Multi-Scale Cascade Network for Salient Object Detection

Xin Li
UESTC
Chengdu, China
xinli_uestc@hotmail.com

Fan Yang*
UESTC
Chengdu, China
fanyang_uestc@hotmail.com

Hong Cheng
UESTC
Chengdu, China
hcheng@uestc.edu.cn

Junyu Chen
NCSU
Raleigh, USA

Yuxiao Guo
UESTC
Chengdu, China

Leiting Chen
UESTC
Chengdu, China

ABSTRACT

In this paper we present a novel network architecture, called Multi-Scale Cascade Network (MSC-Net), to identify the most visually conspicuous objects in an image. Our network consists of several stages (sub-networks) for handling saliency detection across different scales. All these sub-networks form a cascade structure (in a coarse-to-fine manner) where the same underlying convolutional feature representations are fully shared. Compared with existing CNN-based saliency models, the MSC-Net can naturally enable the learning process in the finer cascade stages to encode more global contextual information while progressively incorporating the saliency prior knowledge obtained from coarser stages and thus lead to better detection accuracy. We also design a novel refinement module to further filter out errors by considering the intermediate feedback information. Our MSC-Net is highly integrated, end-to-end trainable, and very powerful. The proposed method achieves state-of-the-art performance on five widely-used salient object detection benchmarks, outperforming existing methods and also maintaining high efficiency. Code and pre-trained models are available at https://github.com/lixin666/MSC-NET.

CCS CONCEPTS

• **Computing methodologies** → *Object detection*; Supervised learning;

KEYWORDS

Salient Object Detection; Deep Learning; Computer Vision

ACM Reference Format:
Xin Li, Fan Yang, Hong Cheng, Junyu Chen, Yuxiao Guo, and Leiting Chen. 2017. Multi-Scale Cascade Network for Salient Object Detection. In *Proceedings of MM '17, Mountain View, CA, USA, October 23–27, 2017,* 9 pages.
https://doi.org/10.1145/3123266.3123290

*Corresponding author.

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123290

1 INTRODUCTION

Saliency detection can be mainly classified into three directions: (i) fixation prediction, (ii) objectness estimation, and (iii) salient object detection. In this paper, we focus on the last problem, *i.e.* identifying the most visually distinctive object(s) in a natural scene. Salient object detection has drawn increasing attention for its wide application in computer vision and multimedia tasks including object recognition [28], image summarization [27], visual tracking [2], dense semantic correspondences [35], image retrieval [6], *etc.*

In recent years, deep Convolutional Neural Networks (CNNs) have played a dominant role in salient object detection. Compared with traditional methods [17] [37] [34] that employ low-level handcrafted image features, CNN-based methods [18] [32] [16] are able to encode high-level semantic information, thus largely improving the detection accuracy. However, general Convolutional Neural Networks can only capture limited local context [40] [38] from an image, but precise inference of salient object is impossible without much more global contextual information. Furthermore, the useful saliency prior knowledge (which can be obtained from previous stages) is ignored by most existing CNN-based methods. As a result, they tend to produce blurred saliency maps without fine details, making errors inevitable. To improve the quality of their results, most existing methods apply fully connected pairwise CRFs [3] as post-processing refinement steps to enforce spatial consistency [16] [32], or adopt objectness proposals to preserve boundary information [31]. However, adding post-processing steps may not only drag down the efficiency, but also produce only suboptimal results when handling challenging cases.

This paper is pursuing further improvements on salient object detection by designing a highly integrated and more powerful network that can directly produce reliable results without using DenseCRF or any additional post-processing refinement steps. The key idea is to consider both multi-scale contextual information and saliency prior knowledge during training and inferring. The multi-scale features contain useful information including spatial relation and context information, which can help to overcome ambiguity and avoid

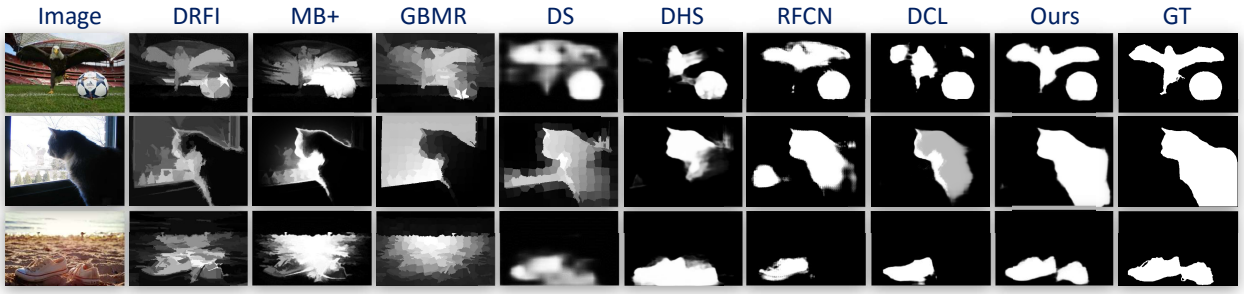


Figure 1: Sample saliency maps produced by several leading methods including three top-ranked traditional methods (DRFI [10], MB+ [37] and GBMR [34]) and five CNN-Based methods (DS [18], DHS [21], RFCN [32], DCL [16] and ours). Our method can generate very reliable saliency maps close to ground truth (GT) without needing any extra post-processing step (e.g., dense CRFs).

context information loss. Also, saliency prior knowledge can make it easier to locate the salient region(s) in an image. To effectively exploit the benefits above, we introduce a novel Multi-Scale Cascade Network for salient object detection. This network adopts a cascade structure, where each stage is designed to perform saliency detection at a specific scale. To encode useful information, each cascade stage receives feature information as well as saliency predictions processed by its preceding stage (coarser stage), and then outputs its learned features and intermediate result to the next stage (finer stage). In addition, all stages share the same underlying convolutional representations. Hence, multi-scale features and intermediate saliency scores can be jointly learned in an end-to-end network, which also makes our MSC-Net superior to previous models [31] [39] [30] that need to train multiple networks separately for collecting multi-scale contextual information.

Our MSC-Net is capable of producing reliable saliency map for the novel input (see Fig. 1). Without the use of any extra post-processing refinement methods, we report currently best results on five widely used salient object detection benchmarks. Moreover, a MSC-Net with the ResNet [7] takes only 0.26s to perform one image, which is faster than most existing CNN-based saliency models.

In summary, the main contributions of this work are three folds:

i) We propose a novel MSC-Net for salient object detection. Thanks to the cascade structure, MSC-Net is highly integrated and end-to-end trainable. More importantly, the proposed network is *naturally* capable of capturing and consolidating multi-scale contextual information and intermediate saliency prior knowledge so that it can handle challenging cases better than existing models.

ii) We fuse both shallow and deep features of the recently proposed Residual Network (ResNet) in a nonlinear manner to build a powerful underlying feature map. This combination enriches the underlying representations with more information (both low-level and high-level features), which is helpful for salient object detection.

iii) We design a refinement module for each cascade stage to further filter out errors. The module employs a recurrent architecture to encode intermediate saliency knowledge for refining the result at each stage. With the intermediate feedback signal, our network is able to locate salient regions more accurately.

2 RELATED WORK

Salient object detection is a research area that has greatly evolved over the past decade. Here we give a brief overview of recent works. We refer readers to a survey paper [1] for a comprehensive literature review.

Traditional Methods. Traditional saliency detection methods mainly depend on handcrafted image features such as color, intensity, depth, orientation, *etc.* For example, Cheng *et al.* [4] propose to use global color contrast to identity salient regions, which is based on their observation that human attention tend to be attracted by certain unique regions. Yang *et al.* [34] use image boundary color information to find the most distinctive regions on the assumption that the regions along the four boundaries of an image are usually non-salient. According to the studies of Gestalt Psychology, Zhang *et al.* [37] explore the surroundedness cue for salient object detection through Boolean Maps computed by using image intensities. Beyond that, the depth maps [25] [5], shapes [12] [11] and orientation features [13] are also used for salient object detection. Furthermore, different low-level features can also be integrated by a feature vector to distinguish between salient regions and non-salient regions. Jiang *et al.* [10] use a feature vector to learn a saliency detector from a set of training samples with manual annotations. Li *et al.* [17] propose to use the feature vector to build dense correspondences between salient regions of two different images so that the groundtruth annotations can be transferred from existing exemplars to the novel input. In general, by using different low-level features, traditional methods can produce accurate saliency maps in many simple cases, but can hardly handle complex images well due to the lack of high-level semantic knowledge.

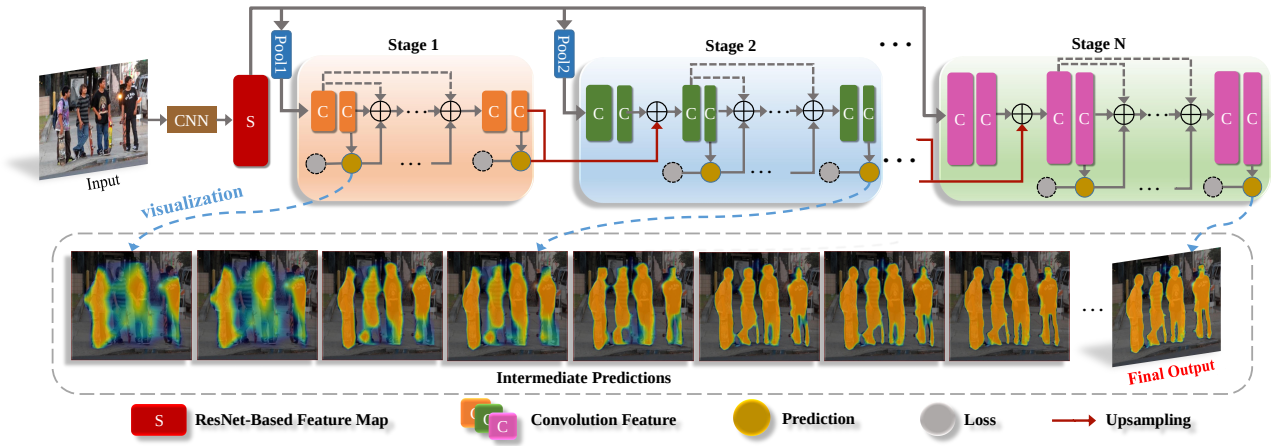


Figure 2: Overview of the Multi-Scale Cascade Network. Our MSC-Net is built upon a ResNet-based feature map and has N cascade stages. At each stage, we add several units that are used to refine the intermediate saliency map. Our MSC-Net enables the later stages (finer stages) to benefit from the contextual information and intermediate priors of the earlier stages (coarser stages), without suffering from the context information loss. The output of the last stage is our final saliency map.

CNN-Based Models. Recently, salient object detection has been made more accurate through the use of Convolutional Neural Networks (CNNs) for learning high-level feature representations. In [18], a fully convolutional network (FCN) is trained under a two-stream learning framework to identify salient objects. After that, Wang *et al.* [32] propose to detect saliency using a recurrent FCN, which can encode saliency prior knowledge and thus produce a better result. Although these CNN-based methods surpass the best performance of traditional methods, yet they ignore multi-scale contextual information, the key to salient object detection. Therefore, their results in many challenging cases are unreliable. To solve this problem, many researchers propose to employ several independent CNNs to capture multi-level information. Zhao *et al.* [39] train two CNNs to extract global and local context information, respectively. Wang *et al.* [31] employ two separate deep CNNs to capture local information and global contrast. Recently, Tang *et al.* [30] introduced a multiple CNN framework to extract pixel and region information of images and combine all information for saliency detection. However, these models involve a number of different networks that cannot be optimized jointly. Consequently, they prolong the training time and produce an only suboptimal solution.

3 METHOD

This section starts with an overview of the proposed model in Sec. 3.1, followed by a description of how to build a powerful underlying feature map based on the pre-trained ResNet [7] in Sec. 3.2. Then, Sec. 3.3 gives a detailed introduction to the MSC-Net. Finally, in Sec. 3.4, we explain how to further improve detection accuracy by using an intermediate refinement strategy (our refinement module) at each stage.

3.1 Architecture Overview

The network architecture is inspired by the importance of capturing multi-scale contextual information and saliency prior knowledge (that can be produced by a coarser stage). As can be seen in Fig. 2, our model takes a single RGB image as input and produces a saliency map of the same size as the output. The MSC-Net is built upon a shared feature map and consists of several cascade stages (in a coarse-to-fine manner) with intermediate supervision. The shared feature map, built by fusing both shallow and deep features of a pre-trained ResNet [7] model in a nonlinear manner, is pooled into varied sizes. In order to form underlying stage-wise feature maps, the dimension of the pooled features is then reduced by using one 1×1 convolution layer. The feature learning at each stage (except for the first stage) is based on (i) the stage-wise feature maps, (ii) the feature information of its preceding stage, and (iii) the previous intermediate saliency predictions. In this structure, the last stage encodes all the contextual information and saliency knowledge of all scales so that it can produce a very accurate saliency map as the final output.

This architecture has several advantages. Firstly, multi-scale contextual information can be reasonably consolidated by one single network, which favors more accurate prediction. Secondly, the intermediate supervision produces a series of saliency priors, making it easier to train a deep model and better distinguish salient objects from non-salient regions. Thirdly, the powerful underlying feature map combines both shallow and deep CNN features of a pre-trained ResNet so that it can provide rich spatial information for subsequent feature learning. Finally, because of its powerful cascade architecture, our MSC-Net can produce reliable results *without* doing any post-processing refinement, which guarantees the overall efficiency.

3.2 ResNet-Based Feature Map

Almost all existing salient object detection models are based on VGG network [29], but researchers have found that using a very deep Residual Net (ResNet) can achieve even better accuracy for many dense prediction tasks [20] [38] [3]. Hence, we adopt a pre-trained ResNet-101 with the dilated network technique [36] to build the underlying feature map. To avoid detailed information loss, the input image is scaled to 480×480 for feature learning, and the dilated network technique [36] [3] is adopted to keep last three groups of ResNet have the same resolution (*i.e.,* 60×60). It is observed that shallow and deep features of a very deep network are usually complementary to each other: shallower features are better at capturing rich spatial information while deeper features are more capable of encoding high-level semantic knowledge [8]. Therefore, we fuse multi-level features in a nonlinear manner to improve detection performance. More specifically, since the last three groups have the same resolution, we directly integrate shallower features from *Res3_4* and *Res4_23* layers and deep features from the last feature layer of ResNet into a high-dimension aggregated feature map by using a concatenation layer. After that, we use one 1×1 convolution layer after the concatenation layer to reduce the dimension and produce the final underlying feature map f. Note that the combination is jointly learnable, and trained with our MSC-Net.

3.3 Multi-Scale Cascade Network

The MSC-Net takes a ResNet-based feature map f of input image x, and produces a full-resolution saliency map $S(f;\Theta)$, where Θ represents the model parameter of MSC-Net. As can be seen in Fig. 3, it consists of N cascade stages from coarse to fine, each of which handles a specific scale $s_i \in \{s_1, s_2, ...s_N\}$ with intermediate supervision. To extract multi-scale features, the input feature map f is pooled into N features with different bin sizes (e.g., 10×10, 15×15 ... 60×60), and the dimension of the pooled features is then reduced by using one 1×1 convolution layer (from 2048 to 512). We denote these resulting 512-dimensional features as $f_{s_i} \in \{f_{s_1}, f_{s_2}, ...f_{s_N}\}$. The feature learning for the i_{th} stage encodes the reduced feature map at its own scale s_i, as well as the learned convolutional features and the intermediate saliency prediction of its previous stage s_{i-1}. Therefore, the underlying feature map F_{s_i} for the i_{th} stage is formally written as:

$$F_{s_i} = \begin{cases} f_{s_i} + \alpha_i \mathrm{R}_{s_i}(F'_{s_{i-1}}) + \beta_i \mathrm{R}_{s_i}(M_{i-1}), i \geq 2 \\ f_{s_i}, i = 1 \end{cases} \quad (1)$$

where F'_{s_i} and M_i denote the learned feature and the intermediate saliency prediction of the i_{th} stage, respectively. α and β are the combination weights for the previous learned features and saliency prediction. $\mathrm{R}_{s_i}(.)$ is a function that resizes the feature map and saliency prediction into the same size as scale s_i via bilinear interpolation.

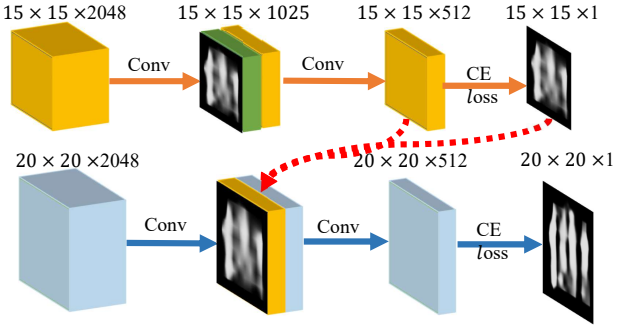


Figure 3: Illustration of the sub-network cascade between two adjacent stages. The red dashed line denotes an up-sampling operation.

We formulate the salient object detection task as per-pixel regression to groundtruth annotations. Based on their underlying feature maps F, all stages (sub-networks) jointly learn the intermediate saliency maps at their own scales. The objective of stage s_i (i_{th} sub-network) is to minimize pixel-wise discrepancy between the groundtruth map and estimated intermediate saliency map $S_{s_i}(F_{s_i};\theta_{s_i})$, and is given below:

$$L_{s_i}(\theta_{s_i}) = \min_{\theta_{s_i}} \sum_j l_{s_i}(G^j_{s_i}, S^j_{s_i}(F^j_{s_i};\theta_{s_i})), \quad (2)$$

where θ_{s_i} denotes the model parameter of the i_{th} stage, and $G^j_{s_i}$ denotes pixel-wise manual annotations of the j_{th} training image with scale of s_i. The scaled groundtruth G_{s_i} is generated through the bilinear interpolation operation for down-sampling. l_{s_i} is the salient object detection loss of $S^j_{s_i}(F_{s_i};\theta_{s_i})$ with respect to $G^j_{s_i}$. More specifically, we adopt a cross-entropy loss in Eq. 2, which is typically used in salient object task. Our loss function is defined as follows:

$$\begin{aligned} l_{s_i} = -\sum_{p=1}^{|I_{s_i}|} & G_{s_i}(p) \log \Pr(S_{s_i}(p) = 1 | I_{s_i};\theta_{s_i}) \\ & + (1 - G_{s_i}(p)) \log \Pr(S_{s_i}(p) = 0 | I_{s_i};\theta_{s_i}). \end{aligned} \quad (3)$$

We compute the loss function over all pixels in a training image I at the i_{th} stage. $G_{s_i}(p)$ and $S_{s_i}(p)$ represent the groundtruth label and estimated intermediate saliency score at pixel p, respectively.

Our model is a highly-integrated network, where all sub-networks (stages) are integrated in a single-step framework. Let $\theta = (\theta_{s_1}, \theta_{s_2} ... \theta_{s_N})$ represent the model parameters of all stages. The loss function of the whole MSC-Net is given as:

$$L_{all}(\Theta, \theta) = \sum_{i=1}^{N} \gamma_i l_{s_i}(\Theta, \theta_{s_i}), \quad (4)$$

where γ_i is the balance weight of the i_{th}-stage loss.

Our model successively integrates the learned features and intermediate saliency prediction from earlier stages. Although each stage produces its own saliency map under the intermediate supervision, we only use the saliency map produced

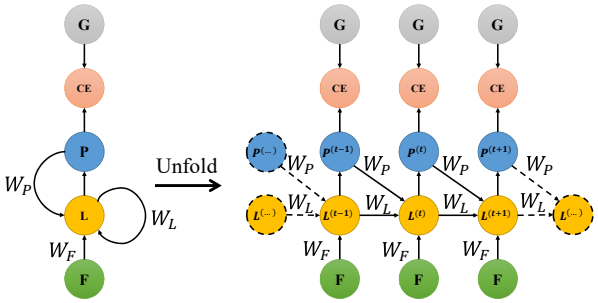


Figure 4: Illustration of the refinement module. The refinement module is a recurrent network that feeds the intermediate predictions and learned features back to the first convolution layer of each stage by recurrent connections. "F" denotes the underlying feature map; "L" means the learned features; "CE" denotes cross-entropy loss that measures the distance from the groundtruth "G" to intermediate prediction "P".

by the last stage as our final result (see Fig. 2), at which all contextual information and intermediate results have been encoded. From the standpoint of architecture and function, our model can be also considered as a novel coarse-to-fine detection approach: the coarser stages (earlier stages) roughly locate the regions likely to contain salient object(s); the finer stages (later stages) progressively update the results of their pervious stage to generate a more accurate saliency map.

3.4 Refinement Module

To make our model more robust, we consider the impact of feedback signal in deep CNN network. The key idea is to build a recurrent architecture at each stage to feed the intermediate result and learned features back to the first convolutional layer of this stage. In practice, we adopt several units under supervision, and input the intermediate predictions of one unit together with the underlying feature map and learned features into the next (see Fig. 4). To this end, we modify the first convolution layer of i_{th} stage as:

$$f(F_{s_i}) = W_{F_{s_i}} F_{s_i} + W_{L_{s_i}} L_{s_i} + W_{P_{s_i}} P_{s_i} + b, \quad (5)$$

where L and P are the learned feature and intermediate saliency map, respectively. $W_{F_{s_i}}$, $W_{L_{s_i}}$ and $W_{P_{s_i}}$ denote corresponding convolution kernels; b is the bias parameter. In the first time-step, the sub-network at i_{th} stage takes the underlying feature map F_{s_i} as input and produces the intermediate saliency map $P^1_{s_i} = S_{s_i}(F_{s_i}; \theta_{s_i})$. In the following each time-step, the learned features $L^{t-1}_{s_i}$ and the intermediate saliency map $P^{t-1}_{s_i}$ are fed back to the input. The sub-network then takes the underlying feature map F_{s_i}, the learned features $L^{t-1}_{s_i}$ and intermediate saliency map $P^{t-1}_{s_i}$ to update the intermediate result:

$$P^t_{s_i} = S_{s_i}(F_{s_i}, L^{t-1}_{s_i}, P^{t-1}_{s_i}; \theta_{s_i}). \quad (6)$$

In each unit, the network has a 3×3 convolutional layer with one padding, BN, ReLU, and one 1×1 convolutional layer with zero padding, BN, ReLU, followed by a concatenation layer. The intermediate output for t_{th} unit at i_{th} stage is a saliency map $P^t_{s_i}$ with scale s_i. The result and learned features of each unit are sent to a concatenation layer, whose goal is to encode this feedback information for the next unit. The output of the final unit at the i_{th} stage is considered as the saliency prior for the next stage (the $(i+1)_{th}$ stage), which is then scaled to the resolution with scale s_{i+1}.

Different from [32] that applies a recurrent strategy for the entire network, we apply the recurrent architecture to each cascade stage. On one hand, compared with [32], it can largely reduce computational complexity and memory cost, because this structure involves fewer parameters. On the other hand, our refinement module enables the MSC-Net to enforce more intermediate supervision, and therefore the performance of the entire network can be further improved.

4 EXPERIMENTS

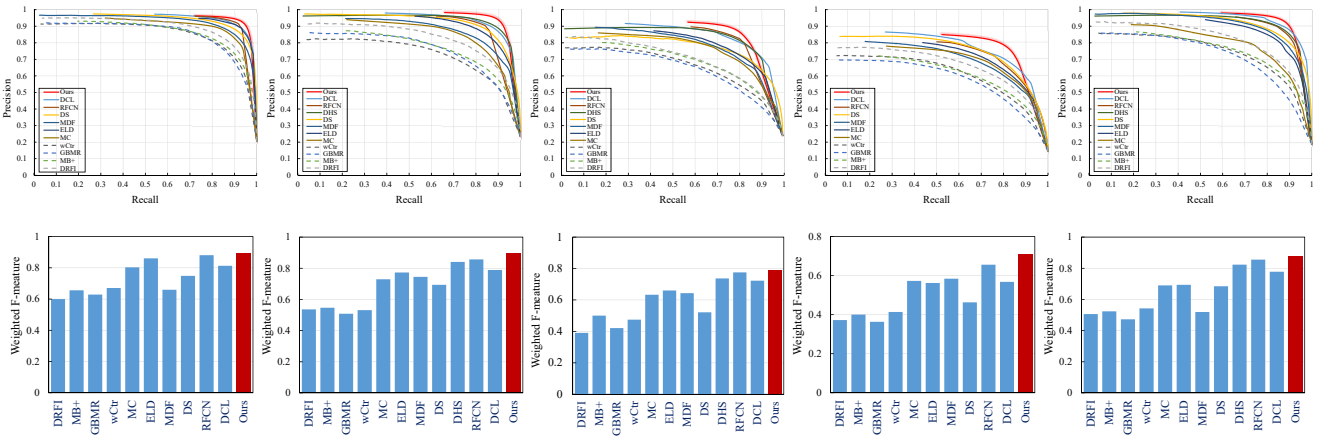
4.1 Implementation Details

Our network is based on the public platform Caffe [9]. As described above, we use ResNet-101 [7] with the dilated network strategy [36] as the pre-trained model. We use five cascade stages for handling multiple scales (*i.e.*, 10×10, 15×15, 20×20, 30×30 and 60×60) in our MSC-Net, and each stage contains two refinement units for both training and inferring.

Training. For each training image, we first resize it to 480×480 pixels. We define $\gamma_i = 1$ in Eq. 4, and utilize the "poly" learning rate policy [23], where the learning rate is scaled by $(1 - \frac{iter}{max_{iter}})^{power}$. We set the initial learning rate to 10^{-9} and the power to 0.9. The maximum number of iterations is set to 30000. The Stochastic Gradient Descent (SGD) is employed for optimization. The groundtruth is scaled to five different sizes to supervise the learning of each stage-wise saliency prediction. All stages are jointly trained in a single-step framework. The training takes about 56 hours on a NVIDIA GTX Titan X GPU with 12G memory.

We use the training set provided by [10], which includes 2,500 images selected from MSRA-B dataset [22]. Our MSC-Net has millions of parameters when only 2,500 training images are available. To reduce overfitting, the training data is augmented by flipping all the training images horizontally, following [8] [16]. We notice that many existing approaches build a stronger training set than ours, which includes much more training samples selected from different datasets. Indeed, a larger training set may lead to better performance because the model can benefit from richer information. However, we want to focus on the performance of our model itself, and thus we do not build a large training set or conduct any post-processing as they do.

Inferring. During test time, the input image is simply forwarded through the MSC-Net to generate a full-resolution saliency map. We directly take the output of the last stage



(a) MSRA-B (b) ECSSD (c) PASCAL-S (d) DUT-OMRON (e) HKU-IS

Figure 5: From top to bottom, Precision-recall (PR) curves and *weighted* F^w-measure (F^w_β) of various approaches on five popular benchmark datasets. Our approach consistently achieves the best performance in all these metrics.

as the final result of our model and do not use any post-processing procedures.

4.2 Datasets and Evaluation Metrics

4.2.1 Datasets. We evaluate our method on five widely-used benchmark datasets: MSRA-B [22], ECSSD [33], PASCAL-S [19], DUT-OMRON [34] and HKU-IS [15]. All of them contain annotations and are available online.

MSRA-B is one of the most widely used dataset, which includes 5,000 images containing various objects. Most of the images have a single salient object. Since 2,500 images are randomly sampled from MSRA-B to train our model, the remaining images are used as the test dataset for all methods.

ECSSD is a publicly available salient object detection dataset. It includes 1,000 structurally complex images and is more challenging than MSRA-B.

PASCAL-S contains 850 very challenging images, all of which are chosen from the validation set of the PASCAL VOC 2010 segmentation challenge. PASCAL-S is less biased than most of the other saliency datasets.

DUT-OMRON is a large dataset that contains 5,168 challenging images with high background clutter and one or more salient objects. It is a particularly challenging dataset. So far, none of the existing methods has achieved very high accuracy on this dataset.

HKU-IS contains 4,447 images with low contrast and less bias. Images mostly have multiple salient objects with low color contrast, which makes it very challenging.

4.2.2 Evaluation Metrics. Four commonly-used metrics are used to evaluate the performance of our model, including Precision-Recall (PR) curves, F-measure(F_β), weighted F^w-measure(F^w_β) and Mean Absolute Error (MAE). First, we can convert each saliency map S to a binary mask B using a threshold and compare the resulting binary mask against the groundtruth G. Therefore, its precision P and recall R can be computed by $P = \frac{|B \bigcap G|}{|B|}$ and $R = \frac{|B \bigcap G|}{|G|}$, respectively. The PR curve is obtained by averaging the precision and recall values over all saliency maps of a given dataset. Then, following [16], the F-measure metric is defined as:

$$F_\beta = (1+\beta^2)\frac{P \times R}{\beta^2 P + R} \tag{7}$$

where β^2 is set to 0.3 to stress the importance of the precision value, as suggested by [16] [8].

Additionally, we adopt the recently proposed weighted F^w-measure [24] for a more balanced comparison, which is defined as:

$$F^w_\beta = (1+\beta^2)\frac{P^w \times R^w}{\beta^2 P^w + R^w} \tag{8}$$

Table 1: Analysis of the MSC-Net. Our results are obtained on ECSSD dataset. "SDC" denotes the nonlinear combination of both shallow and deep features used in this paper; "MSC" refers to the multi-scale cascade architecture we designed; "RM" means that refinement module is used at each stage. " * " denotes the method used in this paper. F^w_β: the higher the better; MAE: the lower the better.

Method	F^w_β	MAE
ResNet101(Baseline)	0.808	0.066
ResNet101+SDC	0.821	0.063
ResNet101+SDC+MSC2	0.854	0.055
ResNet101+SDC+MSC5	0.868	0.051
***ResNet101+SDC+MSC5+RM**	**0.886**	**0.045**

Table 2: Quantitative comparisons with 11 leading methods on five widely-used benchmarks, including four top-ranked traditional methods and seven CNN-based models. The top three results are shown in Red, Green, and Blue, respectively. F_β: the higher the better; MAE: the lower the better.

Methods	MSRA-B		ECSSD		PASCAL-S		DUT		HKU-IS	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
DRFI [10]	0.851	0.120	0.786	0.164	0.684	0.226	0.665	0.155	0.781	0.143
MB+ [37]	0.826	0.110	0.739	0.171	0.679	0.196	0.624	0.168	0.733	0.149
GBMR [34]	0.825	0.125	0.738	0.187	0.648	0.233	0.610	0.189	0.715	0.172
wCtr [41]	0.819	0.109	0.716	0.171	0.657	0.120	0.630	0.144	0.726	0.141
MC [39]	0.872	0.062	0.822	0.107	0.721	0.147	0.703	0.088	0.781	0.098
MDF [15]	0.885	0.104	0.832	0.105	0.764	0.142	0.694	0.092	0.860	0.129
DS [18]	0.898	0.068	0.882	0.122	0.761	0.175	0.745	0.120	0.866	0.079
ELD [14]	0.914	0.042	0.869	0.098	0.777	0.121	0.720	0.091	0.767	0.071
DHS [21]	-	-	0.907	0.061	0.826	0.092	-	-	0.892	0.053
RFCN [32]	0.922	0.041	0.893	0.074	0.832	0.097	0.741	0.075	0.889	0.055
DCL [16]	0.929	0.054	0.901	0.075	0.822	0.107	0.757	0.086	0.907	0.055
Ours	0.934	0.034	0.937	0.045	0.864	0.082	0.801	0.056	0.923	0.037

where P^w and R^w are weighted precision and weighted recall, respectively. This evaluation metric overcomes three flawed assumptions of previous measures, including interpolation, dependency and equal-importance. Therefore, it can better evaluate the performance of all methods.

Finally, the most widely-used MAE score [26] is also adopted to evaluate our model. It computes the average absolute pixel-wise difference between saliency map S and groundtruth G:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|, \qquad (9)$$

where W and H are the width and the height of the saliency map, respectively.

4.3 Ablation Study

We explore three aspects of our design: the effect of combining both shallow and deep features of ResNet, the effectiveness of multi-scale cascade architecture, and the necessity of refinement module. The ResNet101-based FCN [3] is used as the baseline of our method to show the importance of our design. The overall result on ECSSD dataset is shown in Tab. 1.

Based on the baseline, we analyze the proposed components, *i.e.,* the ResNet-Based feature map (which integrates both shallow and deep features), Multi-Scale Cascade (MSC) architecture (with two and five stages) and Refinement Module (RM) by comparing the weighted F^w-measure and MAE. Firstly, we evaluate the effect of feature combination. By combining both shallow and deep features with a nonlinear manner, our method achieves 1.6% improvements according to F_β^w, and lowers the MAE score 4.5% over the baseline. Secondly, to verify the effect of the multi-scale cascade architecture, we build two architectures based on the ResNet-Based feature map: one containing two cascade stages (30×30 and 60×60) and the other one containing five cascade stages (10×10, 15×15, 20×20, 30×30 and 60×60). The experiments show that the network with multi-scale cascade architecture can largely improve the accuracy of salient object detection. Specifically, the MSC-Net with two cascade stages improves the F_β^w by more than 4.0% and lowers the MAE by 12.7%; the MSC-Net with five cascade stages results in a further 1.6% improvement of F_β^w and lowers the MAE by 7.3%. Finally, by adding the refinement module, we get the highest F_β^w 0.886 and the lowest MAE score 0.045. We find that the improvements should be mostly attributed to the multi-scale cascade architecture and refinement module, which is also the novel design of this work.

4.4 Comparison to Other Methods

We compare the proposed method with four top-ranked traditional methods and seven recent CNN-Based models. The 11 leading methods are DRFI [10], MB+ [37], GBMR [34], wCtr [41], MC [39], MDF [15], DS [18], ELD [14], DHS [21], RFCN [32] and DCL [16]. In all cases, we use the code or the saliency maps provided by the authors.

Quantitative Comparison. We compare the performance of our model, MSC-Net, to the other state-of-the-art methods on five commonly used datasets. Firstly, we compare results of all methods in terms of the F-measure and MAE scores. As can be seen in Tab. 2, our MSC-Net achieves the best performance in the two metrics above. Specifically, on MSRA-B, ECSSD, PASCAL-S, DUT-OMRON and HKU-IS, our MSC-Net improves the current best F-measure by 0.5%, 3.3%, 3.8%, 5.8% and 1.8% respectively, and lowers the MAE by 17.1%,

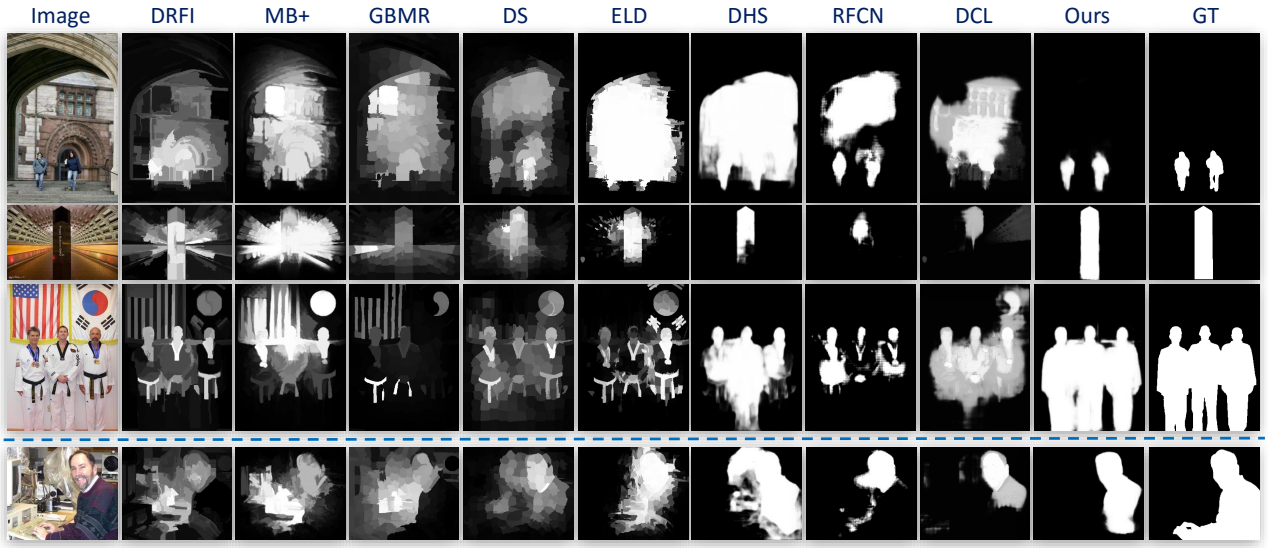


Figure 6: Qualitative comparisons of our MSC-Net and the state-of-the-art methods on some challenging scenes. The failed cases are shown in the last line.

Table 3: Comparison of running times. Mean run-times were measured on 400×300 pixel images.

Method	DRFI	MB+	GBMR	wCtr	MC	MDF	DS	ELD	DHS	RFCN	DCL	**Ours**
Times(s)	6.34	0.03	0.87	0.52	2.38	8.04	0.73	0.59	0.04	1.29	1.17	0.26

26.2%, 10.9%, 25.3% and 30.2% respectively. Furthermore, we compare our approach with the existing methods in terms of PR curve. As can be seen in Fig. 5(top), our model consistently outperforms all the state-of-the-art methods. Finally, we evaluate all methods in terms of the weighted F-measure. As shown in Fig. 5(bottom), our method still achieves the best performance. To be more specific, it improve the best existing weighted F-measure by 1.7%, 4.6%, 2.3%, 8.4% and 2.5% respectively on MSRA-B, ECSSD, PASCAL-S, DUT-OMRON and HKU-IS. Note that we omit the results of DHS [21] on DUT-OMRON and HKU-IS to avoid overrating its performance on these datasets, because its training samples are mainly selected from these test datasets. Also, it is worth mentioning that our model is trained on a relatively smaller and simpler training set and directly generates the final saliency map without using any post-processing steps, which indicates that our MSC-Net is not suffering much from the dataset bias and boundary information loss problem.

Qualitative Comparison. A qualitative comparison is shown in Fig. 6. As can be seen, our method is able to generate very reliable and accurate results even in many challenging images with strong background clutters. Compared to the saliency maps produced by existing methods, our saliency maps focus more on saliency regions, and the object boundaries are also better preserved. When handling complex images where other methods fail, our method can achieve very accurate detection.

Speed Performance. At last, we show the speed performance of the compared methods in Tab. 3. The evaluation of the traditional methods is conducted on a PC with an i7 2.50 GHz CPU and 8 GB RAM, while the CNN-based methods are accelerated by a NVIDIA GTX Titan X GPU with 12G memory. Since our method needs no post-processing procedures, it takes only about 0.26 seconds to generate a saliency map for a 400×300 input image.

5 CONCLUSIONS

In this paper, we have proposed MSC-Net, a novel CNN-based model for salient object detection. The proposed network employs a multi-scale cascade structure that can better consolidate contextual information and intermediate saliency priors. We also introduce a refinement module to further filter out errors. Our MSC-Net can generate a very reliable saliency map for an input image without needing any post-processing steps. Extensive experiments on five commonly-used benchmark datasets show that our MSC-Net can surpass the currently best performance. We believe that our approach will facilitate the development of many other computer vision and multimedia tasks.

ACKNOWLEDGMENTS. This work was supported in part by the NSFC (No.61573084, No.U1613223, No.61370073), "863" program (No.2015AA016010), and the MSTP of Dongguan (No.2015215102).

REFERENCES

[1] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. 2014. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878* (2014).

[2] Ali Borji, Simone Frintrop, Dicky N Sihite, and Laurent Itti. 2012. Adaptive object tracking by learning background context. In *Computer Vision and Pattern Recognition Workshops*. IEEE, 23–30.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016).

[4] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. 2015. Global Contrast based Salient Region Detection. *IEEE TPAMI* 37, 3 (2015), 569–582.

[5] Karthik Desingh, Madhava Krishna K, Deepu Rajan, and CV Jawahar. 2013. Depth really Matters: Improving Visual Salient Region Detection with Depth. In *BMVC*.

[6] Junfeng He, Jinyuan Feng, Xianglong Liu, Tao Cheng, Tai-Hsu Lin, Hyunjin Chung, and Shih-Fu Chang. 2012. Mobile product search with bag of hash bits and boundary reranking. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 3005–3012.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition(CVPR)*. 770–778.

[8] Qibin Hou, Ming-Ming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. 2017. Deeply supervised salient object detection with short connections. In *Computer Vision and Pattern Recognition(CVPR)*.

[9] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM on Multimedia Conference(ACMMM)*. ACM, 675–678.

[10] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient Object Detection: A Discriminative Regional Feature Integration Approach. In *Computer Vision and Pattern Recognition(CVPR)*. 2083–2090.

[11] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. 2013. Object discovery in 3d scenes via shape analysis. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2088–2095.

[12] Jaechul Kim and Kristen Grauman. 2012. Shape sharing for object segmentation. *European Conference on Computer Vision(ECCV)* (2012), 444–458.

[13] Dominik A Klein and Simone Frintrop. 2011. Center-surround divergence of feature statistics for salient object detection. In *International Conference on Computer Vision(ICCV)*. IEEE, 2214–2219.

[14] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. 2016. Deep saliency with encoded low level distance map and high level features. In *Computer Vision and Pattern Recognition(CVPR)*.

[15] Guanbin Li and Yizhou Yu. 2015. Visual saliency based on multiscale deep features. In *Computer Vision and Pattern Recognition(CVPR)*. 5455–5463.

[16] Guanbin Li and Yizhou Yu. 2016. Deep contrast learning for salient object detection. In *Computer Vision and Pattern Recognition(CVPR)*. 478–487.

[17] Xin Li, Fan Yang, Leiting Chen, and Hongbin Cai. 2016. Saliency transfer: an example-based method for salient object detection. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*.

[18] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. 2016. DeepSaliency: Multi-task deep neural network model for salient object detection. *TIP* 25, 8 (2016), 3919–3930.

[19] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. 2014. The secrets of salient object segmentation. In *Computer Vision and Pattern Recognition(CVPR)*. 280–287.

[20] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. 2016. RefineNet: Multi-Path Refinement Networks with Identity Mappings for High-Resolution Semantic Segmentation. *arXiv preprint arXiv:1611.06612* (2016).

[21] Nian Liu and Junwei Han. 2016. DHSNet: Deep hierarchical saliency network for salient object detection. In *Computer Vision and Pattern Recognition(CVPR)*. 678–686.

[22] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. 2011. Learning to detect a salient object. *TPAMI* 33, 2 (2011), 353–367.

[23] Wei Liu, Andrew Rabinovich, and Alexander C Berg. 2015. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579* (2015).

[24] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. 2014. How to evaluate foreground maps?. In *Computer Vision and Pattern Recognition(CVPR)*. 248–255.

[25] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. 2012. Leveraging stereopsis for saliency analysis. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 454–461.

[26] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. 2012. Saliency filters: Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition(CVPR)*. IEEE, 733–740.

[27] Paul L Rosin and Yu-Kun Lai. 2013. Artistic minimal rendering with lines and blocks. *Graphical Models* 75, 4 (2013), 208–229.

[28] Ueli Rutishauser, Dirk Walther, Christof Koch, and Pietro Perona. 2004. Is bottom-up attention useful for object recognition?. In *Computer Vision and Pattern Recognition(CVPR)*.

[29] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations(ICLR)*.

[30] Youbao Tang and Xiangqian Wu. 2016. Saliency Detection via Combining Region-Level and Pixel-Level Predictions with CNNs. In *European Conference on Computer Vision(ECCV)*. Springer, 809–825.

[31] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2015. Deep networks for saliency detection via local estimation and global search. In *Computer Vision and Pattern Recognition(CVPR)*. 3183–3192.

[32] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. 2016. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision(ECCV)*. Springer, 825–841.

[33] Yulin Xie, Huchuan Lu, and Ming-Hsuan Yang. 2013. Bayesian saliency via low and mid level cues. *TIP* 22, 5 (2013), 1689–1698.

[34] Chuan Yang, Lihe Zhang, Huchuan Lu, Ruan Xiang, and Ming Hsuan Yang. 2013. Saliency Detection via Graph-Based Manifold Ranking. In *Computer Vision and Pattern Recognition(CVPR)*. 3166–3173.

[35] Fan Yang, Xin Li, Hong Cheng, Jianping Li, and Leiting Chen. 2017. Object-Aware Dense Semantic Correspondence. In *Computer Vision and Pattern Recognition(CVPR)*.

[36] Fisher Yu and Vladlen Koltun. 2016. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations(ICLR)*.

[37] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomír Měch. 2015. Minimum Barrier Salient Object Detection at 80 FPS. In *IEEE International Conference on Computer Vision(ICCV)*.

[38] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2016. Pyramid Scene Parsing Network. *arXiv preprint arXiv:1612.01105* (2016).

[39] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2015. Saliency detection by multi-context deep learning. In *Computer Vision and Pattern Recognition(CVPR)*. 1265–1274.

[40] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2014. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014).

[41] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. 2014. Saliency optimization from robust background detection. In *Computer Vision and Pattern Recognition(CVPR)*. 2814–2821.