-aware image completion. Guen and Thome [49] disentangled physical dynamics to achieve unsupervised video prediction.

In this paper, we extend disentangled representation to solve the problem of POR saliency. Specifically, we disentangle

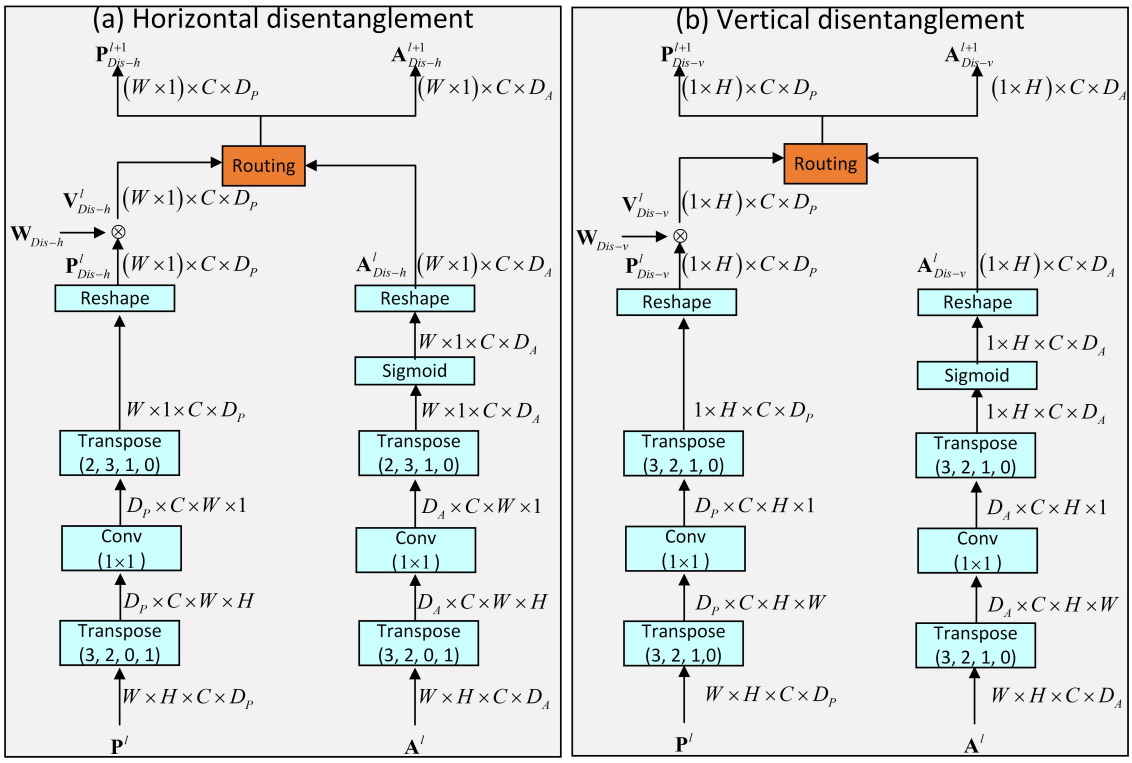


Fig. 4: Primary capsule disentanglement. The pose and activation are disentangled into horizontal pose and activation ((a)) and vertical pose and activation ((b)), respectively, which are further fed into the EM routing algorithm for capsules routing. W, H, and C represent the width, height, and capsule type number, respectively. $\mathbf{W}_{Dis-h}$ and $\mathbf{W}_{Dis-v}$ are learned weight matrices. $\otimes$ is the operation of matrix multiplication. $\mathbf{P} \in R^{W\times H\times C\times D_P}$ and $\mathbf{A} \in R^{W\times H\times C\times D_A}$ represent the pose matrix and the activation of the capsule maps. $D_P = 16$ and $D_A = 1$. The superscript l is the layer index.

orthogonal 1D routing from omnidirectional 2D routing for the sake of exploring better POR cues for saliency inference.

III. The Disentangled Capsule Routing

In this section, we illustrate the proposed Disentangled Capsule Routing (DCR), which is designed for fast part-object relational cues exploration. It consists of two phases, *i.e.*, primary capsule disentanglement and capsule matrix entanglement.

A. Primary Capsule Disentanglement

Primary capsule disentanglement is designed to disentangle vertical and horizontal capsules from the 2D full-resolution primary capsule maps. Fig. 4 shows the details of the disentanglement process, which is composed of two streams along the vertical and horizontal directions, respectively.

Suppose $\mathbf{P} \in R^{W\times H\times C\times D_P}$ and $\mathbf{A} \in R^{W\times H\times C\times D_A}$ are the pose matrix and the activation of the capsule maps, respectively, where W, H, and C represent the width, height, and capsule type number, respectively. $D{=}\{D_P = 16, D_A = 1\}$ is the dimension of the pose matrix and the activation. Fig. 4 details the disentanglement process, which will be illustrated as follows. The disentanglement pipeline consists of two main procedures, including horizontal/vertical disentanglement for capsules and horizontal/vertical votes computation.

Step 1: Horizontal/vertical disentanglement for capsules. As shown in Fig. 4, the straight pipeline of the horizontal disentanglement can be given as

$$\begin{aligned}\mathbf{P}^l/\mathbf{A}^l \in \Re^{W\times H\times C\times D} \xrightarrow{T} \Re^{D\times C\times W\times H} \xrightarrow{\circledast} \Re^{D\times C\times W\times 1}\\ \xrightarrow{T} \Re^{W\times 1\times C\times D} \xrightarrow{R} \mathbf{P}^l_{Dis-h}/\mathbf{A}^l_{Dis-h} \in \Re^{(W\times 1)\times C\times D},\end{aligned} \tag{1}$$

where $\mathbf{P}^l$ and $\mathbf{A}^l$ represents the pose matrix and the activation values of capsules in layer l, respectively, and $/$ means "or". T and R represent the operations of transpose and reshape, respectively. $\circledast$ means a convolution operation with the kernel size of 1×1. The superscript l is the layer index. D can be taken as $D_P = 16$ and $D_A = 1$ to disentangle the pose matrix $\mathbf{P}^l_{Dis-h}$ and the activation values $\mathbf{A}^l_{Dis-h}$ of the capsules in layer l, respectively. It is noted that the sigmoid function is used for $\mathbf{A}^l_{Dis-h}$.

Similarly, as shown in Fig. 4, the straight pipeline of the vertical disentanglement can be given as

$$\begin{aligned}\mathbf{P}^l/\mathbf{A}^l \in \Re^{W\times H\times C\times D} \xrightarrow{T} \Re^{D\times C\times H\times W} \xrightarrow{\circledast} \Re^{D\times C\times H\times 1}\\ \xrightarrow{T} \Re^{1\times H\times C\times D} \xrightarrow{R} \mathbf{P}^l_{Dis-v}/\mathbf{A}^l_{Dis-v} \in \Re^{(1\times H)\times C\times D},\end{aligned} \tag{2}$$

Also, the Sigmoid function is adopted to activate $\mathbf{A}^l_{Dis-v}$.

Step 2: Horizontal/vertical votes computation. The vote matrix can be computed by multiplying the pose matrix and

a learned weight matrix, *i.e.*,

$$\mathbf{V}^{l}_{Dis-h} \in \Re^{(W\times 1)\times C\times\sqrt{D_P}\times\sqrt{D_P}} = \tilde{\mathbf{P}}^{l}_{Dis-h} \times \mathbf{W}^{l}_{Dis-h}\ , \tag{3}$$

$$\mathbf{V}^{l}_{Dis-v} \in \Re^{(1\times H)\times C\times\sqrt{D_P}\times\sqrt{D_P}} = \tilde{\mathbf{P}}^{l}_{Dis-v} \times \mathbf{W}^{l}_{Dis-v}\ , \tag{4}$$

where, $\tilde{\mathbf{P}}^{l}_{Dis-h} \in \Re^{(W\times 1)\times C\times\sqrt{D_P}\times\sqrt{D_P}}$ and $\tilde{\mathbf{P}}^{l}_{Dis-v} \in \Re^{(1\times W)\times C\times\sqrt{D_P}\times\sqrt{D_P}}$ are obtained by reshaping $\mathbf{P}^{l}_{Dis-h}$ and $\mathbf{P}^{l}_{Dis-v}$, respectively. $\mathbf{W}_{Dis-h}\Re^{(W\times 1)\times C\times\sqrt{D_P}\times\sqrt{D_P}}$ and $\mathbf{W}_{Dis-v}\Re^{(1\times W)\times C\times\sqrt{D_P}\times\sqrt{D_P}}$ are learned weight matrices.

$(\mathbf{V}^{Dis-h}, \mathbf{A}^{Dis-h})$ and $(\mathbf{V}^{Dis-v}, \mathbf{A}^{Dis-v})$ are fed into the Expectation Maximization (EM) routing algorithm [11] for horizontal routing and vertical routing to explore horizontal and vertical POR cues, respectively, *i.e.*, $(\mathbf{P}^{l+1}_{Dis-h}, \mathbf{A}^{l+1}_{Dis-h})$ and $(\mathbf{P}^{l+1}_{Dis-v}, \mathbf{A}^{l+1}_{Dis-v})$.

B. Capsule Matrix Entanglement

Capsule matrix entanglement is designed to recover the full-resolution pose matrix from horizontal and vertical pose matrices ($\mathbf{P}^{l+1}_{Dis-h}$ and $\mathbf{P}^{l+1}_{Dis-v}$), and recover the full-resolution activation from horizontal and vertical activations ($\mathbf{A}^{l+1}_{Dis-h}$ and $\mathbf{A}^{l+1}_{Dis-v}$). Fig. 5 details the process of the capsule matrix entanglement, which consists of two streams in terms of pose matrix and activation.

As shown in Fig. 5(a) and (b), the entanglement is achieved by multiplying vertical and horizontal semantics. Before the matrix multiplication, dimension matching is necessary. As shown in Fig. 4(a) and (b), the straight pipeline of dimension matching for $\mathbf{P}^{l+1}_{Dis-h}$ and $\mathbf{A}^{l+1}_{Dis-h}$ can be illustrated as

$$\begin{aligned}\mathbf{P}^{l+1}_{Dis-h}/\mathbf{A}^{l+1}_{Dis-h} \in \Re^{(W\times 1)\times C\times D} &\xrightarrow{R} \Re^{W\times 1\times C\times D} \\ &\xrightarrow{T} \hat{\mathbf{P}}^{l+1}_{Dis-h}/\hat{\mathbf{A}}^{l+1}_{Dis-h} \in \Re^{C\times D\times W\times 1},\end{aligned} \tag{5}$$

where D can be taken as $D_P = 16$ and $D_A = 1$ for the pose matrix and the activation, respectively.

Similarly, as shown in Fig. 4(a) and (b), the straight pipeline of dimension matching for $\mathbf{P}^{l+1}_{Dis-v}$ and $\mathbf{A}^{l+1}_{Dis-v}$ can be illustrated as

$$\begin{aligned}\mathbf{P}^{l+1}_{Dis-v}/\mathbf{A}^{l+1}_{Dis-v} \in \Re^{(1\times H)\times C\times D} &\xrightarrow{R} \Re^{1\times H\times C\times D} \\ &\xrightarrow{T} \hat{\mathbf{P}}^{l+1}_{Dis-v}/\hat{\mathbf{A}}^{l+1}_{Dis-v} \in \Re^{C\times D\times 1\times H}.\end{aligned} \tag{6}$$

On top of that, the entangled pose matrix can be computed by matrix multiplication as

$$\hat{\mathbf{P}}^{l+1} \in {}^{C\times D_P\times W\times H} = \hat{\mathbf{P}}^{l+1}_{Dis-h} \otimes \hat{\mathbf{P}}^{l+1}_{Dis-v}. \tag{7}$$

The full-resolution pose matrix $\mathbf{P}^{l+1} \in {}^{W\times H\times C\times D_P}$ can be achieved by reshaping $\hat{\mathbf{P}}^{l+1}$.

Similarly, the entangled activation can be computed by matrix multiplication as

$$\hat{\mathbf{A}}^{l+1} \in {}^{C\times D_A\times W\times H} = Sigmoid\left(\hat{\mathbf{A}}^{l+1}_{Dis-h} \otimes \hat{\mathbf{A}}^{l+1}_{Dis-v}\right), \tag{8}$$

where $Sigmoid(\cdot)$ means the sigmoid function. The full-resolution activation $\mathbf{A}^{l+1} \in {}^{W\times H\times C\times D_A}$ can be achieved by reshaping $\hat{\mathbf{A}}^{l+1}$.

To this end, the capsule maps of layer $(l+1)$, *i.e.*, $\mathbf{P}^{l+1}$ and $\mathbf{A}^{l+1}$, can be obtained. Algorithm 1 illustrate the DCR based CapsNet.

Algorithm 1 DCR based CapsNet. X is the feature maps of the input image. P^*_* and A^*_* are the pose matrices and activation values, respectively. R is the reshape operation.

Procedure Disentangled capsule routing (X)
1. Primary capsules generation
   $P^l, A^l = PrimaryCaps(X)$
2. Primary capsules disentanglement:
   Horizontal disentanglement:
   $\mathbf{P}^{l}_{Dis-h}/\mathbf{A}^{l}_{Dis-h}$ = Eq. 1(P^l, A^l)
   Vertical disentanglement:
   $\mathbf{P}^{l}_{Dis-v}/\mathbf{A}^{l}_{Dis-v}$ = Eq. 2(P^l, A^l)
3. Vote matrix computation:
   Horizontal vote:
   $\mathbf{V}^{l}_{Dis-h}$ = Eq. 3$(\mathbf{P}^{l}_{Dis-h})$
   Vertical vote:
   $\mathbf{V}^{l}_{Dis-v}$ = Eq. 4$(\mathbf{P}^{l}_{Dis-v})$
4. EM routing:
   Horizontal routing:
   $\mathbf{P}^{l+1}_{Dis-h}, \mathbf{A}^{l+1}_{Dis-h} = EM(\mathbf{V}^{l}_{Dis-h}, \mathbf{A}^{l}_{Dis-h})$
   Vertical routing:
   $\mathbf{P}^{l+1}_{Dis-v}, \mathbf{A}^{l+1}_{Dis-v} = EM(\mathbf{V}^{l}_{Dis-v}, \mathbf{A}^{l}_{Dis-v})$
5. Capsule matrix entanglement:
   $\hat{\mathbf{P}}^{l+1}_{Dis-h}, \hat{\mathbf{A}}^{l+1}_{Dis-h}$ = Eq. 5$(\mathbf{P}^{l+1}_{Dis-h}, \mathbf{A}^{l+1}_{Dis-h})$
   $\hat{\mathbf{P}}^{l+1}_{Dis-v}, \hat{\mathbf{A}}^{l+1}_{Dis-v}$ = Eq. 6$(\mathbf{P}^{l+1}_{Dis-v}, \mathbf{A}^{l+1}_{Dis-v})$
   $\mathbf{P}^{l+1}$ = R(Eq. 7$(\hat{\mathbf{P}}^{l+1}_{Dis-h}, \hat{\mathbf{P}}^{l+1}_{Dis-v})$)
   $\mathbf{A}^{l+1}$ = R(Eq. 8$(\hat{\mathbf{A}}^{l+1}_{Dis-h}, \hat{\mathbf{A}}^{l+1}_{Dis-v})$)

IV. Network Architecture for Salient Object Detection

In this section, we will detail the proposed deep salient object detection method. Fig. 6 shows the proposed DPORTNet architecture for salient object detection, consisting of two main compositions: backbone feature maps generation and DCR. Concretely, at each stage, backbone feature maps are generated by the backbone network and AtrouS Pyramid Pooling (ASPP) [50]. In addition, backbone feature maps are fed into DCR for POR cues exploration at the three deepest stages that contain high-level semantics. Furthermore, a residual learning module is designed to integrate the contrast cues of the backbone feature maps and the POR cues by DCR to attend to the salient regions. Finally, multi-level semantics are integrated in a deep-to-shallow manner to infer the salient object. Details of the proposed salient object detector will be illustrated in the following.

A. Backbone Feature Maps Generation

As shown in Fig. 6, the input image first goes through five stacked convolutional layers, which are implemented by $Conv1\_2$, $Conv2\_2$, $Conv3\_3$, $Conv4\_3$, and $Conv5\_3$ of the pre-trained VGG16 [51] model. Besides, to capture richer context of the input image, ASPP [50] with multiple dilation

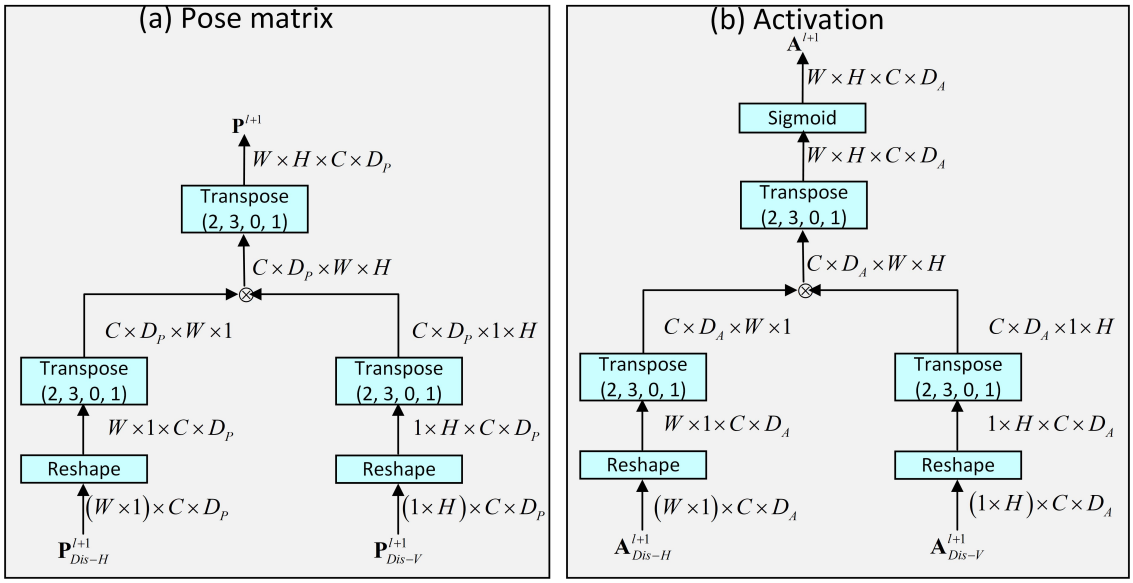


Fig. 5: Capsule matrix entanglement. The horizontal and vertical poses are entangled into the 2D capsule pose ((a)). Likewise, the horizontal and vertical activations are entangled into the 2D capsule activation ((b)). The interpretations of the mathematical symbols can be found in the caption of Fig. 4.

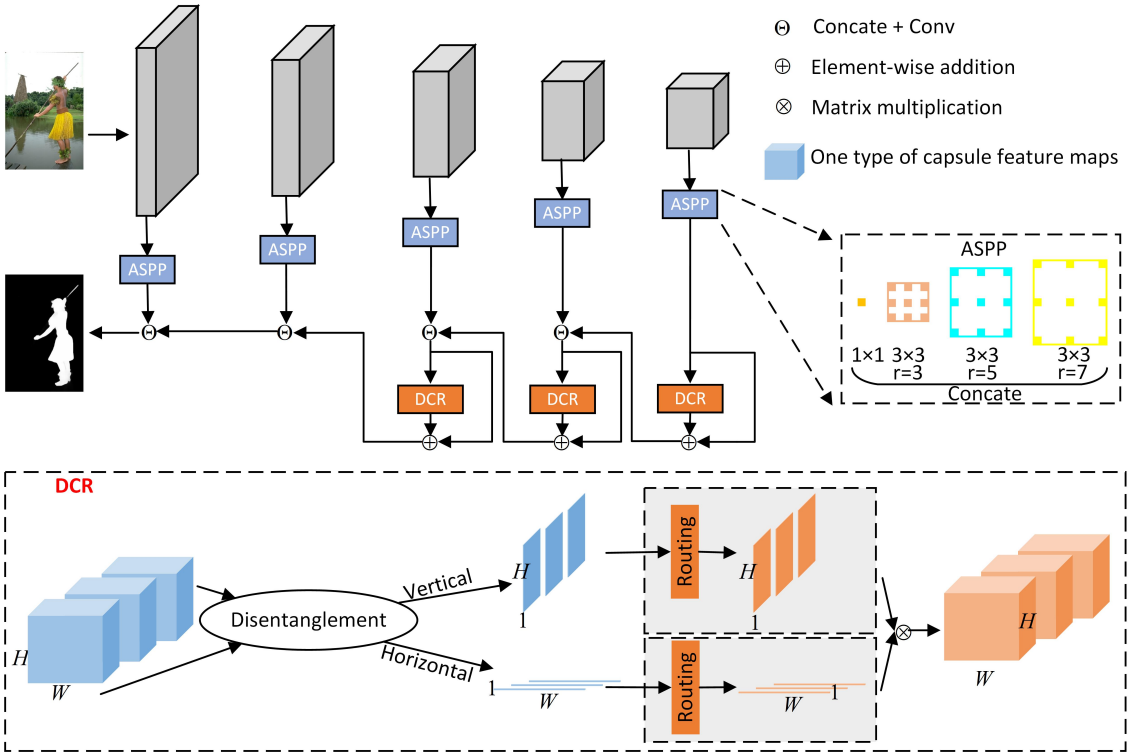


Fig. 6: Proposed salient object detection network architecture, *i.e.*, DPORTNet. The top is the framework of our DPORTNet. The bottom is the framework of our DCR. W and H represent the width and height of the capsule maps. At the three deeper stages, the backbone network and ASPP [50] are employed to learn rich backbone feature maps, which are fed into DCR for POR cues exploration. On top of that, a residual learning integrates the contrast cues from the backbone feature maps and the POR cues from DCR. Finally, multi-level feature maps are integrated in a deep-to-shallow manner to compute the saliency map.

rates (1, 3, 5, 7) is adopted at each stage to generate multi-scale backbone feature maps, which contain rich context information under various receptive fields without increasing the kernel scales.

B. DCR for part-object relational cues exploration

In view of the lightweight of DCR, we adopt it to explore multi-scale POR cues for saliency prediction. Specifically in Fig. 6, we integrate DCR at three deeper stages that contain high-level semantics for sake of multi-scale POR cues exploration. On top of that, a residual learning combines contrast cues captured by backbone feature maps and POR cues explored by DCR, *i.e.*,

$$q_{out}^{i} = q_{in}^{i} + f_{DCR}\left(q_{in}^{i}\right)(i = 3, 4, 5), \tag{9}$$

where q_{in}^{i} and q_{out}^{i} are the input features and output features of DCR at layer i. f_{DCR} represents the DCR operation.

For the shallow two layers with large scales, the backbone feature maps obtained by ASPP are directly integrated with the deep part-object relational cues in a deep-to-shallow manner via concatenation for saliency inference, *i.e.*,

$$q_{out}^{i} = f_{conv}(f_{cat}(q_{out}^{i+1}, q_{ASPP}^{i}), \mathbf{W}_{conv})(i = 1, 2), \tag{10}$$

where f_{conv}, f_{cat}, and $\mathbf{W}_{conv}$ represent the operations of convolution, concatenation, and the parameters of convolution, respectively.

C. Loss Function

We use the cross-entropy loss function (l_{ce}) and the Intersection over Union (IoU) loss function (l_{iou}) to jointly train our salient object detection network, *i.e.*, $l_{ce} + l_{iou}$. Suppose B and G are the predicted saliency map and corresponding ground truth. l_{ce} is formulated as

$$l_{ce}\left(B, G\right) = -\sum_{i}\left[G_i \log\left(B_i\right) + \left(1 - G_i\right)\log\left(1 - B_i\right)\right], \tag{11}$$

where i is the pixel index.

l_{iou} is defined as

$$l_{iou}\left(B, G\right) = 1 - \frac{\sum_{i} B\left(i\right) G\left(i\right)}{\sum_{i}\left[B\left(i\right) + G\left(i\right) - B\left(i\right) G\left(i\right)\right]}. \tag{12}$$

D. Insight into DCR induced saliency

1) Visualization for the DCR saliency detector: Fig. 7 visualizes examples for the POR 1D activation maps[2] at the third level. As can be seen from Fig. 7, 1D vertical and horizontal maps can activate the salient rows and columns, respectively. They are further entangled by matrix multiplication to produce a 2D capsule activation map to attend to the salient object. This also supports the rationality of our DCR, in which 2D routing can be disentangled into horizontal and vertical directions and they can be further entangled by matrix multiplication.

[2]In this paper, the saliency map is derived from the activation map.

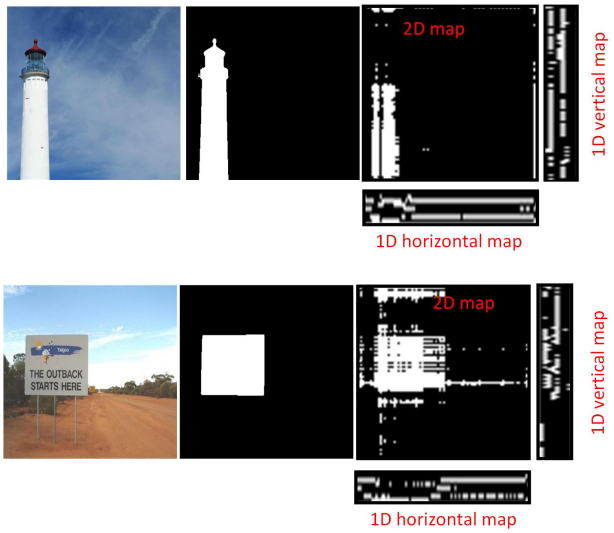


Fig. 7: Visualizations of 1D activation maps at the third level. 1D maps of 8 types of capsules are stacked together for visualizations. The first two columns are images and the ground truth, respectively.

2) Difference between our work and CCNet [52]: CCNet [52] describes a criss-cross attention to extract rich context for the task of semantic segmentation. Specifically, CCNet [52] computes an affinity map as the attention map, which is implemented by multiplying each-position feature vector of query and the row or column feature vectors of key, resulting in a criss-cross attention map. Such a mechanism reduces the parameters and complexity from N^2 to $N\sqrt{N}$, where $N = H \times W$ is the spatial dimension. In contrast, we do not simply separate the row/column information. Instead, we disentangle the input into row vector and column vector. Specifically, the row/column dimension is treated as the channel dimension, which is transformed into one dimension via a convolution. Such a disentanglement mechanism can achieve column/row feature maps of the input, which are fed into the capsule routing for column/row capsules assignment to compute the column/row capsule maps, respectively. By doing so, the 2D capsule routing can be transformed into two 1D capsule routing, of which each routing associates rows or columns together. Usually $H = W$, and our mechanism reduces the parameters and complexity from N^2 to $\frac{N}{2}$. As a result of the lower complexity ($\frac{N}{2} < N\sqrt{N}$), our method is more efficient in terms of computation.

Besides, our disentanglement is essentially different from the row/column separation with the evidence that our vertical and horizontal capsule features are not simply the row and column information. During the disentanglement of vertical capsule features, it can be found in Fig. 4 that the horizontal dimension of the 2D capsule features is transposed into the channel dimension, which is followed by a convolution to achieve our vertical capsule features. Such a mechanism disentangles the vertical capsule features instead of simple row separation. Similarly, our disentanglement of horizontal capsule features from the 2D capsule features is achieved

by a convolution on the vertical dimension instead of simple column separation. The disentangled vertical and horizontal 1D inputs are further fed into the capsule routing algorithm for capsule assignments, producing 1D capsule routing procedures.

V. Experiment and Analysis

In this section, we will carry out abundant experiments and analysis to provide a comprehensive understanding of the proposed method.

A. Dataset

We evaluate the proposed salient object detection network on four public benchmarks.

ECSSD [53] contains 1000 images with complicated structures, which are collected from the Internet.

HKU-IS [12] consists of 3000 training images and 1447 test images, which are with multiple disconnected objects.

DUTS [54] contains 10533 training images and 5019 test images, which are with different scenes and various sizes.

DUT-OMRON [55] has 5168 images with different sizes and complex structures.

In terms of HKU-IS [12] and DUTS [54], only the test images are used for evaluations in our experiments.

B. Evaluation Metric

We evaluate the performance of our model as well as other state-of-the-art methods from both visual and quantitative perspectives. The quantitative metrics include weighted F-measure (F_β) [56], Mean Absolute Error (MAE) [56], S-measure (S_m) [57], and E-measure (E_m) [58]. Given a continuous saliency map, a binary mask $\hat{B}$ is achieved by thresholding the saliency map B. Precision is defined as $Precision = \left|\hat{B} \cap G\right| \Big/ \left|\hat{B}\right|$, and recall is defined as $Recall = \left|\hat{B} \cap G\right| \Big/ |G|$. Then, the PR curve is plotted under different thresholds.

F-measure is an overall performance indicator, which is computed by

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

As suggested in [56], $\beta^2 = 0.3$.

MAE is defined as

$$MAE = \frac{1}{\hat{W} \times \hat{H}} \sum_i \left|B\left(i\right) - G\left(i\right)\right|, \tag{14}$$

where $\hat{W}$ and $\hat{H}$ are the width and height of the image, respectively.

S-measure (S_m) [57] is computed by

$$S_m = \alpha S_o + \left(1-\alpha\right) S_r, \tag{15}$$

where S_o and S_r represent the object-aware and region-aware structure similarities between the prediction and the ground truth, respectively. α is set to 0.5 [57].

E-measure (E_m) [58] combines local pixel values with the image-level mean value to jointly evaluate the similarity between the prediction and the ground truth.

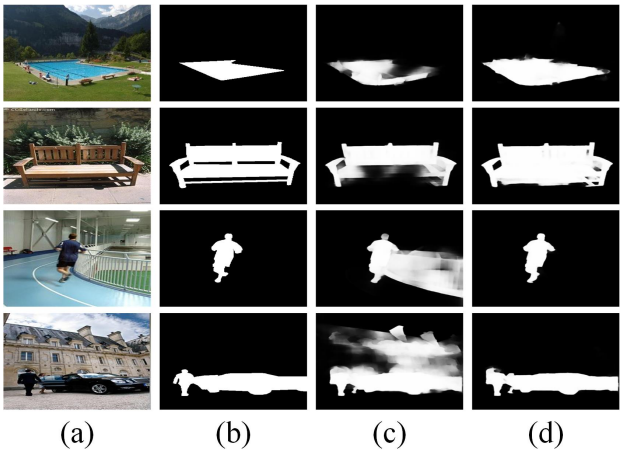


Fig. 8: Ablation visualization for DCR. (a) Images; (b) GT; (c) -DCR; (d) DPORTNet. DCR enables grabbing/capturing the object wholeness (top two rows) and suppressing the confused backgrounds around salient objects (bottom two rows).

C. Implementation Detail

The proposed model is implemented in Tensorflow [59]. To avoid over-fitting caused by training from scratch, the backbone network is initialized by the five stages of the pretrained VGG16 model [51], respectively. The other weights are initialized randomly with a truncated normal ($\sigma = 0.01$) distribution, and the biases are initialized to 0. The Adam optimizer [60] is used to train our model with an initial learning rate of 10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The training dataset of DUTS [54] is used to train our network with horizontal flipping as the data augmentation technique.

D. Ablation Analysis

1) DCR: To verify the effectiveness of the proposed DCR, we compare the entire model with a baseline, which removes DCR from the model in Fig. 6. Table I lists the quantitative values of different metrics for comparison. As shown in Table I, the involvement of DCR can effectively improve the performance. Besides, Fig. 8 shows the visual illustration of the proposed capsule routing. Specifically, as shown in Fig. 8, DCR enables grabbing/capturing the object wholeness (as shown in the top two rows of Fig. 8) and suppressing the confused backgrounds around salient objects (as shown in the bottom two rows of Fig. 8). These improvements benefit from the orthogonal POR cues captured by DCR, which help to detect relevant object parts and learn the object wholeness for better saliency prediction.

2) Different POR Cues Explorations for Saliency: To take a thorough study on different POR cues explorations for POR saliency, we replace the two-stream capsule routing in TSPOANet [9][3] with our DCR, called TSPOANet-DCR, to compare with TSPOANet [9]. As shown in Table I, our DCR can improve the performance of POR saliency, compared

[3]The existing POR saliency detectors [9], [10] explore POR cues by using the same capsule routing during the testing stage, *i.e.*, two-stream routing. Therefore, we select TSPOANet [9] for comparison.

TABLE I: F_β, MAE, S_m, and E_m values for different ablation studies.

	ECSSD [53]				HKU-IS [12]				DUTS [54]				DUT-OMRON [55]			
	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$
DPORTNet	**0.9186**	**0.0381**	**0.9143**	**0.9202**	**0.9051**	**0.0320**	**0.9080**	**0.9476**	**0.8236**	**0.0414**	**0.8702**	**0.8945**	**0.7458**	**0.0553**	**0.8205**	**0.8539**
-DCR	0.8999	0.0441	0.9131	0.9166	0.8879	0.0377	0.9072	0.9390	0.7636	0.0518	0.8585	0.8588	0.7164	0.0644	0.8185	0.8395
TSPOANet-DCR	**0.8929**	**0.0457**	**0.9082**	**0.9132**	**0.8845**	**0.0372**	**0.9037**	**0.9429**	0.7824	**0.0476**	**0.8590**	**0.8827**	**0.7249**	**0.0597**	**0.8165**	**0.8505**
TSPOANet [9]	0.8873	0.0515	0.8684	0.9020	0.8795	0.0391	0.8656	0.9263	**0.7971**	0.0482	0.8202	0.8748	0.7030	0.0628	0.7692	0.8232
DPORTNet	**0.9186**	**0.0381**	**0.9143**	**0.9202**	**0.9051**	**0.0320**	**0.9080**	**0.9476**	**0.8236**	**0.0414**	**0.8702**	**0.8945**	**0.7458**	**0.0553**	**0.8205**	**0.8539**
TSPOANet-DCR	0.8929	0.0457	0.9082	0.9132	0.8845	0.0372	0.9037	0.9429	0.7824	0.0476	0.8590	0.8827	0.7249	0.0597	0.8165	0.8505
DPORTNet	**0.9186**	**0.0381**	**0.9143**	0.9202	**0.9051**	**0.0320**	**0.9080**	**0.9476**	**0.8236**	**0.0414**	**0.8702**	**0.8945**	**0.7458**	**0.0553**	**0.8205**	0.8539
DPORTNet-OS	0.9057	0.0432	0.9058	0.9220	0.8981	0.0343	0.9034	0.9440	0.8069	0.0432	0.8666	0.8908	0.7419	0.0564	0.8167	**0.8610**
DPORTNet-TS	0.9148	0.0414	0.9120	**0.9235**	0.8996	0.0341	0.9060	0.9452	0.8084	0.0447	0.8657	0.8878	0.7372	0.0577	0.8185	0.8576
DPORTNet	**0.9186**	**0.0381**	**0.9143**	0.9202	**0.9051**	**0.0320**	**0.9080**	0.9476	**0.8236**	**0.0414**	**0.8702**	**0.8945**	0.7458	**0.0553**	**0.8205**	0.8539
DPORTNet-V	0.9156	0.0405	0.9126	**0.9209**	0.9037	0.0334	0.9053	**0.9488**	0.8130	0.0442	0.8648	0.8899	**0.7472**	0.0556	0.8192	**0.8569**

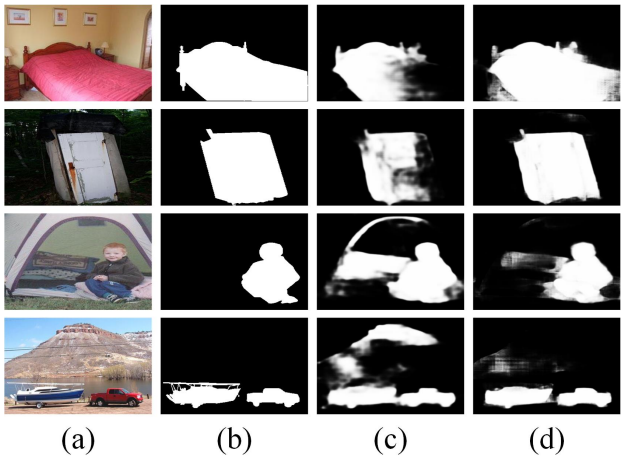


Fig. 9: Ablation visualization of different POR cues explorations for POR saliency. (a) Image; (b) GT; (c) TSPOANet [9]; (d) TSPOANet-DCR, which is implemented by replacing the two-stream capsule routing in TSPOANet [9] with our DCR.

to TSPOANet [9]. Besides, as shown in Fig. 9, compared to TSPOANet [9], our DCR improves the wholeness of the salient objects (as shown in the top two rows of Fig. 9) and background suppression (as shown in the bottom two rows of Fig. 9). To our best knowledge, capsules are much more complex than the neurons in conventional CNNs in terms of the number of parameters. Thus, the current training data may be sufficient for training CNN-based salient object detection models but becomes insufficient for training networks based on CapsNet. Under this circumstance, by reducing the routing complexity between capsules, our DCR can ease the optimization process of capsule routing, thus making the whole learning process much easier when training on the current data.

3) Multi-Scale POR Cues: Unlike the existing POR saliency methods that explore single-scale POR cues for saliency prediction, we explore multi-scale POR cues for saliency inference, which enables learning rich POR cues with different receptive fields of the input image. To understand the superiority of the proposed multi-scale POR cues, we compare our method with TSPOANet-DCR. As shown in Table I, our method improves the performance over TSPOANet-DCR that explores single-scale POR cues like TSPOANet [9]. Besides, as shown in Fig. 10, compared to the single-scale POR saliency method, *i.e.*, TSPOANet-DCR, our multi-scale POR cues achieve better background suppression (as shown in the top row of Fig. 10) and better object wholeness (as shown in the second row of Fig. 10). Furthermore, multi-scale POR cues help to detect salient objects of different sizes (as shown in the bottom two rows of Fig. 10).

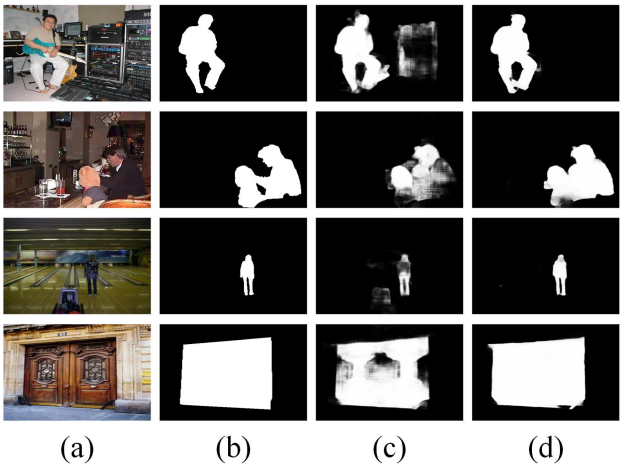


Fig. 10: Ablation visualization for mutli-scale POR cues. (a) Image; (b) GT; (c) TSPOANet-DCR; (d) DPORTNet. Compared to TSPOANet-DCR, our DPORTNet achieves better background suppression (top row) and better object wholeness (the second row). Furthermore, our DPORTNet helps to detect salient objects of different sizes (bottom two rows).

To have a deeper understanding of our multi-scale POR cues, we compare one-scale (*i.e.*, DPORTNet-OS), two-scale (*i.e.*, DPORTNet-TS), and three-scale (*i.e.*, DPORTNet) POR cues for saliency detection. As shown in the fourth block of Table I, our DPORTNet that extracts three-scale POR cues achieves superior performance compared to DPORTNet-OS and DPORTNet-TS. Moreover, as can be seen in Fig. 11, our three-scale DPORTNet, compared to DPORTNet-OS and DPORTNet-TS, can detect the whole salient objects while suppressing the background (as shown in the first three rows of Fig. 11), and identify multiple salient objects (as shown in the last row of Fig. 11), which thanks to the rich POR cues explored by using DCR at three scales.

4) DCR vs. Vanilla CapsNet: To better understand the ability of DCR for POR cues exploration, we compare two models, including our DPORTNet and DPORTNet-V, which is a modified version by replacing our DCR with the vanilla capsule routing at the last stage of ASPP. As shown in Table

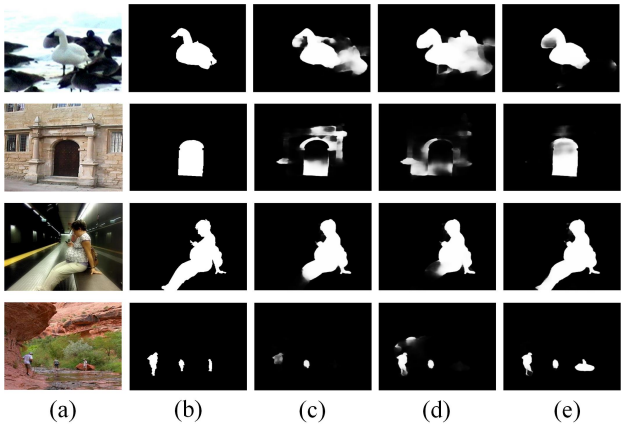


Fig. 11: Ablation visualization of different-scale DPORTNet. (a) Image; (b) GT; (c) DPORTNet-OS; (d) DPORTNet-TS; (e) DPORTNet. Compared to DPORTNet-OS and DPORTNet-TS, our DPORTNet can detect the whole salient objects while suppressing the background (first three rows), and identify multiple salient objects (last row), which thanks to the rich POR cues explored by using DCR at three scales.

I, by comparing DPORTNet and DPORTNet-V, it can be found that our DPORTNet beats DPORTNet-V on most of the evaluation metrics. It indicates that the sparse connection in our DCR is capable of capturing details, compared with the dense connection of the vanilla capsule routing. Fig. 12 visualizes the detection results. As shown in the first three rows of Fig. 12, the dense connection of vanilla capsule routing causes some details lost because of the noise of dense-position routing. In contrast, our DCR can make up for these lost details and detect the whole salient objects. As shown in the last row of Fig. 12, the dense connection of vanilla capsule routing misses one salient object, which can be identified by our DCR.

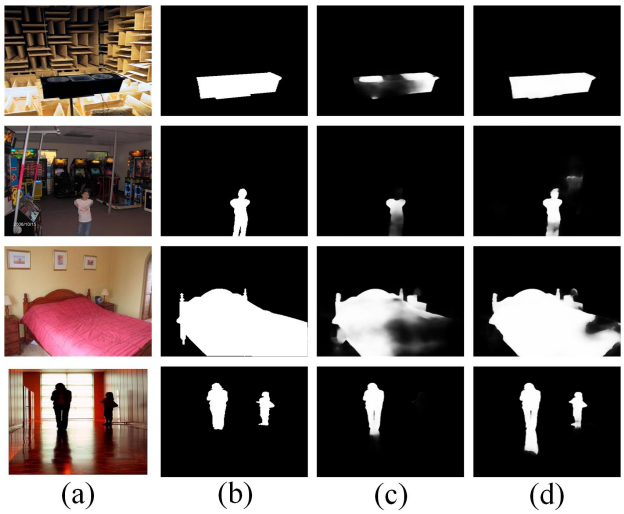


Fig. 12: Ablation visualization of DCR *vs.* vanilla capsule routing. (a) Image; (b) GT; (c) DPORTNet-V; (d) DPORTNet. DPORTNet-V causes some details lost and even salient object missed. In contrast, our DPOETNet can tackle these issues.

TABLE II: Inference time of different POR saliency methods. The input image of CapsNet, TSPOANet [9], TSPORTNet [10], TSPOANet-DCR, and DPORTNet is cropped to 352×352. DPORTNetv1 is a modified version of DPORTNet by cropping the input image to 176×176.

Method	CapsNet	TSPOANet [9]	TSPORTNet [10]	TSPOANet-DCR	DPORTNet	DPORTNetv1
Time (s)	0.53	0.32	0.35	0.07	0.06	0.04

5) Inference Speed of Different POR Saliency: To highlight the inference speed improvement of our proposed method, we list the inference time of different POR saliency methods in Table II. First, we replace the two-stream capsule routing in TSPOANet [9] with the original capsule routing [11] (called CapsNet) and the proposed DCR (called TSPOANet-DCR) for comparisons. As shown in Table II, our DCR achieves $5\times$ faster inference speed, compared to CapsNet and TSPOANet [9]. Secondly, compared with the existing POR saliency methods, *i.e.*, TSPOANet [9] and TSPORTNet [10], our method (*i.e.*, DPORTNet and DPORTNetv1) achieves $(5 \sim 9)\times$ faster inference speed. The speed improvement benefits from the proposed DCR that disentangles orthogonal 1D routing for fast POR cues exploration.

6) FLOPs of different methods: As shown in Fig. 13(a) and (b), when comparing solely POR saliency detection approaches, we reduce the number of parameters by 3.25M, 0.33M, and 2.51M, compared to CapsNet, TSPOANet [9], and TSPORTNet [10], respectively. Likewise, DPORTNet reduces FLOPs by 78.68G, 37G, and 106.74G when comparing with CapsNet, TSPOANet [9], and TSPORTNet [10], respectively. Overall, such reductions are quite substantial. We believe this is a significant improvement towards realizing a fast POR saliency modeling.

E. Comparison with the State-of-the-Art Methods

In this section, we compare our method with 18 state-of-the-art methods, including 2 POR saliency methods (TSPORTNet [10] and TSPOANet [9]) and 16 state-of-the-art saliency methods (PurNet [28], SCA [26], CIG [23], ITSD [64], SAMNet [63], ToHR [65], AFNet [66], BANet [67], JointCRF [68], NLDF [61], PiCANet [69], BMP [21], Amulet [20], UCF [70], DLS [71], and ELE [72]).

1) Quantitative Comparison: Table III lists the values of F_β, MAE, S_m, and E_m of different methods. Altogether, the proposed approach achieves 10 top-1, 12 top-2, and 14 top-3 places in terms of 16 metrics on four benchmarks. Specifically, our method outperforms the best general saliency method, *i.e.*, TSPORTNet [10], which is also the best POR saliency method and obtains 2 top-1, 7 top-2, and 12 top-3 places. Based on the above illustrations, we outperform the current state-of-the-art methods consistently across multiple test sets. Besides, Fig. 14 plots the PR curves on different datasets. Similar to Table III, as shown in Fig. 14, our method also achieves competitive performance compared with the other approaches.

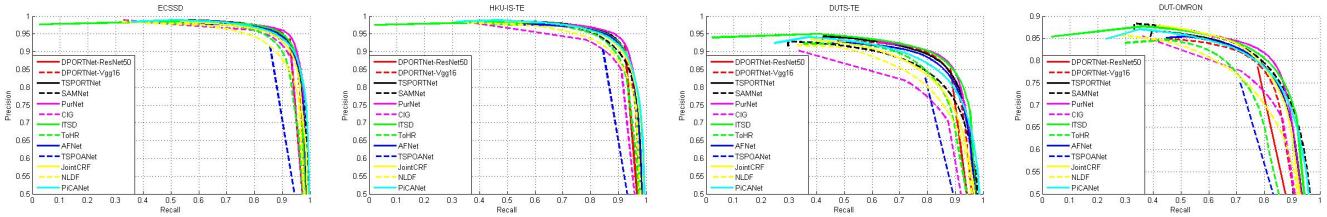
2) Visual Comparison: Fig. 15 shows the visual comparisons of different methods on various scenes, including large object, small object, multiple objects, low contrast between foreground and background, center bias, and complex scenes.

TABLE III: F_β, MAE, S_m, and E_m values of different methods. Top three methods are marked by red, blue, and green, respectively. "-" means that the corresponding authors do not provide the detection results of the dataset. In view of the fact that the compared methods use either Vgg16 [51] (*e.g.*, NLDF [61]) or ResNet50 [62] (*e.g.*, PurNet [28]) as the backbone networks, we list our performance using the ResNet50 [62] and Vgg16 [51] as the backbone networks for fair comparisons, *i.e.*, DPORTNet-ResNet50 and DPORTNet-Vgg16, respectively.

Benchmark	Metric	DPORTNet -ResNet50 (OURS)	DPORTNet -Vgg16 (OURS)	TSPORTNet [10]	SAMNet [63]	SCA [26]	PurNet [28]	CIG [23]	ITSD [64]	ToHR [65]	AFNet [66]	BANet [67]	TSPOANet [9]	JointCRF [68]	NLDF [61]	PiCANet [69]	BMP [21]	Amulet [20]	UCF [70]	DLS [71]	ELE [72]
ECSSD [53]	$F_\beta \uparrow$	0.9244	0.9186	0.9135	0.8913	0.8595	0.9210	0.9028	0.8746	0.9023	0.9076	0.9098	0.8873	0.8956	0.8783	0.8847	0.8682	0.8683	0.8439	0.8219	0.7545
	$MAE \downarrow$	0.0334	0.0381	0.0410	0.0501	0.0700	0.0347	0.0494	0.0401	0.0544	0.0418	0.0409	0.0515	0.0493	0.0626	0.0464	0.0447	0.0589	0.0690	0.0860	0.1201
	$S_m \uparrow$	0.9114	0.9143	0.9129	0.9071	0.8416	0.9245	0.8935	0.9142	0.8829	0.9134	0.9127	0.8684	0.9068	0.8747	0.9138	0.9108	0.8941	0.8834	0.8064	0.7426
	$E_m \uparrow$	0.9286	0.9202	0.9229	0.9114	0.8800	0.9252	0.9220	0.9168	0.9171	0.9180	0.9241	0.9020	0.9152	0.9095	0.9103	0.9137	0.9011	0.8923	0.8655	0.8201
HKU-IS [12]	$F_\beta \uparrow$	0.9117	0.9051	0.9010	0.8770	0.8539	0.8998	0.8732	0.8910	0.8923	0.8891	0.8871	0.8795	0.8817	0.8721	0.8698	0.8705	0.8426	0.8233	0.8081	0.7053
	$MAE \downarrow$	0.0281	0.0320	0.0324	0.0445	0.0597	0.0302	0.0466	0.0346	0.0420	0.0355	0.0362	0.0391	0.0394	0.0480	0.0415	0.0389	0.0501	0.0612	0.0696	0.1118
	$S_m \uparrow$	0.9071	0.9080	0.9091	0.8981	0.8417	0.9158	0.8663	0.9068	0.8827	0.9058	0.9030	0.8656	0.9032	0.8782	0.9054	0.9065	0.8860	0.8742	0.7986	0.7127
	$E_m \uparrow$	0.9536	0.9476	0.9502	0.9341	0.8959	0.9493	0.9267	0.9465	0.9357	0.9424	0.9433	0.9263	0.9384	0.9287	0.9329	0.9373	0.9122	0.9027	0.8788	0.8097
DUTS [54]	$F_\beta \uparrow$	0.8418	0.8236	0.8092	0.7448	0.7977	0.8173	0.7381	0.7977	0.7932	0.7924	0.7890	0.7971	0.7444	0.7389	0.7491	0.7453	0.6775	0.6307	-	0.5765
	$MAE \downarrow$	0.0361	0.0414	0.0433	0.0578	0.0657	0.0391	0.0709	0.0423	0.0512	0.0458	0.0460	0.0482	0.0588	0.0651	0.0541	0.0490	0.0846	0.1122	-	0.1272
	$S_m \uparrow$	0.8662	0.8702	0.8707	0.8487	0.8236	0.8810	0.7957	0.8771	0.8291	0.8670	0.8609	0.8202	0.8358	0.8163	0.8607	0.8616	0.8039	0.7823	-	0.6704
	$E_m \uparrow$	0.9041	0.8945	0.8877	0.8493	0.8676	0.8950	0.8480	0.8918	0.8835	0.8787	0.8773	0.8748	0.8477	0.8543	0.8518	0.8599	0.7939	0.7625	-	0.7479
DUT-OMRON [55]	$F_\beta \uparrow$	0.7448	0.7458	0.7436	0.7110	0.7816	0.7561	0.7026	0.7446	0.7079	0.7385	0.7310	0.7030	0.7379	0.6836	0.7100	0.6917	0.6472	0.6206	0.6453	0.5752
	$MAE \downarrow$	0.0541	0.0553	0.0579	0.0652	0.0634	0.0513	0.0746	0.0632	0.0660	0.0574	0.0614	0.0628	0.0574	0.0796	0.0679	0.0636	0.0976	0.1204	0.0895	0.1215
	$S_m \uparrow$	0.8062	0.8205	0.8230	0.8299	0.8175	0.8414	0.7822	0.8288	0.7718	0.8263	0.8229	0.7692	0.8207	0.7704	0.8264	0.8093	0.7805	0.7599	0.7249	0.6763
	$E_m \uparrow$	0.8483	0.8539	0.8557	0.8399	0.8754	0.8683	0.8260	0.8552	0.8411	0.8533	0.8508	0.8232	0.8571	0.8162	0.8344	0.8375	0.7787	0.7647	0.8016	0.7502

TABLE IV: Parameters, FLOPs, and speed of some good methods.

Metric	DPORTNetv1 (OURS)	DPORTNet (OURS)	TSPORTNet [10]	ITSD [64]	AFNet [66]	BANet [67]	TSPOANet [9]	NLDF [61]	PiCANet [69]	BMP [21]	Amulet [20]	UCF [70]
Parameter (M)↓	18.84	18.84	21.35	17.08	37.11	55.90	19.17	35.49	32.85	23.98	33.15	23.98
FLOPs (G) ↓	40.49	60.78	267.50	57.47	38.40	121.60	197.78	263.90	37.1	239.46	45.30	61.4
Speed (fps) ↑	25	16	3	48	21.6	12.5	3	18.5	15.6	22	9.7	12.0



(a) ECSSD [53] (b) HKU-IS-TE [12] (c) DUTS [54] (d) DUT-OMRON [55]

Fig. 14: PR curves of some good methods. Our method can achieve competitive performance compared with the other approaches.

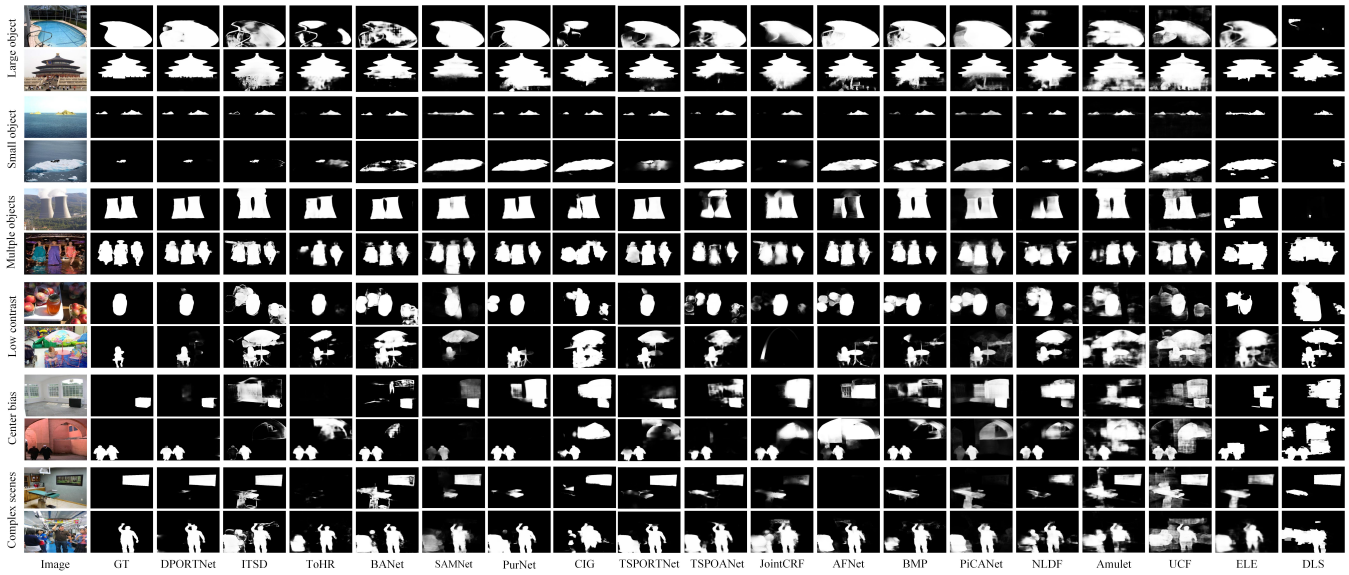
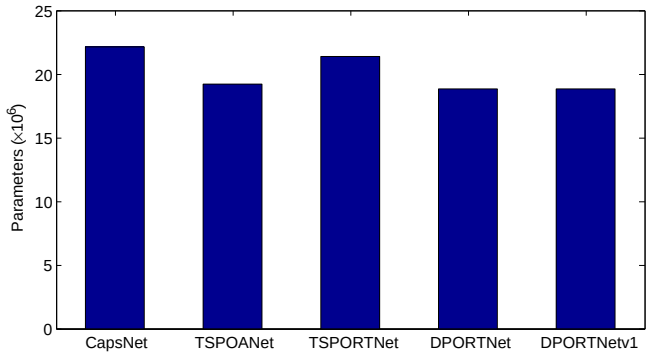
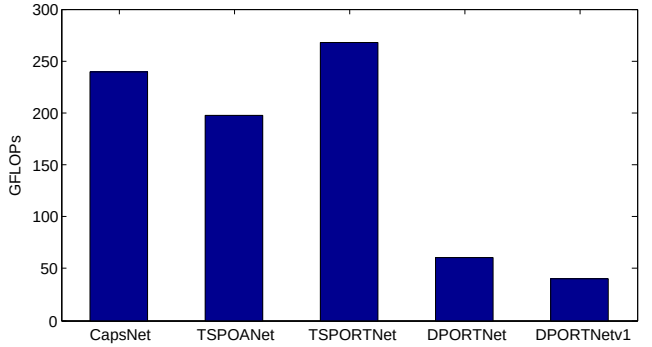


Fig. 15: Detection results of some good methods. We choose several scenes, including large object, small object, multiple objects, low contrast between foreground and background, center bias, and complex scenes, to visualize the detection results of different methods. Compared with the other methods, our model can detect the salient objects under various circumstances with good wholeness and uniformity.



(a) Parameters



(b) FLOPs

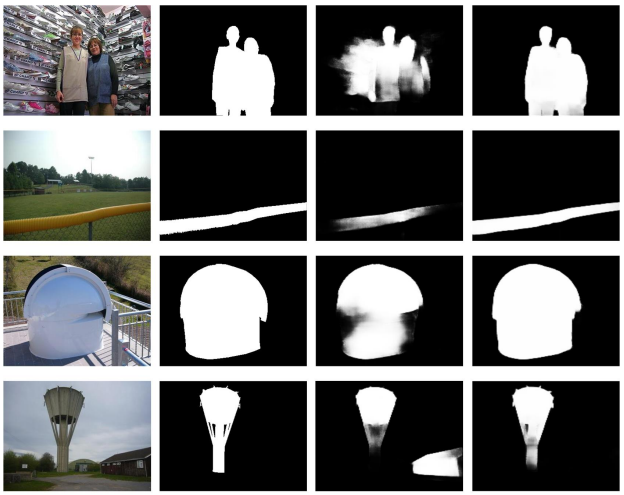
Fig. 13: Parameters (top) and FLOPs (bottom) of different POR saliency methods.

For large objects, our method can detect better object wholeness than the other methods. For small objects, we can locate the small objects and suppress the surrounding backgrounds, compared to the other methods. For multiple objects, our model can detect all the salient objects with good object wholeness and uniformity, while the other methods miss some object parts and introduce some background noise. For those objects with low contrast between themselves and backgrounds, we can identify the salient object from the misleading surroundings, while the other methods are easily confused by the similar backgrounds. For those objects with center biases, we can locate them accurately with good background suppression, while the other methods mostly introduce background noise at the center of the image into the saliency map. For those objects in complex scenes, the compared methods mostly fail to identify the salient object from the complicated backgrounds, which can be solved by our model well. In view of the above illustrations, our method can detect the salient object well in various scenes.

3) Parameters, FLOPs, and Speed: Table IV illustrates the parameters, FLOPs, and speed of some good methods. In Table IV, compared with the POR saliency detectors, including TSPORTNet [10] and TSPOANet [9], our methods have fewer parameters, significantly smaller FLOPs, and (5-9) faster inference speed. Besides, compared with the CNNs saliency methods, our methods also perform well with respect to parameters, FLOPs, and speed.

F. Plugging-in DCR for performance improvement

Our DCR can be easily plugged into any existing salient object detectors for further performance improvements by exploring the part-object relational semantics. To demonstrate it, we incorporate our DCR into NLDF [61], resulting in NLDF-DCR. Table V illustrates the performance of NLDF [61] and NLDF-DCR. Table V clearly shows that adding in DCR results in significant improvements on various datasets in terms of various evaluation metrics. Fig. 16 shows the visual improvements of plugging-in DCR. Compared with NLDF [61], our DCR helps to segment the whole salient objects (as shown in the first three rows of Fig. 16) while suppressing the confusing background (as shown in the last row of Fig. 16).



(a) (b) (c) (d)

Fig. 16: Visual illustration for the performance improvements of plugging-in DCR. (a) Image; (b) GT; (c) NLDF [61]; (d) NLDF-DCR.

G. Failure Cases

Fig. 17 displays some failure cases of our saliency detector on extremely complex scenes. For example, in the images in the left two columns of Fig. 17, the salient objects are labeled as parts of whole objects, but our saliency detector based on the part-object relationships detects the whole object instead. For those images in the right two columns of Fig. 17, the salient objects have poor objectness, which is a challenge for our method. In the future, we will study the relationships between the part-object relational property and saliency to improve the robustness to the above complicated cases.

VI. Conclusions

In this paper, we have proposed DPORTNet for fast POR saliency by involving the disentangled representation. Concretely, DCR was proposed to disentangle vertical 1D routing and horizontal 1D routing from the original omnidirectional 2D routing for fast POR cues exploration with network parameters and routing complexity reduction. Due to the lightweight capsule routing, DCR was carried out at multiple stages to

TABLE V: Performance improvements for plugging-in DCR.

	ECSSD [53]				HKU-IS [12]				DUTS [54]				DUT-OMRON [55]			
	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$	$F_\beta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$E_m \uparrow$
NLDF-DCR	**0.8989**	**0.0475**	**0.8985**	**0.9160**	**0.8961**	**0.0359**	**0.9025**	**0.9454**	**0.7947**	**0.0478**	**0.8542**	**0.8834**	**0.7272**	**0.0615**	**0.8008**	**0.8495**
NLDF [61]	0.8783	0.0626	0.8747	0.9095	0.8721	0.0480	0.8782	0.9287	0.7389	0.0651	0.8163	0.8543	0.6836	0.0796	0.7704	0.8162

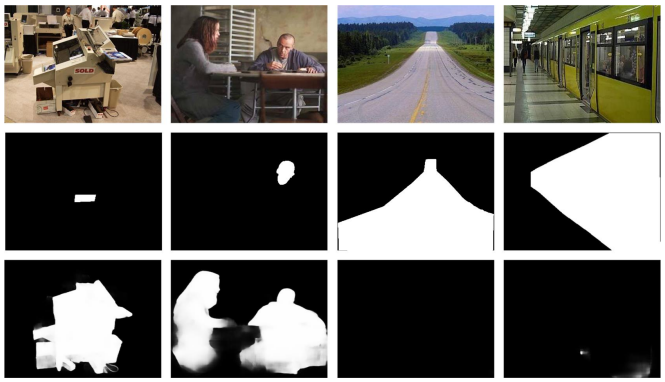


Fig. 17: Failure cases. From top to bottom: Images, GT, and results of our method.

explore multi-scale POR cues. Furthermore, a residual learning method is proposed to integrate contrast cues and POR cues for saliency prediction. Experiments have demonstrated the effectiveness and efficiency of the proposed method. In the future, we will take a further study on more primitive disentangled representation for capsule routing to explore more discriminative POR cues.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 62001341 and the National Natural Science Foundation of Jiangsu Province under Grant No. BK20221379.

REFERENCES

[1] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 141–145, 2006.

[2] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[3] J. Han, E. J. Pauwels, and P. De Zeeuw, "Fast saliency-aware multi-modality image fusion," *Neurocomputing*, vol. 111, pp. 70–80, 2013.

[4] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li, "Salient object detection for searched web images via global saliency," in *Proceedings of IEEE Confernece on Computer Vision and Pattern Recognition*, 2012, pp. 3194–3201.

[5] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 989–1005, 2009.

[6] T. Liu, J. Sun, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proceedings of the IEEE Confernece on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[7] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[8] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2021.3051099*, 2021.

[9] Y. Liu, Q. Zhang, D. Zhang, and J. Han, "Employing deep part-object relationships for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1232–1241.

[10] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2021.3053577*, 2021.

[11] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *Proceedings of the International Conference on Learning Representations*, 2018, pp. 3856–3866.

[12] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5455–5463.

[13] C. Fang, H. Tian, D. Zhang, Q. Zhang, J. Han, and J. Han, "Densely nested top-down flows for salient object detection," *arXiv preprint arXiv:2102.09133*, 2021.

[14] D. Zhang, H. Tian, and J. Han, "Few-cost salient object detection with adversarial-paced learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 236–12 247, 2020.

[15] D. Zhang, J. Han, Y. Zhang, and D. Xu, "Synthesizing supervision for learning deep saliency network without human annotation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1755–1769, 2019.

[16] A. K. Gupta, A. Seal, P. Khanna, E. Herrera-Viedma, and O. Krejcar, "Almnet: Adjacent layer driven multiscale features for salient object detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–14, 2021.

[17] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[19] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[20] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision*, 2017, pp. 202–211.

[21] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1741–1750.

[22] M. Ma12, C. Xia, and J. Li123, "Pyramidal feature shrinking for salient object detection," 2021.

[23] Y. Liu, J. Han, Q. Zhang, and C. Shan, "Deep salient object detection with contextual information guidance," *IEEE Transactions on Image Processing*, vol. 29, pp. 360–374, 2019.

[24] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3264–3274, 2018.

[25] A. K. Gupta, A. Seal, , P. Khanna, A. Yazidi, and O. Krejcar, "Gated contextual features for salient object detection," *Entropy*, vol. 70, 2021.

[26] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4156–4166.

[27] Z. Zhao, C. Xia, C. Xie, and J. Li, "Complementary trilateral decoder for fast and accurate salient object detection," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4967–4975.

[28] J. Li, J. Su, C. Xia, M. Ma, and Y. Tian, "Salient object detection with purificatory mechanism and structural similarity loss," *IEEE Transactions on Image Processing*, vol. 30, pp. 6855–6868, 2021.

[29] S. Yang, W. Lin, G. Lin, Q. Jiang, and Z. Liu, "Progressive self-guided loss for salient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 8426–8438, 2021.

[30] A. K. Gupta, A. Seal, M. Prasad, and P. Khanna, "Salient object detection techniques in computer visionła survey," *Entropy*, vol. 22, no. 10, p. 1174, 2020.

[31] 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 *Proceedings. of the AAAI Conference On Artificial Intelligence*, 2021, pp. 1–9.

[32] L. Tang, B. Li, Y. Zhong, S. Ding, and M. Song, "Disentangled high quality salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3580–3590.

[33] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proceedings of the International Conference on Artificial Neural Networks*, 2011, pp. 44–51.

[34] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

[35] A. R. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, "Stacked capsule autoencoders," in *Advances in Neural Information Processing Systems*, 2019, pp. 1–11.

[36] T. Hahn, M. Pyeon, and G. Kim, "Self-routing capsule networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 1–10.

[37] K. Ahmed and L. Torresani, "Star-caps: Capsule networks with straight-through attentive routing." in *Advances in Neural Information Processing Systems*, 2019, pp. 9098–9107.

[38] J. E. Lenssen, M. Fey, and P. Libuschewski, "Group equivariant capsule networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 1–10.

[39] S. Verma and Z.-L. Zhang, "Graph capsule convolutional neural networks," *arXiv preprint arXiv:1805.08090*, 2018.

[40] K. Duarte, Y. S. Rawat, and M. Shah, "Capsulevos: Semi-supervised video object segmentation using capsule routing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8480–8489.

[41] S. Ramasinghe, C. Athuraliya, and S. H. Khan, "A context-aware capsule network for multi-label classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 1–9.

[42] R. LaLonde and U. Bagci, "Capsules for object segmentation," *arXiv preprint arXiv:1804.04241*, 2018.

[43] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[44] S. Choi, S. Lee, Y. Kim, T. Kim, and C. Kim, "Hi-cmd: hierarchical cross-modality disentanglement for visible-infrared person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 257–10 266.

[45] G. Yin, B. Liu, L. Sheng, N. Yu, X. Wang, and J. Shao, "Semantics disentangling for text-to-image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2327–2336.

[46] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7678–7687.

[47] F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu, "Disentangling features in 3d face shapes for joint face reconstruction and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5216–5225.

[48] A. Gilbert, J. Collomosse, H. Jin, and B. Price, "Disentangling structure and aesthetics for style-aware image completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1848–1856.

[49] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 474–11 484.

[50] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[51] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representation*, 2015, pp. 1–14.

[52] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 603–612.

[53] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.

[54] 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 *Proccedings of the IEEE Conference Computer Vision and Pattern Recognition*, 2017, pp. 136–145.

[55] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.

[56] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.

[57] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4548–4557.

[58] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018, pp. 698–704.

[59] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning," in *Operating System Design and Implementation*, 2016, pp. 265–283.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[61] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6609–6617.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[63] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "Samnet: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 3804–3814, 2021.

[64] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9141–9150.

[65] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7234–7243.

[66] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1623–1632.

[67] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3799–3808.

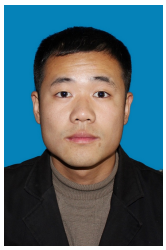
[68] Y. Xu, D. Xu, X. Hong, W. Ouyang, R. Ji, M. Xu, and G. Zhao, "Structured modeling of joint deep feature and prediction refinement for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3789–3798.

[69] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3089–3098.

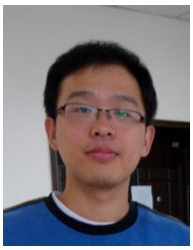
[70] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 212–221.

[71] P. Hu, B. Shuai, J. Liu, and G. Wang, "Deep level sets for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2300–2309.

[72] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, "What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4321–4329.



Yi Liu received the Ph. D. degree from Xidian University, China, in 2019. He is currently a Professor at Changzhou University. From 2018 to 2019, he was a visiting scholar at Lancaster University. His research interests include machine learning and computer vision, especially on saliency detection, capsule network, 3D point cloud, and object detection.



Dingwen Zhang received his Ph.D. degree from the Northwestern Polytechnical University, Xian, China, in 2018. He is currently a Professor in the School of Automation, Northwestern Polytechnical University. He is also an adjunct researcher at Hefei Comprehensive National Science Center, Institute of Artificial Intelligence, Hefei, China. From 2015 to 2017, he was a visiting scholar at the Robotic Institute, Carnegie Mellon University. His research interests include computer vision and multimedia processing, especially on saliency detection, video object segmentation, temporal action localization and weakly supervised learning.

Nian Liu is currently a research scientist with Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. He received the Ph.D. degree and the B.S. degree from the School of Automation at Northwestern Polytechnical University, Xi'an, China, in 2020 and 2012, respectively. His research interests include computer vision and deep learning, especially on saliency detection and few shot learning.

Shoukun Xu received the Ph. D. degree from China University of Mining and Technology, China in 2001. He is currently a Professor and the Vice President at Changzhou University. He is Chair of China Computer Federation Changzhou Branch. He is the distinguished member of China Computer Federation.His research interests include digital twins, computer vision and blockchain.

Jungong Han is currently a Chair Professor and the Director of the Research of Computer Science, Aberystwyth University, U.K. He also holds an Honorary Professorship with the University of Warwick, U.K. He is Fellow of the IAPR. His research interests include computer vision, artificial intelligence, and machine learning.