

Attentive Feedback Network for Boundary-Aware Salient Object Detection

Mengyang Feng¹, Huchuan Lu¹, and Errui Ding²

¹Dalian University of Technology, China

²Department of Computer Vision Technology (VIS), Baidu Inc.

mengyang_feng@mail.dlut.edu.cn, lhchuan@dlut.edu.cn, dingerrui@baidu.com

Abstract

Recent deep learning based salient object detection methods achieve gratifying performance built upon Fully Convolutional Neural Networks (FCNs). However, most of them have suffered from the boundary challenge. The state-of-the-art methods employ feature aggregation technique and can precisely find out wherein the salient object, but they often fail to segment out the entire object with fine boundaries, especially those raised narrow stripes. So there is still a large room for improvement over the FCN based models. In this paper, we design the Attentive Feedback Modules (AFMs) to better explore the structure of objects. A Boundary-Enhanced Loss (BEL) is further employed for learning exquisite boundaries. Our proposed deep model produces satisfying results on the object boundaries and achieves state-of-the-art performance on five widely tested salient object detection benchmarks. The network is in a fully convolutional fashion running at a speed of 26 FPS and does not need any post-processing.

1. Introduction

Different from other dense-labeling tasks, *e.g.* semantic segmentation and edge detection, the goal in salient object detection is to identify the visually distinctive regions or objects in an image and then segment the targets out. Such a useful processing is usually served as the first step to benefit other computer vision tasks including content-aware image editing [6] and image resizing [2], visual tracking [3], person re-identification [35] and image segmentation [7].

Along with the breakthrough of deep learning approaches, convolutional neural networks (CNNs, *e.g.* VGG [25] and ResNet [9]) trained for the image recognition task have been further developed to other computer vision fields via transfer learning. One successful transform is the fully convolutional neural networks (FCNs) in semantic segmentation. In order to perform predictions for every image pixel, [23, 24] usually bring in up-sampling operations by interpolation or learning deconvolutional filters be-

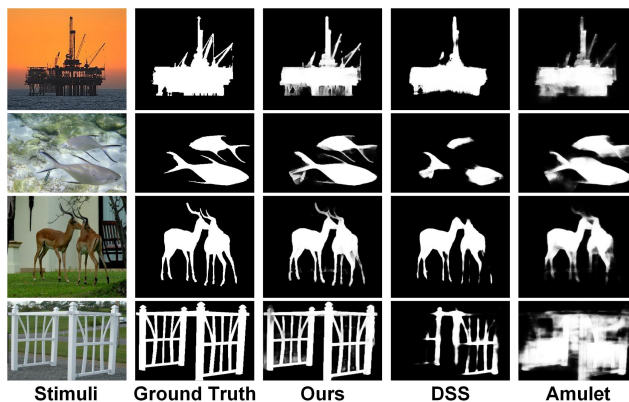


Figure 1. previous multi-scale aggregation methods fail to segment exquisite boundaries.

fore the inferences. Benefiting from such an effective and efficient approach, a large margin of performance gain for dense-labeling tasks has been achieved compared to previous traditional methods. However, the imperfection of the generic fully convolutional frameworks is that they suffer from the scale space problem. The repeated stride and pooling operations in CNN architectures result in the loss of essential fine details (low-level visual cues), which cannot be reconstructed by up-sampling operations.

To address the aforementioned problem, multi-scale mechanisms for conducting communications among hierarchical deep features are introduced to reinforce the spatial information, such as *skip-connections*, *short-connections* and *feature aggregations* proposed in [30], [10] and [33], respectively. These mechanisms are based on the observations that 1) deeper layers encode high-level knowledge and are good at locating objects; 2) shallower layers capture more spatial representations and can better reconstruct spatial details. Although these strategies have brought satisfactory improvements, they fail to predict the overall structures and have difficulties in detecting fine boundaries (see Fig. 1). For the sake of getting exquisite object boundaries, some of the researchers have to employ time-consuming CRF to refine the final saliency maps.

This paper concentrates on proposing a boundary-aware

network for salient object detection, which does not need any costly post-processing. We construct a novel encoder-decoder network in the fully convolutional style. First, we implement a Global Perception Module (GPM) on top of the encoder network to generate a low-resolution saliency map for roughly capturing the salient object. Then we introduce Attentive Feedback Modules (AFMs) which are built by adopting every encoder block and the corresponding decoder block to refine the coarse prediction scale-by-scale. The AFMs contribute to capturing the overall shape of targets. Moreover, a Boundary-Enhanced Loss (BEL)—serving to produce exquisite boundaries—is employed to assist the learning of saliency predictions on the object contours. Our model has the abilities to learn to produce precise and structurally complete salient object detection results, in the meanwhile, the contours of targets can be clearly cut out without post-processing.

Our main contributions are as follows:

- We propose an Attentive Feedback Network (AFNet) to deal with the boundary challenge in saliency detection. Multi-scale features from encoder blocks are transmitted into corresponding decoder ones and used to produce preferable segmentation results through the proposed Attentive Feedback Modules (AFMs). The AFNet learns to predict precise and structurally complete segmentations scale-by-scale and ends up getting the finest-resolution saliency map.
- We introduce a Boundary-Enhanced Loss (BEL) as an assistant to learn exquisite object contours. Thus the AFNet does not need any post-processing or extra parameters to refine the boundaries of the salient object.
- The proposed model can run at a real-time speed of 26 FPS and achieves state-of-the-art performance on five large-scale salient object detection datasets including ECSSD [31], PASCAL-S [19], DUT-OMRON [32], HKU-IS [15], and test set from DUTS [27].

2. Related Work

Multi-scale fusion methods. For the task of salient object detection, the earlier improvements [10, 33] are all benefited from the fully convolutional neural networks (FCNs). The authors attempt to find the optimal multi-scale fusion solutions to deal with the scale space problem caused by down-sampling operations. Both of [10, 33] conduct connections among hierarchical deep features into multiple subnetworks, and each predicts finest-resolution saliency map. The uncovering problem is that linking features from different layers may suffer from the boundary challenge. Although the features from deeper layers could help locate the target, the loss of spatial details might obstruct the features from shallower layers for recovering the object bound-

aries. A more proper way is to employ the multi-scale features in a coarse-to-fine fashion and gradually predict the final saliency map.

Coarse-to-fine solution. Considering that simply concatenating features from different scales may fail if disordered by the ambiguous information, coarse-to-fine solutions are employed in recent state-of-the-art methods such as RefineNet [20], PiCANet [22] and RAS [5]. The authors address this limitation by introducing a recursive aggregation method which fuses the coarse features to generate high-resolution semantic features stage-by-stage. In this paper, we similarly integrate hierarchical features from coarse to fine scales by constructing *skip-connections* between scale-matching encoder and decoder blocks. However, we think the weakness of the recursive aggregation method is that the coarse information may still mislead the finer one without proper guidance. Thus, we build Attentive Feedback Modules (AFMs) to guide the message passing among encoder and decoder blocks.

Attention models. Attention models are popularly used in recent neural networks to mimic the visual attention mechanism in the human visual system. The G-FRNet proposed by Islam *et al.* [11] applies gate units between each encoder and decoder blocks as attention models. These gate units control the feedforward message passing for the sake of filtering out ambiguous information. However, the message passing is controlled by Hadamard product, which means that once the previous stage makes a mistake, the inaccurate guidance and the overuse of these features may cause unexpected drift on segmenting the salient targets. To circumvent this barrier, our attentive feedback module uses ternary attention maps as the guidance between encoder and decoder blocks. Inspired by morphological dilation and erosion, the ternary attention map—indicating the confident foreground, confident background, and inconclusive regions—is constructed according to the initial saliency predictions. The experiments show that the inconclusive regions in the ternary attention map are mainly around the object boundaries. We apply the ternary attention maps on the input of each encoder block, and then generate the **updated multi-scale features** for saliency prediction so that the network can make further efforts on those inconclusive pixels. Thus, by employing the attention models, our network not only integrates the features among different stages by guidance but also has an opportunity for error corrections at each stage through the attentive feedback modules.

3. Proposed Method

In this paper, we propose an Attentive Feedback Network (AFNet) with novel multi-scale Attentive Feedback Modules and Boundary-Enhanced Loss to predict salient objects with entire structure and exquisite boundaries. The following subsections start from the backbone network first

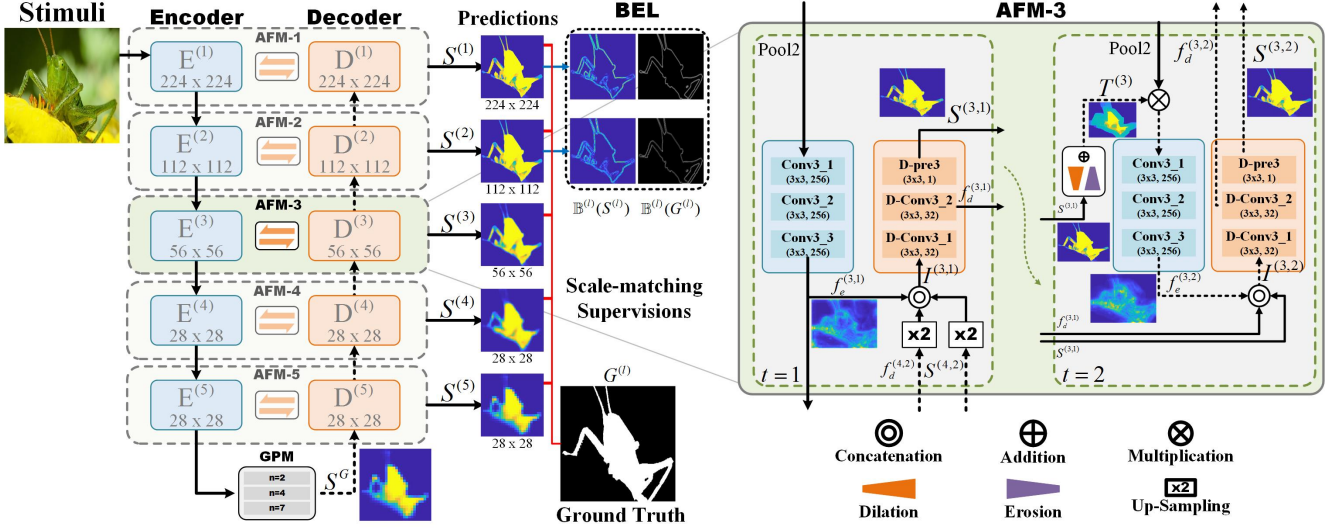


Figure 2. Network pipeline. Our network is in an encoder-decoder style, and we denote the l^{th} -scale encoder and decoder block as $E^{(l)}$ and $D^{(l)}$, respectively. The input image is first passed over $E^{(1)} \sim E^{(5)}$ to extract multi-scale convolutional features. Then, a Global Perception Module (GPM) is built on top of the $E^{(5)}$ to give a global saliency prediction S^G . The decoder network takes as inputs S^G and multi-scale convolutional features to generate finer saliency predictions $S^{(5)} \sim S^{(1)}$ scale-by-scale. We control the message passing among $E^{(l)}$ and $D^{(l)}$ through the Attentive Feedback Modules (AFMs, one illustration is on the right.), in which the built-in ternary attention maps $T^{(l)}$ guide the boundary-aware learning progress. We generate ground truth with multiple resolutions and use cross-entropy loss as supervision. Besides, in order to produce exquisite boundaries, extra Boundary-Enhanced Loss (BEL) is applied to the last two AFMs.

and then the detailed implementations of each component.

3.1. Network Overview

Similar to most previous approaches for salient object detection, we choose the VGG-16 [25] as our backbone network and develop it in an encoder-decoder style. The network illustration is shown in Fig. 2. Five pairs of encoder and decoder blocks are denoted as $E^{(l)}$ and $D^{(l)}$, respectively ($l \in \{1, 2, 3, 4, 5\}$ represents the scale).

Encoder Network. Similar to [14], we modify the VGG-16 network into a fully convolutional network by casting away the last two fully-connected layers along with the last pooling layer. In another hand, we skip the down-scaling operations before the last convolutional block $E^{(5)}$ and use dilated convolutions [4] with $rate=2$ for $E^{(5)}$ to maintain the original receptive field of the filters. We operate such surgery to avoid losing excessive spatial details.

Global Perception Module. The GPM, described in Sec. 3.2, takes the advantages of semantic resourceful features learned from $E^{(5)}$ and predicts the global saliency map S^G which will be fed into the decoder blocks for refinement.

Decoder Network. The decoder network comprises of five convolutional blocks. We apply $2 \times$ up-sampling layers between the decoder blocks to ensure having the same scale with corresponding encoder blocks. Each $D^{(l)}$ has three 3×3 convolutional layers with output number of 32, 32 and 1, respectively. Among the scale-matching pairs, the learned multi-level information can be transmitted through the Attentive Feedback Modules (AFMs) with

built-in ternary attention map $T^{(l)}$. We discuss the implementation details in Sec. 3.3. When training the network, every $D^{(l)}$ recursively estimates two saliency maps ($S^{(l,1)}$ and $S^{(l,2)}$) and each is supervised by the same scale ground truth $G^{(l)}$ via cross-entropy loss. Particularly, we add extra Boundary-Enhanced Loss (see Sec. 3.4) on $D^{(1)}$ and $D^{(2)}$ to enforce their distinguishing capacity on object boundaries.

3.2. Global Perception Module

As for global saliency prediction, Liu *et al.* [21] straightly employ a fully connected layer in its Global-View CNN. The problems are 1) the neighboring elements in the deepest layer have large overlapped receptive fields, meaning that a same pixel on the input image contributes a lot of redundant times to compute a single saliency value; (2) employing all pixels is useful for determining general locations, but the local patterns got lost. These facts motivate us to propose a Global Perception Module (see Fig. 3) to take full use of the local and global information.

Let \mathbf{X} of size $N \times N \times C$ be the feature maps extracted from $E^{(5)}$ (the channel number is reduced to $C = 16$ through one 1×1 convolution). We first split \mathbf{X} into $n \times n$ cells $\{\mathbf{x}_1, \dots, \mathbf{x}_{n \times n}\}$, and then conduct global convolution with kernel size $K_g \times K_g$ on \mathbf{X} to get the global features $\mathbf{F}_{(n)} \in \mathbb{R}^{N \times N \times C}$. The Fig. 3 gives an illustration when $n = 2$ and $K_g = 6$. As can be observed, in our global convolution operation, each element (the red one) in a certain cell is connected to $K_g \times K_g - 1$ ‘neighbors’, *i.e.* the other elements in blue in every cell. It is similar to introduce holes

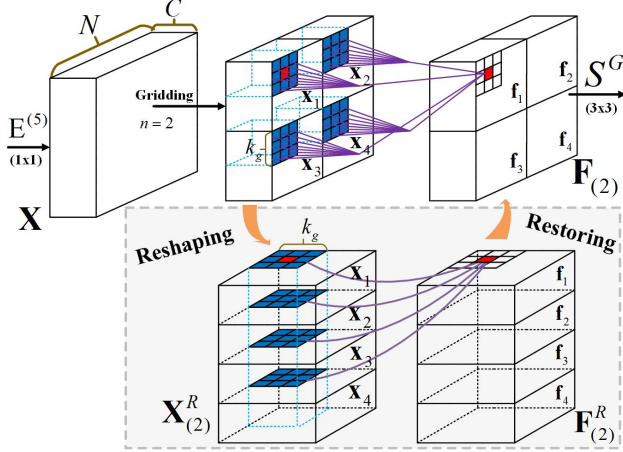


Figure 3. Illustration for Global Perception Module.

in dilated convolution. The difference is that we also consider the local neighbors. In this way, the local patterns and global diagram are simultaneously guaranteed. The global saliency map S^G is generated from $F_{(n)}$ and then delivered into decoder network for refinement along with the multi-scale convolutional features from $E^{(1)} \sim E^{(5)}$.

Implementation. We implement the global convolution in a step-by-step style. First, the split cells $\{x_1, \dots, x_{n \times n}\}$ are stored along the channels via concatenation, resulting a reshaped version $X_{(n)}^R$. Then, the global convolutional features $F_{(n)}$ can be generated through a $k_g \times k_g$ convolution on $X_{(n)}^R$ and restoring the results to size $N \times N \times C$. The final step in GPM is applying a 3×3 convolution to generate our global saliency prediction S^G .

3.3. Attentive Feedback Module

We control the message passing between scale-matching encoder and decoder blocks via Attentive Feedback Modules. The right part in Fig. 2 zooms in a detailed architecture, and the AFM works in a two-step recurrent style. To explain how it works more clearly, we illustrate the message passing streams in two time-steps using solid and dashed lines, respectively. We denote the features from $E^{(l)}$ as $f_e^{(l,t)}$, the input $I^{(l,t)}$, the features from $D^{(l)}$ as $f_d^{(l,t)}$, and output prediction as $S^{(l,t)}$, where t represents the time-step.

When $t = 1$, the decoder block $D^{(l)}$ takes as inputs $f_e^{(l,1)}$ from the l -th encoder block together with $S^{(l+1,2)}$ and $f_d^{(l+1,2)}$ from $D^{(l+1)}$. One 1×1 convolution is applied on $f_e^{(l,1)}$ to reduce its channels to 32 for saving memory. The outputs from $D^{(l+1)}$ are up-sampled by the factor of 2 for matching the spatial resolution with $f_e^{(l,1)}$. Then we concatenate all the input elements, and this formulates an attentive feature $I^{(l,1)}$ guided by the coarse prediction from the last scale. The refined prediction $S^{(l,1)}$ in the first time-step can be easily generated through three convolutional layers with batch normalization and ReLU. The whole stream is

illustrated in Fig. 2 (AFM-3, $t = 1$). However, after the first time-step refinement, we could not guarantee the quality of the results since the guidance from the former block involves an up-scaling operation which pulls in many inaccurate values, especially on object boundaries. Beyond that, supposing the previous block failed to segment out the entire target, the subsequent ones would never take a chance to perform a structurally complete detection.

The AFM provides an opportunity for error corrections using a ternary attention map in the second time-step feedback stream. We introduce to provide credible templates of foreground and background for reference. A proper way for our end-to-end training strategy is to exploit the refined prediction $S^{(l,1)}$ in the first time-step as a reference. Reviewing the morphological dilation and erosion, the former can gain weight for lightly drawn figures, and the latter is a dual operation which allows the thicker figures to get skinny. Motivated by that, we can ease the negative effects on boundaries by thinning down the salient regions through erosion. On the other hand, we can expand the salient regions to pull in more around pixels via dilation operation. Thus when $t = 2$, the ternary attention map—indicating the confident background, confident foreground, and inconclusive regions—is generated by operating dilation and erosion on $S^{(l,1)}$. We achieve morphological dilation $\mathbb{D}^{(l)}(\cdot)$ and erosion $\mathbb{E}^{(l)}(\cdot)$ utilizing the max-pooling operation $\mathbb{P}_{M_e^{(l)}}^{max}(\cdot)$, written as,

$$\begin{aligned} \mathbb{D}^{(l)}(S^{(l,1)}) &= \mathbb{P}_{M_d^{(l)}}^{max}(S^{(l,1)}), \\ \mathbb{E}^{(l)}(S^{(l,1)}) &= -\mathbb{P}_{M_e^{(l)}}^{max}(-S^{(l,1)}), \end{aligned} \quad (1)$$

where $M_d^{(l)}$ and $M_e^{(l)}$ represent the kernel size of the pooling layer at level l . The ternary attention map $T^{(l)}$ is then calculated as the average of $\mathbb{D}^{(l)}(S^{(l,1)})$ and $\mathbb{E}^{(l)}(S^{(l,1)})$. As a consequence, 1) the pixels' value in eroded saliency regions are approaching 1; 2) the margin between these two transformation have scores close to 0.5; 3) and the remaining areas are almost 0 as can be observed in Fig. 2 (AFM-3). Then the $T^{(l)}$ goes to weight the input of $E^{(l)}$ via pixel-wise multiplication and an updated attentive feature map $f_e^{(l,2)}$ is produced from the encoder. Likewise, the $S^{(l,1)}$, $f_d^{(l,1)}$ and $f_e^{(l,2)}$ are collected resulting the updated features $I^{(l,2)}$. In the end, the decoder block performs the refinement process once again to generate $S^{(l,2)}$, which has more outstanding boundaries, and goes to the next level. The whole stream is illustrated in Fig. 2 (AFM-3, $t = 2$). We take the output $S^{(1,2)}$ from the last decoder block as our final saliency map.

3.4. Boundary-Enhanced Loss

Along with the increased spatial resolution, the overall structure of objects gradually appears with the help of AFMs. Even though, the convolutional network still

holds a common problem that they usually generate blurred boundaries and has troubles in distinguishing the narrow background margins between two foreground areas (such as the space between two legs). We apply a Boundary-Enhanced Loss to work together with the cross-entropy loss for saliency detection to overcome this problem. The average-pooling operation $\mathbb{P}_{A^{(l)}}^{ave}(\cdot)$ with kernel size $A^{(l)}$ is employed to extract the smooth boundaries in the predictions. We avoid directly predicting the boundaries since it is really a tough task and the object contour map should be consistent with its saliency mask. We use $\mathbb{B}^{(l)}(X)$ to denote the operation for producing object contour map given a saliency mask X , as follows,

$$\mathbb{B}^{(l)}(X) = |X - \mathbb{P}_{A^{(l)}}^{ave}(X)|, \quad (2)$$

where $|\cdot|$ remarks the absolute value function. We visualize the $\mathbb{B}^{(l)}(G^{(l)})$ and $\mathbb{B}^{(l)}(S^{(l,t)})$ in Fig. 2 (BEL). The loss function for $l = 1, 2$ can be written as,

$$\mathcal{L}(S^{(l,t)}, G^{(l)}) = \lambda_1 \cdot \mathcal{L}_{ce}(S^{(l,t)}, G^{(l)}) + \lambda_2 \cdot \mathcal{L}_e(\mathbb{B}^{(l)}(S^{(l,t)}), \mathbb{B}^{(l)}(G^{(l)})). \quad (3)$$

The first term $\mathcal{L}_{ce}(\cdot, \cdot)$ stands for the cross-entropy loss for saliency detection, while the second term is our Boundary-Enhanced Loss. $\mathcal{L}_e(\cdot, \cdot)$ represents the Euclidean loss. We use λ_1 and λ_2 to control the loss weights, and we set $\lambda_1 : \lambda_2 = 1 : 10$ to strengthen the learning progress on object contours in our implementation. For $l = 3, 4, 5$, the loss function just contains the first term, *i.e.* the cross-entropy loss for saliency detection. It is because that these layers do not maintain the details needed for recovering exquisite outlines. By extracting boundaries from the saliency predictions themselves, the boundary-enhanced loss enhances the model to take more efforts on boundaries.

4. Experiments

4.1. Datasets and Evaluation Metrics

We carry out experiments on five public salient object detection datasets which are **ECSSD** [31], **PASCAL-S** [19], **DUT-OMRON** [32], **HKU-IS** [15] and **DUTS** [27]. The first four are widely used in saliency detection field while the last DUTS dataset is a recently released large-scale benchmark with the explicit training (10533)/test (5019) evaluation protocol. We train our model on the training set from DUTS and test on its test set along with other four datasets. We evaluate the performance using the following metrics.

Precision-Recall curves. It is a standard metric to evaluate saliency performance. One should binarize the saliency map with a threshold sliding from 0 to 255 and then compare the binary maps with the ground truth.

Table 1. Parameter settings for AFM and BEL.

	l	5	4	3	2	1
AFM	$M_d^{(l)}$	11	11	13	13	15
	$M_e^{(l)}$	5	5	5	7	7
BEL	$A^{(l)}$	—	—	—	3	5

F-measure. As an overall measurement, it can be computed both from precision and recall by thresholding the saliency map via $2 \times$ mean saliency value, as follows:

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}, \quad (4)$$

where β^2 is set to 0.3 as suggested in [1] to emphasize the precision. We also report the maximum F-measure (F_β^{max}) computed from all precision-recall pairs.

Mean Absolute Error. MAE is a complementary to PR curves and measures the average difference between the prediction and the ground truth quantitatively in pixel level.

S-measure. S-measure is proposed by Fan *et al.* [8], and it can be used to evaluate non-binary foreground maps. This measurement simultaneously evaluates region-aware and object-aware structural similarity between the saliency map and ground truth.

4.2. Implementation Details

We do data augmentation by horizontal- and vertical-flipping and image cropping to relieve over-fitting inspired by Liu *et al.* [21]. When fed into the AFNet, each image is warped to size 224×224 and subtracted using a mean pixel provided by VGG net at each position.

Our system is built on the public platform Caffe [12] and the hyper-parameters are set as follows: We train our network on two GTX 1080 Ti GPUs for 40K iterations, with a base learning rate (0.01), momentum parameter (0.9) and weight decay (0.0005). The mini-batch is set to 8 on each GPU. The ‘step’ policy with $gamma = 0.5$ and $stepsize = 10K$ is used. The parameters of the first 13 convolutional layers in encoder network are initialized by the VGG-16 model [25] and their learning rates are multiplied by 0.1. For other convolutional layers, we initialize the weights using ‘gaussian’ method with $std = 0.01$. The SGD method is selected to train our neural networks.

4.3. Parameter Settings

Parameters for AFM and BEL. The Table. 1 shows the kernel size of pooling layers implemented in AFM and BEL. All of the strides are fixed to 1, and the padding widths are set for maintaining the spatial resolution. These parameters are adjusted following the observations: 1) for predictions with low-resolution, the ternary attention map should

Table 2. Quantitative comparisons. The best three results are shown in **red**, **green** and **blue**. The method DHS use 3500 images from DUT-OMRON to train, so its results here are excluded in this table. The *index* in the first column regards the publication year.

Method	ECSSD			PASCAL-S			DUT-OMRON			HKU-IS			DUTS-test		
	F_{β}^{max}	S_m	MAE	F_{β}^{max}	S_m	MAE	F_{β}^{max}	S_m	MAE	F_{β}^{max}	S_m	MAE	F_{β}^{max}	S_m	MAE
LEGS ₁₅	.827	.787	.118	.762	.725	.155	.669	.714	.133	.766	.742	.119	.655	.694	.138
RFCN ₁₆	.890	.860	.095	.837	.808	.118	.742	.774	.095	.892	.858	.079	.784	.792	.091
ELD ₁₆	.867	.839	.079	.773	.757	.123	.715	.750	.092	.739	.820	.074	.738	.753	.093
DCL ₁₆	.890	.828	.088	.805	.754	.125	.739	.713	.097	.885	.819	.072	.782	.735	.088
DS ₁₆	.882	.821	.122	.765	.739	.176	.745	.750	.120	.865	.852	.080	.777	.793	.090
DHS ₁₆	.907	.884	.059	.829	.807	.094	—	—	—	.890	.870	.053	.807	.817	.067
Amulet ₁₇	.915	.894	.059	.837	.820	.098	.742	.780	.098	.895	.883	.052	.778	.803	.085
DSS ₁₇	.916	.882	.052	.836	.797	.096	.771	.788	.066	.910	.879	.041	.825	.822	.057
C2SNet ₁₈	.911	.895	.053	.852	.838	.080	.757	.798	.072	.898	.887	.046	.809	.828	.063
RAS ₁₈	.921	.893	.056	.837	.795	.104	.786	.814	.062	.913	.887	.045	.831	.839	.060
DGRL ₁₈	.922	.903	.041	.854	.836	.072	.774	.806	.062	.910	.895	.036	.829	.841	.050
PiCANet ₁₈	.931	.914	.047	.868	.850	.077	.794	.826	.068	.921	.906	.042	.851	.861	.054
AFNet	.935	.914	.042	.868	.850	.071	.797	.826	.057	.923	.905	.036	.862	.866	.046

involve in enough regions in case of excluding the target object. Thus the kernel size should be relatively large to the spatial size. With the increasing of spatial resolution, we could decrease the kernel size cause the overall shape of targets could be recognized already; 2) the kernel size of the Erosion $M_e^{(l)}$ should be smaller than the kernel size of the Dilation $M_d^{(l)}$ because that we need to perceive as much as possible details around the boundary regions. The $M_d^{(l)}$, $M_e^{(l)}$ and $A^{(l)}$ are experimentally set according to the above observations.

Parameters for GPM. The kernel size $K_g = n \times k_g$ in the global convolution, and we fix the local convolutional kernel size k_g in GPM to 3. Regarding the number of split cells $n \times n$, we do ablation studies in Sec. 4.5. In our final implemented version, we employ the multi-scale strategy to form the global prediction module by combining 3 GPMs with different settings. Each GPM receives the features from $E^{(5)}$ as input, and their output features are concatenated for producing S^G through one 3×3 convolution.

4.4. Comparisons with State-of-the-art Results

We compared our algorithm with other 12 state-of-the-art deep learning methods, which are LEGS [26], RFCN [28], ELD [13], DCL [16], DS [18], DHS [21], Amulet [33], DSS [10], C2SNet [17], RAS [5], DGRL [29] and PiCANet [22]. The saliency maps of other methods are provided by the authors or computed by their released codes with default settings for fair comparisons.

Quantitative Evaluation. 1) We evaluate our saliency maps using the standard PR curves in Fig. 4. In the first two rows, the five figures compare the proposed method (red) with other state-of-the-art algorithms. As can be observed, our method performs comparably with PiCANet and much better than the other algorithms. In the last two rows, we calculate PR curves on image boundaries to prove

our boundary-aware approach. The ground truth boundary mask is obtained by subtracting the dilated saliency mask with the eroded one. The structuring element is a 5×5 diamond matrix. We can produce the predicted boundary map in the same way and then compute PR curves. The curves demonstrate that our predicted saliency map has finer object boundaries and can better capture the overall shapes than PiCANet and other methods. Note that PGRL and C2SNet also propose ways to refine the object boundaries. PGRL adopts extra parameters (the BRN) to refine the boundaries while C2SNet needs extra contour/edge ground truth to train another branch (contour branch). Our proposed AFNet does not need extra parameters or training data of edges and can produce saliency and contour maps using the same set of parameters. We also achieve better performance on the PR curves of the object boundaries. 2) Table. 2 presents maximum F-measure, S-measure, and MAE on five datasets. Our AFNet ranks comparable with PiCANet or even better, but is much faster. The speed (without post-processing) in FPS (frames-per-second) of PiCANet, DGRL, C2SNet, DSS, Amulet are 6, 4, 18, 13 and 12, respectively. Ours achieves the real-time speed of 26 FPS which is comparable with RAS (33 FPS) and DHS (28 FPS). Some methods like DCL and DSS apply CRF to refine their final saliency maps while our AFNet does not need any post-processing.

Qualitative Evaluation. We illustrate the visual comparison with other approaches in Fig. 5. In the first four rows, the raised narrow stripes such as tentacles and horns are highlighted in ours but missed in all other methods. Besides, AFNet can produce the knife-edge-shaped boundaries and more close to the ground truth compared with PGRL and C2SNet which use extra parameters or edge data to refine the boundaries. For the two arm-stretched girls in the last two rows, almost all the other methods generate blurred responses on arms while ours gives clear decisions.

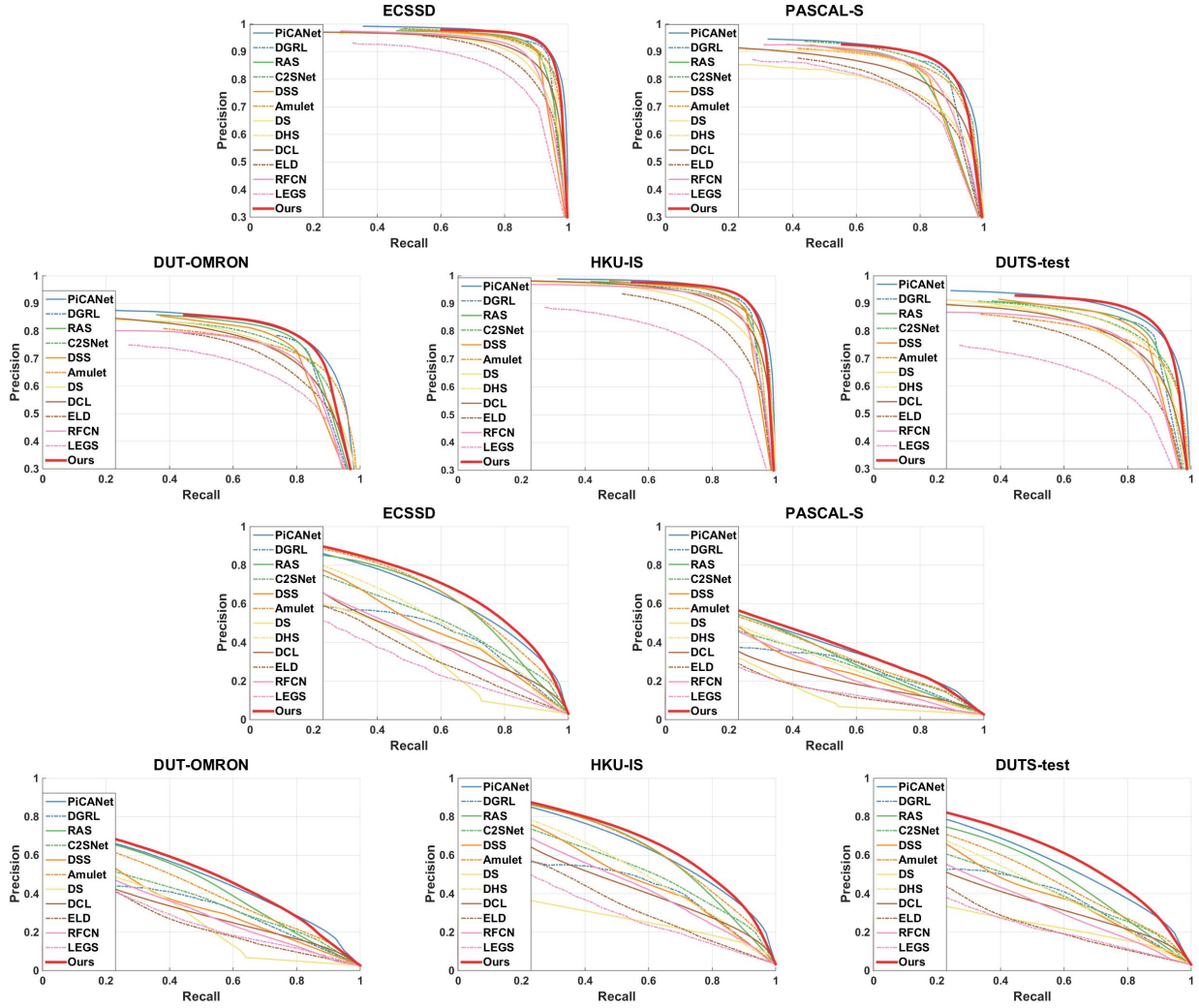


Figure 4. The PR Curves of the proposed algorithm and other state-of-the-art methods over five datasets. First two rows: evaluation on saliency maps. Last two rows: evaluation of boundary maps extracted from saliency predictions via dilation and erosion.

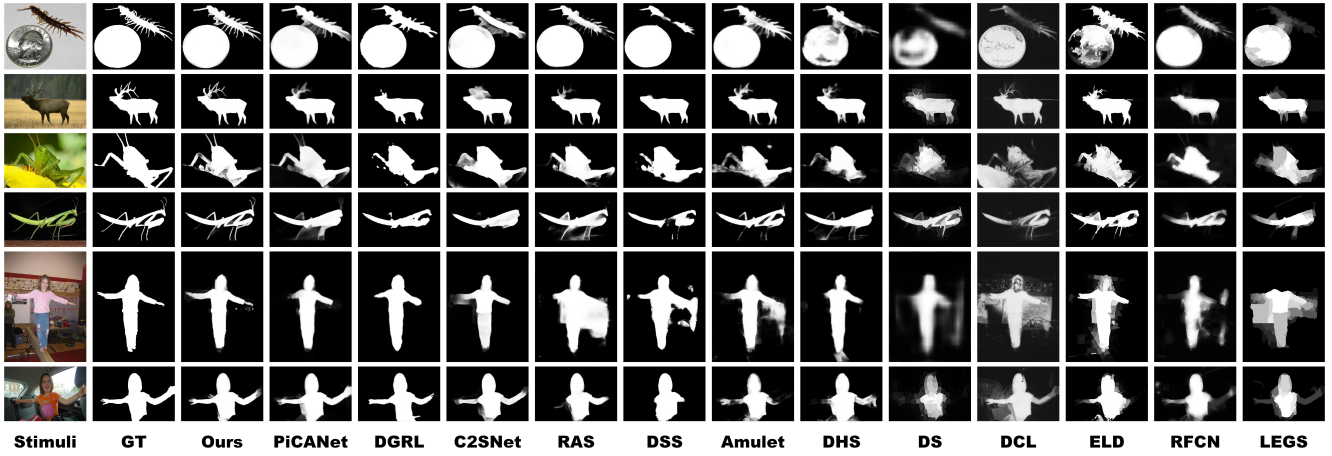


Figure 5. Visual comparison with state-of-the-art methods.

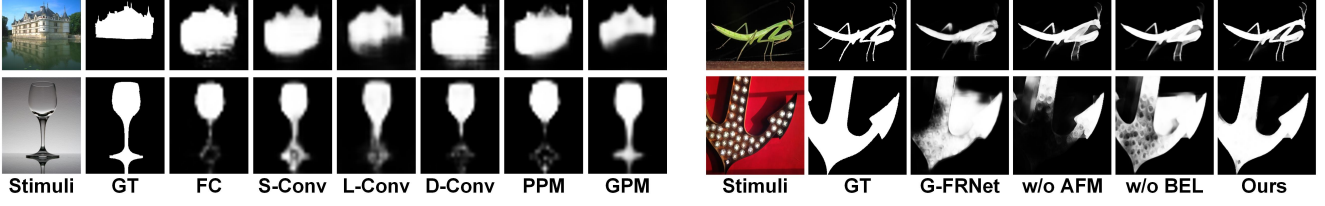


Figure 6. Visual comparison among different model settings. **Left columns** show the comparison of global models. **Right columns** illustrate the effectiveness of AFM and BEL.

Table 3. The effectiveness of Global Perception Module. (x) stands for the different number of split cells in GPMs.

	FC	S-Conv	L-Conv	D-Conv	PPM	GPM(2 ²)	GPM(4 ²)	GPM(7 ²)	GPM(2 ² + 4 ² + 7 ²)
mean IOU	.534	.523	.500	.541	.550	.551	.555	.552	.558
IOU(@F _β ^{max})	.571	.568	.560	.580	.585	.594	.597	.583	.603
F _β ^{max}	.769	.768	.759	.779	.781	.783	.787	.780	.791

Table 4. The effectiveness of Attentive Feedback Module (AFM) and Boundary-Enhanced Loss (BEL).

Model Name	Metric	Precision	Recall	F _β	F _β ^{max}	S _m	MAE	IOU(@F _β ^{max})	mean IOU
G-FRNet		.7573	.8709	.7579	.8319	.8381	.0562	.6667	.6500
AFNet	w/o AFM	.7742	.8899	.7771	.8586	.8621	.0512	.7069	.6960
	w/o BEL	.7829	.8907	.7872	.8557	.8618	.0493	.7140	.7001
	full implementation	.7925	.8941	.7974	.8624	.8664	.0461	.7213	.7186

4.5. Ablation Studies

We conduct ablation studies on DUTS-test dataset, and we use some extra metrics for a better demonstration.

The effectiveness of GPM. We design some simple modules to produce S^G for comparisons with GPMs. The results are shown in the Table. 3. We use some short names for convenience, *i.e.* FC: fully-connected layer; S-Conv: convolutional layer with small kernels (3×3); L-Conv: convolutional layer with large kernels (7×7); D-Conv: dilated convolutional layer with small kernels (3×3) and large rates ($rate = 7$); PPM: pyramid pooling module in PSPNet [34]. We calculate the F_{β}^{max} , the *intersection over union* (IOU) at F_{β}^{max} and the mean IOU. Our GPMs achieve better results and their combination performs best. The visual effects in Fig. 6 (**left columns**) also illustrate that GPM can better capture the overall shapes and local patterns.

The effectiveness of AFM and BEL. The Table. 4 shows add-on effectiveness of AFM and BEL. We also implement G-FRNet [11] for a better demonstration in this part, and the G-FRNet is also trained on DUTS-train dataset in the same environment. We calculate 8 evaluations—the F_{β} , the precision and recall at F_{β} , the three scores used in Table 2, the *intersection over union* (IOU) at F_{β}^{max} and the mean IOU—for detailed comparisons. As stated in Sec. 2, G-FRNet performs less than satisfactory on saliency detection as its gated unit drives the network excessively rely on the coarse results at previous stages, which might make mistakes. The

first row for ‘AFNet’ in the Table.4 excludes the feedback path with the trimap (the dashed lines, $t = 2$), the second row is the AFNet trained without the assistance of BEL, and the last row is our full implemented version. The AFM and BEL both contribute alone and perform better than the G-FRNet. From the visualizations in Fig. 6 (**right columns**) we could observe that AFM helps to recognize the structure of the targets while BEL takes charge of capturing boundary details.

5. Conclusion

In this paper, we have introduced a scale-wise solution for boundary-aware saliency detection. A novel and lightweight Global Perception Module is employed for global saliency prediction, then the encoder and decoder networks are communicated through Attentive Feedback Modules for refining the coarse prediction and predicting the final saliency map. The whole network can learn to capture the overall shape of objects, and experimental results demonstrate that the proposed architecture achieves state-of-the-art performance on five public saliency benchmarks. Our AFNet does not need any post-processing and runs at a real-time speed of 26 FPS.

Acknowledgements. This work was supported by the Natural Science Foundation of China under Grant 61725202, 61829102 and 61751212.

References

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.
- [2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007.
- [3] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti. Adaptive object tracking by learning background context. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 23–30, 2012.
- [4] 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. *arXiv:1606.00915*, 2016.
- [5] S. Chen, X. Tan, B. Wang, and X. Hu. Reverse attention for salient object detection. In *European Conference on Computer Vision*, 2018.
- [6] M. Cheng, F. Zhang, N. J. Mitra, X. Huang, and S. Hu. Repfinder: finding approximately repeated scene elements for image editing. *ACM Trans. Graph.*, 29(4):83:1–83:8, 2010.
- [7] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 817–824, 2009.
- [8] 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.
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [10] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] M. A. Islam, M. Rochan, N. D. B. Bruce, and Y. Wang. Gated feedback refinement network for dense image labeling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4877–4885, 2017.
- [12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pages 675–678, 2014.
- [13] G. Lee, Y. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–668, 2016.
- [14] G. Li, Y. Xie, L. Lin, and Y. Yu. Instance-level salient object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [15] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5455–5463, 2015.
- [16] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 478–487, 2016.
- [17] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen. Contour knowledge transfer for salient object detection. In *Proceedings of European Conference on Computer Vision*, 2018.
- [18] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930, 2016.
- [19] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.
- [20] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. 2016.
- [21] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.
- [22] 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*, pages 3089–3098, 2018.
- [23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [24] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
- [25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [26] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.
- [27] 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 *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [28] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *Proceedings of European Conference on Computer Vision*, pages 825–841, 2016.
- [29] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji. Detect globally, refine locally: A novel approach to saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3127–3135, 2018.

- [30] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.
- [31] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013.
- [32] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2013.
- [33] 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 International Conference on Computer Vision*, 2017.
- [34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [35] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593, 2013.