

# A Simple Pooling-Based Design for Real-Time Salient Object Detection

Jiang-Jiang Liu<sup>1\*</sup>   Qibin Hou<sup>1\*</sup>   Ming-Ming Cheng<sup>1 †</sup>   Jiashi Feng<sup>2</sup>   Jianmin Jiang<sup>3</sup>

<sup>1</sup>TKLNDST, College of CS, Nankai University   <sup>2</sup>NUS   <sup>3</sup>Shenzhen University

{j04.liu, andrewhoux}@gmail.com

## Abstract

We solve the problem of salient object detection by investigating how to expand the role of pooling in convolutional neural networks. Based on the U-shape architecture, we first build a global guidance module (GGM) upon the bottom-up pathway, aiming at providing layers at different feature levels the location information of potential salient objects. We further design a feature aggregation module (FAM) to make the coarse-level semantic information well fused with the fine-level features from the top-down pathway. By adding FAMs after the fusion operations in the top-down pathway, coarse-level features from the GGM can be seamlessly merged with features at various scales. These two pooling-based modules allow the high-level semantic features to be progressively refined, yielding detail enriched saliency maps. Experiment results show that our proposed approach can more accurately locate the salient objects with sharpened details and hence substantially improve the performance compared to the previous state-of-the-arts. Our approach is fast as well and can run at a speed of more than 30 FPS when processing a  $300 \times 400$  image. Code can be found at <http://mmcheng.net/poolnet/>.

## 1. Introduction

Benefiting from the capability of detecting the most visually distinctive objects from a given image, salient object detection plays an important role in many computer vision tasks, such as visual tracking [8], content-aware image editing [4], and robot navigation [5]. Traditional methods [11, 25, 14, 31, 2, 12, 41, 3] mostly rely on hand-crafted features to capture local details and global context separately or simultaneously, but the lack of high-level semantic information restricts their ability to detect the integral salient objects in complex scenes. Luckily, convolutional neural networks (CNNs) greatly promote the development of salient object detection models because of their capability of extracting both high-level semantic information and low-level

detail features in multiple scale space.

As pointed out in many previous approaches [9, 28, 44], because of the pyramid-like structural characteristics of CNNs, shallower stages usually have larger spatial sizes and keep rich, detailed low-level information while deeper stages contain more high-level semantic knowledge and are better at locating the exact places of salient objects. Based on the aforementioned knowledge, a variety of new architectures [9, 17, 38, 10] for salient object detection have been designed. Among these approaches, U-shape based structures [32, 22] receive the most attentions due to their ability to construct enriched feature maps by building top-down pathways upon classification networks.

Despite the good performance achieved by this type of approaches, there is still a large room for improving it. First, in the U-shape structure, high-level semantic information is progressively transmitted to shallower layers, and hence the location information captured by deeper layers may be gradually diluted at the same time. Second, as pointed out in [47], the receptive field size of a CNN is not proportional to its layer depth. Existing methods solve the above-mentioned problems by introducing attention mechanisms [46, 24] into U-shape structures, refining feature maps in a recurrent way [23, 46, 36], combining multi-scale feature information [9, 28, 44, 10], or add extra constraints to saliency maps like the boundary loss term in [28].

In this paper, different from the methods mentioned above, we investigate how to solve these problems by expanding the role of the pooling techniques in U-shape based architectures. In general, our model consists of two primary modules on the base of the feature pyramid networks (FPNs) [22]: a global guidance module (GGM) and a feature aggregation module (FAM). As shown in Fig. 1, our GGM composes of a modified version of pyramid pooling module (PPM) and a series of global guiding flows (GGFs). Unlike [37] which directly plugs PPM into the U-shape networks, our GGM is an individual module. More specifically, the PPM is placed on the top of the backbone to capture global guidance information (where the salient objects are). By introducing GGFs, high-level semantic information collected by PPM can be delivered to feature maps at

\*Indicates equal contributions.

†M.M. Cheng (cmm@nankai.edu.cn) is the corresponding author.

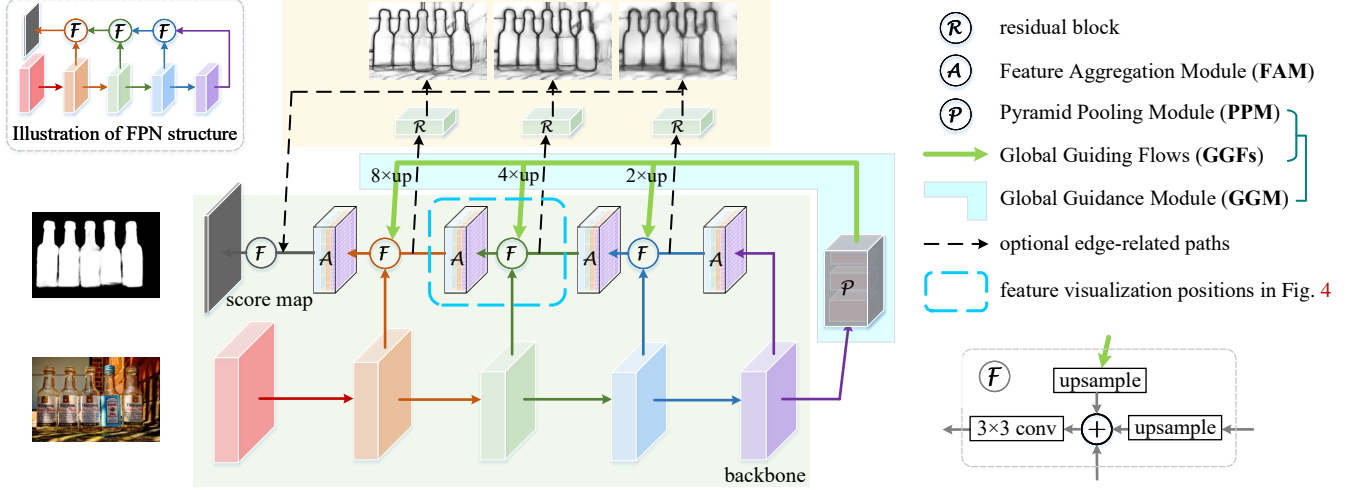


Figure 1. The overall pipeline of our proposed approach. For clarity, we also place a standard U-shape FPN structure [22] at the top-left corner. The top part for edge detection is optional.

all pyramid levels, remedying the drawback of U-shape networks that top-down signals are gradually diluted. Taking into account the fusion problem of the coarse-level feature maps from GGFs with the feature maps at different scales of the pyramid, we further propose a feature aggregation module (FAM), which takes the feature maps after fusion as input. This module first converts the fused feature maps into multiple feature spaces to capture local context information at different scales and then combines the information to weigh the compositions of the fused input feature maps better.

As both the above modules are based on the pooling techniques, we call our method PoolNet. *To the best of our knowledge, this is the first paper that aims at studying how to design various pooling-based modules to assist in improving the performance for salient object detection.* As an extension of this work, we also equip our architecture with an edge detection branch to further sharpen the details of salient objects by joint training our model with edge detection. To evaluate the performance of our proposed approach, we report results on multiple popular salient object detection benchmarks. Without bells and whistles, our PoolNet surpasses all previous state-of-the-art methods in a large margin. In addition, we conduct a series of ablation experiments to let readers better understand the impact of each component in our architecture on the performance and show how joint training with edge detection helps enhance the details of the predicted results.

Our network can run at a speed of more than 30 FPS on a single NVIDIA Titan Xp GPU for an input image with size  $300 \times 400$ . When the edge branch is not incorporated, training only takes less than 6 hours on a training set of 5,000 images, which is quite faster than most of the previous methods [24, 43, 28, 44, 45, 9]. This is mainly due

to the effective utilization of pooling techniques. PoolNet, therefore, can be viewed as a baseline to help ease future research in salient object detection.

## 2. Related Work

Recently, benefiting from the powerful feature extraction capability of CNNs, most of the traditional saliency detection methods based on hand-crafted features [3, 12, 20, 31] have been gradually surpassed. Li *et al.* [18] used the multi-scale features extracted from a CNN to compute the saliency value for each super-pixel. Wang *et al.* [34] adopted two CNNs, aiming at combining local super-pixel estimation and global proposal searching together, to produce saliency maps. Zhao *et al.* [48] presented a multi-context deep learning framework which extracts both local and global context information by employing two independent CNNs. Lee *et al.* [6] combined low-level heuristic features, such as color histogram and Gabor responses, with high-level features extracted from CNNs. All these methods take image patches as the inputs of CNNs and hence are time-consuming. Moreover, they ignore the essential spatial information of the whole input image.

To overcome the above problems, more research attentions are put on predicting pixel-wise saliency maps, inspired by the fully convolutional networks [27]. Wang *et al.* [36] generated saliency prior maps using low-level cues and further exploited it to guide the prediction of saliency recurrently. Liu *et al.* [23] proposed a two-stage network which produces coarse saliency maps first and then integrates local context information to refine them recurrently and hierarchically. Hou *et al.* [9] introduced short connections into multi-scale side outputs to capture fine details. Luo *et al.* [28] and Zhang *et al.* [44] both advanced the U-shape structures and utilized multiple levels of context in-

formation for accurate detection of salient objects. Zhang *et al.* [46] and Liu *et al.* [24] combined attention mechanisms with U-shape models to guide the feature integration process. Wang *et al.* [38] proposed a network to recurrently locate the salient object and then refine them with local context information. Zhang *et al.* [43] used a bi-directional structure to pass messages between multi-level features extracted by CNNs for better predicting saliency maps. Xiao *et al.* [39] adopted one network to tailor the distracting regions first and then used another network for saliency detection.

Our method is quite different from the above approaches. Instead of exploring new network architectures, we investigate how to apply the simple pooling techniques to CNNs to simultaneously improve the performance and accelerate the running speed.

### 3. PoolNet

It has been pointed out in [23, 9, 37, 38] that high-level semantic features are helpful for discovering the specific locations of salient objects. At the meantime, low- and mid-level features are also essential for improving the features extracted from deep layers from coarse level to fine level. Based on the above knowledge, in this section, we propose two complementary modules that are capable of accurately capturing the exact positions of salient objects and meanwhile sharpening their details.

#### 3.1. Overall Pipeline

We build our architecture based on the feature pyramid networks (FPNs) [22] which are a type of classic U-shape architectures designed in a bottom-up and top-down manner as shown at the top-left corner of Fig. 1. Because of the strong ability to combine multi-level features from classification networks [7, 33], this type of architectures has been widely adopted in many vision tasks, including salient object detection. As shown in Fig. 1, we introduce a global guidance module (GGM) which is built upon the top of the bottom-up pathway. By aggregating the high-level information extracted by GGM with into feature maps at each feature level, our goal is to explicitly notice the layers at different feature levels where salient objects are. After the guidance information from GGM is merged with the features at different levels, we further introduce a feature aggregation module (FAM) to ensure that feature maps at different scales can be merged seamlessly. In what follows, we describe the structures of the above mentioned two modules and explain their functions in detail.

#### 3.2. Global Guidance Module

FPNs provide a classic architecture for combining multi-level features from the classification backbone. However, because the top-down pathway is built upon the bottom-up

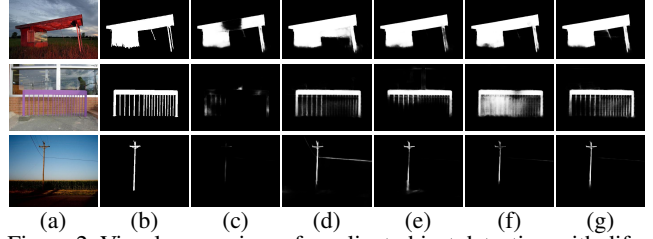


Figure 2. Visual comparisons for salient object detection with different combinations of our proposed GGM and FAMs. (a) Source image; (b) Ground truth; (c) Results of FPN baseline; (d) Results of FPN + FAMs; (e) Results of FPN + PPM; (f) Results of FPN + GGM; (g) Results of FPN + GGM + FAMs.

backbone, one of the problems to this type of U-shape architectures is that the high-level features will be gradually diluted when they are transmitted to lower layers. It is shown in [49, 47] that the empirical receptive fields of CNNs are much smaller than the ones in theory especially for deeper layers, so the receptive fields of the whole networks are not large enough to capture the global information of the input images. The immediate effect on this is that only parts of the salient objects can be discovered as shown in Fig. 2c. Regarding the lack of high-level semantic information for fine-level feature maps in the top-down pathway, we introduce a global guidance module which contains a modified version of pyramid pooling module (PPM) [47, 37] and a series of global guiding flows (GGFs) to explicitly make feature maps at each level be aware of the locations of the salient objects.

To be more specific, the PPM in our GGM consists of four sub-branches to capture the context information of the input images. The first and last sub-branches are respectively an identity mapping layer and a global average pooling layer. For the two middle sub-branches, we adopt the adaptive average pooling layer<sup>1</sup> to ensure the output feature maps of them are with spatial sizes  $3 \times 3$  and  $5 \times 5$ , respectively. Given the PPM, what we need to do now is how to guarantee that the guidance information produced by PPM can be reasonably fused with the feature maps at different levels in the top-down pathway.

Quite different from the previous work [37] which simply views the PPM as a part of the U-shape structure, our GGM is independent of the U-shape structure. By introducing a series of global guiding flows (identity mappings), the high-level semantic information can be easily delivered to feature maps at various levels (see the green arrows in Fig. 1). In this way, we explicitly increase the weight of the global guidance information in each part of the top-down pathway to make sure that the location information will not be diluted when building FPNs.

To better demonstrate the effectiveness of our GGM, we

<sup>1</sup><https://pytorch.org/docs/stable/nn.html#adaptiveavgpool2d>

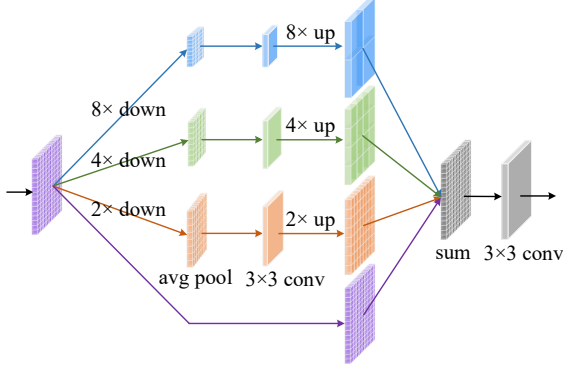


Figure 3. Detailed illustration of our feature aggregation module (FAM). It comprises four sub-branches, each of which works in an individual scale space. After upsampling, all sub-branches are combined and then fed into a convolutional layer.

show some visual comparisons. As depicted in Fig. 2c, we show some saliency maps produced by a VGGNet version of FPNs<sup>2</sup>. It can be easily found that with only the FPN backbone, it is difficult to locate salient objects for some complex scenes. There are also some results in which only parts of the salient object are detected. However, when our GGM is incorporated, the quality of the resulting saliency maps are greatly improved. As shown in Fig. 2f, salient objects can be precisely discovered, which demonstrates the importance of GGM.

### 3.3. Feature Aggregation Module

The utilization of our GGM allows the global guidance information to be delivered to feature maps at different pyramid levels. However, a new question that deserves asking is how to make the coarse-level feature maps from GGM seamlessly merged with the feature maps at different scales of the pyramid. Taking the VGGNet version of FPNs as an example, feature maps corresponding to  $C = \{C_2, C_3, C_4, C_5\}$  in the pyramid have downsampling rates of  $\{2, 4, 8, 16\}$  compared to the size of the input image, respectively. In the original top-down pathway of FPNs, feature maps with coarser resolutions are upsampled by a factor of 2. Therefore, adding a convolutional layer with kernel size  $3 \times 3$  after the merging operation can effectively reduce the aliasing effect of upsampling. However, our GGFs need larger upsampling rates (e.g., 8). It is essential to bridge the big gaps between GGFs and the feature maps of different scales effectively and efficiently.

To this end, we propose a series of feature aggregation modules, each of which contains four sub-branches as illustrated in Fig. 3. In the forward pass, the input feature map is first converted to different scale spaces by feeding it into

<sup>2</sup> Similarly to [22], we use the feature maps outputted by conv2, conv3, conv4, conv5 which are denoted by  $\{C_2, C_3, C_4, C_5\}$  to build the feature pyramid upon the VGGNet [33]. The channel numbers corresponding to  $\{C_2, C_3, C_4, C_5\}$  are set to  $\{128, 256, 512, 512\}$ , respectively.

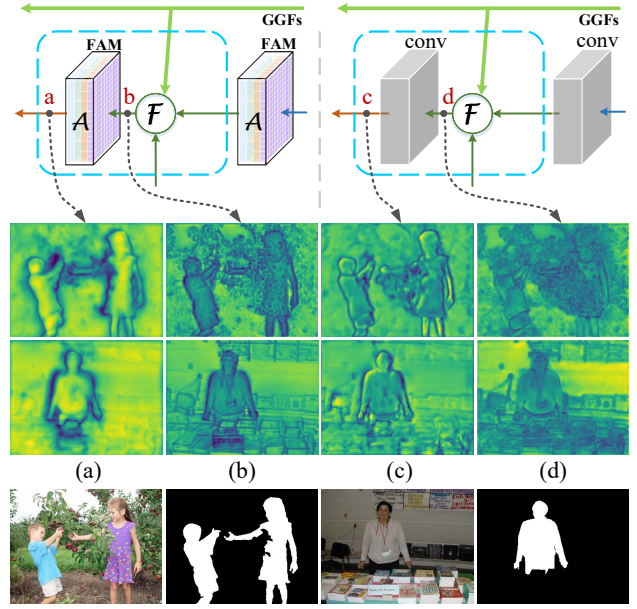


Figure 4. Visualizing feature maps around FAMs. Feature maps shown on the left are from models with FAMs, while feature maps displayed on the right are from models replacing FAMs with two convolution layers. The last row are source images and the corresponding ground-truth annotations. (a-d) are visualizations of feature maps at different places. As can be seen, when our FAMs are used, feature maps after FAMs can more precisely capture the location and detail information of salient objects (Column a), compared to those after two convolution layers (Column c).

average pooling layers with varying downsampling rates. The upsampled feature maps from different sub-branches are then merged together, followed by a  $3 \times 3$  convolutional layer.

Generally speaking, our FAM has two advantages. First, it assists our model in reducing the aliasing effect of upsampling, especially when the upsampling rate is large (e.g., 8). In addition, it allows each spatial location to view the local context at different scale spaces, further enlarging the receptive field of the whole network. *To the best of our knowledge*, this is the first work revealing that FAMs are helpful for reducing the aliasing effect of upsampling.

To verify the effectiveness of our proposed FAMs, we visualize the feature maps near the FAMs in Fig. 4. By comparing the left part (w/ FAMs) with the right part (w/o FAMs), feature maps after FAMs (Column a) can better capture the salient objects than those without FAMs (Column c). In addition to visualizing the intermediate feature maps, we also show some saliency maps produced by models with different settings in Fig. 2. By comparing the results in Column f (w/o FAMs) and Column g (w/ FAMs), it can be easily found that introducing FAM multiple times allows our network to better sharpen the details of the salient objects. This phenomenon is especially clear by observing the sec-



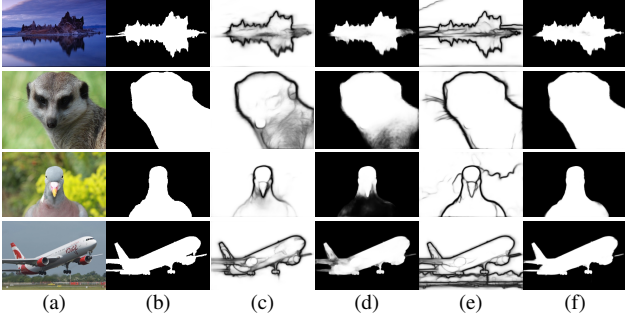


Figure 5. Visual results by joint training with edge detection. (a) Source image; (b) Ground truth; (c-d) Edge maps and saliency maps using the boundaries of salient objects as ground truths of the edge branch; (e-f) Edge maps and saliency maps by joint training with the edge dataset [1, 29]. By comparing the results in Column d and Column f, we can easily observe that joint training with high-quality edge datasets substantially improves the details of the detected salient objects.

ond row of Fig. 2. All the aforementioned discussions verify the significant effect of our FAMs on better fusing feature maps at different scales. In our experiment section, we will give more numerical results.

#### 4. Joint Training with Edge Detection

The architecture described in Sec. 3 has already surpassed all previous state-of-the-art single-model results on multiple popular salient object detection benchmarks. Despite so, by observing the resulting saliency maps produced by our model, we find out that many inaccurate (incomplete or over-predicted) predictions are caused by unclear object boundaries.

At first, we attempt to solve this problem by adding an extra prediction branch built upon the architecture presented in Sec. 3 to estimate the boundaries of the salient objects. The detailed structure can be found on the top side of Fig. 1. We add three residual blocks [7] after the FAMs at three feature levels in the top-down pathway, which are used for information transformation. These residual blocks are similar to the design in [7] and have channel numbers of  $\{128, 256, 512\}$  from the fine level to the coarse level. As done in [26], each residual block is then followed by a 16-channel  $3 \times 3$  convolutional layer for feature compression plus a one-channel  $1 \times 1$  convolutional layer for edge prediction. We also concatenate these three 16-channel  $3 \times 3$  convolutional layers and feed them to three consecutive  $3 \times 3$  convolutional layers with 48 channels to transmit the captured edge information to the salient object detection branch for detail enhancement.

Similar to [17], during the training phase, we use the boundaries of the salient objects as our ground truths for joint training. However, this procedure does not bring us any performance gain, and some results are still short of de-

tail information of the object boundaries. For example, as demoed in Column c of Fig. 5, the resulting saliency maps and boundary maps are still ambiguous for scenes with low contrast between the foreground and background. The reason for this might be that the ground-truth edge maps derived from salient objects still lack most of the detailed information of salient objects. They just tell us where the outermost boundaries of salient objects are, especially for cases where there are overlaps between salient objects.

Taking the aforementioned argument into account, we attempt to perform joint training with the edge detection task using the same edge detection dataset [1, 29] as in [26]. During training, images from the salient object detection dataset and the edge detection dataset are inputted alternatively. As can be seen in Fig. 5, joint training with the edge detection task greatly improves the details of the detected salient objects. We will provide more quantitative analysis in our experiment section.

### 5. Experimental Results

In this section, we first describe the experiment setups, including the implementation details, the used datasets and the evaluation metrics. We then conduct a series of ablation studies to demonstrate the impact of each component of our proposed approach on the performance. At last, we report the performance of our approach and compare it with previous state-of-the-art methods.

#### 5.1. Experiment Setup

**Implementation Details.** The proposed framework is implemented based on the PyTorch repository<sup>3</sup>. All the experiments are performed using the Adam [13] optimizer with a weight decay of  $5e-4$  and an initial learning rate of  $5e-5$  which is divided by 10 after 15 epochs. Our network is trained for 24 epochs in total. The backbone parameters of our network (e.g., VGG-16 [33] and ResNet-50 [7]) are initialized with the corresponding models pretrained on the ImageNet dataset [16] and the rest ones are randomly initialized. By default, our ablation experiments are performed based on the VGG-16 backbone and the union set of MSRA-B [25] and HKU-IS [18] datasets as done in [17] unless special explanations. We only use the simple random horizontal flipping for data augmentation. In both training and testing, the sizes of the input images are kept unchanged as done in [9].

**Datasets & Loss Functions.** To evaluate the performance of our proposed framework, we conduct experiments on 6 commonly used datasets, including ECSSD [41], PASCAL-S [21], DUT-OMRON [42], HKU-IS [18], SOD [30] and DUTS [35]. Sometimes, for convenience, we use the initials of the datasets as their abbreviations if there is no ex-

<sup>3</sup><https://pytorch.org>

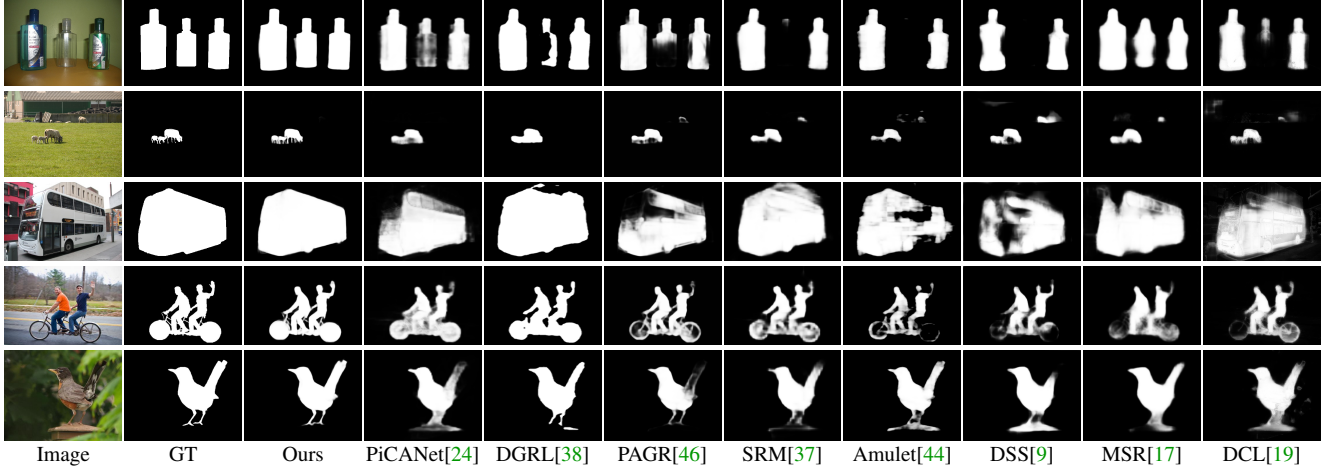


Figure 6. Qualitative comparisons to previous state-of-the-art methods. Obviously, compared to other methods, our approach is capable of not only locating the integral salient objects but also refining the details of the detected salient objects. This makes our resulting saliency map very close to the ground-truth annotations.

plitic conflict. We use standard binary cross entropy loss for salient object detection and balanced binary cross entropy loss [40] for edge detection.

**Evaluation Criteria.** We evaluate the performance of our approach and other methods using three widely-used metrics: precision-recall (PR) curves, F-measure score, and mean absolute error (MAE). F-measure, denoted as  $F_\beta$ , is an overall performance measurement and is computed by the weighted harmonic mean of the precision and recall:

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

where  $\beta^2$  is set to 0.3 as done in previous work to weight precision more than recall. The MAE score indicates how similar a saliency map  $S$  is compared to the ground truth  $G$ :

$$MAE = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |S(x, y) - G(x, y)| \quad (2)$$

where  $W$  and  $H$  denote the width and height of  $S$ , respectively.

## 5.2. Ablation Studies

In this subsection, we investigate the effectiveness of our proposed GGM and FAMs first. Then, we conduct more experiments on the configurations of our GGM and FAMs. Finally, we show the effect of joint training with edge detection on the performance.

**Effectiveness of GGM and FAMs.** To demonstrate the effectiveness of our proposed GGM and FAMs, we conduct ablation experiments based on the FPN baseline with the VGG-16 backbone. Except for different combinations of

No.	GGM + FAMs			DUT-O [42]		SOD [30]	
	PPM	GGFs	FAMs	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$
1				0.770	0.076	0.838	0.124
2	✓			0.783	0.071	0.847	0.125
3		✓		0.772	0.076	0.843	0.121
4	✓	✓		0.790	0.069	0.855	0.120
5			✓	0.798	0.065	0.852	0.118
6	✓	✓	✓	<b>0.806</b>	<b>0.063</b>	<b>0.861</b>	<b>0.117</b>

Table 1. Ablation analysis for the proposed architecture on two popular datasets. All experiments are based on the VGG-16 backbone and trained on the union set of MSRA-B [25] and HKU-IS [18]. By default, our baseline is the VGG-16 version of FPN [22]. As can be observed, each component in our architecture plays an important role and contributes to the performance. Best result in each column are highlighted in **red**.

GGM and FAMs, all other configurations are the same. Table 1 shows the performance on two challenging datasets: DUT-O and SOD. The corresponding visual comparisons can be found in Fig. 2.

- **GGM Only.** The addition of GGM (the 4th row in Table 1) gives performance gains in terms of both F-measure and MAE on the two datasets over the FPN baseline. The global guidance information produced by GGM allows our network to focus more on the integrity of salient objects, greatly improving the quality of the resulting saliency maps. Therefore, the details of the salient objects can be sharpened, which might be wrongly estimated as background for models with limited receptive fields (e.g., the last row in Fig. 2).
- **FAMs Only.** Simply embedding FAMs (the 5th row of Table 1) into the FPN baseline as shown in Fig. 1

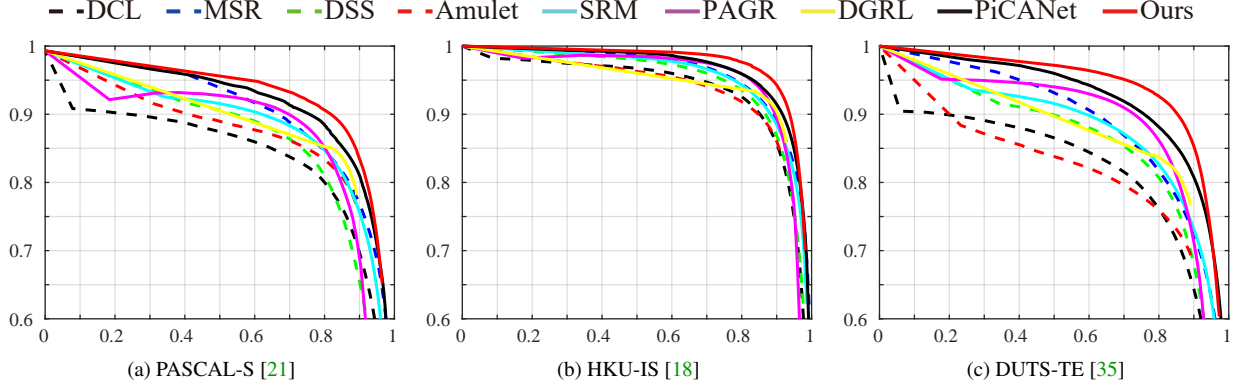


Figure 7. Precision (vertical axis) recall (horizontal axis) curves on three popular salient object datasets.

also helps improve the performance on both F-measure and MAE scores on the same two datasets. This might be because the pooling operations inside FAMs also enlarge the receptive field of the whole network compared to the baseline, and the FPN baseline still needs to merge feature maps from different levels, which indicates the effectiveness of our FAMs for solving the aliasing effect of upsampling.

- **GGM & FAMs.** By introducing both GGM and FAMs into the baseline (the last row of Table 1), the performance compared to the above two cases can be further enhanced on both F-measure and MAE scores. This phenomenon demonstrates that our GGM and FAM are two complementary modules. The utilization of them allows our approach to possess the strong capability of accurately discovering the salient objects and refining the details as illustrated in Fig. 2. More qualitative results can be found in Fig. 6 as well.

**Configuration of GGM.** To have a better understanding of the constitution of our proposed GGM, we perform two ablation experiments, which correspond to the 2nd and 3rd rows of Table 1, respectively. We alternatively remove one of the PPM and GGFs while keeping the other one unchanged. As can be seen, both operations make the performance decline compared to the results with both of them considered (the 4th row). These numerical results indicate that both PPM and GGFs play an important role in our GGM. The absence of any one of them is harmful for the performance of our approach.

**The Impact of Joint Training.** To further improve the quality of saliency maps produced by our approach, we attempt to combine edge detection with salient object detection in a joint training manner. In Table 2, we list the results when two kinds of boundary information are considered. As can be seen, using the boundaries of salient objects as supervision results in no improvement while using standard boundaries can greatly boost the performance on all three

Settings	PASCAL-S [21]		DUT-O [42]		SOD [30]	
	MaxF	MAE	MaxF	MAE	MaxF	MAE
Baseline (B)	0.838	0.093	0.806	0.063	0.861	0.117
<b>B + SalEdge</b>	0.835	0.096	0.805	0.063	0.863	0.120
<b>B + StdEdge</b>	<b>0.849</b>	<b>0.077</b>	<b>0.808</b>	<b>0.059</b>	<b>0.872</b>	<b>0.105</b>

Table 2. Ablation analysis of our approach when different kinds of boundaries are used. The baseline here refers to the VGG-16 version of FPN plus GGM + FAMs. We also use the combination of MSRA-B [25] and HKU-IS [18] as the training set. ‘SalEdge’ refers to the boundaries of salient objects and ‘StdEdge’ refers to the standard datasets for edge detection, which include BSDS500 [1] and PASCAL VOC Context [29] as done in [26, 15].

datasets especially on the MAE metric. This indicates that involving detailed edge information is helpful for salient object detection.

### 5.3. Comparisons to the State-of-the-Arts

In this section, we compare our proposed PoolNet with 13 previous state-of-the-art methods, including DCL [19], RFCN [36], DHS [23], MSR [17], DSS [9], NLDF [28], UCF [45], Amulet [44], GearNet[10], PAGR [46], PiCANet [24], SRM [37], and DGRL [38]. For fair comparisons, the saliency maps of these methods are generated by the original code released by the authors or directly provided by them. Moreover, all results are directly from single-model test without relying on any post-processing tools and all the predicted saliency maps are evaluated with the same evaluation code.

**Quantitative Comparisons.** Quantitative results are listed in Table 3. We consider both VGG-16 [33] and ResNet-50 [7] as our backbones and show results on both of them. Additionally, we also conduct experiments on different training sets to eliminate the potential performance fluctuation. From Table 3, we can observe that our PoolNet surpasses almost all previous state-of-the-art results on all datasets with the same backbone and training set. Average speed (FPS) comparisons among different methods (tested in the same

Model	Training		ECSSD [41]		PASCAL-S [21]		DUT-O [42]		HKU-IS [18]		SOD [30]		DUTS-TE [35]	
	#Images	Dataset	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$	MaxF $\uparrow$	MAE $\downarrow$
<b>VGG-16 backbone</b>														
DCL [19]	2,500	MB	0.896	0.080	0.805	0.115	0.733	0.094	0.893	0.063	0.831	0.131	0.786	0.081
RFCN [36]	10,000	MK	0.898	0.097	0.827	0.118	0.747	0.094	0.895	0.079	0.805	0.161	0.786	0.090
DHS [23]	9,500	MK+DTO	0.905	0.062	0.825	0.092	-	-	0.892	0.052	0.823	0.128	0.815	0.065
MSR [17]	5,000	MB + H	0.903	0.059	0.839	0.083	0.790	0.073	0.907	0.043	0.841	0.111	0.824	0.062
DSS [9]	2,500	MB	0.906	0.064	0.821	0.101	0.760	0.074	0.900	0.050	0.834	0.125	0.813	0.065
NLDF [28]	3,000	MB	0.903	0.065	0.822	0.098	0.753	0.079	0.902	0.048	0.837	0.123	0.816	0.065
UCF [45]	10,000	MK	0.908	0.080	0.820	0.127	0.735	0.131	0.888	0.073	0.798	0.164	0.771	0.116
Amulet [44]	10,000	MK	0.911	0.062	0.826	0.092	0.737	0.083	0.889	0.052	0.799	0.146	0.773	0.075
GearNet[10]	5,000	MB + H	0.923	0.055	-	-	0.790	0.068	0.934	0.034	0.853	0.117	-	-
PAGR [46]	10,553	DTS	0.924	0.064	0.847	0.089	0.771	0.071	0.919	0.047	-	-	0.854	0.055
PiCANet [24]	10,553	DTS	0.930	0.049	0.858	0.078	0.815	0.067	0.921	0.042	0.863	<b>0.102</b>	0.855	0.053
PoolNet (Ours)	2,500	MB	0.918	0.057	0.828	0.098	0.783	0.065	0.908	0.044	0.846	0.124	0.819	0.062
PoolNet (Ours)	5,000	MB + H	0.930	0.053	0.838	0.093	0.806	0.063	<b>0.936</b>	<b>0.032</b>	0.861	0.118	0.855	0.053
PoolNet (Ours)	10,553	DTS	0.936	0.047	0.857	0.078	0.817	0.058	0.928	0.035	0.859	0.115	0.876	0.043
PoolNet <sup>†</sup> (Ours)	10,553	DTS	<b>0.937</b>	<b>0.044</b>	<b>0.865</b>	<b>0.072</b>	<b>0.821</b>	<b>0.056</b>	0.931	0.033	<b>0.866</b>	0.105	<b>0.880</b>	<b>0.041</b>
<b>ResNet-50 backbone</b>														
SRM [37]	10,553	DTS	0.916	0.056	0.838	0.084	0.769	0.069	0.906	0.046	0.840	0.126	0.826	0.058
DGRL [38]	10,553	DTS	0.921	0.043	0.844	0.072	0.774	0.062	0.910	0.036	0.843	0.103	0.828	0.049
PiCANet [24]	10,553	DTS	0.932	0.048	0.864	0.075	0.820	0.064	0.920	0.044	0.861	0.103	0.863	0.050
PoolNet (Ours)	10,553	DTS	0.940	0.042	0.863	0.075	0.830	0.055	0.934	0.032	0.867	<b>0.100</b>	0.886	0.040
PoolNet <sup>†</sup> (Ours)	10,553	DTS	<b>0.945</b>	<b>0.038</b>	<b>0.880</b>	<b>0.065</b>	<b>0.833</b>	<b>0.053</b>	<b>0.935</b>	<b>0.030</b>	<b>0.882</b>	0.102	<b>0.892</b>	<b>0.036</b>

MB: MSRA-B [25], MK: MSRA10K [3], DTO: DUT-OMRON [42], H: HKU-IS [18], DTS: DUTS-TR [35].

Table 3. Quantitative salient object detection results on 6 widely used datasets. The best results with different backbones are highlighted in **blue** and **red**, respectively. <sup>†</sup>: joint training with edge detection. As can be seen, our approach achieves the best results on nearly all datasets in terms of F-measure and MAE.

	Ours	PiCANet [24]	DGRL [38]	SRM [37]	Amulet [44]
Size	400 × 300	224 × 224	384 × 384	353 × 353	256 × 256
FPS	32	7	8	14	16
	UCF [45]	NLDF [28]	DSS [9]	MSR [17]	DHS [23]
Size	224 × 224	400 × 300	400 × 300	400 × 300	224 × 224
FPS	23	12	12	2	23

Table 4. Average speed (FPS) comparisons between our approach (ResNet-50, w/ edge) and the previous state-of-the-art methods.

environment) are also reported in Table 4. Obviously, our approach runs in real time and faster than other methods.

**PR Curves.** Other than numerical results, we also show the PR curves on three datasets as shown in Fig. 7. As can be seen, the PR curves by our approach (red ones) are especially outstanding compared to all other previous approaches. As the recall score approaches 1, our precision score is much higher than other methods. This phenomenon reveals that the false positives in our saliency map are low.

**Visual Comparisons.** To further explain the advantages of our approach, we show some qualitative results in Fig. 6. From top to bottom, the images correspond to scenes with transparent objects, small objects, large objects, complex texture, and low contrast between foreground and background, respectively. It can be easily seen that our approach can not only highlight the right salient objects but also main-

tain their sharp boundaries in almost all circumstances.

## 6. Conclusion

In this paper, we explore the potentials of pooling on salient object detection by designing two simple pooling-based modules: global guidance module (GGM) and feature aggregation module (FAM). By plugging them into the FPN architecture, we show that our proposed PoolNet can surpass all previous state-of-the-art approaches on six widely-used salient object detection benchmarks. Furthermore, we also reveal that joint training our network with the standard edge detection task in an end-to-end learning manner can greatly enhance the details of the detected salient objects. Our modules are independent of network architectures and hence can be flexibly applied to any pyramid-based models. These directions also provide promising ways to improve the quality of saliency maps.

**Acknowledgements.** This research was supported by NSFC (61620106008, 61572264), the national youth talent support program, Tianjin Natural Science Foundation (17JCJJC43700, 18ZXZNGX00110) and the Fundamental Research Funds for the Central Universities (Nankai University, NO. 63191501).



## References

- [1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011. 5, 7
- [2] Ali Borji and Laurent Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, pages 478–485, 2012. 1
- [3] Ming Cheng, Niloy J Mitra, Xumin Huang, Philip HS Torr, and Song Hu. Global contrast based salient region detection. *IEEE TPAMI*, 2015. 1, 2, 8
- [4] Ming-Ming Cheng, Fang-Lue Zhang, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Repfinder: finding approximately repeated scene elements for image editing. *ACM TOG*, 29(4):83, 2010. 1
- [5] Celine Craye, David Filliat, and Jean-François Goudou. Environment exploration for object-based visual saliency learning. In *ICRA*, pages 2303–2309, 2016. 1
- [6] Lee Gayoung, Tai Yu-Wing, and Kim Junmo. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016. 2
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5, 7
- [8] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597–606, 2015. 1
- [9] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019. 1, 2, 3, 5, 6, 7, 8
- [10] Qibin Hou, Jiang-Jiang Liu, Ming-Ming Cheng, Ali Borji, and Philip HS Torr. Three birds one stone: A unified framework for salient object segmentation, edge detection and skeleton extraction. *arXiv preprint arXiv:1803.09860*, 2018. 1, 7, 8
- [11] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998. 1
- [12] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2083–2090, 2013. 1, 2
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5
- [14] Dominik A Klein and Simone Frintrap. Center-surround divergence of feature statistics for salient object detection. In *ICCV*, 2011. 1
- [15] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 7
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5
- [17] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, 2017. 1, 5, 6, 7, 8
- [18] Guanbin Li and Yizhou Yu. Visual saliency based on multi-scale deep features. In *CVPR*, pages 5455–5463, 2015. 2, 5, 6, 7, 8
- [19] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016. 6, 7, 8
- [20] Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013. 2
- [21] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 5, 7, 8
- [22] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2, 3, 4, 6
- [23] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016. 1, 2, 3, 7, 8
- [24] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. 1, 2, 3, 6, 7, 8
- [25] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2011. 1, 5, 6, 7, 8
- [26] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *CVPR*, 2017. 5, 7
- [27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2
- [28] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, 2017. 1, 2, 7, 8
- [29] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. 5, 7
- [30] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR*, pages 49–56, 2010. 5, 6, 7, 8
- [31] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012. 1, 2
- [32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015. 1
- [33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 4, 5, 7
- [34] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015. 2

- [35] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 5, 7, 8
- [36] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016. 1, 2, 7, 8
- [37] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017. 1, 3, 6, 7, 8
- [38] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. Detect globally, refine locally: A novel approach to saliency detection. In *CVPR*, pages 3127–3135, 2018. 1, 3, 6, 7, 8
- [39] Huaxin Xiao, Jiashi Feng, Yunchao Wei, Maojun Zhang, and Shuicheng Yan. Deep salient object detection with dense connections and distraction diagnosis. *IEEE Transactions on Multimedia*, 2018. 3
- [40] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 6
- [41] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 1, 5, 8
- [42] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. 5, 6, 7, 8
- [43] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *CVPR*, pages 1741–1750, 2018. 2, 3
- [44] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, 2017. 1, 2, 6, 7, 8
- [45] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, 2017. 2, 7, 8
- [46] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018. 1, 3, 6, 7, 8
- [47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 3
- [48] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015. 2
- [49] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015. 3