A Stagewise Refinement Model for Detecting Salient Objects in Images

Tiantian Wang[1], Ali Borji[2], Lihe Zhang[1], Pingping Zhang[1], Huchuan Lu[1]
[1] Dalian University of Technology, China
[2] University of Central Florida, USA
Tiantianwang.ice@gmail.com, aliborji@gmail.com,
jssxzhpp@mail.dlut.edu.cn, zhanglihe@dlut.edu.cn, lhchuan@dlut.edu.cn

Abstract

Deep convolutional neural networks (CNNs) have been successfully applied to a wide variety of problems in computer vision, including salient object detection. To detect and segment salient objects accurately, it is necessary to extract and combine high-level semantic features with low-level fine details simultaneously. This happens to be a challenge for CNNs as repeated subsampling operations such as pooling and convolution lead to a significant decrease in the initial image resolution, which results in loss of spatial details and finer structures. To remedy this problem, here we propose to augment feedforward neural networks with a novel pyramid pooling module and a multi-stage refinement mechanism for saliency detection. First, our deep feedward net is used to generate a coarse prediction map with much detailed structures lost. Then, refinement nets are integrated with local context information to refine the preceding saliency maps generated in the master branch in a stagewise manner. Further, a pyramid pooling module is applied for different-region-based global context aggregation. Empirical evaluations over six benchmark datasets show that our proposed method compares favorably against the state-of-the-art approaches.

1. Introduction

Saliency detection, a fundamental topic in computer vision, aims to identify and segment objects that attract human attention in images. Early saliency models, inspired by the human visual attention mechanisms, attempted to predict spatial locations where an observer may fixate during free-viewing of natural scenes (e.g., [21, 20, 39, 5, 6], which are vital in understanding scenes (e.g., describing a scenes, navigation, etc). A strand of saliency research has focused on object-level segmentation since the pioneering works of Liu et al., [34] and Achanta et al., [1]. A large number of follow up works have been reported based on different ideas (e.g., [14, 9, 40, 24, 57, 28, 7]. Salient object detection models have gained broad interest recently due to their applications in several fields such as tracking [19], image understanding [54, 45], person re-identification [4], image captioning [51, 13, 12], visual question answering [32], and object proposal generation [2].

Saliency models can be roughly divided into two categories: unsupervised stimuli-driven and learning-based task-driven approaches. Unsupervised methods mainly exploit low-level visual features and cues, such as color, motion and center-surround contrast to construct a saliency map. However, purely utilizing low-level cues can hardly capture high-level semantic knowledge between the objects and their context. In contrast, learning-based approaches incorporate high-level information to better distinguish salient objects from the background clutter. This, however, often requires supervised learning with manually labeled ground truth maps.

Recently, deep learning based approaches, in particular the convolutional neural networks (CNNs), have delivered remarkable performance in many recognition tasks. However, these approaches pose clear limitations when dealing with dense prediction tasks such as semantic segmentation [8, 36], scene parsing [35] and saliency detection. Multiple stages of spatial pooling and convolutional layers progressively downsample the initial image which results in losing much of the fine image structure. This general architecture results in low-resolution feature maps that are invariant to pixel-level variations thus useful for extracting object-level information. This is beneficial for the classification task which does not need spatial information, but presents challenges for densely segmenting salient objects.

To resolve the above-mentioned limitation, in this paper, we propose a novel stage-wise model based on spatial pyramid pooling module [55] which aggregates multiscale global context priors. The stage-wise refinement network efficiently merges high-level semantic knowledge encoded in the master network layers with the spatially rich information of low-level features encoded in the refinement net. The master net helps locate salient objects, while the refinement

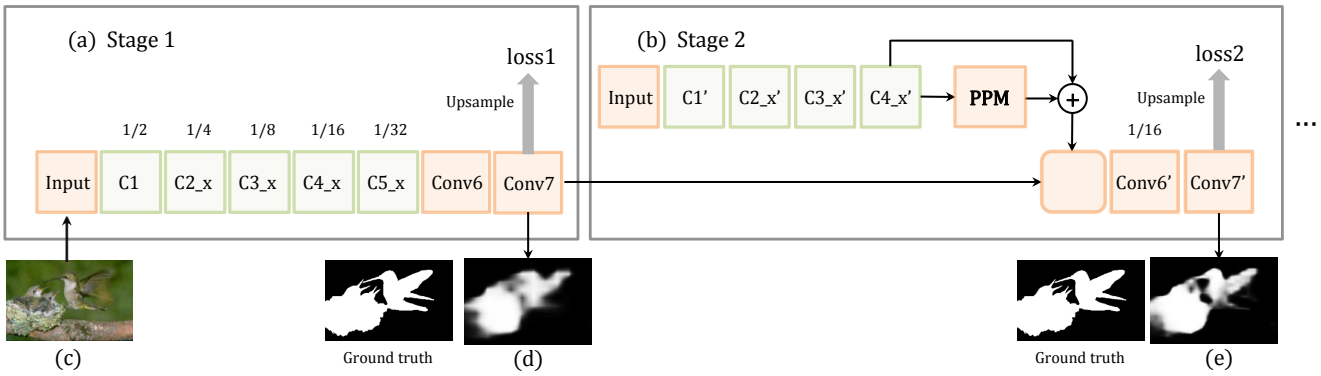


Figure 1. The pipeline of our proposed saliency detection algorithm. Each green box is considered as a residue block. Given an input image (c), an intermediate saliency map (d) is generated in stage 1 with the size 1/32 of the original resolution. Here, we upsample the map to the original size. More refinement nets in stage t ($t \geq 2$) are gradually connected to refine the saliency map generated at the preceding stage. It is noted that a pyramid pooling module (PPM) is attached to C4_x′ of stage 2 for different sub-region representation.

net with a spatial pyramid module helps gradually generate finer details and embed global context information.

Fig. 1 illustrates the pipeline of our model. In the first stage, our approach generates a mask encoding in a feedforward manner, which is simply a semantically meaningful saliency map (Fig. 1.d). Then, in the next stages, refinement nets are utilized to successively refine the preceding saliency maps stage by stage (Fig. 1.e). As will be shown in the Experiments section, the proposed method performs better compared to the other state of the art deep learning based approaches.

In summary, we offer the following contributions in this work:

- We propose a novel stage-wise refinement network where the refinement nets help renovate sharp and detailed boundaries in coarse saliency maps for high-resolution salient object segmentation.
- A pyramid pooling module is adopted to exploit global context information, where different spatial statistics provide varying-scale feature representations.
- Compared with previous works based on CNNs, the proposed method demonstrates consistent performance improvements on ECSSD, THUR15K, DUT-OMRON, HKU-IS and DUTS test benchmark datasets.

2. Related Work

2.1. Saliency Detection

Early saliency models mainly concentrated on the low-level features dating back to the feature integration theory [44]. These models integrate different kinds of visual features for modeling focused attention of humans. The most widely used feature is contrast, which is based on the fundamental hypothesis that salient objects are usually in high contrast with the background. Itti *et al.* [21] propose to measure center-surround contrast using color, intensity and orientation features over different scales. Cheng *et al.* [10] propose a region contrast based algorithm that simultaneously considers spatial coherence across nearby regions and the global contrast over the entire image.

In addition, background prior has been utilized to compute saliency based on the observation that image boundary regions usually tend to belong to the background. In [30], dense and sparse reconstruction errors based on background prior are utilized for saliency detection. Wei *et al.* [50] focus on the saliency detection from two background priors including boundary and connectivity. In [16, 15, 53, 23, 27, 38], saliency is measured by label propagation where initial labeling is propagated from the labeled elements to the unlabeled ones based on their pairwise affinities.

The low-level saliency cues are often effective in simple scenarios but they are not always robust in some challenging cases. Therefore, it is necessary to consider high-level image information and context for saliency prediction.

2.2. Deep Networks for Saliency Detection

Recently, deep convolutional neural networks (CNNs) have achieved near human-level performance in some computer vision tasks [18]. Instead of constructing hand-craft features, deep networks extract high-level semantic features in various scales.

CNNs have also achieved state-of-the-art performance when applied to saliency detection. For instance, in [46], Wang *et al.* train a DNN-L and a DNN-G network by using local patch features and global candidate features to measure saliency. In [28], multiscale features are extracted first and then a fully connected regressor network is trained to infer the saliency score of each image segment. In [56],

superpixels in local and global contexts are considered for saliency detection in a unified deep learning framework. These methods measure saliency at the patch level where CNNs are run thousands of times to obtain the saliency score of every patch, which is computationally very expensive. Further, all pixels of the same patch share the same saliency score, which is not always the case. To address the above-mentioned issues, fully convolutional networks (FCN) haven been trained end-to-end for densely segmenting salient objects. In [29], Li *et al.* integrate multiple saliency maps across multiscale convolutional layers. Kuen *et al.* [25] utilize a convolutional-deconvolutional network to generate a coarse map and then refine it with a recurrent attentional network. The goal is iteratively refine saliency predictions on arbitrary-size image sub-regions. In [48], Wang *et al.* infuse prior knowledge into a recurrent fully convolutional network for accurate saliency inference.

Perhaps, the most similar work to ours is by Liu *et al.* [33]. In spite of having a similar spirit, our proposed approach is significantly different in three aspects. Firstly, a fully connected layer after the conv5_3 layer in [33] results in losing spatial information of salient objects. In our work, we utilize fully convolutional networks to overcome this problem. Secondly, a stage-wise hierarchical refinement network is utilized to progressively refine the intermediate saliency maps where multiscale nets are optimized to obtain their individual best result. Liu *et al.* refine the preceding maps in one network with low-level features which makes it difficult to learn optimal multiscale information in one network. Besides, we also utilize a pyramid pooling module to gather global context information. Thirdly, each intermediate saliency mask will be upsampled to the size of groundtruth map for computing losses, but in [33] the groundtruth mask is downsampled to meet the needs, which causes spatial information loss.

3. The Proposed Method

We propose a new framework that provides a stage-wise refinement mechanism over which finer structures are gradually renovated by multiple refinement nets. Our framework is trained end-to-end. Figure 1 shows the simplified illustration of the proposed approach.

We begin by describing the generation of the initial coarse saliency map in Section 3.1, followed by a detailed description of our multi-stage refinement strategies equipped with pyramid pooling module in Section 3.2.

3.1. Feedforward Network for Coarse Prediction

Standard feedforward CNNs [42, 18] used for image classification employ a cascade of convolutional and pooling layers followed by fully connected layers. They take an image of fixed spatial size as input and produce a probability vector indicating the category label of the input image. Convolutional and pooling layers control the model capability and increase receptive field size, thereby resulting in a coarse, highly-semantic feature representation. Both the input and the output of convolutional layers are three-dimensional feature maps (a.k.a tensors), where output feature map is obtained by sliding different convolutional kernels on the input feature map as

$$\mathcal{F}_s(X, \{W, b\}) = W *_s X + b, \tag{1}$$

where X is the input map. W and b denote kernel and bias parameters, respectively. $*_s$ represents convolutional operation with stride s. As a result, the resolution of the output feature map $\mathcal{F}_s(X; \{W, b\})$ is downsampled by factor s.

We choose the recently proposed Residual Net (ResNet-50) [18] as our baseline network due to its superior performance in classification and modify it to meet our requirements. Compared to VGG16 [42], the training process based on ResNet-50 can converge faster thanks to skip connections and batch normalization layers. In the subsequent stages described in Section 3.2, we also adopt ResNet-50 as our fundamental building block for saliency detection. The baseline network takes an entire image as input, and outputs a saliency map of equal resolution. ResNet-50 consists of 49 convolutional layers with five convolutional blocks, followed by an average pooling layer and one fully connected layer. To adopt it for our dense image prediction task, we utilize the first five convolutional blocks (the fifth block denoted as C5_x; same convention is followed to denote other blocks) and the final feature maps which have 1/32 of the input image resolution. Then, one 3×3 convolutional layer with 256 channels (Conv6) and one 3×3 convolutional layer (Conv7) with 2 channels (one foreground mask plus one for background) are added to compute saliency confidence for every pixel. Finally, to generate a pixelwise prediction map with the same size as the input image, we directly upsample the low-resolution feature map via bilinear interpolation,

$$P = \mathcal{B}(\mathcal{F}_s(I; W); \theta), \tag{2}$$

where I is the input image and P is the output prediction of the whole network. $\mathcal{F}_s(\cdot)$ denotes the output feature map generated by the convolutional layers with a total stride of s. Bias term b is omitted here. $\mathcal{B}(\cdot)$ denotes the interpolation layer with the parameter θ.

As shown in Figure 1, the feedforward network can roughly localize the birds and the nest but the result has low resolution. However, it has difficulty in generating pixel-accurate segmentations for some image regions such as bird beaks or wings.

3.2. Refinement Networks for Finer Prediction

To mitigate the above limitations, we introduce a multi-stage refinement process that attempts to recover lost local

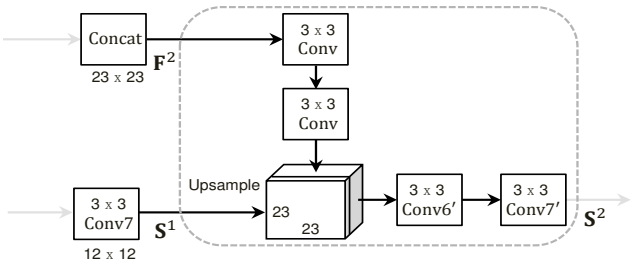


Figure 2. The detailed structure of the refinement model $\mathbf{R}^1$ (gray dotted box).

context information by combining preceding saliency maps with the features fed in the current stage. More specifically, pyramid pooling layers are plugged in to incorporate global context information.

3.2.1 Stage-wise Refinement

The first stage saliency map $\mathbf{S}^1$ generated by the feedforward network is coarse compared to the original resolution ground truth. Thus, in the second stage, we adopt a refinement net used for the subsequent refinement as shown in Figure 1b.

We use a network structure composed of the first four convolutional blocks of ResNet-50 (denoted as C1′, C2_x′, C3_ x′ and C4_x′) with different parameters than those used in stage 1. This allows a more flexible account for different structures and helps learn stage-specific refinements. $\mathbf{S}^1$ serves as the input to the subsequent incorporation module $\mathbf{R}^1$ and is refined to progressively increase the resolution. Similarly, in a subsequent stage t ($t \in \{2, ..., T\}$), each incorporation module $\mathbf{R}^{t-1}$ aggregates information from the preceding coarse map encoding $\mathbf{S}^{t-1}$ and outputs feature $\mathbf{F}^t$ of the refinement net in stage t. Each module $\mathbf{R}^{t-1}$ takes as input a mask encoding $\mathbf{S}^{t-1}$ generated in the master pass, along with matching features $\mathbf{F}^t$ generated in the refinement pass. It learns to merge information in order to generate a new prediction encoding $\mathbf{S}^t$,

$$\mathbf{S}^t = \mathbf{R}^{t-1}(\mathbf{S}^{t-1}, \mathbf{F}^t), \tag{3}$$

where $\mathbf{S}^{t-1}$ and $\mathbf{S}^t$ denote the t-th stage input and output, respectively.

Structure Details. Figure 2 shows a detailed illustration of the first refinement module $\mathbf{R}^1$ adopted in stage 2 (i.e., concatenating a coarse saliency map $\mathbf{S}^1$ from the master pass with a feature map $\mathbf{F}^2$ from a refinement pass) to generate a finer saliency map $\mathbf{S}^2$. Since $\mathbf{S}^1$ (12×12 pixels) is coarser than $\mathbf{F}^2$ (23×23), we first upsample $\mathbf{S}^1$ to double its size. Then, we combine the upsampled saliency map with the feature maps $\mathbf{F}^2$ to generate $\mathbf{S}^2$. We append two extra convolutional layers behind the fourth convolutional block (C4_x′) to reduce the dimension. The first extra layer has 3×3 kernels and 256 channels while the second extra layer (output feature map) has 3×3 kernels and 64 channels.

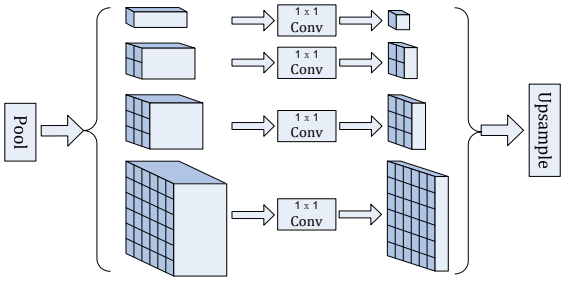


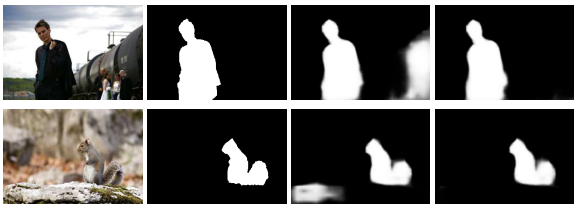
Figure 3. The structure of the pyramid pooling module. From the first row to the last one: 1×1, 2×2, 3×3 and 6×6 feature bins, which are achieved by employing variable-size pooling kernels with different strides on the output feature map (e.g. C4_x′).

3.2.2 Pyramid Pooling Module

The aforementioned stage-wise refinement mechanism can progressively encode local context information for finer saliency prediction. To further distinguish salient objects from the background, we employ a pyramid pooling module (PPM) for gathering global context information.

PPM was first adopted for deep visual recognition by [17] to get rid of the fixed-size input constraint to generate a fixed-length representation. This was accomplished by concatenating three-scale pyramid pooling features. The drawback is losing context structures which are crucial for pixel-level prediction. As a remedy, [55] utilized spatial pyramid pooling layers in which hierarchical features are generated by average pooling and aggregated for different-scale global feature representation.

In this paper, we apply PPM to every refinement stage (t >1) by attaching it to each refinement net. As shown in Figure 1, in the 2-stage refinement network, C4_x′ feature map is passed to the pyramid pooling module. Then, the pyramid module is concatenated with the output feature map of C4_x′. We show the visual comparison in Figure 4 of 3-stage refinement network with or without PPM. It can be seen that the saliency maps generated from the proposed method with PPM can preserve salient object boundaries and suppress background noise.



(a) Image (b) GT (c) without PPM (d) with PPM

Figure 4. Comparison of 3-stage prediction with and without PPM.

Architecture Details. The details of the pyramid pool-

ing module is illustrated in Figure 3. PPM is composed of four-scale feature bins, including 1×1, 2×2, 3×3 and 6×6, respectively. The variable-size feature maps contain global information at different scales. The 1×1 bin is the coarsest representation, which is generated by global pooling. The larger bins with pooled representations over different sub-locations contain richer global context information. Also, a 1×1 convolutional layer with 512 channels is connected to every pyramid level to reduce the dimension of the corresponding context representation for maintaining the weight of global features. Then an upsample layer is constructed aiming to obtain the same size feature map as the output of C4_x′ via bilinear interpolation instead of deconvolution for the limitation of memory. Finally, the four-scale features are concatenated as the final pyramid pooling aggregation features.

3.2.3 Training and Inference.

We use standard stochastic gradient descent algorithm (SGD) to train all T stages of the network end to end. To share features across all subsequent stages, we share the weights of corresponding convolutional layers across stages $t \geq 2$. Each stage of the framework is trained to repeatedly produce a saliency map based on the preceding one with more finer details recovered and added. We encourage the network to repeatedly arrive at such a representation by applying an auxiliary loss function at the output of each stage t ($t < T$). Both master and auxiliary losses help optimize the learning process together. Specifically, we first upsample the map generated at every stage to the size of ground truth saliency mask (achieved in the master branch behind every stage). Then, the pixel-wise cross entropy loss between $\mathbf{S}^t$ and the ground truth saliency mask G is computed as:

$$L(\Psi) = -\sum_{i,j} \sum_{lg \in \{0,1\}} \mathbf{1}(\mathbf{S}^t_{i,j} = lg) \log \Pr(l_{i,j} = lg|\Psi) \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function. The notation $l_g \in \{0, 1\}$ indicates the foreground or background label of the pixel at location (i, j) and $\Pr(l_{i,j} = lg|\Psi)$ represents its corresponding probability of being salient or not. Ψ denotes the parameters of all network layers.

Our final loss function combining master and auxiliary losses can be written as:

$$L_{final}(\Psi) = L_{mas}(\Psi) + \sum_{t=1}^{T-1} \lambda_t L_{aux}(\Psi), \tag{5}$$

where we set $\lambda_t = 1$ to balance all the losses.

The auxiliary loss branches are only used during the training process. They are abandoned in final pixel-wise prediction. We just feed the fixed-size input image to the network to generate a final saliency map without using any pre- or post-processing.

Implementation Details. The proposed refinement network is based on the public platform Caffe [22]. We use the 'fixed' learning rate policy and set the base learning rate to 10^{-10} with a decay of 0.005. Our model is initialized by the ResNet-50 weights [41] and finetuned on the DUT-S [47] training dataset. We test our model on the DUTS test dataset and other five datasets. All input images are resized to 353×353 pixels for training and testing. The source code will be released[1].

4. Experiments and Results

4.1. Experimental Setup

Datasets. To evaluate the effectiveness of the proposed method, we carry out comprehensive experiments on 5 popular benchmark datasets: ECSSD [52], THUR15K [11], DUT-OMRON [53], SED [3], HKU-IS [28], and DUT-S [47]. ECSSD dataset contains 1,000 complex images with objects of different sizes. THUR15K has 6,232 images from five categories including 'butterfly', 'coffee mug', 'dog jump', 'giraffe' and 'plane'. DUT-OMRON is a challenging dataset with high content variety. It includes 5,168 images with relatively complex backgrounds. SED contains two subsets: SED1 has 100 images each with only one salient object and SED2 has 100 images each with two salient objects. HKU-IS has 4,447 images which contain multiple salient objects with low color contrast or overlapping with the image boundary. DUTS is currently the largest saliency detection benchmark containing 10,553 training images and 5,019 test images. All six datasets are human-labeled with pixel-wise ground-truth for quantitative evaluations.

Evaluation Metrics. We first adopt the Precision-Recall (PR) curve to evaluate the performance of our method as it is the most widely-used evaluation metric in the saliency detection literature. All saliency maps are binarized at every integer threshold in the range of [0, 255]. Compared with the binary ground-truth mask, pairs of precision and recall values are computed to plot the PR curve.

In addition, we also compute the average precision, recall, and F-measure values, where every saliency map is binarized with an adaptive threshold proposed by [1]. The threshold is determined to be twice the mean saliency value of the saliency map. The F-measure is an overall performance measurement calculated as,

$$F_\beta = \frac{(1 + \beta^2) \cdot Precssion \cdot Recall}{\beta^2 \cdot Precison + Recall}. \tag{6}$$

Here, as in [1], we set β^2 to 0.3 to emphasize the precision over recall.

[1] http://ice.dlut.edu.cn/lu/publications.html

Table 1. Quantitative comparison of F-measure and MAE scores. The best two scores are shown in red and blue colors, respectively.

*	ECSSD [52]		DUT-OMRON [53]		THUR15K [11]		SED [3]		HKU-IS [28]		DUTS [47]	
	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
Ours	0.892	0.056	0.707	0.069	0.708	0.077	0.851	0.083	0.874	0.046	0.757	0.059
RFCN [48]	0.834	0.109	0.627	0.111	0.695	0.100	0.808	0.115	0.835	0.089	0.712	0.090
KSR [49]	0.782	0.135	0.591	0.131	0.604	0.123	0.745	0.144	0.747	0.120	0.602	0.121
ELD [26]	0.810	0.082	0.611	0.092	0.634	0.098	0.815	0.085	0.769	0.074	0.628	0.093
DS [31]	0.821	0.124	0.603	0.120	0.626	0.116	0.799	0.108	0.785	0.078	0.632	0.091
DHS [33]	0.871	0.063	-	-	0.673	0.082	0.855	0.068	0.852	0.054	0.724	0.067
MCDL [56]	0.796	0.102	0.625	0.089	0.620	0.103	0.817	0.097	0.757	0.092	0.594	0.105
MDF [28]	0.805	0.108	0.644	0.092	0.636	0.109	0.821	0.100	-	-	0.673	0.100
LEGS [46]	0.785	0.119	0.592	0.133	0.607	0.125	0.795	0.113	0.732	0.119	0.585	0.138
DCL [29]	0.827	0.151	0.684	0.157	0.676	0.161	0.825	0.154	0.853	0.136	0.714	0.149
BL [43]	0.684	0.217	0.499	0.239	0.532	0.219	0.746	0.185	0.660	0.207	0.490	0.238
DRFI [24]	0.733	0.166	0.550	0.138	0.576	0.150	0.770	0.142	0.722	0.145	0.541	0.175

Table 2. Ablation analysis using F-measure and MAE scores. 'ppm' stands for pyramid pooling module. 't'-stage denotes that there are t refinement stages in total.

*	ECSSD		DUT-OMRON		THUR15K		SED		HKU-IS		DUTS	
	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
1-stage (baseline)	0.843	0.073	0.651	0.079	0.657	0.089	0.799	0.097	0.816	0.065	0.682	0.072
2-stage	0.867	0.059	0.664	0.082	0.669	0.092	0.847	0.076	0.842	0.051	0.706	0.068
2-stage+ppm	0.878	0.060	0.688	0.073	0.691	0.080	0.845	0.083	0.856	0.049	0.734	0.063
3-stage	0.879	0.055	0.687	0.071	0.689	0.082	0.842	0.079	0.858	0.045	0.735	0.059
3-stage+ppm	0.892	0.056	0.707	0.069	0.708	0.077	0.851	0.083	0.874	0.046	0.757	0.059
4-stage	0.885	0.054	0.690	0.074	0.699	0.078	0.855	0.076	0.869	0.045	0.740	0.061
4-stage+ppm	0.882	0.056	0.693	0.070	0.693	0.082	0.852	0.080	0.864	0.045	0.742	0.057
ResNet-50 based FCN structure												
	0.864	0.070	0.659	0.081	0.674	0.085	0.855	0.078	0.845	0.059	0.689	0.078

Complementary to the PR curve, we also report the Mean Absolute Error (MAE) which is calculated as the average pixelwise absolute difference between the binary groundtruth G and the saliency map S adopted by [37],

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|, \quad (7)$$

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

4.2. Comparison with the State of the Art

We compare our method with eleven state-of-the-art deep learning-based and classic saliency detection methods, including DRFI [24], BL [43], LEGS [46], MDF [28], MCDL [56], DS [31], DHS [33], ELD [26], DCL [29], KSR [49] and RFCN [48]. For a fair comparison, we utilize either the implementations with recommended parameter settings or the saliency maps provided by the authors[2].

Quantitative Evaluation. PR curves, F-measure curves, and F-measure scores are given in Figures 5 6. In all cases (over all datasets and evaluation metrics), our proposed method is among the top contenders.

We also compare the proposed method with the state-of-the-art methods in terms of F-measure and MAE scores as shown in Table 1. Our method outperforms all existing salient object detection algorithms across all datasets except the SED dataset. Using F-measure scores, our 3-stage pyramid refinement model improves the second best algorithm by 1.9%, 2.4%, 2.5%, 3.4% and 4.6% over THUR15K, ECSSD, HKU-IS, DUT-OMRON and DUTS datasets, respectively. Also, our model lowers the MAE scores by 6.1%, 11.1%, 11.9%, 14.8% and 22.5% on THUR15K, ECSSD, DUTS, HKU-IS and DUT-OMRON datasets, respectively. We provide more results on SED and ECSSD datasets in the supplementary material due to the limited space.

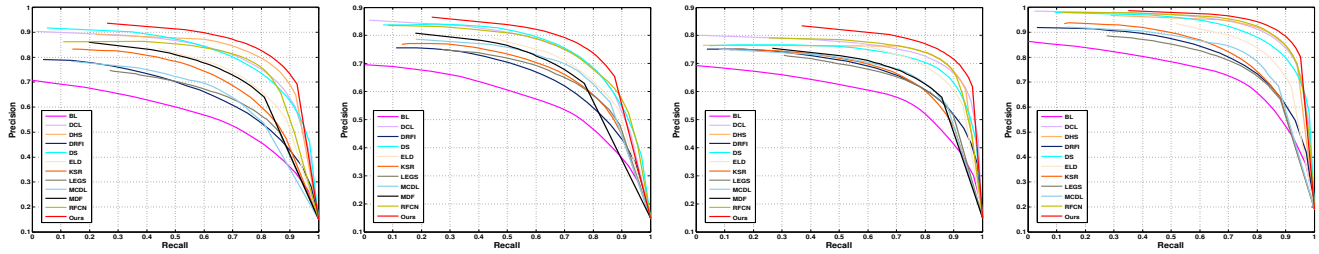
Table 3 shows a comparison of running times. This evaluation was conducted on a machine with a i7-4790 CPU and a TITAN-X GPU. As it can be seen, our method is much faster than other methods. It achieves a speed of 14 FPS.

Qualitative Evaluation. Figure 7 shows a visual comparison of results of our method with respect to others. It can be seen that our method is capable of uniformly highlighting the inner part of salient objects as well as suppressing the background clutter. Further, our saliency maps are much closer to the ground truth maps in various challenging scenarios.

4.3. Ablation Analysis

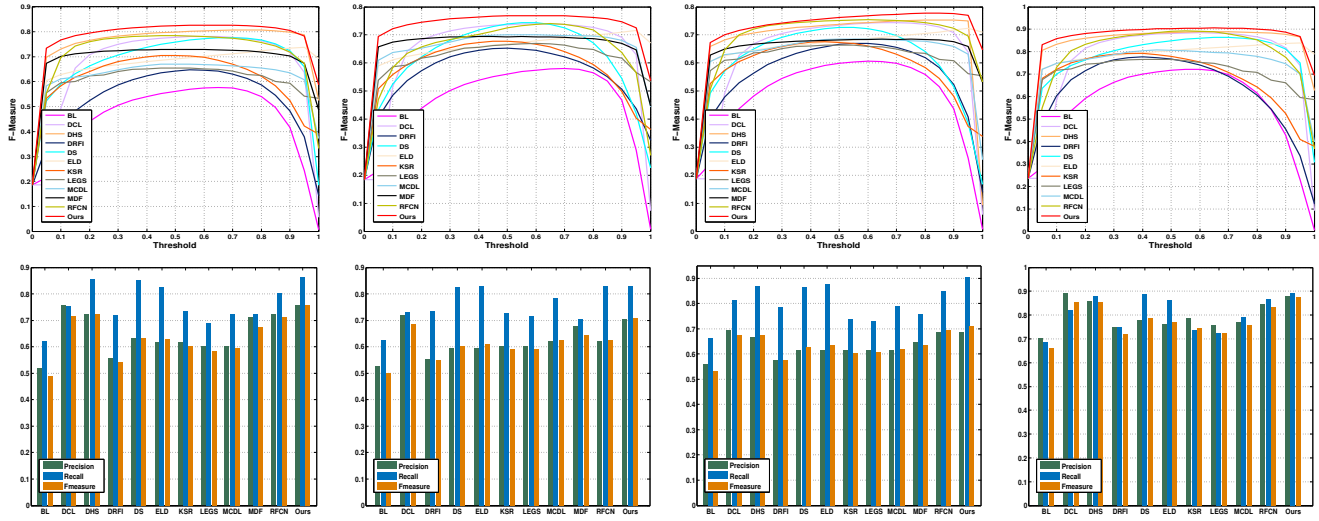
In this section, we analyze the contribution of model components in the final accuracy.

[2]The results of DHS and MDF methods on the DUT-OMRON and HKU-IS datasets are not reported, because they are trained on these datasets.



(a) DUTS dataset (b) DUT-OMRON dataset (c) THUR15K dataset (d) HKU-IS dataset

Figure 5. Comparison of precision-recall curves of 12 state-of-the-art methods over four datasets. The proposed method outperforms other methods on all datasets.

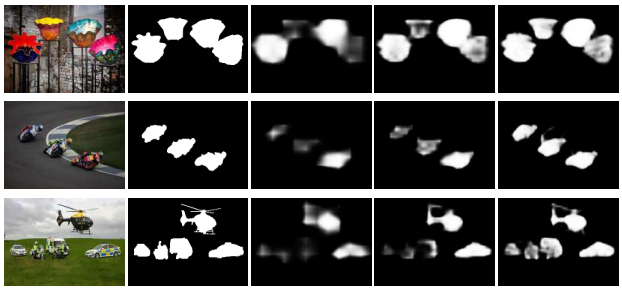


(a) DUTS dataset (b) DUT-OMRON dataset (c) THUR15K dataset (d) HKU-IS dataset

Figure 6. The first row shows the F-measure curves. The second row shows the comparison of precision, recall, and F-measure scores across four datasets. The proposed method achieves the highest F-measure at every threshold on all datasets.

Performance across pyramid pooling modules and refinement stages. As described in Section 3, a stage-wise refinement mechanism plus a pyramid pooling module are utilized to refine the coarse saliency map from the preceding stage. Here, to analyze the relative contributions of different stages and the pyramid pooling module of the proposed methods, we perform a detailed comparison of their performance using F-measure and MAE scores. Results are reported in Table 2.

We find that performance increases by adding more stages. This is because predictors in subsequent stages make use of contextual information from the current auxiliary net on the previous maps to improve detailed structures. With the pyramid pooling module connected to the subsequent stages (not included in the stage 1), the performance also increases. This is because the pyramid pooling module can aggregate global context information which is important for distinguishing salient objects from the background in a global view. We conclude that more stages and pyramid pooling module are both of vital importance for achieving good performance. Compared to the 3-stage pyramid refinement model, the 4-stage one does not improve much on F-measure and MAE scores. So We set the total number of stages of the framework to T=3.



(a) Image (b) GT (b) 1-stage (c) 2-stage (d) 3-stage

Figure 8. Illustration of stage-wise saliency map generation.

Figure 8 shows the qualitative results. We find that the stage-wise refinement scheme progressively improves details of saliency maps. It detects multiple objects, highlights salient objects uniformly, and produces sharp boundaries.

Comparison of training schemes. Here, we explore d-

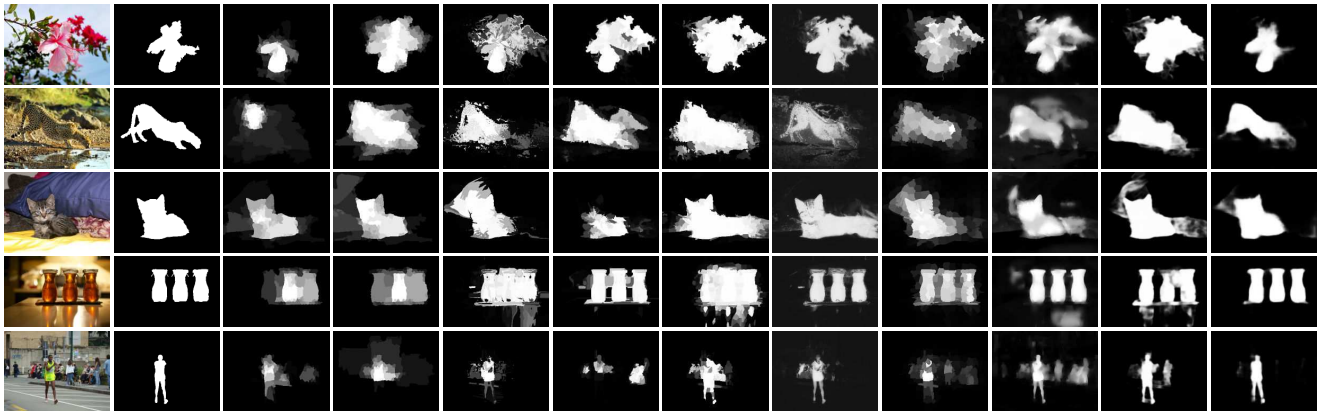


Figure 7. Visual comparison of our results compared with state-of-the-art methods.

Table 3. Run time analysis of the compared methods.

	Ours	BL	DCL	DHS	DRFI	DS	ELD	KSR	LEGS	MCDL	MDF	RFCN
Time (s)	0.07	31.73	0.41	0.04	46.21	0.12	0.57	50.90	1.54	2.27	21.55	4.72

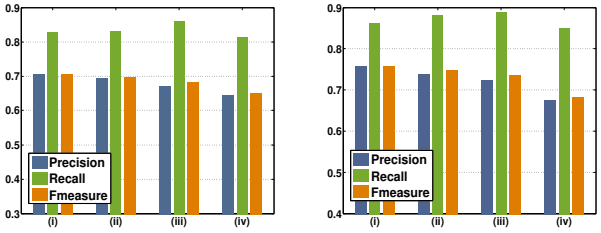


Figure 9. Performance comparison using F-measures with different variants of the proposed method on the DUT-OMRON (left) and DUTS (right) datasets.

ifferent variants of training over a 3-stage refinement network. We train the network in 3 ways: (i) training the entire framework with one master loss and two auxiliary losses, (ii) training as in (i) but without auxiliary losses, and (iii) training as in (i) but utilizing a stage-wise scheme where each stage is trained independently first, and then stacked to jointly train the whole network. In addition, the 1-stage training (iv) is also exhibited as the baseline for comparison.

Figure 9 shows the F-measure scores over the DUT-OMRON and DUTS datasets. It can be seen that the case (i) outperforms all other training cases. This shows that auxiliary losses and additional stages are indeed crucial for gaining better performance. However, the stage-wise training performs worse than direct training. This might be because the training saturates when weights are initialized by the learned parameters of each stage.

Comparison with similar FCN structures. We also compare the proposed method with the ResNet-50 based FCN [36]. To explore the fine-grained local appearance of the input image, skip connections are employed to combine output feature maps of lower convolutional layers with final convolutional layers for more accurate inference. We fine-tune this model on the same training set as in our method and test it on all six datasets. The results are shown in the last row of Table 2. It can be seen that our method outperforms the ResNet-50 based FCN structure.

5. Discussion and Conclusion

In this paper, we propose a novel stagewise end-to-end saliency detection method based on the multi-stage refinement mechanism and pyramid pooling layers. The multi-stage refinement mechanism is able to effectively combine high-level object-level semantics with low-level image features to produce high-resolution saliency maps. Pyramid pooling layers allow our network to take advantage of global contextual information.

Extensive quantitative and qualitative evaluations verify that the above contributions can significantly improve state-of-the-art saliency detection performance over five widely adopted datasets and two evaluation measures.

6. Acknowledge

L. Zhang and H. Lu were supported by the National Natural Science Foundation of China (#61371157, #61472060 and #61528101) and the Fundamental Research Funds for the Central Universities (#DUT2017TB04 and #DUT17TD03)

References

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

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE TPAMI*, 34(11):2189–2202, 2012.

[3] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, pages 1–8, 2007.

[4] S. Bi, G. Li, and Y. Yu. Person re-identification using multiple experts with random subspaces. *Journal of Image and Graphics*, 2(2), 2014.

[5] A. Borji and L. Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, pages 478–485, 2012.

[6] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE TPAMI*, 35(1):185–207, 2013.

[7] A. Borji, D. Sihite, and L. Itti. Salient object detection: A benchmark. In *ECCV*, pages 414–429, 2012.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2014.

[9] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.

[10] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[11] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

[12] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *arXiv preprint arXiv:1606.03556*, 2016.

[13] H. Fang, S. Gupta, F. Iandola, and R. K. Srivastava. From captions to visual concepts and back. In *CVPR*, pages 1473–1482, 2015.

[14] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

[15] V. Gopalakrishnan, Y. Hu, and D. Rajan. Random walks on graphs for salient object detection in images. *IEEE TIP*, 19(12):3232–3242, 2010.

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE TPAMI*, 37(9):1904–16, 2014.

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

[19] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597–606, 2015.

[20] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, pages 1–8, 2007.

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

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

[23] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang. Saliency detection via absorbing markov chain. In *ICCV*, pages 1665–1672, 2013.

[24] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2083–2090, 2013.

[25] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, pages 3668–3677, 2016.

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

[27] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. D. Feng. Robust saliency detection via regularized random walks ranking. In *CVPR*, pages 2710–2717, 2015.

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

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

[30] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013.

[31] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8):3919–3930, 2016.

[32] Y. Lin, Z. Pang, D. Wang, and Y. Zhuang. Task-driven visual saliency and attention-based visual question answering. *arXiv preprint arXiv:1606.03556*, 2017.

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

[34] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2011.

[35] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

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

[37] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[38] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 110–119, 2015.

[39] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua. An eye fixation database for saliency detection in images. In *ECCV*, pages 30–43, 2010.

[40] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860, 2012.

[41] M. Simon, E. Rodner, and J. Denzler. Imagenet pre-trained models with batch normalization. *arXiv preprint arXiv:1612.01452*, 2016.

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

[43] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In *CVPR*, pages 1884–1892, 2015.

[44] A. Triesman and G. Gelade. A feature-integration theory of attention. *Cogn. Psychol.*, 12(1):97–136, 1980.

[45] Z. Tu, Y. Wei, E. Chang, J. Wu, and J. Y. Zhu. Unsupervised object class discovery via saliency-guided multiple class learning. *IEEE TPAMI*, 37(4):862, 2014.

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

[47] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017.

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

[49] T. Wang, L. Zhang, H. Lu, C. Sun, and J. Qi. Kernelized subspace ranking for saliency detection. In *ECCV*, pages 450–466, 2016.

[50] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, pages 29–42, 2012.

[51] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *Computer Science*, pages 2048–2057, 2015.

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

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

[54] F. Zhang, B. Du, and L. Zhang. Saliency-guided unsupervised feature learning for scene classification. *Geoscience & Remote Sensing IEEE Transactions on*, pages 2175–2184, 2015.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

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

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