This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Label Decoupling Framework for Salient Object Detection

Jun Wei[1,2], Shuhui Wang[1]*, Zhe Wu[2,3], Chi Su[4], Qingming Huang[1,2,3], Qi Tian[5]
[1]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China [3]Peng Cheng Laboratory, Shenzhen, China
[4]Kingsoft Cloud, Beijing, China [5]Noah's Ark Lab, Huawei Technologies, China
jun.wei@vipl.ict.ac.cn, wangshuhui@ict.ac.cn, zhe.wu@vipl.ict.ac.cn
suchi@kingsoft.com, qmhuang@ucas.ac.cn, tian.qi1@huawei.com

Abstract

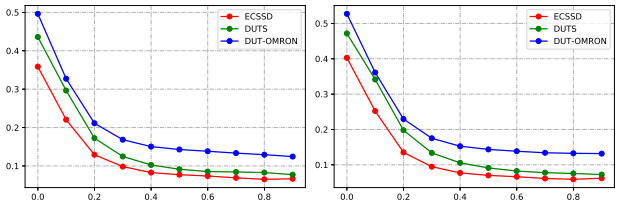
To get more accurate saliency maps, recent methods mainly focus on aggregating multi-level features from fully convolutional network (FCN) and introducing edge information as auxiliary supervision. Though remarkable progress has been achieved, we observe that the closer the pixel is to the edge, the more difficult it is to be predicted, because edge pixels have a very imbalance distribution. To address this problem, we propose a label decoupling framework (LDF) which consists of a label decoupling (LD) procedure and a feature interaction network (FIN). LD explicitly decomposes the original saliency map into body map and detail map, where body map concentrates on center areas of objects and detail map focuses on regions around edges. Detail map works better because it involves much more pixels than traditional edge supervision. Different from saliency map, body map discards edge pixels and only pays attention to center areas. This successfully avoids the distraction from edge pixels during training. Therefore, we employ two branches in FIN to deal with body map and detail map respectively. Feature interaction (FI) is designed to fuse the two complementary branches to predict the saliency map, which is then used to refine the two branches again. This iterative refinement is helpful for learning better representations and more precise saliency maps. Comprehensive experiments on six benchmark datasets demonstrate that LDF outperforms state-of-the-art approaches on different evaluation metrics. Codes can be found at https://github.com/weijun88/LDF.

1. Introduction

Salient object detection (SOD) [1, 6, 10, 11, 12] aims at identifying the most visually attractive objects or parts in an image or video, which is widely applied as a pre-processing procedure in downstream computer vision tasks [29, 35]. During the past decades, researchers have proposed hundreds of SOD methods based on hand-crafted features (*e.g.,* color, texture and brightness) [29]. However, these features can not capture high-level semantic information, which restricts their applications in complex scenes. Recently, convolutional neural networks (CNNs) have demonstrated powerful capability of feature representation and greatly promoted the development of SOD. Many CNNs-based methods [15, 39, 26, 27, 28, 40, 4, 21, 7, 42, 16, 32] have achieved remarkable performance by designing different decoders to aggregate multi-level CNN features. To get better feature representations, these methods focus on mining more context information and devising more effective feature fusion strategies. Besides, introducing the boundary

Table 1. Mean absolute error of the predicted saliency maps (MAE_{global}) and edge areas (MAE_{edge}) of two state-of-the-art methods over three datasets. MAE_{edge} is much larger than MAE_{global}, demonstrating that edge prediction is more difficult.

	EGNet [41]			SCRN [34]		
	ECSSD	DUTS	DUT-O	ECSSD	DUTS	DUT-O
MAE_{global}	0.037	0.039	0.053	0.037	0.040	0.056
MAE_{edge}	0.289	0.292	0.298	0.299	0.297	0.302



(a) EGNet [41] (b) SCRN [34]

Figure 1. Distribution of prediction error with respect to distance from pixel to its nearest edge. Horizontal coordinate represents the distance, which has been normalized to [0,1] and vertical coordinate is the prediction error. As can be seen, the closer the pixel is to the edge, the more difficult it is to be predicted.

*Corresponding author

information is another key point in SOD. Existing methods attempt to take edges as supervision to train SOD models, which significantly improves the accuracy of saliency maps [23, 20, 41, 34, 24, 13].

However, the imbalance between edge pixels and non-edge ones makes it hard to get good edge predictions. Therefore, directly taking edges as supervision may lead to suboptimal solutions. To better elaborate this statement, we calculate the mean absolute error (MAE) of two state-of-the-art methods (*i.e.,* EGNet [41] and SCRN [34]) over three SOD datasets (*i.e.,* ECSSD [36], DUTS [25] and DUT-O [37]) in Tab. 1. Though two methods get low error in global saliency prediction, they perform much worse in edge prediction, which shows that edge pixels are more difficult to predict than others. To further explore the prediction difficulties of pixels, we analyse the distribution of prediction error about the distance to the nearest edge of EGNet and SCRN in Fig. 1.

In Fig. 1, the prediction error curves gradually increases from far away to close to the edge (*i.e.,* the right axis to the left axis). When the distance is larger than 0.4, these curves rise slowly. However, when the distance gets smaller than 0.4, these curves begin to go upwards quickly. Based on this observation, we can divide each of the curves into two parts according to pixel distance from their nearest edges. Pixels near the edges correspond to much larger prediction errors than far-away pixels. These pixels with high prediction errors consists of both edge pixels and many other pixels close to edges that are ignored by recent edge-aware methods. Most of the hard pixels that can greatly improve the performance of SOD are not fully used, while using only edge pixels will lead to difficulties because of the imbalance distribution between edge pixels and background ones. In contrast, pixels far away from edges have relatively low prediction errors, which are much easier to be classified. However, traditional saliency labels treat all pixels inside salient object equally, which may cause pixels with low prediction errors to suffer distractive effects from those near edges.

We propose label decoupling framework to address the above problems. LDF mainly consists of a label decoupling procedure and a feature interaction network. As shown in Fig. 3, a saliency label is decomposed into a body map and a detail map by LD. Different from the pure edge map, the detail map consists of both edges as well as nearby pixels, which makes full use of pixels near edge and thus has a more balanced pixel distribution. The body map mainly concentrates on pixels far away from edges. Without the disturbance of pixels near edges, the body map can supervise the model to learn better representations. Accordingly, FIN is designed with two branches to adapt to body map and detail map respectively. The two complementary branches in FIN are fused to predict the saliency map, which is then used to refine the two branches again. This iterative refinement procedure is helpful for obtaining gradually accurate saliency maps prediction.

We conduct experiments on six popular SOD datasets and demonstrate the superior performance of LDF. In summary, our contributions are as follows:

- We analyse the shortcomings of edge-based SOD methods and propose a label decoupling procedure to decompose a saliency label into body map and detail map to supervise the model, respectively.
- We design a feature interaction network to make full use of the complementary information between branches. Both branches will be enhanced by iteratively exchanging information to produce more precise saliency maps.
- Extensive experiments on six SOD datasets show that our model outperforms state-of-the-art models by a large margin. In particularly, we demonstrate the good performance of LDF in different challenging scenes in the SOC dataset [8].

2. Related Work

During the past decades, a huge body of traditional methods have been developed for SOD. These methods [2, 3, 36] mainly rely on intrinsic cues (*e.g.*, color and texture) to extract features. However, these features cannot capture high-level semantic information and are not robust to variations, which limits their applications in complex scenarios. Recently, deep learning based models have achieved remarkable performance, which can be divided into aggregation-based models and edge-based models.

2.1. Aggregation-based Models

Most of the aggregation-based models adopt the encoder-decoder framework, where the encoder is used to extract multi-scale features and the decoder is used to integrate the features to leverage context information of different levels. Hou *et al.* [15] constructed shortcut connections on fully convolutional networks [22] and integrated features of different layers to output more accurate maps. Chen *et al.* [4] proposed a reverse attention network, which erased the current predicted salient regions to expect the network to mine out the missing parts. Deng *et al.* [7] designed an iterative strategy to learn the residual map between the prediction and ground truth by combining features from both deep and shallow layers. Wu *et al.* [33] found that features of shallow layers greatly increased the computation cost, but only brought little improvement in final results. Liu *et al.* [20] utilized simple pooling and a feature aggregation module to build fast and accurate model. Zhao *et al.* [42] introduced the channel-wise attention and spatial attention to extract valuable features and suppress background noise.

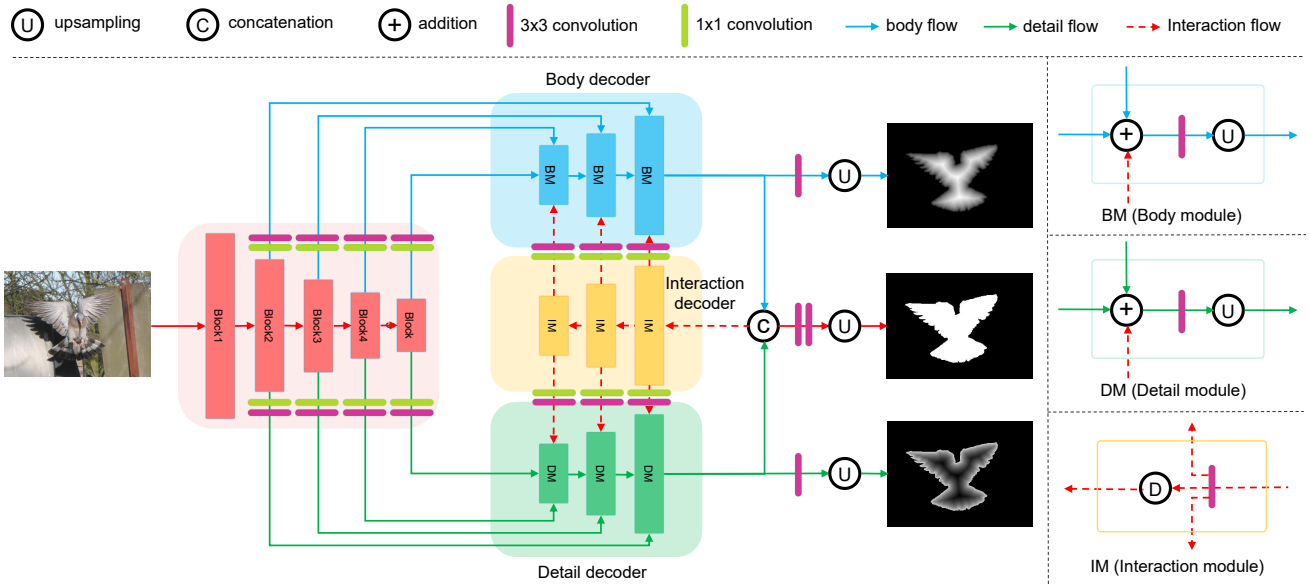


Figure 2. An overview of our proposed label decoupling framework (LDF). LDF is based on ResNet-50 [14] with supervision from body map, detail map and saliency map. LDF consists of two encoders and two decoders, *i.e.*, a backbone encoder for feature extraction, an interaction encoder for exchanging information, a body decoder and a detail decoder to generate body map and detail map respectively. The interaction encoder is not involved until body decoder and detail decoder output features.

Wang *et al.* [30] designed a top-down and bottom-up workflow to infer the salient object regions with multiple iterations. Liu *et al.* [21] proposed a pixel-wise contextual attention network to learn the context of each pixel, and combined the global context and local context for saliency prediction. Zhang *et al.* [38] designed a bi-directional message passing model for better feature selection and integration.

2.2. Edge-based Models

In addition to saliency masks, edge label is also introduced to SOD in [23, 34, 31, 20, 39, 42] to assist the generation of saliency maps. Zhang *et al.* [39] and Zhao *et al.* [42] directly built the edge loss with binary cross-entropy to emphasize the importance of boundaries. Qin *et al.* [23] designed a hybrid loss to supervise the training process of SOD on pixel-level, patch-level and map-level. Liu *et al.* [20] used additional edge dataset for joint training of both edge detection and SOD models. Feng *et al.* [13] applied a boundary-enhanced loss to generate sharp boundaries and distinguish the narrow background margins between two foreground areas. Li *et al.* [18] used a two-branch network to simultaneously predict the contours and saliency maps, which can automatically convert the trained contour detection model to SOD model. Wu *et al.* [34] investigated the logical inter-relations between segmentation and edge maps, which are then promoted to bidirectionally refine multi-level features of the two tasks. Although these methods take into account the relationship between edges and saliency maps, edge prediction is a hard task because of imbalanced pixel distribution. In this paper, we explicitly decouple the saliency label into body map and detail map, as shown in Fig. 3. Detail map helps model learn better edge features and body map decreases the distraction from pixels near edges to center ones.

3. Methodology

In this section, we first introduce the label decoupling method and give the specific steps to decompose the saliency map into body map and detail map. Then, to take advantage of the complementarity between features, we introduce FIN which facilitates the iterative information exchange between branches. The overview of the proposed model is shown in Fig. 2.

3.1. Label Decoupling

As described in Sec. 1, the prediction difficulty of a pixel is closely related to its position. Because of the cluttered background, pixels near the edge are more prone to be mispredicted. In comparison, central pixels have higher prediction accuracy due to the internal consistency of the salient target. Instead of treating these pixels equally, it will be more reasonable to deal with them according to their respective characteristics. Accordingly, we propose to decouple the original label into body label and detail label, as shown in Fig. 3. To achieve this goal, we introduce Distance Transformation (DT) to decouple the original label, which is a traditional image processing algorithm. DT can convert the binary image into a new image where each foreground

pixel has a value corresponding to the minimum distance from the background by a distance function.

Specifically, the input of DT is a binary image I, which can be divided into two groups (*i.e.*, foreground I_{fg} and background I_{bg}. For each pixel p, $I(p)$ is its corresponding value. If $p \in I_{fg}$, $I(p)$ equals 1, and 0 if $p \in I_{bg}$. To get the DT result of image I, we define the metric function $f(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$ to measure the distance between pixels. If pixel p belongs to the foreground, DT will first look up its nearest pixel q in the background and then use $f(p,q)$ to calculate the distance between pixel p and q. If pixel p belongs to the background, their minimum distance is set to zero. We use $f(p,q)$ as the pixels of a newly generated image, and the distance transformation can be expressed as

$$I^{'}(p) = \begin{cases} \min\limits_{q \in I_{bg}} f(p,q), & p \in I_{fg} \\ 0, & p \in I_{bg} \end{cases} \tag{1}$$

After the distance transformation, the original image I has been transformed into $I^{'}$ where pixel value $I^{'}(p)$ no longer equals to 0 or 1. We normalize the pixel values in $I^{'}$ using a simple linear function $I^{'} = \frac{I^{'} - min(I^{'})}{max(I^{'}) - min(I^{'})}$ to map the original value to [0, 1]. Compared with the original image I which treats all pixels equally, pixel value of $I^{'}$ not only depends on whether it belongs to foreground or background, but also is related to its relative position. Pixels located in the center of object have the largest values and those far away from the center or in background have the smallest values. So $I^{'}$ represents the body part of the original image, which mainly focuses on the central pixels that are relatively easy. We use it as the body label in the following experiments. Correspondingly, by removing the body image $I^{'}$ from the original image I, we can get the detail image, which is regarded as the detail label in consequent experiments and mainly concentrates on pixels far away from the main regions. In addition, we multiply the newly generated labels with the original binary image I to remove the background interference as

$$Label \Rightarrow \begin{cases} BL = I * I^{'} \\ DL = I * (1 - I^{'}) \end{cases} \tag{2}$$

where BL means the body label and DL represents the detail label. Now the original label has been decoupled into two different kinds of supervision to assist the network to learn both the body and detail features with different characteristics respectively.

3.2. Feature Extraction

As suggested by [28, 27, 21], we use ResNet-50 [14] as our backbone network. Specifically, we remove the fully connected layer and retain all convolutional blocks. Given an input image with shape $H \times W$, this backbone will generate five scales of features with decreasing spatial resolution by stride 2 due to downsampling. We denote these features as $F = \{F_i | i = 1, 2, 3, 4, 5\}$. The size of the i-th feature is $\frac{W}{2^i} \times \frac{H}{2^i} \times C_i$, where C_i is the channel of the i-th feature. It has been shown that low-level features greatly increase computation cost, but bring limited performance improvement [33]. So we only utilize features from $\{F_i | i = 2, 3, 4, 5\}$, as shown in Fig. 2. Two convolution layers are applied to these features to adapt them seperately to the body prediction task and detail prediction task. Then we get two groups of features $B = \{B_i | i = 2, 3, 4, 5\}$ and $D = \{D_i | i = 2, 3, 4, 5\}$, which all have been squeezed to 64 channels and sent to the decoder network for saliency map generation.

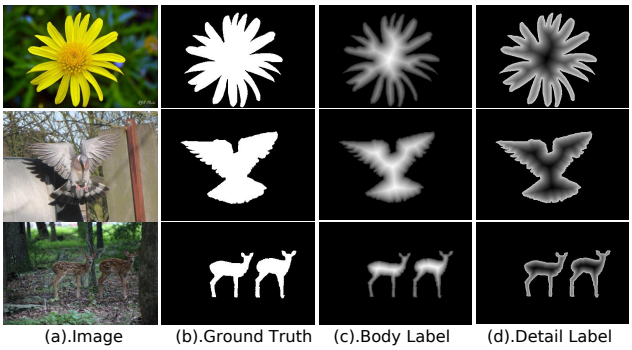


Figure 3. Some examples of label decoupling. (c) represents the body label of the ground truth, where pixels close to the center of the target have larger values. (d) means the detail label of the ground truth, where pixels near the boundary of the target have larger values. The sum of (c) and (d) is equal to (b).

3.3. Feature Interaction Network

Feature interaction network is built to adapt to the label decoupling, as shown in Fig. 2. With label decoupling, the saliency label has been transformed into the body map and the detail map, both of which are taken as supervision for model learning. FIN is designed as a two-branch structure, each of which is responsible for one label kind. Since both the body map and detail map are derived from the same saliency label, there exists a certain level of similarity and complementarity between the features from two branches. We introduce feature interaction between the complementary branches for information exchanging.

On the whole, the proposed framework is made up of one backbone encoder network, one interaction encoder network, one body decoder network and one detail decoder network. As discussed in Sec. 3.2, ResNet-50 [14] is used as the backbone network to extract multi-level features $B = \{B_i | i = 2, 3, 4, 5\}$ and $D = \{D_i | i = 2, 3, 4, 5\}$. For features B, a body decoder network is applied to gen-

erate body maps. Similarly, for features D, a detail decoder network is applied to generate detail maps. After getting the output features of these two branches, the simplest way to deal with them is to concatenate these features and apply a convolutional layer to get final saliency maps. However, this way ignores the relationship between branches. To explicitly promote the information exchange between branches, an interaction encoder network is introduced.

More specifically, interaction decoder takes the concatenated features of the body decoder and detail decoder as input. It stacks multiple convolutions to extract multi-level features. Then these multi-level features will be applied with 3x3 convolution layers to make them appropriate for body decoder and detail decoder respectively. Direct addition is used to fuse the interaction features with features from backbone encoder to produce more accurate saliency maps. On the surface, the whole network is unusual since the latter branch outputs are used in the former decoder. But in fact, feature interaction consists of multiple iterations. At the first iteration, two branches output features without exchanging information. From the second iteration, interaction is involved between branches.

3.4. Loss Function

Our training loss is defined as the summation of the outputs of all iterations as,

$$\mathcal{L} = \sum_{k=1}^{K} \alpha_k \ell^{(k)}, \tag{3}$$

where $\ell^{(k)}$ is the loss of the k-th iteration, K denotes the total number of iterations and α_k is the weight of each iteration. To simplify the problem, we set $\alpha_k = 1$ to treat all iterations equally. For each iteration, we will get three outputs (*i.e.*, body, detail and segmentation) and each of them corresponds to one loss. So $\ell^{(k)}$ can be defined as the combination of three losses as follows:

$$\ell^{(k)} = \ell^{(k)}_{body} + \ell^{(k)}_{detail} + \ell^{(k)}_{segm}, \tag{4}$$

where $\ell^{(k)}_{body}$, $\ell^{(k)}_{detail}$ and $\ell^{(k)}_{segm}$ denote body loss, detail loss and segmentation loss, respectively. We directly utilize binary cross entropy (BCE) to calculate both $\ell^{(k)}_{body}$ and $\ell^{(k)}_{detail}$. BCE is a widely used loss in binary classification and segmentation, which is defined as:

$$\ell_{bce} = -\sum_{(x,y)} [g(x,y)log(p(x,y)) + (1-g(x,y))log(1-p(x,y))], \tag{5}$$

where $g(x,y) \in [0,1]$ is the ground truth label of the pixel (x,y) and $p(x,y) \in [0,1]$ is the predicted probability of being salient object. However, BCE calculates the loss for each pixel independently and ignores the global structure of the image. To remedy this problem, as suggested by [23] we utilize the IoU loss to calculate $\ell^{(k)}_{segmentation}$, which can measure the similarity of two images on the whole rather than a single pixel. It is defined as:

$$\ell_{iou} = 1 - \frac{\sum_{(x,y)} [g(x,y) * p(x,y)]}{\sum_{(x,y)} [g(x,y) + p(x,y) - g(x,y) * p(x,y)]}, \tag{6}$$

where the notations are the same as Eq. 5. We do not apply IoU loss on $\ell^{(k)}_{body}$ and $\ell^{(k)}_{detail}$, because IoU loss requires the ground truth to be binary or it will result in wrong predictions, while body label and detail label do not satisfy this requirement.

4. Experiments

4.1. Datasets and Evaluation Metrics

To evaluate the proposed method, six popular benchmark datasets are adopted, including ECSSD [36] with 1000 images, PASCAL-S [19] with 850 images, HKU-IS [17] with 4447 images, DUT-OMRON [37] with 5168 images, DUTS [25] with 15572 images and THUR15K [5] with 6232 images. Among them, **DUTS** is the largest saliency detection benchmark, which contains 10,553 training images (**DUTS-TR**) and 5,019 testing images (**DUTS-TE**). **DUTS-TR** is used to train the model, other datasets for evaluation. In addition, we also measure the model performance on the challenging SOC dataset [8] of different attributes. Five metrics are used to evaluate the performance of our model and existing state-of-the-art methods. The first metric is the mean absolute error (MAE), as shown in Eq. 7, which is widely adopted in [4, 15, 18, 21]. Mean F-measure (mF), E-measure (E_ξ) [9], weighted F-measure (F^ω_β) and S-measure (S_α) are also widely used to evaluate saliency maps. In addition, precision-recall (PR) and F-measure curves are drawn to show the overall performance.

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |P(i,j) - G(i,j)| \tag{7}$$

where P is the predicted map and G is the ground truth.

4.2. Implementation Details

The proposed model is trained on DUTS-TR and tested on the above mentioned six datasets. For data augmentation, we use horizontal flip, random crop and multi-scale input images. ResNet-50, pretrained on ImageNet, is used to initialize the backbone (*i.e.*, block1 to block5) and other parameters are randomly initialized. We set the maximum learning rate to 0.005 for ResNet-50 backbone and 0.05 for other parts. Warm-up and linear decay strategies are used. The whole network is trained end-to-end by stochastic gradient descent (SGD). Momentum and weight decay are set to 0.9 and 0.0005, respectively. Batchsize is set to 32 and maximum epoch is set to 48. During testing, each image is

Table 2. Performance comparison with state-of-the-art methods on six datasets. MAE (smaller is better), mean F-measure (mF, larger is better) and E-measure (E_ξ, larger is better) are used to measure the model performance. '-' means the author has not provided corresponding saliency maps. The best and the second best results are highlighted in red and blue respectively.

Algorithm	ECSSD 1,000 images			PASCAL-S 850 images			DUTS-TE 5,019 images			HKU-IS 4,447 images			DUT-OMRON 5,168 images			THUR15K 6,232 images		
	MAE	mF	E_ξ	MAE	mF	E_ξ	MAE	mF	E_ξ	MAE	mF	E_ξ	MAE	mF	E_ξ	MAE	mF	E_ξ
BMPM [38]	.044	.894	.914	.073	.803	.838	.049	.762	.859	.039	.875	.937	.063	.698	.839	.079	.704	.803
DGRL [28]	.043	.903	.917	.074	.807	.836	.051	.764	.863	.037	.881	.941	.063	.709	.843	.077	.716	.811
R^3Net [7]	.051	.883	.914	.101	.775	.824	.067	.716	.827	.047	.853	.921	.073	.690	.814	.078	.693	.803
RAS [4]	.055	.890	.916	.102	.782	.832	.060	.750	.861	.045	.874	.931	.063	.711	.843	.075	.707	.821
PiCA-R [21]	.046	.867	.913	.075	.776	.833	.051	.754	.862	.043	.840	.936	.065	.695	.841	.081	.690	.803
AFNet [13]	.042	.908	.918	.070	.821	.846	.046	.792	.879	.036	.888	.942	.057	.738	.853	.072	.730	.820
BASNet [23]	.037	.880	.921	.076	.775	.847	.048	.791	.884	.032	.895	.946	.056	.756	.869	.073	.733	.821
CPD-R [33]	.037	.917	.925	.072	.824	.849	.043	.805	.886	.034	.891	.944	.056	.747	.866	.068	.738	.829
EGNet-R [41]	.037	.920	.927	.074	.823	.849	.039	.815	.891	.032	.898	.948	.053	.755	.867	.067	.741	.829
PAGE [31]	.042	.906	.920	.077	.810	.841	.052	.777	.869	.037	.882	.940	.062	.736	.853	-	-	-
TDBU [30]	.041	.880	.922	.071	.779	.852	.048	.767	.879	.038	.878	.942	.061	.739	.854	-	-	-
SCRN [34]	.037	.918	.926	.064	.832	.857	.040	.808	.888	.034	.896	.949	.056	.746	.863	.066	.741	.833
SIBA [24]	.035	.923	.928	.070	.830	.855	.040	.815	.892	.032	.900	.950	.059	.746	.860	.068	.741	.832
PoolNet [20]	.039	.915	.924	.074	.822	.850	.040	.809	.889	.032	.899	.949	.056	.747	.863	.070	.732	.822
LDF(ours)	.034	.930	.925	.060	.848	.865	.034	.855	.910	.027	.914	.954	.051	.773	.873	.064	.764	.842

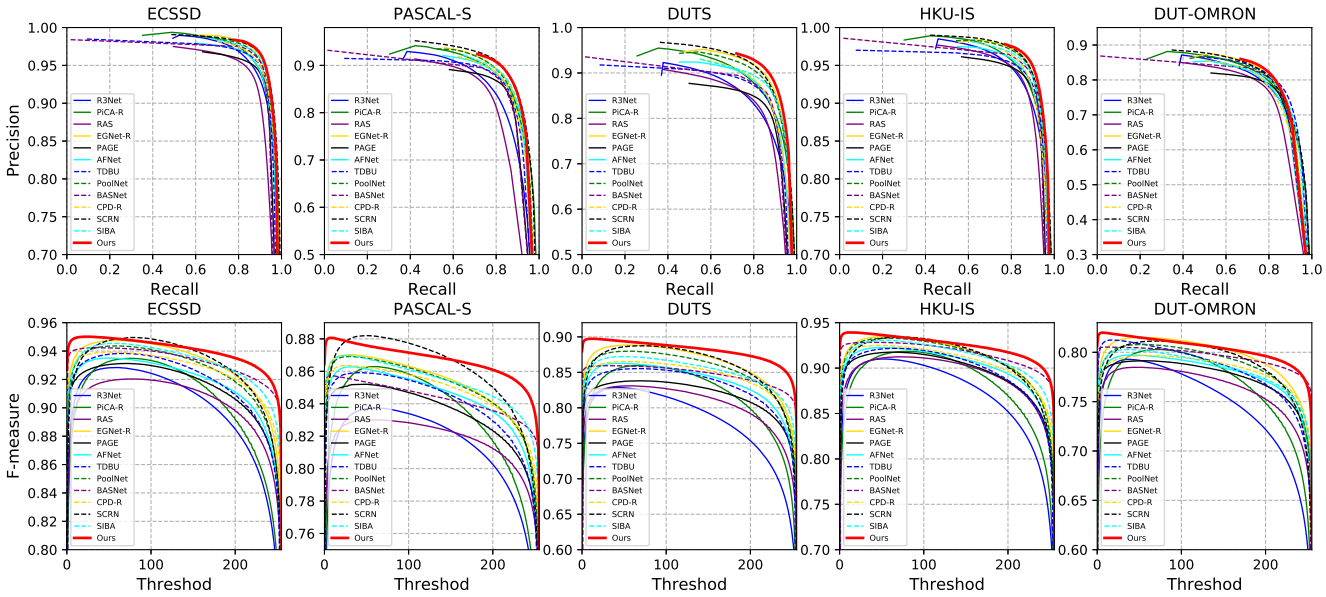


Figure 4. Performance comparison with state-of-the-art methods on five datasets. The first row shows precision-recall curves. The second row shows F-measure curves with different thresholds.

simply resized to 352 x 352 and then fed into the network to get prediction without any post-processing. It is worth noting that the output saliency maps are used as the predictions rather than the addition of predicted body and detail maps.

4.3. Ablation Studies

Number of Feature Interaction. Tab. 4 shows the performance with different numbers of feature interaction. Compared with the baseline which has no feature interaction (Number=0), model with one feature interaction achieves better results. When the number is larger, the performance becomes worse. Because repeated feature interaction makes the network to grow too deeper and harder to optimize. So in all the following experiments, we set the number to 1 to balance the model optmization and performance.

Different Combinations of Supervision. Tab. 5 shows the performance with different combinations of supervision.

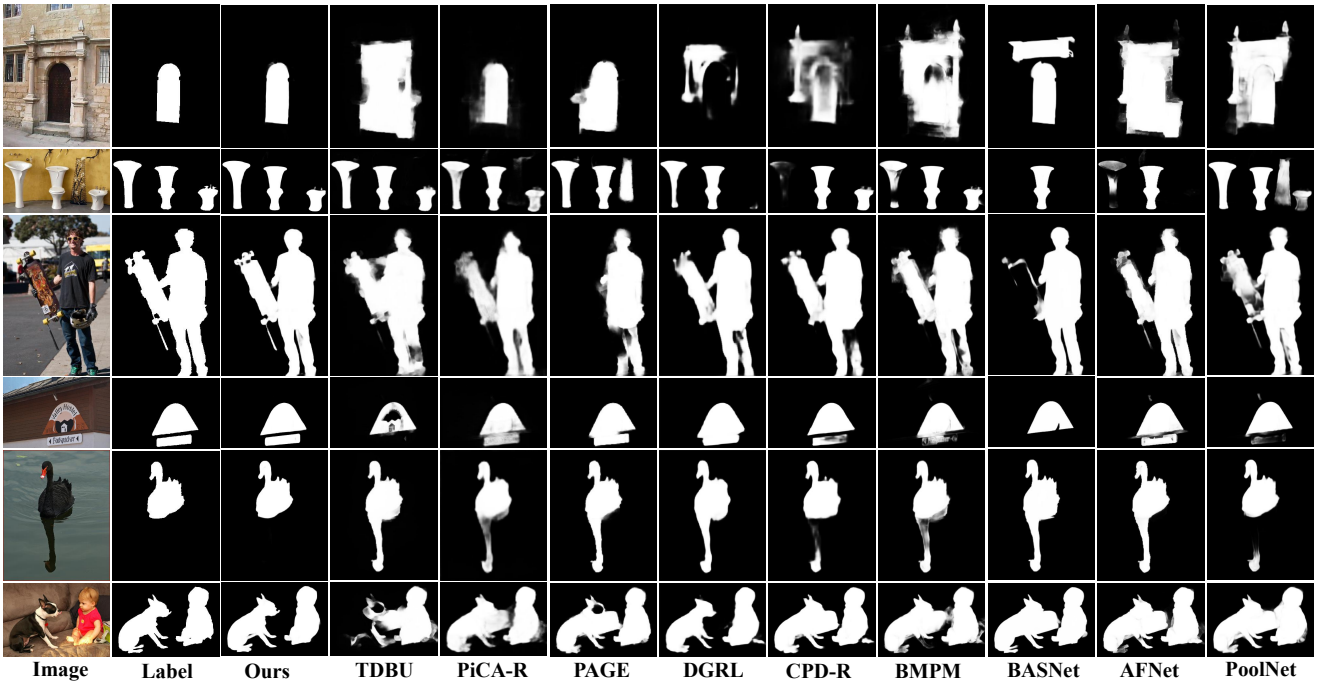


Figure 5. Visual comparison of different algorithms. Each row represents one image and corresponding saliency maps. Each column represents the predictions of one method. Apparently, our method is good at dealing with cluttered background and producing more accurate and clear saliency maps.

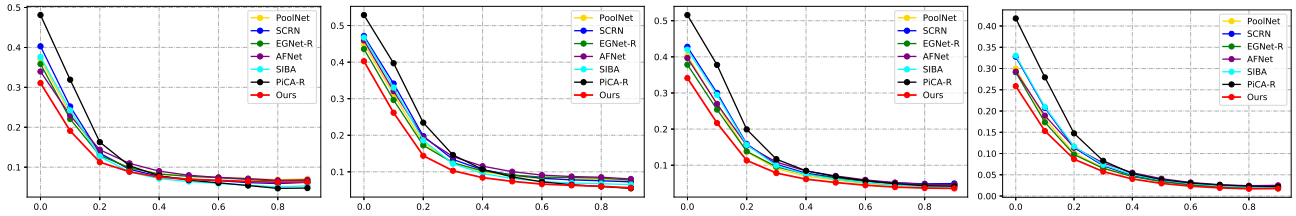
Table 3. Performance on SOC [8] of different attributes. Each row represents one attribute and we report the mean F-measure scores of LDF and state-of-the-art methods. The last row shows the whole performance on the SOC dataset. The best and the second best results are highlighted in red and blue respectively.

Attr	PiCA-R	BMPM	R^3Net	DGRL	RAS	AFNet	BASNet	PoolNet	CPD-R	EGNet-R	SCRN	Ours
AC	0.721	0.727	0.659	0.744	0.664	0.763	0.773	0.746	0.765	0.739	0.770	0.774
BO	0.706	0.802	0.637	0.847	0.654	0.824	0.780	0.677	0.821	0.743	0.743	0.803
CL	0.703	0.708	0.667	0.735	0.616	0.740	0.721	0.723	0.741	0.707	0.751	0.772
HO	0.727	0.738	0.683	0.773	0.682	0.778	0.769	0.768	0.766	0.747	0.775	0.807
MB	0.779	0.757	0.669	0.809	0.687	0.794	0.791	0.784	0.810	0.741	0.815	0.840
OC	0.692	0.711	0.625	0.724	0.608	0.730	0.721	0.713	0.741	0.699	0.732	0.756
OV	0.778	0.783	0.677	0.797	0.666	0.805	0.802	0.774	0.799	0.768	0.801	0.820
SC	0.678	0.702	0.626	0.725	0.645	0.711	0.713	0.723	0.726	0.708	0.738	0.774
SO	0.569	0.588	0.546	0.618	0.560	0.615	0.619	0.631	0.635	0.605	0.639	0.676
Avg	0.662	0.673	0.611	0.698	0.608	0.700	0.697	0.694	0.709	0.680	0.710	0.739

From this table, combinations including detail label perform better than those including edge label, which demonstrates the effectiveness of detail label than edge label. In addition, combinations including body label perform better than those including saliency label (Sal). It confirms that without the interference of edges, center pixels can learn better feature representations.

4.4. Comparison with State-of-the-arts

Quantitative Comparison. To demonstrate the effectiveness of the proposed method, 14 state-of-the-art SOD methods are introduced to compare, including BMPM [38], DGRL [28], R^3Net [7], RAS [4], PiCA-R [21], AFNet [13], BASNet [23], CPD-R [33], EGNet-R [41], PAGE [31], TDBU [30], SCRN [34], SIBA [24] and PoolNet [20]. For fair comparison, we evaluate all the saliency maps provided by the authors with the same evaluation codes. We compare the proposed method with others in terms of MAE, mF and E_ξ, which are shown in Tab. 2. The best results are highlighted with red color. Obviously, compared with other counterparts, our method outperforms previous state-of-the-art methods by a large margin. Besides, Fig. 4



(a) ECSSD (b) DUTS (c) HKU-IS (d) THUR15K

Figure 6. Error-Distance distribution of different methods. The proposed method has the smallest error along the distance. Especially around edge areas, the proposed method performs much better.

Table 4. Performance with different numbers of feature interaction. Number=0 means two branches have no feature interaction.

Number	THUR15K			DUTS-TE		
	MAE	mF	E_ξ	MAE	mF	E_ξ
0	0.069	0.751	0.834	0.038	0.839	0.897
1	0.064	0.764	0.842	0.034	0.855	0.910
2	0.066	0.756	0.837	0.035	0.849	0.903
3	0.068	0.753	0.834	0.037	0.842	0.897

Table 5. Comparison on different combinations of supervision. Body, detail, saliency and edge maps are used, respectively.

Label	THUR15K			DUTS-TE		
	MAE	mF	E_ξ	MAE	mF	E_ξ
Body + Detail	0.064	0.764	0.842	0.034	0.855	0.910
Body + Edge	0.066	0.758	0.836	0.036	0.850	0.904
Sal + Detail	0.066	0.756	0.835	0.037	0.848	0.901
Sal + Edge	0.070	0.752	0.827	0.039	0.844	0.895

presents the precision-recall curves and F-measure curves on five datasets. As can be seen, the curves of the proposed method consistently lie above others. In addition, we calculate the Error-Distance distribution of different methods in Fig. 6, where predictions produced by the proposed method have the minimum error along distance, especially around the edge areas.

Visual Comparison. Some prediction examples of the proposed method and other state-of-the-art approaches have been shown in Fig. 5. We observe that the proposed method not only highlights the correct salient object regions clearly, but also well suppresses the background noises. It is robust in dealing with various challenging scenarios, including cluttered background, manufactured structure and low contrast foreground. Compared with other counterparts, the saliency maps produced by the proposed method are clearer and more accurate.

Performance on SOC of Different Attributes. SOC [8] is a challenging dataset with multiple attributes. Images with the same attribute have certain similarity and reflect the common challenge in real world. We utilize this dataset to test the robustness of model under different scenes. Specifically, we evaluate the mean F-measure score of our model as well as 11 state-of-the-art methods. Each model will get nine scores under nine attributes. In addition, an overall score is calculated to measure the whole performance under all scenes. Tab. 3 shows the scores. We can see the proposed model achieves the best results among most of attributes except "BO", which indicates the good generalization of the proposed method. It can be applied in different challenging scenes.

5. Conclusion

In this paper, we propose the label decoupling framework for salient object detection. By empirically showing that edge prediction is a challenging task in saliency prediction, we propose to decouple the saliency label into body map and detail map. Detail map helps model learn better edge features and body map avoids the distraction from pixels near edges. Supervised by these two kinds of maps, the proposed method achieves better performance than direct supervison with saliency maps. Besides, feature interaction network is introduced to make full use of the complementarity between body and detail maps. Experiments on six datasets demonstrate that the proposed method outperforms state-of-the-art methods under different evaluation metrics.

6. Acknowledgement

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102003, in part by National Natural Science Foundation of China: 61672497, 61620106009, 61836002, 61931008 and U1636214, and in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. Authors would like to thank Kingsoft Cloud for their helpful disscussion and free GPU cloud computing resource support.

References

[1] Radhakrishna Achanta, Sheila S. Hemami, Francisco J. Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009. 1

[2] Jingdong Wang andn Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Xiaowei Hu, and Nanning Zheng. Salient

object detection: A discriminative regional feature integration approach. *International Journal of Computer Vision*, 123(2):251–268, 2017. 2

[3] Ali Borji and Laurent Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, pages 478–485, 2012. 2

[4] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *ECCV (9)*, volume 11213, pages 236–252, 2018. 1, 2, 5, 6, 7

[5] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014. 5

[6] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015. 1

[7] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R^3net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690, 2018. 1, 2, 6, 7

[8] Deng-Ping Fan, Ming-Ming Cheng, Jiangjiang Liu, Shanghua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV (15)*, volume 11219 of *Lecture Notes in Computer Science*, pages 196–212. Springer, 2018. 2, 5, 7, 8

[9] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, pages 698–704. ijcai.org, 2018. 5

[10] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564. Computer Vision Foundation / IEEE, 2019. 1

[11] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[12] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[13] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *CVPR*, pages 1623–1632, 2019. 2, 3, 6, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 4

[15] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):815–828, 2019. 1, 2, 5

[16] Xiaowei Hu, Lei Zhu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. Recurrently aggregating deep features for salient object detection. In *AAAI*, pages 6943–6950, 2018. 1

[17] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015. 5

[18] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *ECCV (15)*, volume 11219, pages 370–385, 2018. 3, 5

[19] 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

[20] Jiangjiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, pages 3917–3926, 2019. 2, 3, 6, 7

[21] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. 1, 3, 4, 5, 6, 7

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[23] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jägersand. Basnet: Boundary-aware salient object detection. In *CVPR*, pages 7479–7489, 2019. 2, 3, 5, 6, 7

[24] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian. Selectivity or invariance: Boundary-aware salient object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 6, 7

[25] 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 3796–3805, 2017. 2, 5

[26] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV (4)*, volume 9908, pages 825–841, 2016. 1

[27] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4039–4048, 2017. 1, 4

[28] 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, 4, 6, 7

[29] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, and Haibin Ling. Salient object detection in the deep learning era: An in-depth survey. *CoRR*, abs/1904.09146, 2019. 1

[30] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In *CVPR*, pages 5968–5977, 2019. 3, 6, 7

[31] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C. H. Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *CVPR*, pages 1448–1457, 2019. 3, 6, 7

[32] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. A mutual learning method for salient object detection with intertwined multi-supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[33] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, June 2019. 2, 4, 6, 7

[34] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 6, 7

[35] Yongjian Xin, Shuhui Wang, Liang Li, Weigang Zhang, and Qingming Huang. Reverse densely connected feature pyramid network for object detection. In *Asian Conference on Computer Vision*, pages 530–545, 2018. 1

[36] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 2, 5

[37] 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. 2, 5

[38] 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. 3, 6, 7

[39] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017. 1, 3

[40] 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

[41] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 6, 7

[42] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *CVPR*, pages 3085–3094, 2019. 1, 2, 3