Learning Deep Relations to Promote Saliency Detection

Changrui Chen,[1] **Xin Sun,**[*,1] **Yang Hua,**[2] **Junyu Dong,**[1] **Hongwei Xv**[1]
[1]Ocean University of China, [2]Queen's University Belfast
{ccr, xhw}@stu.ouc.edu.cn, Y.Hua@qub.ac.uk, {sunxin, junyudong}@ouc.edu.cn

Abstract

Though saliency detectors has made stunning progress recently. The performances of the state-of-the-art saliency detectors are not acceptable in some confusing areas, e.g., object boundary. We argue that the feature spatial independence should be one of the root cause. This paper explores the ubiquitous relations on the deep features to promote the existing saliency detectors efficiently. We establish the relation by maximizing the mutual information of the deep features of the same category via deep neural networks to break this independence. We introduce a threshold-constrained training pair construction strategy to ensure that we can accurately estimate the relations between different image parts in a self-supervised way. The relation can be utilized to further excavate the salient areas and inhibit confusing backgrounds. The experiments demonstrate that our method can significantly boost the performance of the state-of-the-art saliency detectors on various benchmark datasets. Besides, our model is label-free and extremely efficient. The inference speed is 140 FPS on a single GTX1080 GPU.

1 Introduction

In the deep learning era, deep neural networks based models significantly boost the performance of saliency detection. Nevertheless, these models are also unsure about some confusing saliency area. As shown in figure 1c, the saliency detector makes a wavering prediction on the boundary part. Furthermore, component missing, shown in figure 1b, is also a common problem in the saliency prediction.

However, human beings can easily distinguish all parts of an object. Most of the DNNs based saliency detectors are derived versions of the FCNs (Shelhamer, Long, and Darrell 2016). In the common training strategy of the FCNs, the feature vector at each pixel is assigned with an independent ground truth label. The neighbor feature vectors have little communication during the training and inference phases. We call this phenomenon **feature spatial independence**. Different from these FCNs, human beings can utilize the color similarity, material texture, and edge coherence to assist object perception. Consequently, we believe that the feature spatial independence should be one of the root cause of the above-mentioned problems in saliency detection.

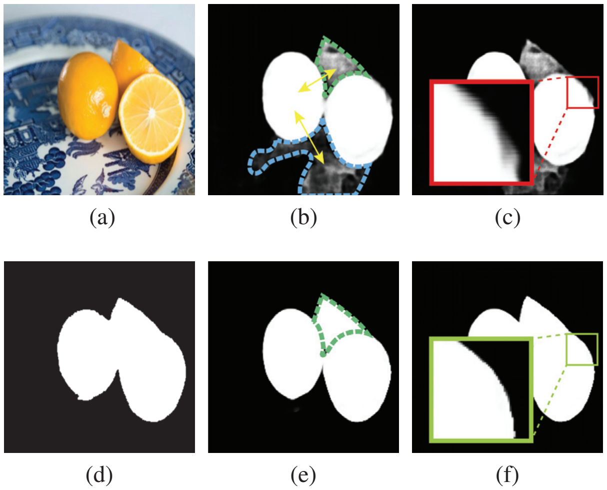

Figure 1: Illustration of the saliency result refinement. (a) The example natural image. (b) The result of the UCF (Zhang et al. 2017b) saliency detector. The green dash line surrounds the missing saliency area. The blue dash line indicates the fake saliency area. We can use relations, marked by the yellow arrow, to remedy the saliency result. (c) The wavering boundary. (d) The ground truth (GT). (e) The result refined by our method. The missing component surrounded by the green dashed line appears. (f) The highlight of the boundary refined by our method

We consider using the relation to break the independence. Some conventional methods such as DenseCRFs (Krähenbühl and Koltun 2013) use the conventional features such as RGB value to build the relation. There is no doubt that the deep features are more representative than the conventional features. In this paper, we build deep relations to break independence between deep feature vectors. Benefiting from the deep relations, our method can not only eliminate the unconfirmed area (e.g., figure 1e is the refined result) but also enhance the blurry edge (e.g., figure 1f). To establish the relations between deep features, we appeal to mutual information (MI) (Kullback 1962), which is widely

[*] Prof.Sun is the corresponding author.

applied in natural language processing. The MI measures the dependence between two variables. To some extent, MI is the same as the *similarity* concept in human's mind. In figure 1b, a high value of MI between the green dashed line area and the salient region can help us refine the results in that area. Therefore, the key is to ensure that the features of similar regions have high mutual information. We propose a simple but efficient model. Through jointly optimizing a feature extractor and a discriminator, the MI between the feature vectors of saliency areas is maximized. During the inference phase, we estimate the MI between the high-confidence foreground feature vectors with all the feature vectors. As a result, we can generate the refined saliency result by merging the prior saliency map into the relation estimation output.

The experiments demonstrate that our method can significantly promote the state-of-the-art saliency detectors including conventional methods and deep-learning based methods. Notably, our method is label-free. Our training can be performed with a self-supervised strategy. Moreover, our method is extremely efficient. The inference speed of our method is 140 fps on a single GTX 1080 GPU.

In summary, the main contributions of this paper are:

- We promote saliency detectors via learning the relations on the deep feature maps by maximizing the mutual information (MI).
- Armed with our method, all the state-of-the-art methods in our experiment are boosted significantly on four benchmark datasets.
- Our method is trained under a self-supervised scheme without any ground truth. Moreover, our method is extremely fast.

2 Related Work

Saliency Detection. In the early days, saliency cues and the handcrafted features were the main driving force of the conventional saliency detectors. For example, Cheng *et al.* (Cheng et al. 2014) utilize the global contrast to generate the saliency map. Zhu *et al.* (Zhu et al. 2014) propose a robust background measure for saliency optimization. Qin *et al.* (Qin et al. 2015) propose cellular automata dynamic evolution model to intuitively detect the salient object. Recently, the deep neural networks (DNNs), specifically the CNNs, have been widely applied in various fields of computer vision. Many papers take advantage of the powerful feature extracting ability of the CNNs to boost the performance of the saliency detection models significantly. Li and Yu (Li and Yu 2016) extract the multi-scale features from the DCNNs to replace the handcrafted feature. Liu *et al.* (Liu et al. 2015) fuse the bottom-up and top-down method. Hou *et al.* (Hou et al. 2017) propose a salient object detection method promoted by the short connections of a skip-layer within the holistically-nested edge architecture. Zhang *et al.* (Zhang et al. 2017a) argue there is no end of fusing the multi-level convolutional features and propose a generic framework to aggregate it. Wang *et al.* (Wang et al. 2017b) propose a multi-stage refinement mechanism for saliency detection. RADF (Hu et al. 2018) use recurrently aggregated deep features to detect saliency object. Zhuge *et al.* (Zhuge, Zeng, and Lu 2019) argue that the noise in some features are harmful to saliency detection. PiCANet (Liu, Han, and Yang 2018), RAS (Chen et al. 2018), and PFA (Zhao and Wu 2019)) both adopt the attention mechanism to get better saliency result. R^{3}Net (Deng et al. 2018) use a recurrent residual refinement to more accurately detect salient regions.

Post Processing. The most relevant approach to ours is Zeng *et al.* (Zeng et al. 2018). They propose a novel model to promote the saliency detectors by embedding the image features to the foreground and background anchors with some ground truth. In contrast, our method can improve all existing saliency detection approaches in an unsupervised way. DenseCRFs (Krähenbühl and Koltun 2013) is a widely used post-processing method, which builds a graph of an image and optimizes the energy function to refine the segmentation prediction. DenseCRFs generates the unary item by deep neural networks and uses some conventional features such as RGB value to estimate the pairwise item. In this paper, we use MI to estimate the pairwise relations between deep features.

Mutual Information. MI is used to measure the mutual dependence between two variables. The InfoMax optimization principle (Bell and Sejnowski 1995; Linsker 1988), which is the objective for the neural network, advocates maximizing the mutual information between the input and output. For so long, mutual information could not be accurately estimated in neural networks. MINE proposed by Belghazi *et al.* (Belghazi et al. 2018) estimates MI by gradient descent with a neural network, and they apply it to promote the generative adversarial networks. The other application of mutual information in the neural network is DIM (Hjelm et al. 2019) which is to learn the satisfactory representation of the input image. They all use a discriminator to train their models but discard it after training. In this paper, we endeavor to construct the relationship between the feature vectors of different image areas via mutual information to improve the saliency detection without any ground truth. In addition, our discriminator is not only a tool for maximizing mutual information but also a key detector to generate the saliency map.

3 MI for Deep Relation Estimation

In this section, we firstly describe our conception of modeling relations by estimating mutual information with deep neural networks. Then, we illustrate the derivation of the conception.

3.1 Mutual Information in Saliency Detection

The existing conventional and deep saliency detectors can almost distinguish the foreground and background areas but they are still indecisive about some indistinguishable areas. As mentioned previously, the relations, such as the material similarity and the edge coherence, can tackle the ambiguous area ascription problem caused by the feature spatial independence.

The widely applied MI measures the dependence between two random variables. It quantifies the information of one random variable we obtained after observing the other random variable. Actually, in a saliency detection task, vague areas which belong to the saliency object should have high mutual information with the confident foreground area. This property can help us eliminate vague areas. Therefore, we can establish relations by maximizing the mutual information between the deep features of the saliency object.

3.2 Derivation

Inspired by Belghazi *et al.* (Belghazi et al. 2018), we train a deep neural network to estimate mutual information. Our network consists of a feature extractor and a discrimiantor.

We use the convolutional feature extractor E_ω with learnable parameters ω to extract the robust and representative feature vectors of the input image. Let X and Y be two random variables. In this paper, X denotes the deep feature vectors of confident foreground areas and Y denotes the feature vectors of random areas. Formally, the mutual information of X and Y (Kullback 1962) can be defined as:

$$I(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p(x,y) log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy, \tag{1}$$

where the X and Y are both extracted by E_ω. So, our target is to obtain the best parameters ω which can maximize the MI:

$$\omega = \underset{\omega}{\text{argmax}}(I(X;Y)), \quad (X, Y \in E_\omega(Image))^1. \tag{2}$$

To estimate MI, we consider the divergence between two distributions: the joint distribution $\mathbb{J}(X,Y)$ and the product of two marginal distribution $\mathbb{P}(X) \otimes \mathbb{P}(Y)$. The mutual information between X and Y can be transformed to the ***KL-divergence*** between these two distributions (Kullback 1962):

$$I(X;Y) = \mathcal{D}_{KL}(\mathbb{J}(X,Y)||\mathbb{P}(X) \otimes \mathbb{P}(Y)). \tag{3}$$

Maximizing the mutual information is equivalent to maximizing the *KL-divergence*. Because there is no upper boundary of *KL-divergence*, we use the ***JS-divergence*** to do the maximizing optimization instead:

$$\begin{aligned} &\mathcal{D}_{JS}(\mathbb{J}(X,Y)||\mathbb{P}(X) \otimes \mathbb{P}(Y)) = \\ &\frac{1}{2}\mathcal{D}_{KL}\left(\mathbb{J}(X,Y)||\frac{\mathbb{J}(X,Y)+\mathbb{P}(X)\otimes\mathbb{P}(Y)}{2}\right) + \\ &\frac{1}{2}\mathcal{D}_{KL}\left(\mathbb{P}(X)\otimes\mathbb{P}(Y)||\frac{\mathbb{J}(X,Y)+\mathbb{P}(X)\otimes\mathbb{P}(Y)}{2}\right). \end{aligned} \tag{4}$$

The upper boundary of *JS-divergence* is $\frac{1}{2}log2$. To estimate the *JS-divergence*, we adopt the local variational inference estimation proposed by Nowozin *et al.* (Nowozin, Cseke, and Tomioka 2016):

$$\begin{aligned} &\mathcal{D}_{JS}(\mathbb{J}(X,Y)||\mathbb{P}(X) \otimes \mathbb{P}(Y)) = \\ &\quad \max_F(\mathbb{E}_{(x,y)\sim\mathbb{J}(X,Y)}[log\sigma(F(x,y))] + \\ &\qquad \mathbb{E}_{(x,y)\sim\mathbb{P}(X)\otimes\mathbb{P}(Y)}[log(1-\sigma(F(x,y)))]), \end{aligned} \tag{5}$$

where F indicates a discriminator that can determine which distribution the sample (x, y) belongs to. If we simultaneously optimize the discriminator F and the feature extractor E_ω to maximize the value of the right hand of Eq. 5, we can maximize $\mathcal{D}_{JS}$, which leads to MI maximization.

3.3 Optimization

Notably, Eq. 5 is very similar to the Binary Cross Entropy Loss Function:

$$CELoss = -(y \log(p) + (1-y)\log(1-p)), \tag{6}$$

where y is a binary indicator and p is the predicted probability of $y = 1$. In this paper, y indicates which distribution the sample (x, y) belongs to. In the saliency detection task, the confident saliency area and the area which probably belongs to the object commonly appear together. So, we suppose that they can be the pair sampled from the joint distribution $\mathbb{J}(X,Y)$ with high MI. On the contrary, the confident saliency area and the area which seems to be the random background is the pair sampled from $\mathbb{P}(X) \otimes \mathbb{P}(Y)$ with low MI. The predicted probability p is calculated by $\sigma(F(x,y))$, where σ is the sigmoid activation function.

We use the mini-batch gradient descent to minimize the binary cross entropy loss with proper training pairs through end-to-end training to optimize the loss function. When the binary cross entropy loss converges, the extractor E_ω can generate the feature vectors of the confident and vague foreground areas, which meet the requirement of mutual information maximizing. We also get a satisfactory discriminator F which can distinguish the pairs sampled from the joint distribution. This discriminator can help us to determine the high MI feature pairs. Therefore, we can promote the saliency detection results by estimating the mutual information between deep feature vectors of confident foreground areas and vague areas.

4 Mutual Information Relation Model

In this section, we introduce the pipeline of the mutual information relation model for promoting the saliency detection results. As shown in figure 2, the network mainly consists of three parts. The first one is a feature extractor. The second part is composed of a multi-scale feature fusing layer and a series of operations for training the whole model. The last part illustrates the testing phase and the generation of the final refined saliency map.

4.1 Multi-scale Feature Extraction

As shown in the left part of figure 2, we firstly feed an image into a fully convolutional neural network to obtain the deep features. In this paper, we use the MobileNet v2 (Sandler et al. 2018) without the fully connected layer as our feature extractor. The low-resolution feature maps that obtained

[1] Here we slightly abuse the $\in$.

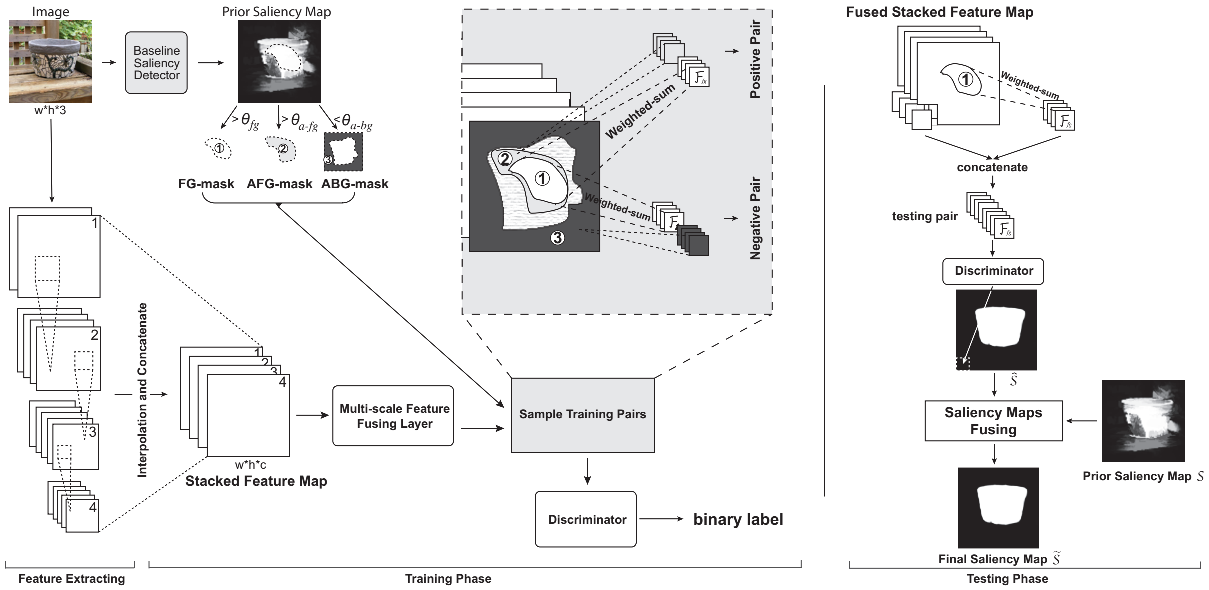

Figure 2: The pipeline of the proposed network to promote the saliency detection via maximizing the mutual information. The fused stacked feature maps in testing phase are also produced by the multi-scale feature fusing layer.

from the high convolutional stages are insufficient for producing the exact saliency map. Thus, we collect the feature maps from convolutional layer 0, 3, 13, and 17. In the multi-scale feature fusing module, we upsample these four groups of feature maps to the same height and width via bilinear interpolation. Then, we concatenate all the high-resolution feature maps. The final feature maps are two times smaller than the input image dimension. Inspired by the Deeplab (Cheng et al. 2014), we apply an ASPP module on the high-resolution feature maps and reduce their channel number to 32.

4.2 Training Samples

Guide masks. To provide the positive and negative training pairs to the discriminator, as shown in the upper-left of figure 2, we use a baseline saliency detector to produce the prior saliency map initially. The prior saliency map can guide to construct the training pairs and suggest the labels of the training pairs. Three kinds of masks are generated by three different thresholds. θ_{fg} is the certain foreground threshold. The first kind of mask generated by θ_{fg} is the **F**ore**G**round mask (**FG-mask**) marked by ① in figure 2. Distinguishing most saliency areas is not a puzzle for existing detectors. Therefore, we assume that all the pixels in the FG-mask area belong to the foreground object. The second kind mask Approximate ForeGround Mask (**AFG-mask**) marked by ② and the third kind mask (**ABG-mask**) marked by ③ are generated by **A**pproximate **F**ore**G**round and **B**ack**G**round thresholds θ_{a-fg} and θ_{a-bg}. We assume that most of the pixels in the AFG-mask or the ABG-mask pertain to the foreground objects or the background.

Construct training pairs. We aggregate all the doubtless foreground feature vectors masked by the FG-mask to one foreground vector $\mathcal{F}_{fg}$ as below,

$$\mathcal{F}_{fg} = \sum_i p_i f_i, \tag{7}$$

where i indicates the pixel location in the FG-mask. f_i is the feature vectors at the location i. p_i is the normalized saliency probability calculated by $p_i = \frac{s_i}{\sum_i s_i}$, where s_i is the prior saliency probability at the location i.

Then, we concatenate the feature vectors f_{afg} located in the approximate foreground area with the $\mathcal{F}_{fg}$ to construct the positive pair u:

$$u = [\mathcal{F}_{fg}, f_{afg}], \tag{8}$$

where $[\cdot,\cdot]$ denotes concatenation. The positive pair is the one which sampled from the joint distribution of $\mathbb{J}$ in Eq. 5. We also generate the negative pair v which are composed of $\mathcal{F}_{fg}$ and the vectors f_{abg} in the area masked by the ABG-mask:

$$v = [\mathcal{F}_{fg}, f_{abg}]. \tag{9}$$

The negative sample is the one which sampled from the product distribution in Eq. 5.

Once we obtain the positive and negative training pairs, we can send them to the discriminator and train the whole network by the binary cross-entropy loss with the binary labels, specifically 1 for positive and 0 for negative. The mutual information between feature vectors is maximized when the loss converges.

4.3 Saliency Map Generating

During inference phase, we concatenate the certain foreground feature vector $\mathcal{F}_{fg}$ with all the feature vectors in the fused stacked feature maps and send them to the trained discriminator. The output of the discriminator represents not only the probability that the pair is the one sampled from the joint distribution but also the probability that the pixel belongs to the salient object.

Finally, we merge the output generated by the discriminator with the prior saliency result. For a saliency prediction s_i at location i, we define the confidence value c_i by:

$$c_i = \begin{cases} s_i, & s_i > 0.5 \\ 1 - s_i, & otherwise. \end{cases} \quad (10)$$

We calculate the confidence value $\widehat{c}_i$ for the new saliency result $\widehat{s}_i$ produced by the discriminator, and $\overline{c}_i$ for the prior saliency result $\overline{s}_i$. We produce the final saliency result according to the confidence value:

$$s_i = \begin{cases} \overline{s}_i, & \overline{c}_i > \widehat{c}_i \\ \widehat{s}_i, & otherwise. \end{cases} \quad (11)$$

5 Experiments

5.1 Datasets and Basic Algorithms

In our experiment, we use four well-known saliency benchmark datasets to evaluate our method. **HKU-IS** (Li and Yu 2016) contains 4447 images with multiple salient objects. **DUT-OMRON** (Yang et al. 2013) includes 5168 complicated images with one or two salient objects. **Pascal-S** (Li et al. 2014) which contains 850 natural images is a subset of the PASCAL VOC2010 dataset. **ECSSD** (Yan et al. 2013) contains 1000 images with multiple objects of varying sizes. The training dataset of our model is **DUTS-TE** (Wang et al. 2017a) which has 5019 images collected from the ImageNet DET dataset (Deng et al. 2009).

We choose nine state-of-the-art deep learning methods (i.e., Amulet (Zhang et al. 2017a), UCF (Zhang et al. 2017b), ELD (Lee, Tai, and Kim 2016), NLDF (Luo et al. 2017), SRM (Wang et al. 2017b), PiCANet (Liu, Han, and Yang 2018), RAS (Chen et al. 2018), R^{3}Net (Deng et al. 2018), and PFA (Zhao and Wu 2019)) and three conventional methods including MB+ (Zhang et al. 2015), wCtrO (Zhu et al. 2014) and BSCA (Qin et al. 2015) as our baseline saliency detectors.

5.2 Evaluation Metrics

We adopt two widely used evaluation metrics. The first one is the F-measure which is a comprehensive performance indicator:

$$F_\beta = \frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}, \quad (12)$$

where the $precision$ indicates the ratio of the correctly labeled saliency pixels in the predicted saliency map. The $recall$ is the ratio of the correctly labeled saliency pixels in the ground truth. Following the suggestion of Achanta *et al.* (Achanta et al. 2009), we use the double mean value of the predicted saliency map as the threshold to measure the F-measure. The β^2 is set to 0.3.

The second metric is the mean absolute error (MAE) which is used to measure the average discrepancy between the saliency result and the ground truth:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |S_{ij} - GT_{ij}|. \quad (13)$$

5.3 Implementation Details

The training is operated on a PC with a GTX 2080ti GPU. We use a single GTX 1080 during the inference phase.

The feature extractor mentioned in Section 4 is a MobileNet v2 (Sandler et al. 2018) pretrained on the ImageNet dataset (Deng et al. 2009). The discriminator consists of 5 convolutional layers. We use a sigmoid function as the output layer to generate the output. We illustrate the detailed architecture of the discriminator in the supplementary material (submitted to the code repository). All the thresholds for generating the masks are set as: $\theta_{fg} = 0.9, \theta_{a-fg} = 0.8, \theta_{a-bg} = 0.3$. The code will be published on https://github.com/ouc-ocean-group/LDPS soon.

We train our model on DUTS-TEST dataset which contains 5019 images without the ground truth. We use the saliency maps generated by PiCANet as the prior saliency maps to construct the positive and negative pairs for optimizing our model. The entire network is trained end-to-end by SGD with backpropagation. We train our model on only 1 GPU for 20k iterations, with a learning rate of 5e-4 for backbone and 5e-3 for the rest components. The learning rates are decreased by the polynomial learning rate policy.

5.4 Performance

We evaluate the trained model on four benchmark datasets. The F-measure and MAE scores of all the baseline saliency detectors and our refined results are reported in table 1. We summarize the significant improvements as follows:

(1) The F-measure scores of all baseline methods increase dramatically after refining with our method. Not only the conventional methods but also the state-of-the-art deep-learning methods such as the RAS and the PFA also benefit from our method.

(2) Our method can decrease MAE of all the methods including the latest methods with ultra-low MAE scores.

(3) Notably, the best results of each benchmark dataset are illustrated in bold respectively in table 1. We can see that our method presents the best performance on all the datasets by refining the state-of-the-art methods without any ground truth.

(3) Our method can help the poor saliency detectors match or even exceed the good detectors. For example, the F-measure of the refined wCtrO on HKU-IS dataset is 0.8585, which is higher than the raw Amulet, UCF, and ELD.

Figure 3 visualizes the results of some state-of-the-art saliency detectors including SRM, RAS, and PFA as well as the refined results with our method. We show more visualizations of all the prior detectors in the supplementary material. Obviously, our method can highlight the neglected object areas. Furthermore, the redundant background areas are also restrained.

Table 1: Improvement of the F-measure (higher is better) and MAE (lower is better) after refining by our method. The Baseline is the basic performance of each method. The best methods are illustrated in bold respectively.

Datasets		**ECSSD**		**DUT-O**		**HKU-IS**		**Pascal-S**	
Methods		$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓
MB+	Baseline	0.6902	0.1708	0.5215	0.1679	0.6677	0.1503	0.6161	0.1983
	Ours	0.8418	0.0841	0.666	0.1048	0.8357	0.0672	0.7504	0.1202
BSCA	Baseline	0.7024	0.1824	0.5087	0.1907	0.6543	0.1748	0.5953	0.2238
	Ours	0.8617	0.0817	0.6909	0.1022	0.8575	0.0680	0.7558	0.1233
wCtrO	Baseline	0.6763	0.1713	0.5277	0.1438	0.6769	0.1424	0.5963	0.2012
	Ours	0.8537	0.0893	0.7015	0.0866	0.8585	0.0658	0.7552	0.1225
Amulet	Baseline	0.8682	0.0588	0.6472	0.0975	0.8408	0.0506	0.7632	0.0997
	Ours	0.9079	0.0473	0.7105	0.0805	0.8912	0.0387	0.8037	0.0839
UCF	Baseline	0.8435	0.0691	0.6205	0.1203	0.8231	0.0619	0.7305	0.1160
	Ours	0.8964	0.0505	0.7125	0.0871	0.8893	0.0411	0.7808	0.0900
ELD	Baseline	0.8157	0.0723	0.6571	0.0876	0.8164	0.0636	0.7126	0.1130
	Ours	0.8884	0.0501	0.7418	0.0715	0.8919	0.0413	0.7901	0.0872
NLDF	Baseline	0.8783	0.0626	0.6836	0.0795	0.8735	0.0477	0.7742	0.0989
	Ours	0.9009	0.0514	0.7260	0.0703	0.8994	0.0395	0.8046	0.0869
SRM	Baseline	0.8922	0.0544	0.7068	0.0693	0.8738	0.0459	0.7961	0.0852
	Ours	0.9158	0.0460	0.7432	0.0637	0.9041	0.0379	0.8244	0.0759
PiCANet	Baseline	0.8872	0.0456	0.7496	0.0653	0.8766	0.0413	0.8033	0.0782
	Ours	0.9096	0.0406	0.7899	0.0603	0.9074	0.0359	0.8288	0.0723
RAS	Baseline	0.8893	0.0564	0.7129	0.0617	0.8705	0.0453	0.7807	0.1037
	Ours	0.9109	0.0499	0.7484	0.0580	0.8993	0.0394	0.8093	0.0937
R^{3}Net	Baseline	0.9148	0.0399	0.7562	0.0623	0.8941	0.0356	0.8029	0.0933
	Ours	**0.9208**	**0.0383**	0.7667	0.0608	0.9038	0.0336	0.8111	0.0897
PFA	Baseline	0.8863	0.0448	0.7842	0.0414	0.8847	0.0324	0.8224	0.0648
	Ours	0.9138	0.0383	**0.8147**	**0.0402**	**0.9127**	**0.0290**	**0.8472**	**0.0601**

Comparison of other refinement method. We compare our method to LPS (Zeng et al. 2018) which is a novel model to promote saliency detectors. The official code and the official pretrained model of LPS are implemented in our experiments. Following LPS, we also use VGGNet (Simonyan and Zisserman 2015) as the feature extractor to guarantee a fair comparison. LPS trains their model on the DUTS-TRAIN datasets (10000 images) with ground truth. The performance comparison of LPS and our model is shown in table 2. All the F-measure scores and the MAE scores of our method are better than LPS. Moreover, our method can refine a result with 256 × 256 resolution at 140+ fps with TensorRT (90+ fps without TensorRT), which is extremely faster than LPS (11 fps).

In the comparison of our method with the DenseCRFs (Krähenbühl and Koltun 2013), for the sake of fairness, we also initialize our model for each image and refine the prior saliency map relying on only one image. The learning rate and the training iteration steps is set to 0.2 and 10 respectively. Some results are shown in table 3. Our method achieves better performance than DenseCRFs. Moreover, If we use the training scheme mentioned in Section 5.2 to train our model, our model can significantly outperform DenseCRFs. More detail of the network architecture and results can be found in supplementary material.

5.5 Ablation Studies

In this section, we choose SRM, which is a stable and outstanding saliency detector, as the prior detector to analyze our method in detail.

Pairs sampling. We binarize the prior saliency maps with the thresholds θ_{fg}. The true foreground ratio (TFR) is investigated, which indicates the ratio of correctly foreground pixels in all binarized salient pixels. We take the SRM on the DUTS-TEST dataset as an example. The TFR with $\theta_{fg} = 0.9$ is 0.9003 which means that almost all confident areas fall into the ground truth area. So, we have confidence to believe that the feature vector $\mathcal{F}_{fg}$ can represent the object robustly.

Moreover, we set θ_{fg} to different values during testing and investigate the different performance. As shown in table

Table 2: Comparison of LPS and our method. The best results on each dataset are illustrated in bold respectively.

Datasets		HKU-IS		DUT-O		Pascal-S		ECSSD	
Methods		$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$
BSCA	LPS	0.7386	0.1075	0.5824	0.1650	0.6690	0.1654	0.7823	0.1043
	Ours	**0.8483**	**0.0897**	**0.6587**	**0.1607**	**0.7572**	**0.1488**	**0.8559**	**0.0929**
Amulet	LPS	0.8772	0.0446	0.6472	0.0975	0.7985	0.0920	0.8963	0.0509
	Ours	**0.8892**	**0.0401**	**0.6965**	**0.0951**	**0.8031**	**0.0885**	**0.9069**	**0.0464**
UCF	LPS	0.8530	0.0546	0.6423	0.1328	0.7703	0.1044	0.8805	0.0560
	Ours	**0.8862**	**0.0443**	**0.6893**	**0.1120**	**0.7869**	**0.0966**	**0.8921**	**0.0530**
ELD	LPS	0.8443	0.0511	0.6614	0.0885	0.7694	0.1022	0.8689	0.0577
	Ours	**0.8977**	**0.0404**	**0.7364**	**0.0797**	**0.7987**	**0.0894**	**0.8912**	**0.0497**
SRM	LPS	0.9042	0.0388	0.6938	0.068	0.8240	0.0810	0.9151	0.0465
	Ours	**0.9106**	**0.0366**	**0.7459**	**0.0655**	**0.8286**	**0.0757**	**0.9193**	**0.0451**
PiCANet	LPS	0.8667	0.0395	0.6825	0.0746	0.8232	0.0802	0.8569	0.0466
	Ours	**0.9159**	**0.0344**	**0.7923**	**0.0627**	**0.8349**	**0.0719**	**0.9164**	**0.0391**

Table 3: Comparison of DenseCRFs(C) and our method(O) over Pascal-S and ECSSD.

Methods		UCF		ELD		NLDF	
Datasets		$F_\beta \uparrow$	MAE$\downarrow$	$F_\beta \uparrow$	MAE$\downarrow$	$F_\beta \uparrow$	MAE$\downarrow$
ECSSD	C	0.847	0.067	0.841	0.071	0.875	0.065
	O	**0.878**	**0.055**	**0.866**	**0.057**	**0.885**	**0.060**
HKU-IS	C	0.842	0.056	0.832	0.061	0.882	0.048
	O	**0.864**	**0.047**	**0.866**	**0.047**	**0.886**	**0.045**

Table 4: Quantitative effect evaluated of different θ_{fg} on ECSSD.

θ_{fg}	$F_\beta \uparrow$	MAE $\downarrow$
0.9	0.9158	0.0460
0.8	0.9154	0.0458
0.7	0.9152	0.0458

Table 5: Quantitative effect evaluated of different θ_{afg} and θ_{abg} on ECSSD.

θ_{afg}	θ_{abg}	$F_\beta \uparrow$	MAE $\downarrow$
0.7	0.4	0.9145	0.0457
0.8	0.3	0.9154	0.0458
0.9	0.2	0.9149	0.0471

4, our method performs steadily with various θ_{fg}. By the way, we don't use the approximate masks during the inference phase. Therefore, θ_{a-fg} and θ_{a-bg} have no effect on the refined results generation. We adopt different θ_{a-fg} and θ_{a-bg} during training and analyze the performance. The results are shown in table 5. There is no big fluctuation of the F_β and the MAE.

6 Conclusion

In this paper, we proposed an efficient method to promote saliency detectors. We build the ubiquitous relations in the deep features to break the feature spatial independence. To the best of our knowledge, it is the first time of employing the relation of the deep features to promote saliency detectors without any ground truth. We proved that the mutual information can be used as the measure to estimate the relation and apply it into our method. Our experiments demonstrated that existing saliency detectors are boosted on four benchmark datasets by our method, which means that the deep relation is significantly profitable for saliency detection.

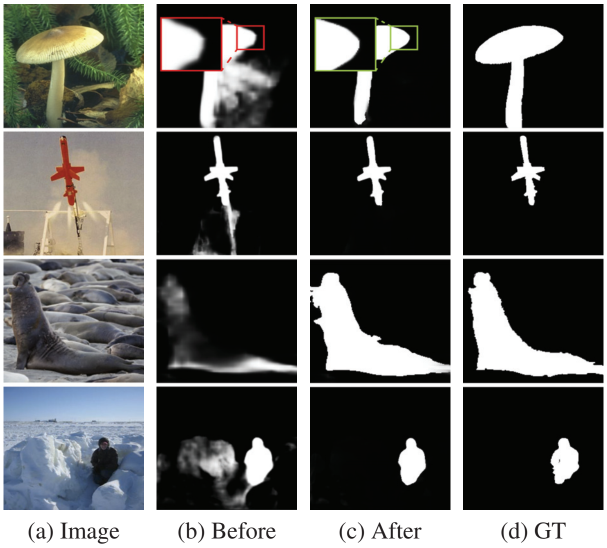
(a) Image (b) Before (c) After (d) GT

Figure 3: We present the result of some state-of-the-art detectors. GT means the ground truth. Our method can make the edge more clear and eliminate fake saliency areas.

7 Acknowledgments

We would like to sincerely thank Dr. Yang Hua and the anonymous reviewers for their helpful comments. This work was partially supported by the National Natural Science Foundation of China (No. 61971388, U1706218, 41576011, 41741007), and Key Research and Development Program of Shandong Province (No. GG201703140154).

References

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1597–1604.

Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. MINE: Mutual Information Neural Estimation. In *Proceedings of the 35th International Conference on International Conference on Machine Learning*.

Bell, A. J., and Sejnowski, T. J. 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* 7(6):1129–1159.

Chen, S.; Tan, X.; Wang, B.; and Hu, X. 2018. Reverse Attention for Salient Object Detection. In *European Conference on Computer Vision*, 236–252.

Cheng, M. M.; Mitra, N. J.; Huang, X.; Torr, P. H. S.; and Hu, S.-M. 2014. Global Contrast Based Salient Region Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3):569–582.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Deng, Z.; Hu, X.; Zhu, L.; Xu, X.; Qin, J.; Han, G.; and Heng, P.-A. 2018. R^{3}Net: Recurrent Residual Refinement Network for Saliency Detection. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 684–690.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hou, Q.; Cheng, M.-M.; Hu, X.; Borji, A.; Tu, Z.; and Torr, P. H. S. 2017. Deeply Supervised Salient Object Detection with Short Connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(4):815–828.

Hu, X.; Zhu, L.; Qin, J.; Fu, C.-W.; and Heng, P.-A. 2018. Recurrently aggregating deep features for salient object detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Krähenbühl, P., and Koltun, V. 2013. Parameter Learning and Convergent Inference for Dense Random Fields. In *Proceedings of the 30th International Conference on Machine Learning*.

Kullback, S. 1962. *Information Theory and Statistics*. Courier Corporation.

Lee, G.; Tai, Y.-W.; and Kim, J. 2016. Deep Saliency with Encoded Low Level Distance Map and High Level Features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 660–668.

Li, G., and Yu, Y. 2016. Visual Saliency Detection Based on Multiscale Deep CNN Features. *IEEE Transactions on Image Processing* 25(11):5012–5024.

Li, Y.; Hou, X.; Koch, C.; Rehg, J. M.; and Yuille, A. L. 2014. The Secrets of Salient Object Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 280–287.

Linsker, R. 1988. Self-organization in a perceptual network. *Computer* 21(3):105–117.

Liu, N.; Han, J.; Zhang, D.; Wen, S.; and Liu, T. 2015. Predicting eye fixations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 362–370.

Liu, N.; Han, J.; and Yang, M.-H. 2018. PiCANet: Learning Pixel-Wise Contextual Attention for Saliency Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3089–3098.

Luo, Z.; Mishra, A.; Achkar, A.; Eichel, J.; Li, S.; and Jodoin, P.-M. 2017. Non-Local Deep Features for Salient Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 271–279. USA: Curran Associates Inc.

Qin, Y.; Lu, H.; Xu, Y.; and Wang, H. 2015. Saliency detection via Cellular Automata. In *IEEE Conference on Computer Vision and Pattern Recognition*, 110–119.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Shelhamer, E.; Long, J.; and Darrell, T. 2016. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(4):640–651.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Wang, L.; Lu, H.; Wang, Y.; Feng, M.; Wang, D.; Yin, B.; and Ruan, X. 2017a. Learning to Detect Salient Objects with Image-Level Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3796–3805.

Wang, T.; Borji, A.; Zhang, L.; Zhang, P.; and Lu, H. 2017b. A Stagewise Refinement Model for Detecting Salient Objects in Images. In *IEEE International Conference on Computer Vision*, 4039–4048.

Yan, Q.; Xu, L.; Shi, J.; and Jia, J. 2013. Hierarchical Saliency Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1155–1162.

Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2013. Saliency Detection via Graph-Based Manifold Ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3166–3173.

Zeng, Y.; Lu, H.; Zhang, L.; Feng, M.; and Borji, A. 2018. Learning to Promote Saliency Detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1644–1653.

Zhang, J.; Sclaroff, S.; Lin, Z.; Shen, X.; Price, B.; and Mech, R. 2015. Minimum Barrier Salient Object Detection at 80 FPS. In *IEEE International Conference on Computer Vision*, 1404–1412.

Zhang, P.; Wang, D.; Lu, H.; Wang, H.; and Ruan, X. 2017a. Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection. In *IEEE International Conference on Computer Vision*, 202–211.

Zhang, P.; Wang, D.; Lu, H.; Wang, H.; and Yin, B. 2017b. Learning Uncertain Convolutional Features for Accurate Saliency Detection. In *IEEE International Conference on Computer Vision*, 212–221.

Zhao, T., and Wu, X. 2019. Pyramid Feature Attention Network for Saliency detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhu, W.; Liang, S.; Wei, Y.; and Sun, J. 2014. Saliency Optimization from Robust Background Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2814–2821.

Zhuge, Y.; Zeng, Y.; and Lu, H. 2019. Deep embedding features for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9340–9347.