Cascaded Partial Decoder for Fast and Accurate Salient Object Detection

Zhe Wu[1,2], Li Su[1,2,3], and Qingming Huang[1,2,3]

[1]School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS),
Beijing, China
[2]Key Lab of Big Data Mining and Knowledge Management, UCAS, Beijing, China
[3]Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy
of Sciences, Beijing, China
zhe.wu@vipl.ict.ac.cn, {suli, qmhuang}@usas.ac.cn

Abstract

Existing state-of-the-art salient object detection networks rely on aggregating multi-level features of pretrained convolutional neural networks (CNNs). Compared to high-level features, low-level features contribute less to performance but cost more computations because of their larger spatial resolutions. In this paper, we propose a novel Cascaded Partial Decoder (CPD) framework for fast and accurate salient object detection. On the one hand, the framework constructs partial decoder which discards larger resolution features of shallower layers for acceleration. On the other hand, we observe that integrating features of deeper layers obtain relatively precise saliency map. Therefore we directly utilize generated saliency map to refine the features of backbone network. This strategy efficiently suppresses distractors in the features and significantly improves their representation ability. Experiments conducted on five benchmark datasets exhibit that the proposed model not only achieves state-of-the-art performance but also runs much faster than existing models. Besides, the proposed framework is further applied to improve existing multi-level feature aggregation models and significantly improve their efficiency and accuracy.

1. Introduction

Recently, deep learning has achieved surprising performance in salient object detection for it providing abundant and discriminative image representations. The early deep saliency methods [15, 16, 32] utilize CNNs to predict saliency scores of image regions and obtain accurate saliency maps with high computational complexity. In the following works, fully convolutional network (FCN) [24] based
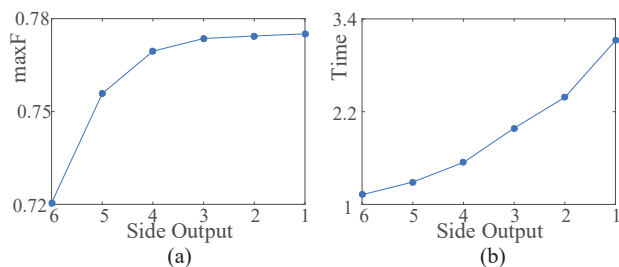

Figure 1: *(a) Maximum F-measure of six side outputs of the original DSS [9] model in PASCAL-S [19] dataset. (b) We set inference time of backbone network as 1, and show inference time of each side output here. The performance growth is getting slower and the inference time rapidly increases when gradually integrating features from high-level 6 to low-level 1.*

encoder-decoder architecture is widely applied for salient object detection. The encoder is the pre-trained image classification model (*e.g.* VGG [29] and ResNet [8]) which provides multi-level deep features: the high-level features with low resolutions represent semantic information, and the low-level features with high resolutions represent spatial details. In the decoder, these features are combined to generate accurate saliency maps. Researchers have developed various decoders [9, 17, 20, 21, 25, 41, 42] to integrate low-level and high-level features.

However, two drawbacks exist in these deep aggregation methods. On the one hand, compared to high-level features, low-level features contribute less to the performance of deep aggregation methods. In Fig. 1(a), we present performances of different side outputs of the DSS [9] model. It is obvious that the performance tends to saturate quickly when gradually aggregating features from high-level to low-level. On the other hand, due to the large resolutions
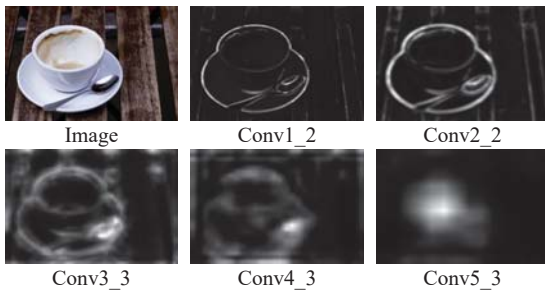
Figure 2: *The original image and five-level feature maps from VGG*16 *[29]. The Conv*3_3 *feature still retains edge information. Hence the Conv*1_2 *and Conv*2_2 *features with large resolutions are not under consideration in the proposed framework.*

of low-level features, integrating them with high-level features obviously enlarges the computational complexity as shown in Fig. 1(b). However, detecting and segmenting salient objects should be fast since this process is often a preprocessing stage to more complex operations [3]. In consequence, it is essential to design a mechanism to eliminate the impact of low-level features on computational complexity while ensuring the performance.

When CNNs go deep, feature gradually changes from low-level representation to high-level representation. Hence deep aggregation models may recover spatial details of saliency maps when only integrating features of deeper layers. In Fig. 2, we show examples of multi-level feature maps of VGG16 [29]. Compared to low-level features of Conv1_2 and Conv2_2 layers, the feature of Conv3_3 layer also reserve edge information. Besides, background regions in feature maps may result in inaccuracy of saliency maps. Previous works make use of adaptive attention mechanism [21, 41] to solve this problem. However, the effect of this mechanism relies on the accuracy of the attention map. Since fusing features of deeper layers will generate relatively precise saliency map, we can directly use this map to refine features.

In this paper, we propose a novel cascaded partial decoder framework, which discards features of shallower layers to ensure high computational efficiency and then refine features of deeper layers to improve their representation ability. We modify the standard straight backbone network to a bifurcated one. This new backbone network contains two branches with the same architecture. We construct partial decoder which only aggregates features in each branch. In order to further accelerate the model, we design a fast and efficient context module to abstract discriminative features and integrate them in an upsampling-concatenating way. Then we propose a cascaded optimization mechanism which utilizes initial saliency map of the first branch to

refine features of the second branch. In order to uniformly segment the whole salient objects, we propose a holistic attention module to allow the initial saliency map cover more useful information. In addition, the proposed framework can be utilized to improve existing deep aggregation models. When embedding their decoders in our framework, the accuracy and efficiency will be significantly improved. Our contributions are summarized as follows:

(1) We propose a novel cascaded partial decoder framework, which discards low-level features to reduce the complexity of deep aggregation models, and utilizes generated relatively precise attention map to refine high-level features to improve the performance.

(2) Experimental results on five benchmark datasets demonstrate that the proposed model not only achieves state-of-the-art performance but also runs much faster than existing models.

(3) Our framework can be applied to improve existing deep aggregation models. The efficiency and accuracy of improved models will both be significantly improved compared to the original models.

2. Related Work

Over the past two decades, researchers have developed a large amount of saliency detection algorithms. Traditional models extract hand-crafted features and are based on various saliency assumptions [2, 6, 11, 46]. More details about traditional methods are concluded in [3, 4]. Here we mainly discuss deep learning based saliency detection models.

Early works utilize CNNs to determine whether image regions are salient or not [15, 16, 32, 44]. Although these models have achieved much better performance than traditional methods, it is time-consuming to predict saliency scores for image regions. Then researchers develop more effective models based on the successful fully convolutional network [24]. Li *et al.* [18] set up a unified framework for salient object detection and semantic segmentation to effectively learn the semantic properties of salient objects. Wang *et al.* [34] leverage cascaded fully convolutional networks to continuously refine previous prediction maps.

Recently, researchers have proved that fusing multi-level features further improves the performance of dense prediction tasks [7, 27]. In CNNs, high-level features provide semantic information, and low-level features contains spatial details which are helpful for refining object boundaries. Many works [9, 17, 20, 21, 25, 41, 42] follow this strategy and precisely segment salient objects. Li *et al.* [17] directly integrate multi-level features to obtain more advanced feature representation. Liu and Han [20] first make a coarse global prediction, and then hierarchically and progressively refine the details of saliency maps step by step via in-
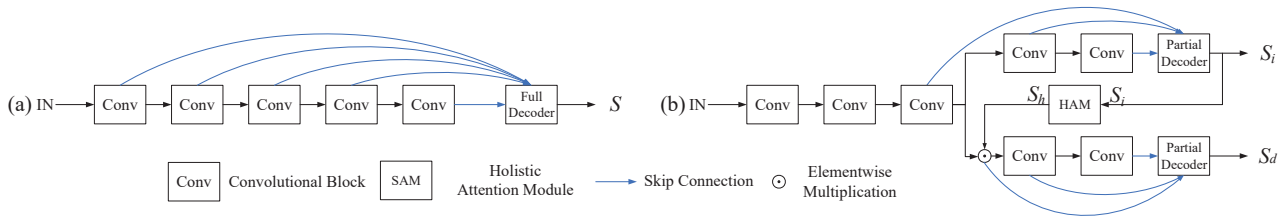
Figure 3: *(a) Traditional encoder-decoder framework, (b) The proposed cascaded partial decoder framework. We use VGG16 [29] as the backbone network. Traditional framework generates saliency map S by adopting full decoder which integrates all level features. The proposed framework adopts partial decoder, which only integrates features of deeper layers, and generates an initial saliency map S_i and the final saliency map S_d.*

tegrating local context information. Hou *et al.* [9] introduce short connections to the skip-layer structures within the HED [38] architecture. Luo *et al.* [25] segment salient objects by combining local contrast feature and global information through a multi-resolution 4×5 grid network. Zhang *et al.* [42] first integrate multi-level feature maps into multiple resolutions, which simultaneously incorporate semantic information and spatial details. Then this work predicts the saliency map in each resolution and fuses them to generate the final saliency map. In [41], the work extracts context-aware multi-level features and then utilizes a bidirectional gated structure to pass messages between them. Liu *et al.* [21] leverage global and local pixel-wise contextual attention network to capture global and local context information. Then these modules are incorporated with U-Net architecture to segment salient objects. In this paper, we argue that low-level features always contribute less than high-level features. However, they need more computation cost than high-level features owing to their larger spatial resolutions. Hence we propose a novel cascaded partial decoder framework for salient object detection, which does not consider low-level features and utilizes generated saliency map to refine high-level features.

3. The Proposed Framework

In this paper, we propose a novel cascaded partial decoder framework which contains two branches. In each branch, we design a fast and effective partial decoder. The first branch generates an initial saliency map which is utilized to refine the features of the second branch. Besides, we propose a holistic attention module to segment the whole objects uniformly.

3.1. Mechanism of the Proposed Framework

We design the proposed model on the basis of VGG16 network, which is the most widely utilized backbone network in deep salient object detection models. For an input image I with size $H \times W$, we can abstract features at five levels, which are denoted as $\{f_i, i = 1, ..., 5\}$ with res-

olutions $[\frac{H}{2^{i-1}}, \frac{W}{2^{i-1}}]$. The decoders proposed in previous works [41, 42], which are called full decoder in this paper, integrate all features to generate saliency map S. A unified architecture of the full decoder is shown in Fig. 3(a) and it can be represented by $D_T = g(f_1, f_2, f_3, f_4, f_5)$, where $g(\cdot)$ denotes a multi-level feature aggregation algorithm. Previous works focus on how to develop a more effective integration strategy.

In Fig. 3(b), we show the architecture of the proposed cascaded partial decoder framework. Since that the features of shallower layers contribute less to performance, we construct a partial decoder that only integrates features of deeper layers. In order to utilize generated saliency map to refine features, we design a bifurcated backbone network. We set the Conv3_3 layer as an optimization layer, and use the last two convolutional blocks to construct two branches (an attention one and a detection one). In the attention branch, we design a partial decoder to integrate three-level features which are denoted as $\{f_i^a = f_i, i = 3, 4, 5\}$. Hence the partial decoder is represented by $D_a = g_a(f_3^a, f_4^a, f_5^a)$ and it generates an initial saliency map S_i. After processing of the proposed holistic attention module, we obtain an enhanced attention map S_h which is utilized to refine the feature f_3. Because we can obtain relatively precise saliency map via integrating features of three top layers, the attention map S_h effectively eliminates distractors in feature f_3. Then we obtain the refined feature f_3^d for detection branch via element-wise multiplying the feature and the attention map: $f_3^d = f_3 \odot S_h$. Hence the following two-level features of the detection branch are denoted as $\{f_4^d, f_5^d\}$. Through constructing another partial decoder $D_d = g_d(f_3^d, f_4^d, f_5^d)$ for the detection branch, the proposed model outputs the final saliency map S_d. For convenience, we set $g_a = g_d$ in this paper. The details of the proposed holistic attention module and the partial decoder are described in Section 3.2 and Section 3.3 respectively.

We jointly train the two branches with ground truth. The parameters of the two branches are not shared. Given $\{S_i, S_d\}$ and the corresponding label l, the total loss L_{total}

is formulated as:

$$L_{total} = L_{ce}(S_i, l|\Theta_i) + L_{ce}(S_d, l|\Theta_d). \qquad (1)$$

L_{ce} is the sigmoid cross entropy loss:

$$L_{ce}(\Theta) = -\sum_{j=1}^{N} \sum_{c \in \{0,1\}} \delta(l^j = c) \log p(S^j = c|\Theta), \quad (2)$$

where N is the pixel number, δ is the indicator function, j denotes pixel coordinate and $\Theta = \{\Theta_i, \Theta_d\}$ are parameter sets corresponding to the saliency maps $S = \{S_i, S_d\}$. It is obvious that Θ_i is a proper subset of Θ_d, which indicates that the two branches work in an alternating way. On the one hand, the attention branch provides precise attention map for the detection branch, which leads to that the detection branch segments more accurate salient objects. On the other hand, the detection branch could be considered as an auxiliary loss of the attention branch, which also helps the attention branch to focus on salient objects. Joint training the two branches makes our model uniformly highlights salient objects while suppressing distractors.

In addition, we can leverage the proposed framework to improve existing deep aggregation models when we integrate the features of each branch by using the aggregation algorithms of these works. Even though we raise the computation cost of the backbone network and add one more decoder when compared to the traditional encoder-decoder architecture, the total computation complexity is still significantly reduced because of discarding low-level features in decoders. Moreover, the cascaded optimization mechanism of the proposed framework promotes the performance, and the experiments show that the two branches both outperform the original models.

3.2. Holistic Attention Module

Given the feature map from the optimization layer and the initial saliency map from attention branch, we can use a initial attention strategy which means directly multiplying the feature map with the initial saliency map. When we obtain an accurate saliency map from the attention branch, this strategy will efficiently suppress distractors of the feature. On the contrary, if distractors are classified as salient regions, this strategy results in abnormal segmentation results. As a result, we need to improve the effectiveness of the initial saliency map. More specially, the edge information of salient objects may be filtered out by the initial saliency map because it is difficult to be precisely predicted. In addition, some objects in complex scenes are hard to be completely segmented. Therefore we propose a holistic attention module which aims to enlarge the coverage area of the initial saliency map, and it is defined as follows:

$$S_h = MAX(f_{min\_max}(Conv_g(S_i, k)), S_i) \qquad (3)$$

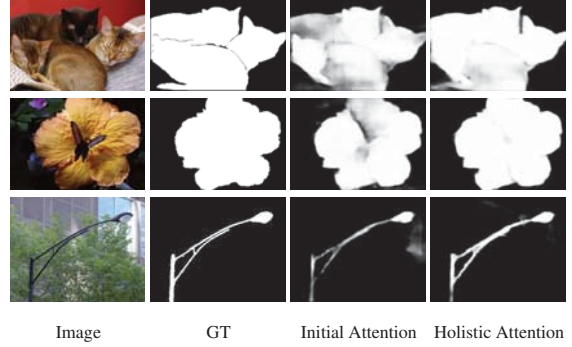Image   GT   Initial Attention   Holistic Attention

Figure 4: *GT is the ground truth. As we can see, the proposed holistic attention module is helpful for segmenting the whole salient objects and refining more precise boundaries.*

where $Conv_g$ is a convolution operation with a Gaussian kernel k and zero bias, $f_{min\_max}(\cdot)$ is a normalization function to make the blurred map ranges in $[0, 1]$, and $MAX(\cdot)$ is a maximum function which tends to increase the weight coefficient of salient regions of S_i because that the convolution operation will blur S_i. Compared to the initial attention, the proposed holistic attention mechanism hardly increases computation cost, and it further highlights the whole salient objects as shown in Fig. 4. Moreover, the size and standard deviation of Gaussian kernel k are initialized with 32 and 4. Then it is jointly trained with the proposed model.

3.3. The Proposed Decoder

Since that the proposed framework consists of two decoders, we need to construct a fast integration strategy to ensure low complexity. Meanwhile, we need to generate saliency map as accurate as possible. Firstly, in order to capture global contrast information, we design an effective context module which is inspired by the receptive field block (RFB) [22]. Compared to the original RFB, we add one more branch to enlarge the receptive field further. Our context module consists of four branches $\{b_m, m = 1, ..., 4\}$. For acceleration, in each branch, we use a 1×1 convolutional layer to reduce channel number to 32. For $\{b_m, m > 1\}$, we add two layers: a $(2m-1) \times (2m-1)$ convolutional layer and a 3×3 convolutional layer with $(2m-1)$ dilation [5]. We concatenate the outputs of these branches and reduce the channel to 32 by an additional 1×1 convolutional layer. Then a short connection is added as the original RFB. In general, given features $\{f_i^c, i \in [l, ..., L], c \in [a, d]\}$ from the bifurcated backbone network, we obtain discriminative features $\{f_i^{c_1}\}$ from the context module. Then we use multiplication operation to reduce the gap between multi-level features. Especially, for the top-most feature ($i = L$), we set $f_L^{c_2} = f_L^{c_1}$. For feature $\{f_i^{c_1}, i < L\}$, we update it to

f_i^{c2} via element-wise multiplying itself with all features of deeper layers. This operation is defined as follows:

$$f_i^{c2} = f_i^{c1} \odot \Pi_{k=i+1}^{L} Conv(Up(f_k^{c1})), i \in [l, ..., L-1], \quad (4)$$

where $Up(\cdot)$ is upsampling feature by a factor 2^{k-j}, and $Conv$ is a 3×3 convolutional layer. At last, we utilize an upsampling-concatenating strategy to integrate multi-level features. When we construct a partial decoder and set the Conv3_3 layer as the optimization layer ($l = 3, L = 5$), we obtain a feature map with $[\frac{H}{4}, \frac{W}{4}]$ size and 96 channel number. With 3×3 layer and 1×1 convolutional layers, we obtain the final feature map and resize it to $[H, W]$.

4. Experiment

4.1. Salient Object Detection

4.1.1 Experimental Setup

Evaluation Datasets. We evaluate the proposed model on five benchmark datasets: ECSSD [39], HKU-IS [16], PASCAL-S [19], DUTS [33], DUT-OMRON [40].
Evaluation Metrics. We adopt two metrics: mean absolute error (MAE) and F-measure (maxF). We adopt mean absolute error (MAE) and F-measure as our evaluation metrics. According to the different ways for saliency map binarization, there exist two ways to compute F-measure [4]. One is maximum F-measure (denoted as maxF), which is adopted in [9, 21, 25, 41]. The other is average F-measure (denoted as avgF), which is adopted in [35, 36, 42, 43]. For fairly comparison, we compute both maxF and avgF.
Implementation Details. We implement the proposed model based on the Pytorch[1] framework and a GTX 1080Ti GPU is used for acceleration. Following previous works [21, 35, 36, 41, 43], we train the proposed model on the training set of DUTS [33] dataset. The parameters of the bifurcated backbone network are initialized by VGG16 [29]. We initialize the other convolutional layers using the default setting of the Pytorch. All training and test images are resized to 352×352. Any post-processing procedure (*e.g.* CRF [14]) is not applied in this paper. The proposed model is trained by Adam optimizer [13]. The batch size is set as 10 and the initial learning rate is set as 10^{-4} and decreased by 10% when training loss reaches a flat. It takes nearly six hours for training the proposed model. The code is available at https://github.com/wuzhe71/CPD.

4.1.2 Comparison with State-of-the-arts

We compare the proposed model with eight state-of-the-art deep salient object detection algorithms, including NLDF [25], Amulet [42], DSS [9], SRM [35], BMPM [41], PAGR [43], DGRL [36] and PiCANet [21]. We implement

---

[1]https://pytorch.org/

these models with available source codes or directly evaluate saliency maps provided by authors. Especially, NLDF, Amulet and DSS are originally trained on MSRA10K [6] dataset or MSRA-B [23] dataset (there is a large overlap between these two datasets). Hence we re-train these three models on DUTS dataset as other models for fairly comparison. We find that training on DUTS dataset will make deep models work better in complex scenes. Besides, we also train the proposed model on MSRA-B dataset to compare with these three original models, and the results are reported in supplementary material.

In Table. 1, we show the quantitative comparison results. Considering some works use ResNet50 as the backbone, we also train the proposed model on the basis of this backbone network. ResNet50 contains four convolutional blocks, and we set the last layer of the second block as the optimization layer. Then we utilize the last two blocks to design the two branches. In Table. 1, the results of the attention branch (denoted as "-A") of the proposed model are also reported. Moreover, we compare the average execution time with the other models on DUTS dataset, and all scores are tested on our platform (PAGR only provides saliency maps). It is obvious that the proposed model outperforms all other models in most cases and it runs much faster than existing models. Only PiCANet-R obtains higher maxF score than the proposed model on DUT-OMRON dataset. However, our model runs about 12 times faster than PiCANet-R. More specially, compared to the improvements on maxF and MAE, we obtain a larger improvement on avgF. This demonstrates that the proposed model works much better in uniformly highlighting salient objects. In addition, we can find that the results of our attention branch also achieve comparable results with other models. Meanwhile, the proposed model only with the attention branch runs faster. This indicates that the proposed model provides two-level saliency maps for real-time applications.

In Fig. 5, we show the qualitative comparison on some challenging cases: small object, complex scenes, multiple objects and large object. Even though we discard the low-level features of backbone network, our model still recovers precise boundaries of salient objects, and the small object is still accurately segmented. Moreover, the proposed model segments more uniform salient objects than the compared models. It is consistency with the results in Table. 1 that our model achieves more improvement in avgF score than MAE and maxF. This phenomenon is owing to the joint training strategy of the proposed model. On one hand, the supervised attention map of the attention branch makes the detection branch further concentrate on salient objects. On the other hand, when training the proposed model, the gradient of the detection branch also back propagates to the attention branch. This training mechanism gradually promotes the proposed model to focus on salient objects. More visual

Method	Backbone	FPS	ECSSD [39]			HKU-IS [16]			DUT-OMRON [40]			DUTS [33]			PASCAL-S [19]		
			maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
Amulet [42]	VGG16	21	0.922	0.881	0.057	0.909	0.863	0.047	0.791	0.699	0.072	0.832	0.738	0.062	0.839	0.780	0.095
NLDF [25]	VGG16	20	0.915	0.886	0.051	0.908	0.871	0.041	0.759	0.694	0.071	0.830	0.759	0.055	0.840	0.792	0.083
DSS [9]	VGG16	23	0.928	0.889	0.051	0.915	0.867	0.043	0.781	0.692	0.065	0.858	0.757	0.050	0.859	0.796	0.081
BMPM [41]	VGG16	28	0.928	0.894	0.044	0.920	0.875	0.039	0.775	0.693	0.063	0.850	0.768	0.049	0.862	0.770	**0.074**
PAGR [43]	VGG19	-	0.927	0.894	0.061	0.918	0.886	0.048	0.771	0.711	0.072	0.855	0.788	0.055	0.851	0.803	0.092
PiCANet [21]	VGG16	7	0.931	0.885	0.046	0.921	0.870	0.042	**0.794**	0.710	0.068	0.851	0.749	0.054	0.862	0.796	0.076
CPD-A (ours)	VGG16	**105**	0.928	0.906	0.045	0.918	0.884	0.037	0.781	0.721	0.061	0.854	0.787	0.047	0.859	0.814	0.077
CPD (ours)	VGG16	66	**0.936**	**0.915**	**0.040**	**0.924**	**0.896**	**0.033**	**0.794**	**0.745**	**0.057**	**0.864**	**0.813**	**0.043**	**0.866**	**0.825**	**0.074**
SRM [35]	ResNet50	37	0.917	0.892	0.054	0.903	0.871	0.047	0.769	0.707	0.069	0.827	0.757	0.059	0.847	0.796	0.085
DGRL [36]	ResNet50	6	0.925	0.903	0.043	0.914	0.882	0.037	0.779	0.709	0.063	0.834	0.764	0.051	0.853	0.807	0.074
PiCANet-R [21]	ResNet50	5	0.935	0.886	0.046	0.919	0.870	0.043	**0.803**	0.717	0.065	0.860	0.759	0.051	0.863	0.798	0.075
CPD-RA (ours)	ResNet50	**104**	0.934	0.907	0.043	0.918	0.882	0.038	0.783	0.725	0.059	0.852	0.776	0.048	0.855	0.807	0.077
CPD-R (ours)	ResNet50	62	**0.939**	**0.917**	**0.037**	**0.925**	**0.891**	**0.034**	0.797	**0.747**	**0.056**	**0.865**	**0.805**	**0.043**	**0.864**	**0.824**	**0.072**

Table 1: *Comparison of different methods on five benchmark datasets and four metrics including FPS, MAE (lower is better), max F-measure (higher is better) and average F-measure. The comparison is under two settings (with VGG [29] and ResNet50 [8] backbone netowrk). The best result of each setting is shown in **Red**. "-R" means using ResNet50 as the backbone. "-A" means the results of the attention branch. All method are the trained on training set of DUTS [33]. There is not available code of PAGR [43] and the author only provides the saliency maps.*
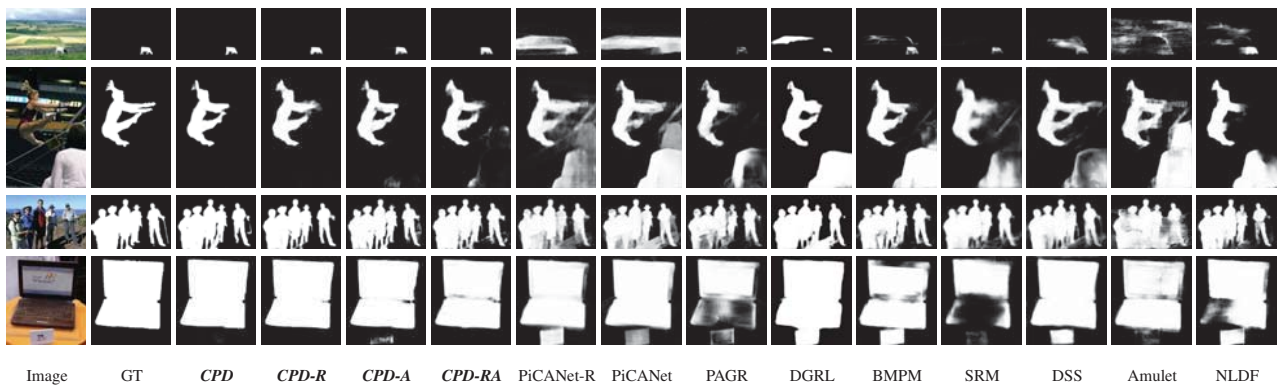

Figure 5: *Visual comparisons of the proposed model and existing state-of-the-art algorithms in some challenging cases: small object, complex scene, multiple objects and large object.*

comparison results are shown in supplementary material.

4.1.3 Application in Existing Models

Through integrating features of each branch via using aggregation algorithms proposed in existing models, our framework can be utilized to improve these works. In this paper, we apply the proposed framework in three deep aggregation models (BMPM, Amulet, NLDF). NLDF adopts a typical U-Net architecture, BMPM proposes a bi-directional decoder with gate function and Amulet integrates multi-level feature maps in multiple resolutions. We implement the improved models in their respectively default deep learning library (tensorflow [1] for BMPM and NLDF, caffe [12] for Amulet). For BMPM and NLDF, we train the improved models (denoted as BMPM-CPD and NLDF-CPD) by using default settings, and it only needs to change the learning rate from the original 10^{-6} to 10^{-5}. For A-mulet, we train the improved model (denoted as Amulet-CPD) by using the completely same settings as the original model.

In Table. 2, we show the quantitative results of the original models and the improved models (-CPD-A, -CPD) on five benchmark datasets. We can see that each improved model outperform its original model. More specially, the improved models obtain a large improvement on the two most challenging DUT-OMRON and DUTS datasets. In addition, the improved models (-CPD and -CPD-A) runs about 2 and 3 times faster than the original models respectively. In conclusion, the proposed cascaded partial decoder framework can be used to improve deep aggregation models with different kinds of decoders. In Fig. 6, we show the qualitative results on challenge cases: multiple objects, small object, large object and complex scene. The upper two rows show that the improved models further focus on target regions and suppress distractions. The under two rows

Method	FPS	ECSSD [39]			HKU-IS [16]			DUT-OMRON [40]			DUTS [33]			PASCAL-S [19]		
		maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
BMPM [41]	28	0.928	0.894	0.044	0.920	0.875	0.039	0.775	0.693	0.063	0.850	0.768	0.049	0.862	0.803	0.074
BMPM-CPD-A	82	0.932	0.901	0.046	0.920	0.875	0.037	0.796	0.731	0.057	0.864	0.799	0.046	0.861	0.817	0.074
BMPM-CPD	47	**0.935**	**0.907**	**0.043**	**0.925**	**0.888**	**0.035**	**0.804**	**0.740**	**0.056**	**0.870**	**0.808**	**0.044**	**0.868**	**0.822**	**0.072**
NLDF [25]	21	0.915	0.886	0.051	0.908	0.871	0.041	0.759	0.694	0.071	0.830	0.759	0.055	0.840	0.792	0.083
NLDF-CPD-A	75	0.918	0.889	0.049	0.914	0.873	0.039	0.775	0.710	0.061	0.837	0.773	0.050	0.841	0.793	0.083
NLDF-CPD	48	**0.922**	**0.896**	**0.044**	**0.916**	**0.880**	**0.036**	**0.781**	**0.721**	**0.060**	**0.842**	**0.786**	**0.048**	**0.843**	**0.800**	**0.080**
Amulet [42]	21	0.922	0.881	0.057	0.909	0.863	0.047	0.791	0.699	0.072	0.832	0.738	0.062	0.839	0.780	0.095
Amulet-CPD-A	61	0.925	0.889	0.053	0.910	0.864	0.045	0.790	0.708	0.070	0.832	0.747	0.060	0.842	0.784	0.091
Amulet-CPD	45	**0.934**	**0.901**	**0.047**	**0.920**	**0.878**	**0.040**	**0.805**	**0.735**	**0.063**	**0.845**	**0.771**	**0.055**	**0.851**	**0.801**	**0.085**

Table 2: *Comparison of the original models and the improved models (-CPD-A and -CPD).*

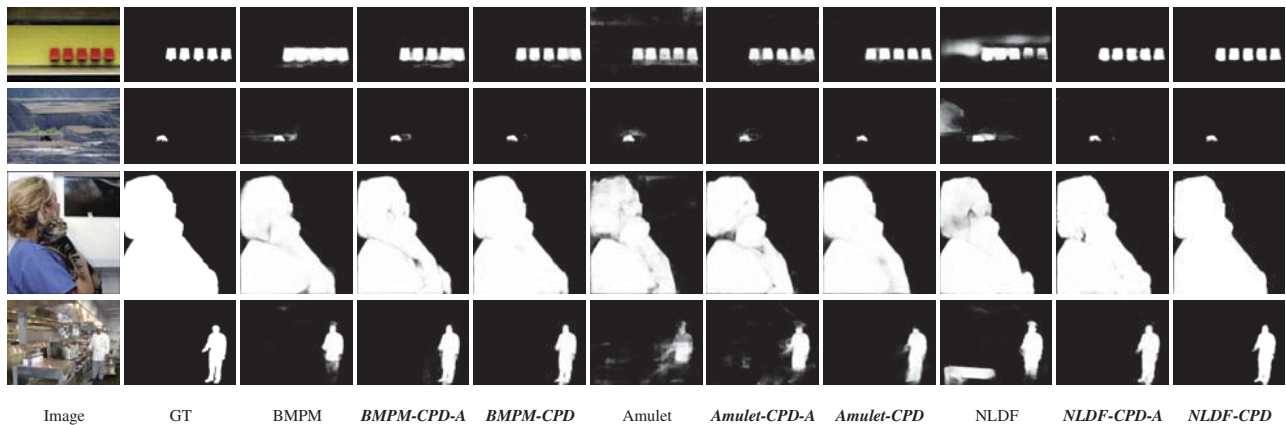| Image | GT | BMPM | **BMPM-CPD-A** | **BMPM-CPD** | Amulet | **Amulet-CPD-A** | **Amulet-CPD** | NLDF | **NLDF-CPD-A** | **NLDF-CPD** |

Figure 6: *Visual comparisons of original models (BMPM, Amulet, NLDF) with improved models (-CPD-A, -CPD).*

Settings	DUTS [33]			PASCAL-S [19]		
	maxF	avgF	MAE	maxF	avgF	MAE
CPD (with ia)	0.862	0.803	0.045	0.862	0.821	0.075
CPD (with ha)	**0.864**	**0.813**	**0.043**	**0.866**	**0.825**	**0.074**
Amulet-CPD (with ia)	0.842	0.763	0.056	0.849	0.794	0.087
Amulet-CPD (with ha)	**0.845**	**0.771**	**0.055**	**0.851**	**0.801**	**0.085**
BMPM-CPD (with ia)	0.865	0.791	0.045	0.867	0.818	**0.072**
BMPM-CPD (with ha)	**0.870**	**0.808**	**0.044**	**0.868**	**0.822**	**0.072**
NLDF-CPD (with ia)	0.838	0.777	0.051	0.840	0.793	0.084
NLDF-CPD (with ha)	**0.842**	**0.786**	**0.048**	**0.843**	**0.800**	**0.080**

Table 4: *Comparison of initial attention (ia) and holistic attention (ha) in four models (the proposed model and three improved models).*

show that the improved model further highlights the whole objects.

4.1.4 Analysis of the Proposed Framework

Effectiveness of holistic attention. Here we demonstrate the effectiveness of the proposed holistic attention model in the proposed model and the three improved models. We compare these models with holistic attention and the models with initial attention, and the results are shown in Table.4. It

is shown that holistic attention outperforms initial attention. **Selection of Optimization Layer.** In the proposed model, we set Conv3_3 layer as the optimization layer. Here we compared the proposed model with different optimization layers (Conv2_2 and conv4_3). In addition, we also report the results of no optimization layer, which means integrating all-level features via the proposed decoder. We do not test the proposed model with Conv1_2 optimization layer because this setting will increase the computation cost via adding one more full decoder; thus requirements of reducing computation cost will not be achieved. The comparison results on five benchmark datasets are shown in Table. 3. In conclusion, we set the conv3_3 layer as the optimization layer considering its best performance. When we refine the shallower feature (Conv2_2), the computation complexity increases and the performance decreases. The reason might be that the feature of shallower layer has not been enough trained. When we refine the deep feature (Conv4_3), the computation cost and the performance both decrease. This is because that resolution of the feature in the Conv4_3 layer is smaller. The accuracy and efficiency of settings (Conv2_2 and Conv4_3) both outperform the full decoder, which validates the effectiveness of the proposed framework.

Settings	FPS	ECSSD [39]			HKU-IS [16]			DUT-OMRON [40]			DUTS [33]			PASCAL-S [19]		
		maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE	maxF	avgF	MAE
Conv2_2	38	**0.936**	0.903	0.042	**0.925**	0.884	0.036	0.792	0.720	0.063	0.861	0.778	0.048	0.865	0.810	0.076
Conv3_3	66	**0.936**	**0.915**	**0.040**	0.924	**0.896**	**0.033**	**0.794**	**0.745**	**0.057**	**0.864**	**0.813**	**0.043**	**0.866**	**0.825**	0.074
Conv4_3	**90**	0.931	0.910	0.041	0.920	0.890	0.034	0.787	0.737	0.059	0.855	0.801	0.045	0.863	0.824	**0.072**
Full Decoder	30	0.922	0.891	0.051	0.911	0.873	0.042	0.758	0.692	0.070	0.843	0.766	0.050	0.853	0.807	0.077

Table 3: *Comparison of the proposed model with different optimization layers and no optimization layer (full decoder).*
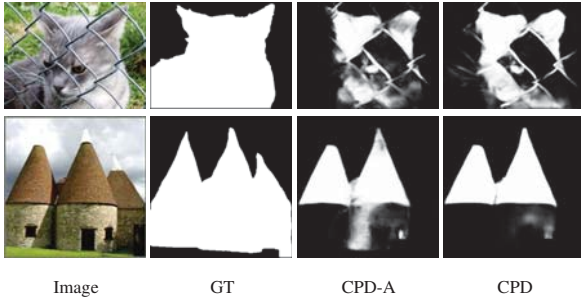

Figure 7: *Some Failure examples of the proposed model. When the attention branch only localizes a small part of target regions, our model performs poorly.*

Failure Examples. The performance of the proposed model relies on the accuracy of the attention branch. When the attention branch detects clutters as target regions, our model will obtain wrong results. In Fig. 7, we show some failure examples of our model. When a large target region is not segmented correctly, the proposed model is unable to segment the whole objects.

4.2. Application in Other Tasks

In this paper, we also evaluate the proposed model on other two binary segmentation tasks: shadow detection and portrait segmentation.

Shadow Detection. We re-train our model on the training set of SBU [30] dataset and test the model on three public shadow detection datasets: test set of SBU, ISTD [31] and UCF [45]. Moreover, we apply the widely-used metric BER (balanced error rate) for quantitative comparison. We compare our method with five deep shadow detection methods: JDR [31], DSC [10], DC-DSPF [37], scGAN [26], Stacked-CNN [30]. In addition, we re-train three salient object detection models for shadow detection: NLDF [25], DSS [9], BMPM [41]. The results are shown in Table. 5, and the proposed model outperforms the other models in all cases.

Portrait Segmentation. We use the data from [28]. And we re-train NLDF, DSS, BMPM on this dataset. The results are shown in Table 6. It can be seen that the proposed model outperforms existing algorithms.

	SBU [30]	ISTD [31]	UCF [45]
Method	BER↓	BER↓	BER↓
NLDF [25]	7.02	7.50	7.69
DSS [9]	7.00	10.48	10.56
BMPM [41]	6.17	7.10	8.09
scGAN [26]	9.10	8.98	11.50
StackedCNN [30]	11.00	10.45	13.00
JDR [31]	8.14	7.35	11.23
DC-DSPF [37]	4.90	-	7.90
DSC [10]	5.59	8.24	8.10
CPD (ours)	**4.19**	**6.76**	**7.21**

Table 5: *Comparing the proposed method with state-of-the-arts for shadow detection (DSC, DC-DSPF, JDR, StackedC-NN, scGAN), and for salient object detection (Amulet, NLD-F, BMPM, DSS).*

Methods	PFCN+ [28]	NLDF [25]	DSS [9]	BMPM [42]	***CPD*** (Ours)
Mean IoU	95.90%	95.60%	96.20%	96.20%	**96.60%**

Table 6: *Quantitative Comparison on Portrait Segmentation.*

5. Conclusion

In this paper, we propose a novel cascaded partial decoder framework for fast and accurate salient object detection. When constructing decoders, the proposed framework discards features of shallower layers to improve the computational efficiency, and utilizes generated saliency map to refine features to improve the accuracy. We also propose a holistic attention module to further segment the whole salient objects and an effective decoder to abstract discriminative features and quickly integrate multi-level features. The experiments show that our model achieves state-of-the-art performance on five benchmark datasets and runs much faster than existing deep models. To prove the generalization of the proposed framework, we apply it to improve existing deep aggregation models and significantly improve their accuracy and efficiency. Besides, we validate the effectiveness of the proposed model in two tasks of shadow detection and portrait segmentation.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878*, 2014.

[4] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2018.

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[7] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019.

[10] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, pages 7454–7462, 2018.

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[14] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[15] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, pages 660–668, 2016.

[16] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015.

[17] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016.

[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 TIP*, 25(8):3919–3930, 2016.

[19] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[20] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016.

[21] N. Liu, J. Han, and M.-H. Yang. Picanet: Learning pixelwise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018.

[22] S. Liu, D. Huang, and Y. Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, 2018.

[23] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, pages 1–8, 2007.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[25] Z. Luo, A. K. Mishra, A. Achkar, J. A. Eichel, S. Li, and P.-M. Jodoin. Non-local deep features for salient object detection. In *CVPR*, pages 6593–6601, 2017.

[26] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras. Shadow detection with conditional generative adversarial networks. In *ICCV*, pages 4520–4528, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[28] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. L. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. *Comput. Graph. Forum*, 35(2):93–102, 2016.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[30] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *ECCV*, pages 816–832, 2016.

[31] J. Wang, X. Li, and J. Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, pages 1788–1797, 2018.

[32] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015.

[33] 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 *CVPR*, pages 136–145, 2017.

[34] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.

[35] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017.

[36] 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 *CVPR*, pages 3127–3135, 2018.

[37] Y. Wang, X. Zhao, Y. Li, X. Hu, K. Huang, and N. CRIPAC. Densely cascaded shadow detection network via deeply supervised parallel fusion. In *IJCAI*, pages 1007–1013, 2018.

[38] S. Xie and Z. Tu. Holistically-nested edge detection. In *CVPR*, pages 1395–1403, 2015.

[39] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[40] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[41] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang. A bi-directional message passing model for salient object detection. In *CVPR*, pages 1741–1750, 2018.

[42] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.

[43] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018.

[44] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.

[45] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, pages 223–230, 2010.

[46] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014.