Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection

Pingping Zhang[†] Dong Wang[†] Huchuan Lu[†*] Hongyu Wang[†] Xiang Ruan[‡]
[†]Dalian University of Technology, China [‡]Tiwaki Co.Ltd
jssxzhpp@mail.dlut.edu.cn {wdice,lhchuan,whyu}@dlut.edu.cn ruanxiang@tiwaki.com

Abstract

Fully convolutional neural networks (FCNs) have shown outstanding performance in many dense labeling problems. One key pillar of these successes is mining relevant information from features in convolutional layers. However, how to better aggregate multi-level convolutional feature maps for salient object detection is underexplored. In this work, we present ***Amulet****, a generic aggregating multi-level convolutional feature framework for salient object detection. Our framework first integrates multi-level feature maps into multiple resolutions, which simultaneously incorporate coarse semantics and fine details. Then it adaptively learns to combine these feature maps at each resolution and predict saliency maps with the combined features. Finally, the predicted results are efficiently fused to generate the final saliency map. In addition, to achieve accurate boundary inference and semantic enhancement, edge-aware feature maps in low-level layers and the predicted results of low resolution features are recursively embedded into the learning framework. By aggregating multi-level convolutional features in this efficient and flexible manner, the proposed saliency model provides accurate salient object labeling. Comprehensive experiments demonstrate that our method performs favorably against state-of-the-art approaches in terms of near all compared evaluation metrics.*

1. Introduction

Salient object detection, which aims to identify the most conspicuous objects or regions in an image, has received considerable amount of attention in recent years. As a pre-processing step in computer vision, saliency detection has shown a great success in ranges of visual applications, *e.g.* object retargeting [7, 38, 40], scene classification [35, 33], visual tracking [4, 29], image retrieval [14, 11] and semantic segmentation [8]. Despite decades of valuable research, salient object detection still remains an unsolved research problem because there are large variety of aspects that can contribute to define visual saliency, and it's hard to combine all hand-tuned factors or cues in an appropriate way.

Inspired by human visual attention mechanisms, many early existing methods [17, 12, 10, 19, 46, 47] in salient object detection leverage low-level visual features (*e.g.* color, texture and contrast) with heuristic priors to model and approximate human saliency. These generic techniques are known to be useful for keeping fine image structures and reducing computation. Representative methods have set the benchmark on several saliency detection datasets. However, such low-level features and priors can hardly capture high-level semantic knowledge about the object and its surroundings. Thus, these low-level feature based methods are very far away from distinguishing salient objects from the clutter background and can not generate satisfied predictions.

In recent years, fully convolutional networks (FCNs), adaptively extracting high-level semantic information from raw images, have shown impressive results in many dense labeling tasks, such as image segmentation [28, 31, 6], generic object extraction [25, 13], pose estimation [48] and contour detection [45]. Motivated by these achievements, several attempts to utilize high-level features of FCNs, have been performed and delivered superior performance in predicting saliency maps [20, 21, 27, 41, 50]. Nevertheless, these state-of-the-art models mainly focus on the non-linear combination of high-level features extracted from the last convolutional layers. Due to the lack of low-level visual information such as object edge, the predicted results of these methods tend to have poorly localized object boundaries.

From above discussions, we note that 1) how to simultaneously utilize multi-level potential saliency cues, 2) how to conveniently find the optimal multi-level feature aggregation strategy, and 3) how to efficiently preserve salient objects' boundaries should become the most intrinsic problems in salient object detection. To resolve these problems, in this paper, we propose a generic aggregating multi-level convolutional feature framework, namely **Amulet**, which effectively utilizes multi-level features of FCNs for salient object detection.

[*] Prof.Lu is the corresponding author.

Our main contributions are summarized as follows:

- We propose a multi-level feature aggregation network, dubbed AmuletNet, which utilizes convolutional features from multiple levels as saliency cues for salient object detection. AmuletNet integrates multi-level features into multiple resolutions, learns to combine these features at each resolution and predicts saliency maps in a recursive manner.
- We propose a deeply recursive supervision learning framework. It effectively incorporates edge-aware feature maps in low-level layers and the predicted results from low resolution features, to achieve accurate object boundary inference and semantic enhancement. The resulting framework can be trained by end-to-end gradient learning, which uses single-resolution ground truth without additional annotations.
- The proposed model (only trained on the MSRA10K dataset [5]) achieves new state-of-the-art performance on other large-scale salient object detection datasets, including the recent DUTS [42], DUT-OMRON [47], ECSSD [46], HKU-IS [50], PASCAL-S [26], SED [2] and SOD [46]. In addition, the model is fast on modern GPUs, achieving a near real-time speed of 16 fps.

2. Related Work

In this section, we briefly review existing representative models for salient object detection. We also discuss the multi-level feature aggregation methods based on FCNs.

2.1. Salient object detection

Over the past decades, lots of salient object detection methods have been developed. The majority of salient object detection methods are based on low-level hand-crafted features, *e.g*., image contrast [10, 19], color [26, 2], texture [46, 47]. A complete survey of these methods is beyond the scope of this paper and we refer the readers to a recent survey paper [3] for details.

Recently, deep learning based approaches, in particular the convolutional neural networks (CNNs), have delivered remarkable performance in many recognition tasks. A lot of research efforts have been made to develop various deep architectures for useful features that characterize salient objects or regions. For instance, Wang *et al*. [41] first propose two deep neural networks to integrate local pixel estimation and global proposal search for salient object detection. Li *et al*. [21] predict the saliency degree of each superpixel by taking multi-scale features in multiple generic CNNs. Zhao *et al*. [50] also predict the saliency degree of each superpixel by taking global and local context into account, and detect salient objects in a multi-context deep CNN. Though these methods achieve better results than traditional counterparts, none of them handle low-level details perfectly, and all of their models include several fully connected layers, which are computationally expensive and drop spatial information of input images. To remedy above problems, Lee *et al*. [20] propose to encode low-level distance map and high-level sematic features of deep CNNs for salient object detection. Liu *et al*. [27] propose a deep hierarchical saliency network to learn enough global structures and progressively refine the details of saliency maps step by step via integrating local context information. In addition, Li *et al*. [22] design a pixel-level fully convolutional stream and a segment-level spatial pooling stream to produce pixel-level saliency predictions. Wang *et al*. [44] develop deep recurrent FCNs to incorporate the coarse predictions as saliency priors and stage-wisely refine the generated predictions. In contrary to the above methods only used specific-level features, we observe that features from all levels are potential saliency cues and helpful for salient object detection. In light of this observation, we develop a new multi-level feature aggregation approach based on deep FCNs, and show that beyond refining the predicted saliency map, the approach can also jointly learn to preserve object boundaries.

2.2. Feature aggregation in FCNs

Several works on visualizing deep CNNs [36, 49, 30, 43] indicate that convolutional features at different levels describe the object and its surroundings from different views. High-level semantic features helps the category recognition of image regions, while low-level visual features help to generate sharp, detailed boundaries for high-resolution prediction. However, how to effectively and efficiently exploit multi-level convolutional features remains an open question. To this end, several valuble attempts have been performed. The seminal FCN method [28] introduces skip-connections and adds high-level prediction layers to intermediate layers to generate pixel-wise prediction results at multiple resolutions. The Hypercolumn method [13] also integrates convolutional features from multiple middle layers and learns high-level dense classification layers. The SegNet [1] and DeconvNet [31] employ a convolutional encoder-decoder network with pooling index guided deconvolution modules to exploit the features from multi-level convolutional layers. Similarly, the U-Net [34] apply multiple skip-connections to construct a contracting path to capture context and a symmetric expanding path that enables precise localization. The HED model [45] employs deeply supervised structures, and automatically learns rich hierarchical representations that are fused to resolve the challenging ambiguity in edge and object boundary detection.

Our proposed approach clearly differs from the above-mentioned methods in three aspects. Firstly, our method aggregates multi-level features at multiple resolutions. We use a pre-trained FCN and integrate all level features into multiple resolutions at once. Our method can simultaneously incorporate coarse semantics and fine details. Although all above methods seem to be useful for aggregating multi-level features, their aggregation is carried out in a stage-wise manner rather than jointly integrating. Secondly, our method employs a bidirectional information stream, which facilitates complement effect in prediction. In contrary, all above-mentioned methods simply aggregate multiple level features from one direction, i.e., low to high or high to low. Thirdly, our method is able to refine the coarse high-level semantic predictions by exploiting low-level visual features. In particular, our method employs edge-aware feature maps of low-level layers into the prediction modules which help to preserve objects' boundaries.

3. Aggregating Convolutional Feature Model

In this section, we begin by describing the components of our proposed AmuletNet architecture in Section 3.1. Then we give the detailed formulas of our bidirectional information aggregating learning method in Section 3.2. In the end, we construct saliency inference based on the multi-level predictions of the proposed **Amulet**.

3.1. AmuletNet architecture

Our proposed AmuletNet consists of four components: multi-level feature extraction, resolution-based feature integration, recursive saliency map prediction and boundary preserved refinement. The four main components are jointly trained to optimize the output saliency detection quality. The overall architecture is illustrated in Fig. 1.

Multi-level feature extraction. The first component of our architecture is a deep feature extraction network, which takes the input image and produces feature maps for convolutional feature integration. We build our architecture on the VGG-16 model from [37], which is well known for its elegance and simplicity, and at the same time yields nearly state-of-the-art results in image classification and good generalization properties. In the VGG-16 model there are five max-pooling stages with kernel size 2 and stride 2. Given an input image with size $W \times H$, the output feature maps have size $\lfloor \frac{W}{2^5}, \frac{H}{2^5} \rfloor$, thus a FCN model built upon the VGG-16 would output feature maps reduced by a factor of 32. To balance the semantic context and fine image details, we remove the last pooling stage and enlarge the size of the input image. This way, the output feature maps of our feature extraction network are rescaled by a factor of 16 with respect to the input image. We take feature maps at five levels from the VGG-16 model: conv1-2 (which contains 64 feature maps), conv2-2 (128 feature maps), conv3-3 (256 feature maps), conv4-3 (512 feature maps) and conv5-3 (512 feature maps). Note that our feature extraction network is extremely flexible in that it can be replaced and modified in various ways, such as using different layers or networks, *e.g.* VGG-19 [37] and ResNet [16].

Resolution-based feature integration. Considering the inconsistent resolution of multi-level convolutional features, we propose a novel resolution-based feature combination structure, named RFC. The RFC structure consists of both **shrink** and **extend** branches. Assume $\mathbf{I}$ is the input image; $\tau = \lfloor \frac{W}{2^l}, \frac{H}{2^l} \rfloor$ is the target resolution of integrated feature maps, and identified by feature level $l (= 0, 1, ..., L)$; $\mathbf{F}_n(\mathbf{I})$ denotes a 3D tensor, i.e., the feature maps generated by the feature extraction network with $n \times \tau$ resolution. Thus, the proposed RFC generates the integrated feature maps by

$$\begin{aligned}\mathbf{F}^\tau = \mathbf{W}^\tau * \mathbf{Cat}(&S_n(\mathbf{F}_n(\mathbf{I}); \psi_n), ..., S_1(\mathbf{F}_1(\mathbf{I}); \psi_1), \\ &E_1(\mathbf{F}_1(\mathbf{I}); \varphi_1), ..., E_m(\mathbf{F}_m(\mathbf{I}); \varphi_m)),\end{aligned} \tag{1}$$

where $*$ represents convolution operation;$S_n(\cdot; \psi_n)$ denotes the shrink operator parameterized by ψ_n that aims to down-sample the input high-resolution feature maps by a factor of n, while the extend operator $E_m(\cdot; \varphi_m)$ aims to up-sample the low-resolution ones by a factor of m. The shrink operators can be convolution or pooling. The extend operators can be deconvolution or interpolation. **Cat** is the cross-channel concatenation. $\mathbf{W}^\tau$ is the parameter for combining the concatenated feature maps. The details of RFC are shown in Fig. 2. For our proposed AmuletNet, we take feature maps at five different levels ($L = 4$) from the above feature extraction network. We utilize RFCs to resize all level feature maps into the five spatial resolution by performing 64 convolution or deconvolution operations. The generated features are concatenated into a tensor with 320 channels at each resolution. Then we use a convolutional layer with 1×1 kernel size to weight the importance of each feature map. For computational efficiency, 64 convolutional kernels are used to combine each tensor into 64 integrated feature maps. This way, each integrated feature map will simultaneously incorporate coarse semantics and fine details.

Recursive saliency map prediction. The integrated feature maps already contains various saliency cues, so we can use them to predict the saliency map. A direct method is to deconvolute the integrated feature maps at each level into the size of the input image, and add a new convolutional layer to produce the predicted saliency map. Although this method can detect salient objects from different levels, the inner connection of different-level predictions is missing. As a result, the independent prediction is not satisfactory enough, both quantitatively and visually, and further post-processing is needed [24, 44]. To facilitate the interaction of multiple predictions, we propose a recursive prediction architecture, i.e. Deep Recursive Supervision (DRS) in Fig. 1, to hierarchically and progressively absorb high-level predictions and render pixel-wise supervised information. The proposed DRS includes saliency map prediction modules (SMP) and the deeply supervised learning mechanism [45].

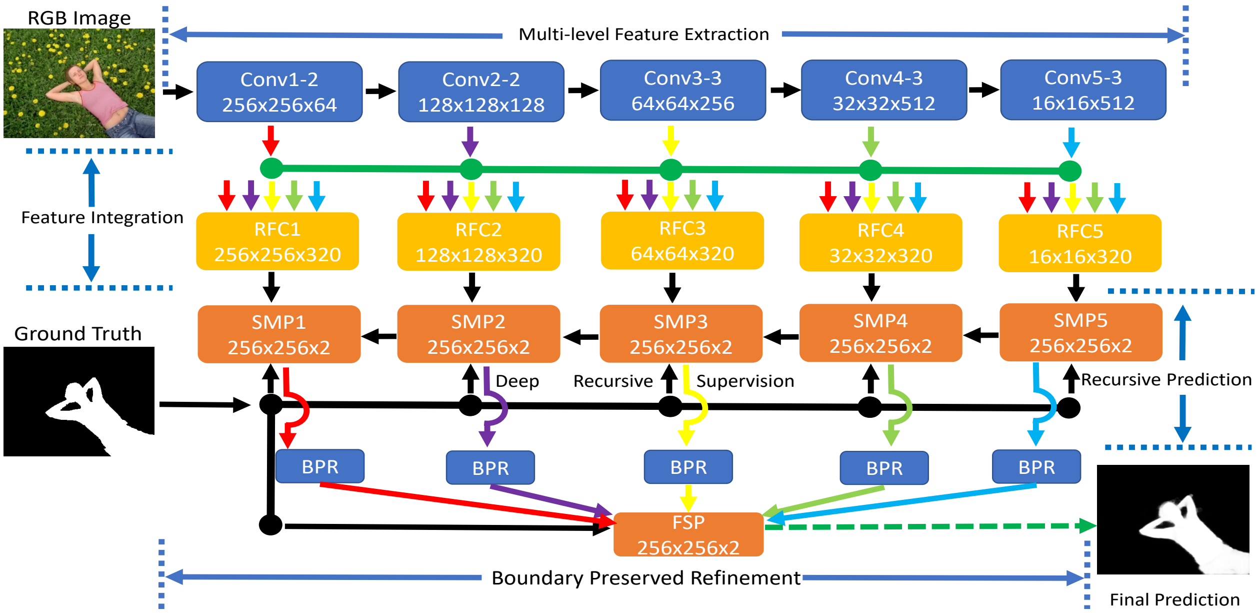


Figure 1. The overall architecture of our proposed **Amulet** model. Each colorful box is considered as a feature block. The arrows between blocks indicate the information stream. Given an input image (256×256×3), multi-level features are first generated by the feature extraction network (VGG-16 [37]). Then feature integration is performed by resolution-based feature combination modules (RFCs). After that, deep recursive supervision (DRS) is employed to improve the interaction of multiple predictions. Finally, boundary preserved refinements (BPRs) are used to refine the predicted saliency maps. The final saliency map is the fused output of multiple predicted saliency maps.

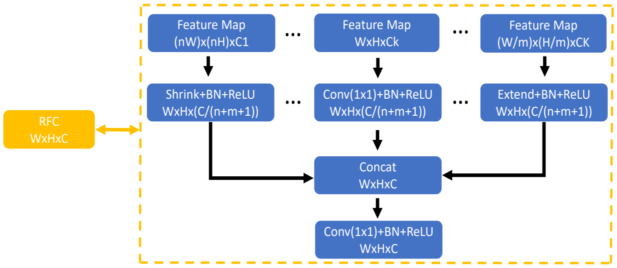


Figure 2. Details of the RFC module. The RFC first takes feature maps with different resolutions and channels as input. Then shrink and extend operators resize the feature maps to the same spatial resolution and equal channels. Finally, the concatenation and 1×1 convolution are used to generate the integrated features.

The SMP incorporates autoregressive recurrent connections into the predictions from high-level to low. In each level l, the SMP takes integrated feature maps $\mathbf{F}^{\tau}$ and the high-level prediction $\mathbf{P}^{l+1}$ as input, and produces the new prediction of this level as

$$\mathbf{P}^{l} = \begin{cases} \mathbf{W}_r * \sigma(\mathbf{W}_{F^{\tau}} \star_s \mathbf{F}^{\tau} + \mathbf{W}_{P^{l+1}} * \mathbf{P}^{l+1} + \mathbf{b}), l < L \\ \mathbf{W}_{F^{\tau}} \star_s \mathbf{F}^{\tau} + \mathbf{b}, l = L \end{cases} \tag{2}$$

where $\star_s$ represents deconvolution operation with stride s to ensure the same spatial size of the output prediction. $\mathbf{W}_{F^{\tau}}$ and $\mathbf{W}_{P^{l+1}}$ are the integrated feature weight and the output prediction weight, respectively. $\mathbf{b}$ is the bias parameter. σ is the ReLU activation function. $\mathbf{W}_r$ is the recursive weight. From Eq.(2) and Fig. 1, we can see that multiple autoregressive recurrent connections ensure that the new prediction has multiple paths from the input to the output, which facilitates effective information exchanges. Besides, we employ deeply supervised learning into the SMPs. This way, the pixel-wise supervised information from ground truth will guide the recursive saliency map prediction at each level, making the SMPs be able to propagate fine details back to the predictions of large contexts. Thus, DRS can build a bidirectional information stream aggregation, which facilitates complement effect in prediction. We will fully elaborate the bidirectional information aggregating learning in Section 3.2. The experiments in Section 4.4 show the superiority of DRS over the deeply supervised learning in [45].

Boundary preserved refinement. To further improve the detection accuracy, we add boundary refinements by introducing short connections to the predicted results. Our approach bases on the observation that low-level feature maps in the conv1-2 layer have edge-preserving properties [49, 30]. We expect that these low-level features help to predict objects' boundary. Besides, the features also have the same spatial resolution with respect to the input image. For boundary refinement, a convolutional layer with 1×1 kernel size is first applied to the conv1-2 layer, yielding boundary predictions $\mathbf{B}^{l}$. Then $\mathbf{B}^{l}$ are added to the raw prediction for better aligned object boundaries,

$$\mathbf{P}_b^{l} = \mathbf{W}_b * \sigma(\mathbf{B}^{l} + \mathbf{P}^{l}), \tag{3}$$

where $\mathbf{W}_b$ is the refinement parameter. ReLU is used so that the boundary prediction is in the range of zero to infinity.

Based on the boundary preserved refinements $\mathbf{P}_b$, a additional convolutional layer is applied and learned to produce the fusion saliency prediction (FSP) as the final output.

3.2. Bidirectional information aggregating learning

Given the salient object detection training dataset $S = \{(X_n, Y_n)\}_{n=1}^{N}$ with N training pairs, where $X_n = \{x_j^n, j = 1, ..., T\}$ and $Y_n = \{y_j^n, j = 1, ..., T\}$ are the input image and the binary ground-truth image with T pixels, respectively. $y_j^n = 1$ denotes the foreground pixel and $y_j^n = 0$ denotes the background pixel. For notional simplicity, we subsequently drop the subscript n and consider each image independently. We denote $\mathbf{W}$ as the parameters of the feature extraction network and RFCs. Supposing the network has M predictions, including one fused prediction and $M-1$ specific-level predictions. In our AmuletNet, we have $M = 6$. For the fused prediction, the loss function can be expressed as

$$\begin{aligned}\mathcal{L}_f(\mathbf{W}, w_f) = &-\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, w_f) \\ &-(1-\beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, w_f),\end{aligned} \tag{4}$$

where w_f is the classifier parameter for the fused prediction. Y_+ and Y_- denote the foreground and background label sets, respectively. The loss weight $\beta = |Y_+|/|Y|$, and $|Y_+|$ and $|Y_-|$ denote the foreground and background pixel number, respectively. $\Pr(y_j = 1|X; \mathbf{W}; w_f) \in [0, 1]$ is the confidence score of the fused prediction that measures how likely the pixel belong to the foreground.

For the prediction at level l, the loss function can be represented by

$$\begin{aligned}\mathcal{L}_l(\mathbf{W}, \theta_l, w_l) = &-\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, \theta_l, w_l) \\ &-(1-\beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, \theta_l, w_l),\end{aligned} \tag{5}$$

where $\theta_l = (w_l^r, w_l^b)$ is the parameter of the recursive prediction component and boundary refinement component in the prediction module. w_l is the classifier parameter for the prediction at level l. Thus, the joint loss function for all predictions is obtained by

$$\mathcal{L}(\mathbf{W}, \theta, w) = \alpha_f \mathcal{L}_f(\mathbf{W}, w_f) + \sum_{l=0}^{L} \alpha_l \mathcal{L}_l(\mathbf{W}, \theta_l, w_l), \tag{6}$$

where α_f and α_l are the loss weights to balance each loss term. For simplicity and fair comparison, we set $\alpha_f = \alpha_l = 1$ as used in [45]. The above loss function is continuously differentiable, so we can use the stochastic gradient descent (SGD) method to obtain the optimal parameters,

$$(\mathbf{W}^*, \theta^*, w^*) = \arg\min \mathcal{L}(\mathbf{W}, \theta, w). \tag{7}$$

Our aggregating learning method has several significant differences with other deeply supervised implementations, i.e., DHS [27] and HED [45]. In DHS and HED, the deep supervision is directly applied on side-outputs, while in our method the deep supervision is applied on multiple same resolution predictions. According to Eq.(2), each recursive prediction contains the information of two predictions at least, endowing our method the capability to propagate the supervised information across deep layers in a bidirectional manner. The bold black arrows in Fig. 1 illustrate the bidirectional information stream. Besides, DHS needs to specify scales for side-outputs to minimize the multi-scale error, which requires additional annotation for each scale. In contrast, the proposed method adaptively unify the scale information into the size of input images, without using multi-scale annotations. In addition, different from the methods used sigmoid classifiers in [27, 45], we use the following softmax classifier to evaluate the prediction scores:

$$\Pr(y_j = 1 | X; \mathbf{W}, \theta, w) = \frac{e^{z_1}}{e^{z_0} + e^{z_1}}, \tag{8}$$

$$\Pr(y_j = 0 | X; \mathbf{W}, \theta, w) = \frac{e^{z_0}}{e^{z_0} + e^{z_1}}, \tag{9}$$

where z_0 and z_1 are the score of each label of training data. In this way, each prediction of the AmultNet is composed of a foreground excitation map ($\mathbf{M}^{fe}$) and a background excitation map ($\mathbf{M}^{be}$). We utilize $\mathbf{M}^{fe}$ and $\mathbf{M}^{be}$ of all-level predictions to generate the final fusion. This strategy not only increases the pixel-level discrimination but also captures context contrast information.

3.3. Saliency inference

Although the architecture we use in this work can produce M predictions computed by Eq.(8) with the optimal parameters $(\mathbf{W}^*, \theta^*, w^*)$, we observe that the quality of the predictions at different levels varies widely. The more lower level, the better they are. The fused prediction generally appears much better than other predictions. For saliency inference, we can simply use the fused prediction as our final saliency map. However, saliency inference emphasize the contrast between foreground and background. Therefore, more biologically we utilize the mean contrast of different predictions to further improve the detection accuracy during saliency inference. Formally, let $\mathbf{M}_l^{fe}(\mathbf{M}_f^{fe})$ and $\mathbf{M}_l^{be}(\mathbf{M}_f^{be})$ denote the foreground excitation map and background excitation map at level l (of the fused prediction), respectively. They can be computed by Eq.(8) and Eq.(9). Thus, the final saliency map can be obtained by

$$\mathbf{S} = \sigma(\mathbf{Mean}(\sum_{l=0}^{L}(\mathbf{M}_l^{fe} - \mathbf{M}_l^{be})) + (\mathbf{M}_f^{fe} - \mathbf{M}_f^{be})), \tag{10}$$

where **Mean** is the pixel-wise mean and σ is the ReLU activation function for clipping the negative values.

4. Experiments

4.1. Experimental Setup

Datasets: For the training, we utilize the MSRA10K dataset [5], which includes 10,000 images with high quality pixel-wise annotations. Most of the images in this dataset contain only one salient object. To improve the varieties, we simply augment this dataset by mirror reflection and rotation techniques ($0^\circ, 90^\circ, 180^\circ, 270^\circ$), producing 80,000 training images totally.

For the performance evaluation, we adopt seven public saliency detection datasets as follows.

DUT-OMRON [47]. This dataset has 5,168 high quality images. Images of this dataset have one or more salient objects and relatively complex background. Thus this dataset is more difficult and challenging, and provides more space of improvement for related research in saliency detection.

DUTS [50]. This dataset is currently the largest saliency detection benchmark, and contains 10,553 training images (DUTS-TR) and 5,019 test images (DUTS-TE) with high quality pixel-wise annotations. Both the training and test set contain very challenging scenarios for saliency detection.

ECSSD [46]. This dataset contains 1,000 natural images, which include many semantically meaningful and complex structures in their ground truth segmentation.

HKU-IS [50]. This dataset has 4,447 images with high quality pixel-wise annotations. Images of this dataset are well chosen to include multiple disconnected salient objects or objects touching the image boundary.

PASCAL-S [26]. This dataset is generated from the PASCAL VOC dataset [9] and contains 850 natural images.

SED [2]. This dataset contains two subsets: **SED1** and **SED2**. The **SED1** has 100 images each containing only one salient object, while the **SED2** has 100 images each containing two salient objects.

SOD [46]. This dataset has 300 images, and it was originally designed for image segmentation. Pixel-wise annotation of salient objects was generated by [19]. This dataset is challenging since many images contain multiple objects either with low contrast or touching the image boundary.

Implementation Details: We implement our approach based on the MATLAB R2014b platform with the Caffe toolbox [18]. We run our approach in a quad-core PC machine with an i7-4790 CPU (with 16G memory) and a NVIDIA Titan X GPU (with 12G memory). We train our model using augmented images from the MSRA10K dataset. We do not use validation set and train the model until its training loss converges. The parameters of multi-level feature extraction layers are initialized from the VGG-16 model [37]. For other convolutional layers, we initialize the weights by the "msra" method [15]. We use the SGD method to train our network with a momentum 0.9 and a weight decay 0.0001. We set the base learning rate to 1e-8 and decrease the learning rate by 10% when training loss reaches a flat. The training process takes almost 16 hours and converges after 200k iterations with mini-batch size 8. When testing, the proposed salient object detection algorithm runs at about **16 fps** with 256×256 resolution. The source code can be found at http://ice.dlut.edu.cn/lu/.

Evaluation Metrics: We utilize three main metrics to evaluate the performance of different salient object detection algorithms, including the precision-recall (PR) curves, F-measure and mean absolute error (MAE) [3]. The precision and recall are computed by thresholding the predicted saliency map, and comparing the binary map with the ground truth. The PR curve of a dataset demonstrates the mean precision and recall of saliency maps at different thresholds. The F-measure is a harmonic mean of average precision and average recall, and can be calculated by

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision \times Recall}. \tag{11}$$

We set β^2 to be 0.3 to weigh precision more than recall as suggested in [46] [41] [3] [47].

We report the performance when each saliency map is binarized with an image-dependent threshold. The threshold is determined to be twice the mean saliency of the image:

$$T = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} S(x, y), \tag{12}$$

where W and H are width and height of an image, $S(x, y)$ is the saliency value of the pixel at (x, y). We report the average precision, recall and F-measure over each dataset.

The above overlapping-based evaluations usually give higher score to methods which assign high saliency score to salient pixel correctly. However, the evaluation on non-salient regions can be unfair especially for the methods which successfully detect non-salient regions, but miss the detection of salient regions. Therefore, we also calculate the mean absolute error (MAE) for fair comparisons as suggested by [3]. The MAE evaluates the saliency detection accuracy by

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x, y) - G(x, y)|, \tag{13}$$

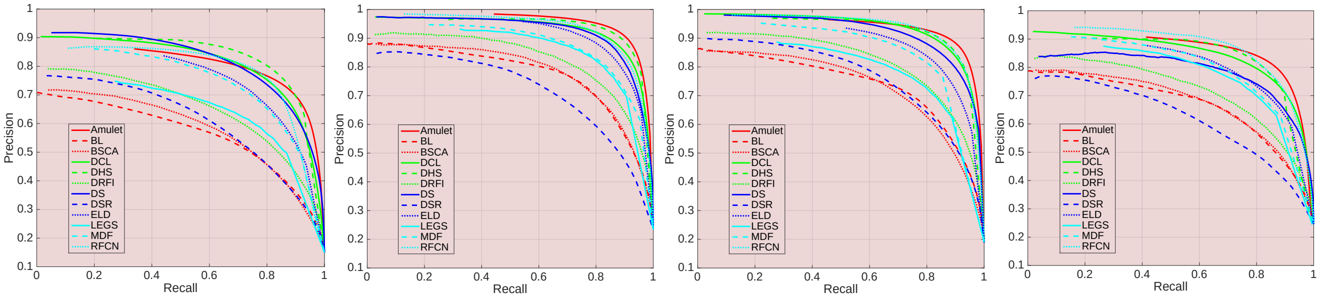
where G is the binary ground truth mask.

4.2. Performance Comparison with State-of-the-art

We compare our algorithm with other 11 state-of-the-art ones including 7 deep learning based algorithms (DCL [22], DHS [27], DS [24], ELD [20], LEGS [41], MDF [50], RFCN [44]) and 4 conventional algorithms (BL[39], BSCA [32], DRFI [19], DSR [23]). For fair comparison, we use either the implementations with recommended parameter settings or the saliency maps provided by the authors.

	DUT-OMRON		DUTS-TE		ECSSD		HKU-IS		PASCAL-S		SOD	
Methods	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
Amulet	0.6471	0.09761	0.7365	0.08517	0.8684	0.05874	0.8542	0.05214	0.7632	0.09824	0.7547	0.13998
Amulet-1/1	0.6413	0.10161	0.7320	0.08796	0.8678	0.05997	0.8460	0.05416	0.7634	0.09948	0.7512	0.14169
Amulet-1/2	0.6408	0.10178	0.7210	0.08807	0.8675	0.05998	0.8456	0.05421	0.7629	0.09965	0.7509	0.14177
Amulet-1/4	0.6392	0.10219	0.7169	0.08851	0.8659	0.06039	0.8439	0.05465	0.7615	0.10001	0.7503	0.14204
Amulet-1/8	0.6356	0.10282	0.6942	0.08933	0.8625	0.06137	0.8397	0.05570	0.7584	0.10067	0.7492	0.14262
Amulet-1/16	0.6266	0.10280	0.6891	0.09110	0.8523	0.06477	0.8327	0.05821	0.7469	0.10273	0.7421	0.14495
Amulet$_{BPR-}$	0.6301	0.12062	0.6912	0.09761	0.8647	0.06572	0.8402	0.06302	0.7533	0.1240	0.7201	0.15340
DCL [22]	0.6842	0.15726	0.7141	0.14928	0.8293	0.14949	0.8533	0.13587	0.7141	0.18073	0.7413	0.19383
DHS [27]	-	-	0.7301	0.06578	0.8675	0.05948	0.8541	0.05308	0.7741	0.09426	0.7746	0.12840
DS [24]	0.6028	0.12038	0.6323	0.09070	0.8255	0.12157	0.7851	0.07797	0.6590	0.17597	0.6981	0.18894
ELD [20]	0.6109	0.09240	0.6277	0.09761	0.8102	0.07955	0.7694	0.07414	0.7180	0.12324	0.7116	0.15452
LEGS [41]	0.5915	0.13335	0.5846	0.13793	0.7853	0.11799	0.7228	0.11934	-	-	0.6834	0.19548
MDF [50]	0.6442	0.09156	0.6732	0.09986	0.8070	0.10491	0.8006	0.09573	0.7087	0.14579	0.7205	0.16394
RFCN [44]	0.6265	0.11051	0.7120	0.09003	0.8340	0.10690	0.8349	0.08891	0.7512	0.13241	0.7426	0.16919
BL [39]	0.4988	0.23881	0.4897	0.23794	0.6841	0.21591	0.6597	0.20708	0.5742	0.24871	0.5798	0.26681
BSCA [32]	0.5091	0.19024	0.4996	0.19614	0.7048	0.18211	0.6544	0.17480	0.6006	0.22286	0.5835	0.25135
DRFI [19]	0.5504	0.13777	0.5407	0.17461	0.7331	0.16422	0.7218	0.14453	0.6182	0.20651	0.6343	0.22377
DSR [23]	0.5242	0.13886	0.5182	0.14548	0.6621	0.17837	0.6772	0.14219	0.5575	0.21488	0.5962	0.23394

Table 1. The F-measure and MAE of different saliency detection methods on six large-scale saliency detection datasets. The best three results are shown in red, green and blue. The proposed methods rank first or second on these datasets.



(a) DUTS-TE (b) ECSSD (c) HKU-IS (d) PASCAL-S

Figure 3. The PR curves of the proposed algorithm and other state-of-the-art methods.

Quantitative Evaluation. As shown in Tab. 1 and Fig. 3, the **Amulet** model can largely outperform other compared counterparts across all the datasets in terms of near all evaluation metrics, which convincingly demonstrates the effectiveness of the proposed method. Results on the SED dataset and PR curves on the DUT-OMRON, SED and SOD datasets appear in the supplemental material due to the limitation of space. From the results, we have other fundamental observations: (1) Our model improves the F-measure with a considerable margin on most of datasets, especially on large-scale datasets, such as DUTS-TE, ECSSD, HKU-IS. And at the same time, our model generally decreases the MAE. This indicates that our model is more convinced of the predicted regions and provides more accurate saliency maps. (2) Although only trained on the MSRA10K dataset, our model significantly outperforms other algorithms that pre-trained on specific saliency datasets, such as LEGS and RFCN on PASCAL-S, MDF on HKU-IS. The superior performance confirms that our model have good generalization abilities on other large-scale datasets. (3) Our method is inferior to DHS on several datasets. However, these datasets are relatively small compared to the era of deep learning.

Qualitative Evaluation. Fig. 4 provides a visual comparison of our approach and other methods. It can be seen that our method generates more accurate saliency maps in various challenging cases, *e.g.*, low contrast between the objects and backgrounds (the first two rows), objects near the image boundary (the 3-4 rows) and multiple disconnected salient objects (the 5-6 rows). What's more, with our BPR component, our saliency maps provide more accurate boundaries of salient objects (the 1, 3, 4, 6 rows).

4.3. Ablation Studies

Feature resolution effects. To verify the importance of resolutions of integrated features, we additionally evaluate several variants of the proposed **Amulet** model with different scales. **Amulet**-$1/n$ denotes the model that takes the integrated features reduced by a factor not larger than n, with respect to the input image. The corresponding performance are also reported in Tab. 1. The results suggest that features of all levels are helpful for saliency detection, and with the increment of resolutions, our approach gradually achieves

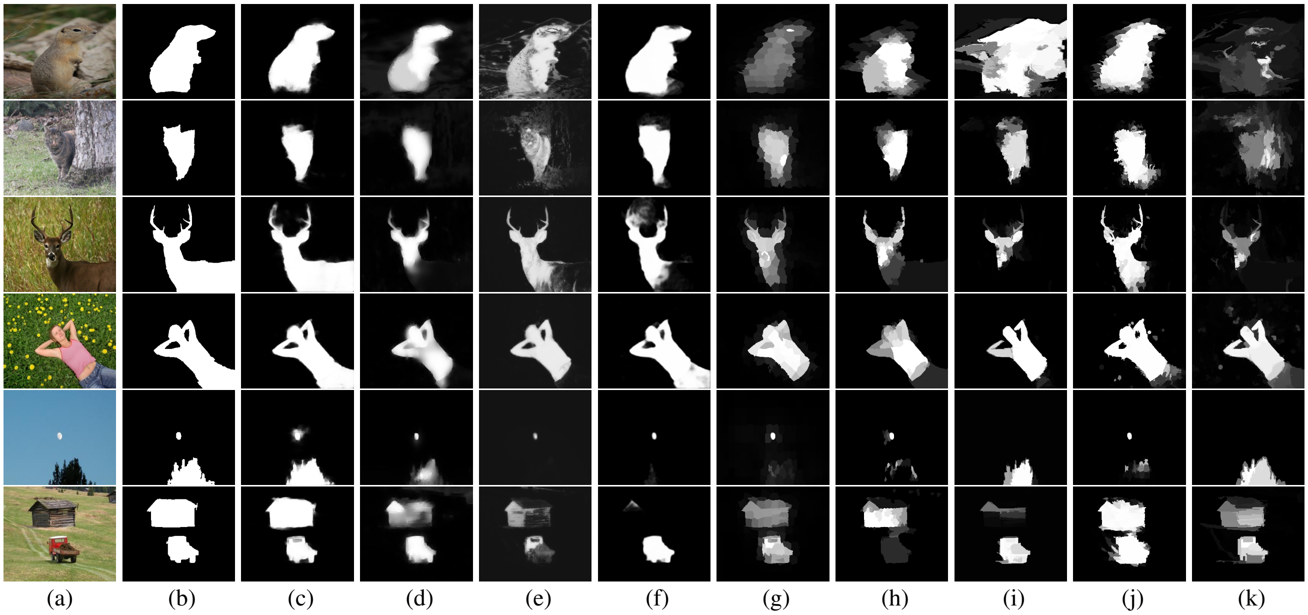


Figure 4. Comparison of saliency maps. (a) Input images; (b) Ground truth; (c) Our method; (d) RFCN; (e) DCL; (f) DHS; (g) DS; (h) LEGS; (i) MDF; (j) ELD; (k) DRFI. The top four row and bottom two row images are from the ECSSD and SED dataset, respectively.

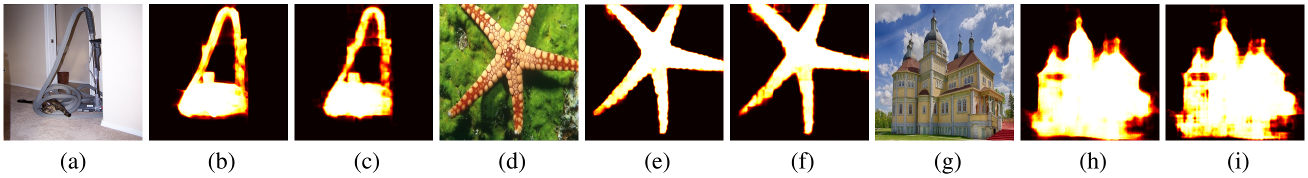


Figure 5. Visual comparison of the Amulet algorithm with /without BPRs. (a)(d)(g) Input images; (b)(e)(h) Predictions of the **Amulet**; (c)(f)(i) Predictions of the **Amulet**$_{BPR-}$. High resolution to see better.

better performance. In addition, even our simplest model (i.e., **Amulet**-1/16) can achieve better results than most of existing methods. This fact further verifies the strength of our proposed methods.

Boundary refinements. To verify the contributions of our proposed BPR, we also implement our proposed approach without BPRs, named **Amulet**$_{BPR-}$, and report the performance in Tab. 1. It can be observed that without BPRs, our approach decreases the performance but not too much in F-measure. But it leads to a large drop in MAE. This indicates that our proposed BPR is capable of detecting and localizing the boundary of most salient objects, while other methods often fail at this fact. Several visual examples are illustrated in Fig. 5.

4.4. Comparison with Other Aggregation Methods

For fair comparison, we perform additional evaluations to verify the detection ability of different aggregation methods. Specifically, we use the same augmented MSRA10K dataset to train the FCN-8s [28], Hypercolumn (HC) [13], SegNet (SN) [1], DeconvNet(DN) [31] and HED [45] for saliency detection task. All compared methods are based on the same VGG-16 model pre-trained on the ImageNet classification task [37]. We drop the unnecessary components in each model and only focus on the feature aggregation part. For our model, we use the simplest model (i.e., **Amulet**-1/16) without BPRs. For each method, we find the optimal parameters to achieve its' best results. The performance on the ECSSD dataset is listed in Tab. 2. As can be seen from Tab. 2, with the aggregation of multi-level features, our approach achieves better performance.

Methods	**FCN-8s**	**HC**	**SN**	**DN**	**HED**	**Ours**
F_β	0.8116	0.8187	0.8145	0.8264	0.8321	0.8521
MAE	0.1343	0.1193	0.0947	0.1435	0.1022	0.0662

Table 2. The performance of different aggregations on ECSSD dataset. Other datasets have the similar performance trend.

5. Conclusion

In this paper, we propose a generic aggregating multi-level convolutional feature framework for salient object detection. Our framework can integrate multi-level feature maps into multiple resolutions, learn to combine feature maps, and predict saliency maps with the integrated features. In addition, edge-aware maps and high-level predictions are embedded into the framework. Experiments demonstrate that our method performs favorably against state-of-the-art approaches in saliency detection.

Acknowledgment. This paper is supported by the Natural Science Foundation of China #61472060, #61502070 and #61528101.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2, 8

[2] A. Borji. What is a salient object? a dataset and a baseline model for salient object detection. *IEEE TIP*, 24(2):742–756, 2015. 2, 6

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015. 2, 6

[4] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti. Adaptive object tracking by learning background context. In *CVPRW*, pages 23–30. IEEE, 2012. 1

[5] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. 2, 6

[6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, pages 534–549, 2016. 1

[7] Y. Ding, J. Xiao, and J. Yu. Importance filtering for image retargeting. In *CVPR*, pages 89–96, 2011. 1

[8] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In *ICCV*, pages 817–824, 2009. 1

[9] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 6

[10] Y. P. Federico Perazzi, Philipp Krähenbühl and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012. 1, 2

[11] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3-d object retrieval and recognition with hypergraph analysis. *IEEE TIP*, 21(9):4290–4303, 2012. 1

[12] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2007. 1

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015. 1, 2, 8

[14] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang. Mobile product search with bag of hash bits and boundary reranking. In *CVPR*, pages 3005–3012, 2012. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[17] 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. 1

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[19] 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. 1, 2, 6, 7

[20] 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. 1, 2, 6, 7

[21] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015. 1, 2

[22] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016. 2, 6, 7

[23] 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. 6, 7

[24] 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. 3, 6, 7

[25] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1

[26] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 2, 6

[27] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016. 1, 2, 5, 6, 7

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 2, 8

[29] V. Mahadevan and N. Vasconcelos. Biologically inspired object tracking using center-surround saliency mechanisms. *IEEE TPAMI*, 35(3):541–554, 2013. 1

[30] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196, 2015. 2, 4

[31] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015. 1, 2, 8

[32] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015. 6, 7

[33] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang. Region-based saliency detection and its application in object recognition. *IEEE TCSVT*, 24(5):769–779, 2014. 1

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 2

[35] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE TPAMI*, 29(2):300–312, 2007. 1

[36] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 4, 6, 8

[38] J. Sun and H. Ling. Scale and object aware image retargeting for thumbnail browsing. In *ICCV*, pages 1511–1518, 2011. 1

[39] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In *CVPR*, pages 1884–1892, 2015. 6, 7

[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2014. 1

[41] 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. 1, 2, 6, 7

[42] 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. 2

[43] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, pages 3119–3127, 2015. 2

[44] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016. 2, 3, 6, 7

[45] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 1, 2, 3, 4, 5, 8

[46] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 1, 2, 6

[47] 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. 1, 2, 6

[48] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, pages 3073–3082, 2016. 1

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014. 2, 4

[50] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015. 1, 2, 6, 7