Salient Object Detection A Combination of State of Art Results

Bohong Wu, *516030910365,F1603303*, Haoqi Zhang, *516030910393,F1603303,*

Abstract—Recently, great progress has been achieved on salient object detection because of the great development of Convolutional Neural Networks (CNNS). We have developed a Fully Convolutional Neural Network (FCNs) and applied the Holistically-Nested Edge Detector (HED) which provides a skip-layer structure. But the performance of our HED is good enough so we applied short connections to the skip-layer structures based on our original HED architecture so that we can combine high-level semantic meaning with low-level boundary details. The enhanced architecture achieved great progress but still have trouble dealing with regions with low contrast ratio and boundary areas. So, we have made another two approaches to overcome these two problems, one is adding Robust Background Detection (RBD) as us prior to guide our model with contrast prior and boundary prior, the other is adding a local Boundary Refinement Network as the subnet after the HED architecture to enhance the boundary areas.

1 Introduction

VISUAL saliency is a very important issue in recent years and has gained a lot of interest. It has a lot of effective usages in a wide range of different applications such as human identification, image and video compression, image segmentation and visual question answering. Our goal in salient object detection is to try to identify the most visually distinctive objects or region in a big image and then we cut them out from the background part. When talking about the image-based salient object detection, there are two major problems need to be solved:

- How to tell the salient part from the cluttered background
- How to preserve the boundaries and details of the salient objects

The first problem has some difficulties because salient objects may have some similar visual attributes with the background parts and sometimes multiple salient objects overlap with each other, which makes it quite difficult to segment the salient objects from the rest parts of the image. The second problem also has a lot of difficulties since there has a huge contradiction between extracting the high-level global features of the whole image and preserving the quite small boundary details with in some pixels.

To solve these problems, earlier salient object detection methods were mainly based on cognitive studies of visual attention where contrast is quite important. And based on this, various hand-crafted features have been designed which are based on the prior knowledge of existing datasets, which means they are limited by these human priors and hard to extend to other data sets.

To overcome these problems in original hand-crafted features, the CNN-based and fully convolutional networks (FCNs) have appeared and have been applied to the salient object detection and have achieved great progress. However, the above two problems remain unsolved and there still has a huge space for improvement.

In this paper we applied a HED (holistically-nested edge detector) based architecture. HED explicitly deals with the scale space problems and has achieved large improvements compared with generic FCN models[1] in the context of edge detection. So, we think it could also work well in our salient object detection problem by fusing multi-level features extracted from different scales. However, our salient object detection relies heavily on high-level semantic feature representations while the edge detection task does not really need to know these high-level features and thus the original HED did not perform well in salient detection tasks. So, we have to improve the original HED so that it can both get the high-level feature and preserving low level details.

To achieve this, after observing we found that

- deeper side outputs encode high-level salient knowledge and hence can better locate where the salient objects are, but due to the down-sampling operations in FCNs, the predicted map are normally with poor details and irregular shapes especially when the input image is complex and cluttered.
- shallower side outputs capture rich spatial information and they are able to capture the boundary details of the salient objects.

Based on these observations, we tried to combine these multilevel features by introducing short connections to the skip-layer structure within the HED architecture.

Our improved network with short connections has two main advantages:

- High-level features can be transformed to shallower side-output layers so it can help them better locate the most salient thing in the image.
- Shallower side-output layers can get rich low-level features which can help us improv e the details.

However, although the salient map in our improved HED architecture with short connections has been improved greatly it still have two problems:

- Regions with low contrast ratio still perform poorly since in our short connections we still dont have applied any extra contrast information to the training network.

- The boundary part still misses some details.

To address these two problems, we have tried two different approaches.

- **RBD approach.** We have applied RBD (robust background detection) salient map as our prior knowledge to guide our model especially in detecting low contrast ratio images. RBD uses super pixel technology and takes both spatial layout and background weighted contrast into account when producing the handcrafted salient map and thus can provide both contrast prior and boundary prior. We add the RBD images as the prior knowledge and join it with our side outputs.
- **BRN approach.** We have applied a BRN (boundary refinement network) sub-network after our main HED architecture with short connections to improve the boundary region. For each position in the image, BRN aims to learn a n n propagation coefficient map with which we can learn local context information for each pixel, and we assume that every pixels saliency is influenced by its $n \times n$ neighbors.

To summarize, our main achievements are described as follows

- We used the HED architecture as our basic architecture and add the short connections between different side outputs.
- We add the RBD image as our prior knowledge to guide the model with both contrast prior and boundary prior.
- We add a BRN sub-network after the main HED architecture to improve the boundary region.

2 Method

2.1 Refined HED

We used the enhanced HED architecture[3] for our salient object detection. Compared to the original HED architecture, we added another sideoutput in the last pooling layer of our network which is pool5 in the VGGNet since we need to get more general information about the whole image. Meanwhile, we found that we have to further analysis the side outputs instead of just using it as the original HED has done since our saliency detection is a relatively harder task for the original edge detection, so there are two additional convolutional layers after we get each side output. So, in all there are 5 side outputs from $conv1\_2$, $conv2\_2$, $conv3\_3$,$conv4\_3$, $conv5\_3$ and pool5 in the VGGNet, and each with 3 convolutional layers and a final up-sampling layer. The filter channels and spatial sizes of the convolutional layers are different for different side outputs, except the last convolutional layer with 1 channel and 1*1 kernel for all of the side outputs. The detail of these convolutional layers are shown in the table below:

No.	Layer	1	2	3
1	conv1_2	128, 3×3	128, 3×3	1, 1×1
2	conv2_2	128, 3×3	128, 3×3	1, 1×1
3	conv3_3	256, 5×5	256, 5×5	1, 1×1
4	conv4_3	256, 5×5	256, 5×5	1, 1×1
5	conv5_3	512, 5×5	512, 5×5	1, 1×1
6	pool5	512, 7×7	512, 7×7	1, 1×1

We use the same up-sampling method as in the original HED architecture which is bilinear interpolation.

For every side output we use the standard cross-entropy loss to compute the loss over all pixels in the training image for every side output layer and we use the weighted-fusion layer similar to the original HED to connect these side activations together as well as calculate the fuse loss between our fused saliency image and the ground truth. We take both the fuse loss and each side outputs side loss into account when calculating the final loss, which is presented as follows.

$$\hat{l}_{\text{side}}^{(m)}\left(\mathbf{W}, \hat{\mathbf{w}}^{(m)}\right) = -\sum_{z_j \in Z} z_j \log \Pr\left(z_j = 1 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\right) \\ + (1 - z_j) \log \Pr\left(z_j = 0 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\right) \tag{1}$$

$A_{\text{side}}^{(m)} = \left\{a_j^{(m)}, j = 1, \ldots, |X|\right\}$are activations of the m-th side output which can help us to calculate the loss of the m-th side output. And we use the weighted-fusion layer similar to the original HED to connect these side activations. The loss function at the fusion layer can be defined as:

$$\hat{L}_{\text{ fuse}}\left(\mathbf{W}, \hat{\mathbf{w}}, \mathbf{f}\right) = \hat{\sigma}\left(Z, \sum_{m=1}^{\hat{M}} f_m \hat{A}_{\text{side}}^{(m)}\right) \tag{2}$$

f_m is the fusion weights and $\hat{\sigma}(\cdot, \cdot)$ represents the distance between the new fused predictions, which share the same method and formula with the side loss given above. We take both the fuse loss and each sideoutputs side loss into account when calculating the final loss. The figure below show the whole structure of our enhanced HED architecture.

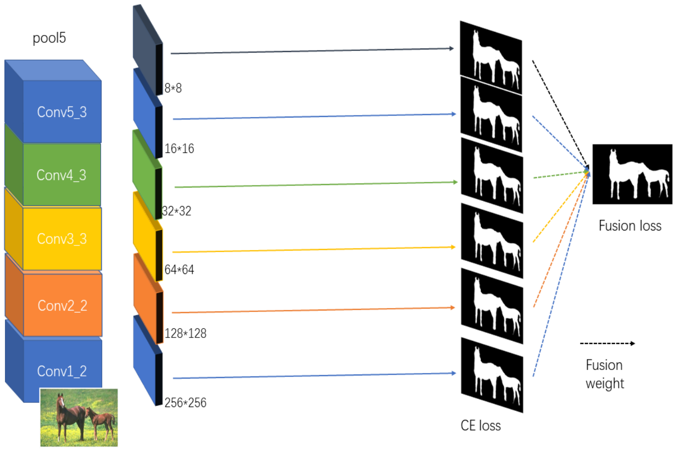

Fig. 1. **The Enhanced HED architecture.** The side outputs.... to be add

2.2 Robust Background Detection

We select RBD[4] as our handcrafted saliency detection model since it is one of the best handcrafted saliency detection methods and it takes both spatial layout and background weighted contrast into consideration.

First of all, we can abstract the image as a set of nearly regular super pixels using the SLIC method. Then we create an undirected weighted graph connect all the adjacent super pixels and assign the distance between two superpixels as the Euclidean distance between their average colors in the CIE-Lab color space. Based on the distance between different pairs of super pixels we then can get the spanning area of a super pixel which includes the super pixels that can be reached with a limited length of edges in the weighted graph. For a spanning area, we can calculate how this area is connected with the images boudaries.

In spatial layout, RBD is developed based on the statistical observation that our target salient objects and background regions in most of the images are very different in their spatial layout. The salient object regions are much less connected with the images boundaries and meanwhile the background regions are very connected with the images boundaries. So, we can define the Background connectivity to represent how much one given region is connected with the image boundaries:

$$\mathrm{BndCon}(p) = \frac{\mathrm{Len}_{bnd}(p)}{\sqrt{\mathrm{Arca}(p)}} \tag{3}$$

In this formula $BndCon(p)$ means the boundary connectivity and it is obvious that super pixel with higher boundary connectivity is more likely to be in the background in the original image. Thus, we can further define the background probability ${}^{b}_{i}g$ which is close to 1 when boundary connectivity is large and close to 0 when it is small.

Furthermore, the background weighted contrast is introduced to compute saliency for a given super pixel p which is defined as:

$$wCtr(p) = \sum_{i=1}^{N} d_{app}\left(p, p_i\right) w_{spa}\left(p, p_i\right) w_i^{bg} \tag{4}$$

$W_{spa}(p, pi)$ represents the spatial distance between the center of p and pi. From this formula the salient object regions get high ω_i^{bg} from the background regions and thus their contrast is enhanced while the background regions receive small ω_i^{bg} from the salient object regions and thus their contrast is attenuated. So, this operation effectively enlarges the contrast differences between the object and the background regions.

Finally, we fuse our handcrafted features with our side outputs to let the handcrafted prior to guide our model.

2.3 Boundary Refinement

Our HED based architecture with short connections can aggregate a lot of useful features by combing different layer side-outputs and preserve some low-level details, but a lot of details in the boundary region are still missing. So, in order to recover continuous details for obtaining spatial precision, we have used a local boundary refinement network (BRN)[2] to rectify our prediction in boundary regions.

First, we concatenate the saliency map produced by our HED based architecture with the original RGB image and take it as the input of our BRN sub network. For each position in the image, our BRN sub network aims at learning a n*n propagation coefficient map with which local context information can be aggregated to the center pixel and thus we could improve the boundary region.

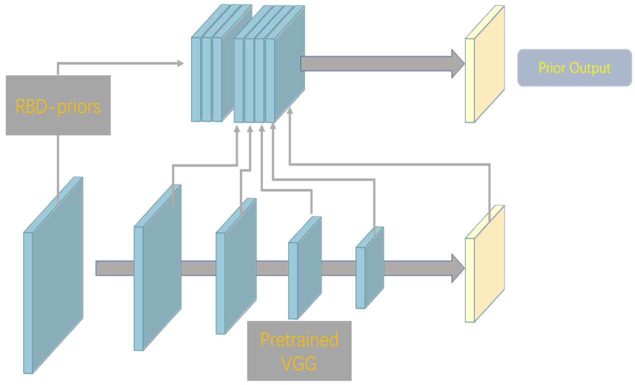

Fig. 2. **The RBD network architecture.** The side-outputs are concated and processed by convolutional layers.

For position i, BRN will first produce a propagation coefficient vector, which is actually a n×n square. The refinement map at position i can be generated by the multiplied sum of the propagation map and the original saliency map in the neighbors of i.

$$\mathbf{s}_i' = \sum_{d=1}^{n\times n} \mathbf{v}_i^d \cdot \mathbf{s}_i^d, d \in 1, 2, \ldots, n \times n \tag{5}$$

In this formula V_i^d is the coefficient vector of the d-th neighbor at position i. $n \times n$ represents the size of pixel $i's$ local neighbors, which we think will have impact on its saliency. S_i^d denotes the prediction vector at location i before the refinement and S_i' denotes the vector after the refinement. Each position in BRN has its own different propagation coefficient and it will be learned during the training.

3 Experiments

In this Section, we have made substantial experiments to validate the effectiveness of our model.

3.1 Experimental Setups

Datasets. The MSRA-B dataset contains 5,000 labeled images. We randomly divide these 5,000 images into a 4,000 training set and a 1,000 validation set. Jittering and mirroring are adopted for data augmentation. The ECSSD dataset contains 1,000 labeled images, and the HKU-IS dataset contains more than 4,000 labeled images. We use both dataset as the test set to verify the generality of our model.

Evaluation Protocols In our experiments, the performance of our model is mainly evaluated by the F_β.

$$F_\beta = \frac{\left(1+\beta^2\right) \text{ Precision } \times \text{Recall}}{\beta^2 \times \text{ Precision } + \text{Recall}} \tag{6}$$

To leverage the importance of precision and recall, β^2 is set to 0.3. We also use MAE as substantial evaluation metric.

$$mae = \frac{N_{pixels\ predicted\ right}}{N_{pixels\ of\ image}}$$

$N_{pixels\ predicted\ right}$ is the number of pixels which are correctly predicted, while $N_{pixels\ of\ image}$ depicts the number of pixels for an image.

3.2 Implementation Details

Network Structure for BRN. For BRN subnet, the implementation details is shown in the table below. BRN has 7 convolutional layers and their kernels are 3*3. We add ReLU operations between two convolutional layers to keep nonlinearity. To keep the same resolution between our original input and output feature maps we do not apply any pooling layers. In BRN subnet, we learn a matrix of size $K * H * W * 25$ as $tmp1$, where K denotes the batch size, and H, W denote the image's height and width respectively. Then, we concat the last prediction of HED architecture and the original image, pass it through convolutional layers to form a matrix of the same size $K * H * W * 25$ as $tmp2$. Then we use the dot product of $tmp1$ and $tmp2$ to form a new boundary refined feature map.

TABLE The architecture of BRN subnet

Layer	Channel	Kernel size	Bias size
1	64	(K+3)×64×3×3	64
2	64	64×64×3×3	64
3	64	64×64×3×3	64
4	64	64×128×3×3	128
5	64	128×128×3×3	128
6	64	128×128×3×3	128
7	64	128×(n×n)×3×3	5×5

Network Structure for RBD. For RBD, we simply concat the RBD output(4-channel) with the original HED side outputs(7-channel). To make sure the original details isn't lost, we also concat the original RGB channels(3-channel) with the side outputs. Therefore, the side output consists of 14 channels. Then, unlike BRN, we use convolutional layers to process the side outputs to get the salient image.

Training Details. For training, we use Adam optimizer to accelerate the converge speed. Also, we used pretrained $vgg16$. We adopt learning rate decay strategy, with the learning rate before epoch 30 set $5e^{-4}$, and after epoch 30 $5e^{-5}$. The weight initialization we adopt is Xavier. Both of the networks are trained with a batch size of 32.

3.3 Results on Different Datasets

Learning curve. The learning curves of both networks are presented as follows. The converge speed is fast, as we have used pre-trained vgg16 in our training.

Statistic Comparison. The statistical comparison is presented as follows. From the table, we could observe that in $Fscore$, both of our approaches reached comparable results with the state of art results. And the F-score of our RBD approach is slightly better than the original DSS.

TABLE F-score of all approaches on different datasets.

	MSRA-B validation	ECSSD	HKU-IS
Original DSS	0.9312	0.9264	0.9295
DSS+RBD	**0.9329**	**0.9268**	**0.9305**
DSS+BRN	0.9302	0.9244	0.9267

The substantial MAE metrics is presented as follows. Our BRN approach has a relatively lower MAE than the original network. This is because the BRN focuses on refining boundary, thus more detail is well preserved, which makes the MAE metric lower.

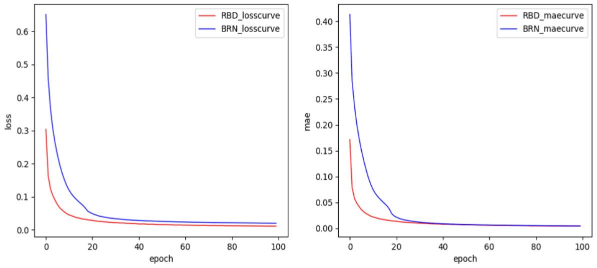

Fig. 3. **The learning curves.** The converge speed of RBD is relatively higher and the original MAE of RBD is also relatively lower. This is because in RBD approach, saliency prior information is adopted, thus the model already has some learning abilities from the very beginning.

TABLE MAE of all approaches on different datasets.

	MSRA-B validation	ECSSD	HKU-IS
Original E-HED	0.0562	0.0428	0.0441
E-HED+RBD	0.0593	0.0414	0.0446
E-HED+BRN	**0.0548**	**0.0408**	**0.0414**

4 Conclusion

In this work, we proposed two different approaches to detect the salient region of an image. Experiments on these two models showed that both of theses approaches reach the state-of-art results, and are efficient in improving the ability to detect salient regions.

References

[1] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Salient object detection with recurrent fully convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[2] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3127–3135, 2018.

[3] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[4] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821, 2014.