Learning to Detect Salient Objects with Image-level Supervision

Lijun Wang[1], Huchuan Lu[1], Yifan Wang[1], Mengyang Feng[1]
Dong Wang[1], Baocai Yin[1], and Xiang Ruan[2]
[1] Dalian University of Technology, China [2] Tiwaki Co., Ltd
E-mails: wlj@mail.dlut.edu.cn lhchuan@dlut.edu.cn

Abstract

Deep Neural Networks (DNNs) have substantially improved the state-of-the-art in salient object detection. However, training DNNs requires costly pixel-level annotations. In this paper, we leverage the observation that image-level tags provide important cues of foreground salient objects, and develop a weakly supervised learning method for saliency detection using image-level tags only. The Foreground Inference Network (FIN) is introduced for this challenging task. In the first stage of our training method, FIN is jointly trained with a fully convolutional network (FCN) for image-level tag prediction. A global smooth pooling layer is proposed, enabling FCN to assign object category tags to corresponding object regions, while FIN is capable of capturing all potential foreground regions with the predicted saliency maps. In the second stage, FIN is fine-tuned with its predicted saliency maps as ground truth. For refinement of ground truth, an iterative Conditional Random Field is developed to enforce spatial label consistency and further boost performance.

Our method alleviates annotation efforts and allows the usage of existing large scale training sets with image-level tags. Our model runs at 60 FPS, outperforms unsupervised ones with a large margin, and achieves comparable or even superior performance than fully supervised counterparts.

1. Introduction

Driven by the remarkable success of deep neural networks (DNNs) in many computer vision areas [23, 14, 11, 47, 48], there has been a recent surge of interests in training DNNs using samples with accurate pixel-level annotations for saliency detection [57, 26, 49]. Compared with unsupervised methods [56, 22], DNNs learned from full supervision are more effective in capturing foreground regions that are salient in the semantic meaning, yielding accurate results under complex scenes. Given the data-hunger nature of DNNs, their superior performance heavily relies on large amounts of data set with pixel-level annotations for training. However, the annotation work is very tedious, and training sets with accurate annotations remain scarce and expensive.

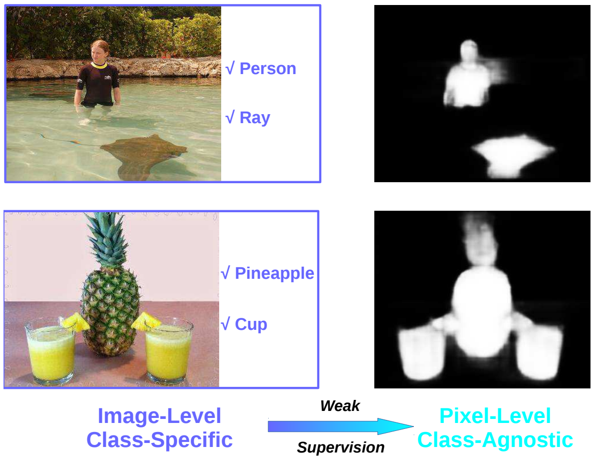


Figure 1. Image-level tags (left panel) provide informative cues of dominant objects, which tend to be the salient foreground. We propose to use image-level tags as weak supervision to learn to predict pixel-level saliency maps (right panel).

To alleviate the need of large scale pixel-wise annotations, we explore the weak supervision of image-level tags to train saliency detectors. Image-level tags indicate the presence or absence of object categories in the images and are much easier to collect than pixel-wise annotations. The task of predicting image-level tags focuses on the object categories in the image and is irresponsible of the object locations (Figure 1 left), whereas saliency detection aims to highlight the full extend of foreground objects and neglects their categories (Figure 1 right). These two tasks seem to be conceptually different but inherently correlated with each other. On the one hand, saliency detection provides object candidates, enabling more accurate category classification. On the other hand, image-level tags provide the category information of dominant objects in the images which are much likely to be the salient foreground. Moreover, recent works [34, 58] have suggested that DNNs trained with only image-level tags are also informative of object locations. It is thus natural to leverage image-level tags as weak supervision to train DNNs for salient object detection. Surprisingly, this idea is largely unexplored in the literature.

In light of the above observations, we propose a new weakly supervised learning method for saliency detection using image-level supervisions only. Our learning method consists of two stages: pre-training with image-level tags and self-training using estimated pixel-level labels.

In the first stage, a deep Fully Convolutional Network (FCN) is pre-trained for the task of image-level tags prediction. To empower FCN with the ability of grounding image-level tags in corresponding object regions, we propose a global smooth pooling (GSP) layer, which aggregates spatial high responses of feature maps into image-level category scores. Compared with global average pooling (GAP) and global max pooling (GMP), GSP alleviates the risks of both overestimating and underestimating the object areas. In addition, GSP takes a more general form of pooling operation, making GAP and GMP its two special cases. Since we focus on generic salient object detection, a new network, named Foreground Inference Net (FIN) is designed. When jointly trained with FCN for image-level tags prediction, the FIN is capable of inferring a foreground heat map capturing all potential category-agnostic object regions, which generalizes well to unseen categories, and provides an initial estimation of the saliency map.

In the second stage, the self-training alternates between estimating ground truth saliency maps and training the FIN using estimated ground truth. To obtain more accurate ground truth estimation, we refine the saliency maps predicted by the FIN with an iterative Conditional Random Field (CRF). Instead of using a fixed unary term as in traditional CRFs, the proposed CRF performs inference by iteratively optimizing the unary term and the prediction results in an EM-like procedure. In practice, the proposed CRF is more robust to input noise and yields higher accuracy.

Our contributions are three folds. Firstly, we provide a new paradigm for learning saliency detectors with weak supervision, which requires less annotation efforts and allows the usage of existing large scale data set with only image-level tags (*e.g.*, ImageNet [7]). Secondly, we propose two novel network designs, *i.e.*, global smooth pooling layer and foreground inference network, which enable the deep model to infer saliency maps by leveraging image-level tags and better generalize to previously unseen categories at test time. Thirdly, we propose a new CRF algorithm, which provides accurate refinement of the estimated ground truth, giving rise to more effective network training. The trained DNN does not require any post-processing step, and yields comparable or even higher accuracy than fully supervised counterparts at a substantially accelerated speed.

2. Related Work

Fully Supervised Saliency Detection. Many supervised algorithms, like CRFs [32], random forests [17, 19, 30], SVMs [35], AdaBoost [60], DNNs [31, 25, 24], *etc*., have been successfully applied to saliency detection. In particular, DNN based methods have substantially improved the performance. Early works [46, 57, 26, 4] use DNNs in a patch-by-patch scanning manner, leading to numerous redundant computations. Recently, FCN based saliency methods [29, 49] have been proposed, with more competitive performance in terms of both accuracy and speed. Nevertheless, training these models requires a large amount of expensive pixel-level annotations. In contrast, ours only relies on image-level tags for training.

Weakly Supervised Learning. Weakly supervised learning has been attracting increasingly more attention from areas like object detection [44], semantic segmentation [37], and boundary detection [18]. In [38], weakly supervised segmentation is formulated as a multiple instance learning problem. A FCN is trained by using a GMP layer to select latent instances. Recently, [58] utilizes GAP to learn CNNs for object localization from image tags. However, both GMP and GAP perform hard selections of latent instances, and are sub-optimal to weakly supervised learning. To address this issue, [39] uses Log-Sum-Exp function to approximate max pooling, while [20] proposes a weighted rank-pooling layer to aggregate spatial responses by their ranked indices. Our method bears a similar spirit but differs from these works in two aspects. Firstly, these methods aim to segment objects of the training categories, whereas we aim to detect generic salient objects, which requires generalization to unseen categories at test time and is more challenging in this sense. The FIN is proposed for this task and has not been explored by these work. Secondly, we study the general form of global pooling operation and propose a new pooling method (*i.e.*, GSP), which explicitly computes the weights of feature responses according to their importance and is better suited to our task. Another line of works is top-down neural attention models [42, 55], which require both forward- and backward-propagation of a trained CNN. In comparison, our method predicts saliency maps by only forward passes. Due to end-to-end training, our predicted saliency maps are also more accurate.

The exploration of weakly supervised learning in saliency detection is limited. In co-saliency [10, 54], the assumption that a group of images contain common objects serves as a form of weak supervision. In [16], binary image tags indicating the existence of salient objects are utilized to train SVMs. To our knowledge, we are the first to leverage object category labels for learning salient object detectors.

3. Weakly Supervised Saliency Detection

A CNN for image-level tags prediction typically consists of a series of convolutional layers followed by several fully connected layers. Let $\boldsymbol{X}$ be a training image, and

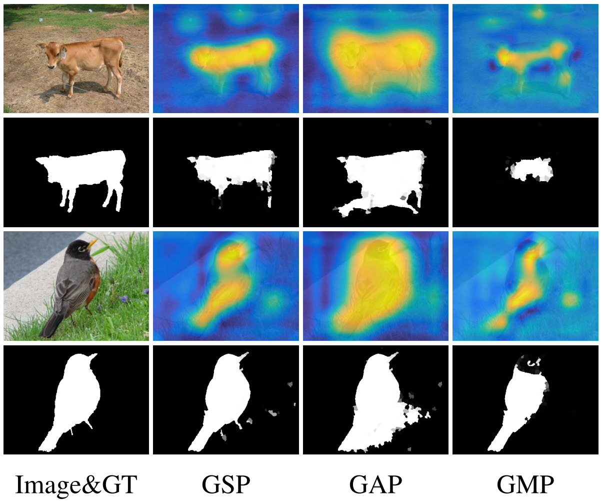


Figure 2. Comparison between different pooling methods. First and third rows: foreground maps produced by the FIN (Section 3.2) with different pooling methods. Second and fourth rows: refined saliency maps (Section 3.4) based on the foreground maps.

$\boldsymbol{l} \subseteq \{1, \ldots, C\}$ denotes its label set. The CNN takes image $\boldsymbol{X}$ as input and predicts a C-dimensional score vector $\hat{\boldsymbol{y}}$. Training the CNN involves minimizing some loss function $L(\boldsymbol{l}, \hat{\boldsymbol{y}})$, which measures the accuracy of the predicted scores based on the ground truth label set. Although the CNN is trained on image-level labels, it has been shown that higher convolutional layers are able to capture discriminative object parts and serve as object detectors. However, the location information encoded in convolutional layers fails to be transferred to fully connected layers.

Based on the above discussions, recent works on dense label prediction tasks (*e.g.*, semantic segmentation) have mostly discarded fully connected layers and explore Fully Convolutional Networks (FCNs) to maintain spatial location information. Given an input image $\boldsymbol{X}$, the FCN produces a subsampled score map $\boldsymbol{S}$, with the k-th channel $\boldsymbol{S}^k$ corresponding to the k-th class. High responses in $\boldsymbol{S}^k$ indicate the potential object regions of class k. The FCN can be easily trained with per-pixel annotations in a fully supervised manner. In weakly supervised settings, where only image-level labels are provided, some form of score aggregation $A(\cdot)$ is required to predict the image-level score $s^k = A(\boldsymbol{S}^k)$ for class k based on the pixel-level score map $\boldsymbol{S}^k$. Image-level supervision can then be injected into the FCN through the predicted class score. Both Global Max Pooling (GMP) and Global Average Pooling (GAP) have been intensively investigated in the literature for this purpose. Next, we provide a discussion of both approaches and propose a new smooth form of global pooling method.

3.1. Aggregation Through Global Smooth Pooling

Global pooling operation is independently performed in each channel of the score map $\boldsymbol{S}$. Without loss of generality, we only consider the score map with one channel, *i.e.*, $\boldsymbol{S} \in \mathbb{R}^{n \times n}$. For a more compact notation, we stack all the columns of the score map $\boldsymbol{S}$ into a vector $\boldsymbol{s}$. Global pooling can then be described in a general form as $s = \boldsymbol{w}^\intercal \boldsymbol{s}$, where $\boldsymbol{w} \in \triangle$ denotes the non-negative weight vector, and $\triangle = \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 = 1, \boldsymbol{x} \succeq 0\}$ denotes the probability simplex. The value of $\boldsymbol{w}$ is determined according to different pooling operations. For GMP, where only the maximum response value is considered, the aggregation is performed by the following maximization problem

$$s = \max_{\boldsymbol{w} \in \triangle} \boldsymbol{w}^\intercal \boldsymbol{s}, \tag{1}$$

which can be simply solved by setting the weight of the highest response to 1, while others to 0. For GAP, all the responses are equally treated with the same value of weight, *i.e.*, $s = \frac{1}{d}\sum_{i=1}^{d} s_i$, where s_i is the i-th element of feature $\boldsymbol{s}$, and $d = n^2$ is the dimension of $\boldsymbol{s}$.

Though, both GMP and GAP have been successfully used for score aggregation, they are sub-optimal for grounding image-level tags in object regions. Considering the fact that higher convolutional layers act as object (part) detectors, we may treat the score maps as spatial responses of an ensemble of these detectors. Since GMP only focuses on a single response, the detectors are trained using the most discriminative object part. As a result, they mostly fail to discover the full extend of objects. In contrast, GAP encourages the detectors to have the same response at all spatial positions, which is irrational and leads to overestimated object areas. See Figure 2 for an example.

We notice that the drawbacks of GMP are mainly caused by the hard selection of the highest response, which involves a non-smooth maximization problem (1). It can be shown that these drawbacks are largely addressed by smoothing the selection operation. To this end, we follow the techniques in [36], and smooth the maximization in (1) by subtracting a strong convex function of the weight vector $\boldsymbol{w}$. For simplicity, we choose the L2 norm as the convex function, and the smoothed GMP is formulated as

$$s = \max_{\boldsymbol{w} \in \triangle} \boldsymbol{w}^\intercal \boldsymbol{s} - \frac{\mu}{2}\|\boldsymbol{w}\|_2^2, \tag{2}$$

where μ is a trade-off parameter to balance the effect of the two terms. As μ approximates 0, (2) is reduced to GMP. When μ is sufficiently large, the maximization of (2) amounts to the minimization of $\|\boldsymbol{w}\|_2^2$, and requires each element of $\boldsymbol{w}$ equals to $\frac{1}{d}$, which has the same effect of GAP.

Since the L2 norm of the weight $\boldsymbol{w}$ does not explicitly contain information of feature responses, we omit this term in the aggregated response s and only use (2) to determine the weight. It turns out that the optimal weight $\hat{\boldsymbol{w}}$ can be computed by a projection of feature $\boldsymbol{s}$ onto the simplex (3). The proposed Global Smooth Pooling (GSP) can then be formulated in the following two steps:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\triangle} \left\| \frac{1}{\mu}\boldsymbol{s} - \boldsymbol{w} \right\|_2^2, \tag{3}$$

$$s = \hat{\boldsymbol{w}}^{\intercal}\boldsymbol{s}. \tag{4}$$

In the first step, the optimal $\hat{\boldsymbol{w}}$ can be computed in $O(d)$ time using the projection algorithm [9]. The second step performs aggregation by a simple inner-production between the feature and weight vectors. For score maps of multi-channels, the GSP is applied to each channel independently.

The proposed GSP is motivated by two insights. Firstly, by smoothing GMP, GSP jointly considers multiple high responses instead of a single one at each time, which is more robust to noisy high responses than GMP and enables the trained deep model to better capture the full extend of the objects rather than only discriminative parts. Secondly, as opposed to GAP, GSP selectively encourages the deep model to fire on potential object regions instead of blindly enforcing high response at every location. As a result, GSP can effectively suppress background responses that tend to be highlighted by GAP. See Figure 2 for an example.

3.2. Foreground Inference Network

When jointly trained with GSP layer on the image-level tags, the score map $\boldsymbol{S}$ generated by FCN can capture object regions in the input image, with each channel corresponding to an object category. For saliency detection, we do not pay special attentions to the object category, and only aim to discover salient object regions of all categories. To obtain such a category-agnostic saliency map, one can simply average the category score map across all the channels. However, there are two potential issues. Firstly, response values in different channels of the score map often subject to distributions of different scales. By simply averaging all the channels, responses in some objects (parts) will be suppressed by regions with higher responses in other channels. In consequence, the generated saliency maps either suffer from background noises (Figure 3 (a)) or fail to uniformly highlight object regions (Figure 3 (b-d)). More importantly, since each channel of the score map is trained to exclusively capture a specific category of the training set, they can hardly generalize to unseen categories (Figure 3 (e)).

The Foreground Inference Net (FIN) is designed to mitigate the above issues by integrating category-specific responses of score map $\boldsymbol{S}$ in a principled manner. The basic architecture of FIN consists of a sequence of convolutional layers followed by one sigmoid layer. It takes the image $\boldsymbol{X}$ as input and predicts a subsampled saliency map $\boldsymbol{F} = [F_{i,j}]^{n\times n}$. Through the final sigmoid layer, each element $F_{i,j}$ is in the range of $[0, 1]$ and measures the saliency degree of the subsampled pixel.

In weakly supervised learning, ground truth saliency maps are not provided for directly training the FIN. We therefore propose an indirect method to jointly train the FIN and FCN[1] for image label prediction. Given the training sample $\{\boldsymbol{X}, \boldsymbol{l}\}$. The input image $\boldsymbol{X}$ is fed forward through both the FIN and FCN to obtain the foreground saliency map $\boldsymbol{F} \in \mathbb{R}^{n\times n}$ and score map $\boldsymbol{S} \in \mathbb{R}^{n\times n\times C}$, respectively. Before score aggregation, we mask each channel of the score map with the foreground saliency map:

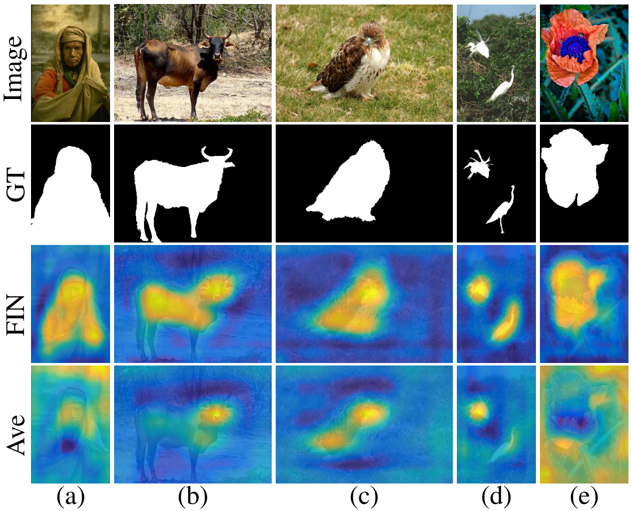


Figure 3. Comparison between the output of FIN and the averaged score maps. The averaged score maps have noisy background responses (a), fail to uniformly highlight foreground (b-d), and cannot generalize to unseen categories (e).

$$\hat{\boldsymbol{S}}_k = \boldsymbol{S}_k \odot \boldsymbol{F}, \tag{5}$$

where $\boldsymbol{S}_k$ denotes the k-th channel of score map $\boldsymbol{S}$; $\odot$ represents the element-wise multiplication; and $\hat{\boldsymbol{S}}_k$ is the k-th channel of the masked score map $\hat{\boldsymbol{S}}$. Score aggregation is then performed on $\hat{\boldsymbol{S}}_k$ using the the proposed GSP to predict image-level score $\hat{s}_k$ for the k-th category. Both FIN and FCN can then be jointly trained by minimizing the loss function $L(\boldsymbol{l}, \hat{\boldsymbol{s}})$. The key motivation is as follows. Each channel of the score map $\boldsymbol{S}$ highlights the region of one object category by spatial high responses. To preserve these high responses in the masked score map $\hat{\boldsymbol{S}}$, the saliency map $\boldsymbol{F}$ is required to be activated at object regions of all categories. Similar ideas can also be found in the attention models [51] and the convolutional feature masking layer [6]. The attention model in [51] adopts GAP to aggregate masked features, whereas we explore GSP. In [6], each mask is generated by bottom-up region proposal methods [45] to characterize one object candidate. In comparison, we aim at learning FIN to automatically infer saliency maps of all categories using weak supervision.

However, one may still concern that the FIN can easily learn the trivial solution of having high responses at all locations. To prevent this trivial solution, we add an additional

[1] Though FIN also has fully convolutional architecture, we exclusively refer to FCN as the network that generates the category score map $\boldsymbol{S}$.

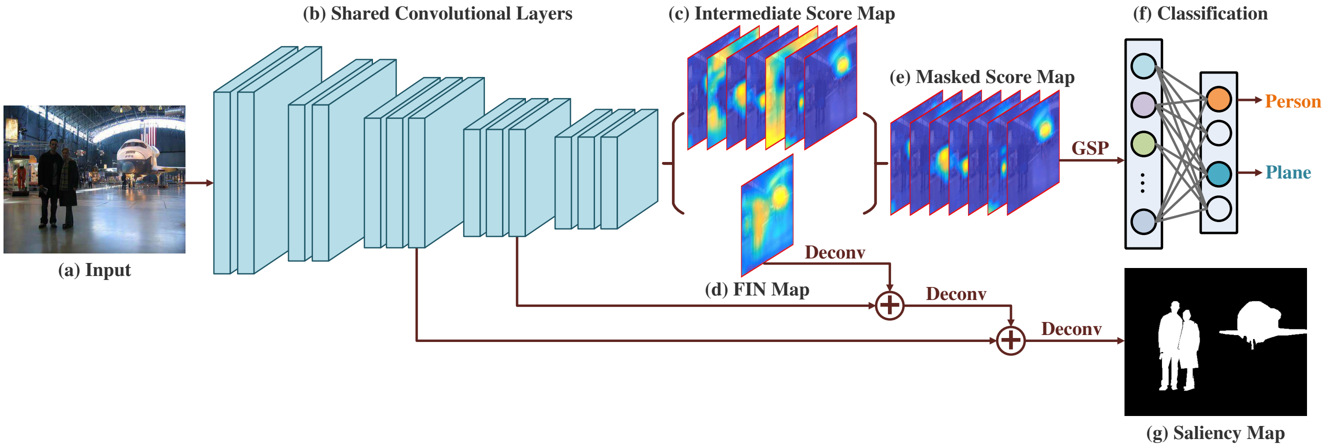


Figure 4. Overview of the network architecture. In the first stage, we jointly train FCN and FIN (b-e) for image categorization (f). In the second stage, the FIN (b,d) is trained for saliency prediction (g).

sparse regularization on the generated foreground saliency map $\boldsymbol{F}$, leading to the following loss function:

$$\min L(\boldsymbol{l}, \hat{\boldsymbol{s}}) + \lambda \|\boldsymbol{f}\|_1, \tag{6}$$

where $\boldsymbol{f}$ denotes the vectorized version of saliency map $\boldsymbol{F}$. The first term encourages $\boldsymbol{F}$ to have high responses at foreground, while the second term penalizes high responses of $\boldsymbol{F}$ at background; λ is a pre-defined trade-off parameter. Note that the regularization term in (6) is imposed on the feature representations rather than the weight parameters, which is reminiscent of a recent work [12], where L1 regularization on feature is used to enforce better generalization from low-shot learning. In contrast, we aim to produce accurate saliency maps with less background noise.

Another lingering concern is that FIN trained on a fixed set of categories may struggle in generalizing to unseen categories. To address this issue, we apply the masking operation (5) to the intermediate score map rather than the final one (See Section 3.3). The intermediate score map does not directly correspond to object categories, As confirmed by [11], it mainly encodes mid-level patterns, *e.g.*, red blobs, triangular structures, specific textures, *etc.*, which are generic in characterizing all categories. Consequently, FIN can capture conspicuous regions of category-agnostic objects/parts and can better generalize to unseen categories.

3.3. Pre-training on Image-level Tags

We now formally describe the first stage of the proposed weakly supervised training method. We train the networks on the ImageNet object detection data set, containing 456k training samples over 200 object categories. Only image-level tags of training images are utilized, while bounding box annotations are discarded for fairness. The training images in the detection data set often contain multiple objects from different categories, as opposed to the image classification data set with only one annotated category in each image. Therefore, the object detection data set is more suitable for solving saliency co-occurrence problem [3].

Network architecture. Figure 4 overviews the network architecture. As discussed in Section 3.2, the FIN for saliency map prediction and the FCN for category score map prediction are jointly trained. Since both networks have highly correlated tasks, they can be trained with shared convolutional features. Specifically, we design the shared network (Figure 4 (b)) following the 16-layer VGG network [43], which consists of 13 convolutional layers inter-leaved by ReLU non-linearity and 4 max pooling layers. The FCN and FIN are built as two sibling sub-networks on top of the shared layers. The FCN consists of a convolutional layer followed by a BN [15] and a ReLU layer. Instead of directly generating the object score map, FCN predicts an intermediate score map (Figure 4 (c)) of 512 channels corresponding to mid-level category-agnostic patterns. The FIN consists of a convolutional layer followed by a BN and a sigmoid layer, and infers a saliency map $\boldsymbol{F}$ (Figure 4 (d)), which is then used to mask the score map to obtain the masked score map (Figure 4 (e)). A GSP layer is used to aggregate the spatial responses in the masked score map into a 512-dimensional image-level score, which is then passed through a fully connected layer to generate a 200-dimensional score $\hat{\boldsymbol{s}}$ for the 200 object categories. The output layer is a sigmoid layer, which converts the category score into category probability $p(k) = \frac{1}{1+\exp(-\hat{s}_k)}$.

Training details. Given a training set $\{\boldsymbol{X}_i, \boldsymbol{l}_i\}_{i=1}^N$ containing N sample pairs, we train the network by minimizing the following objective function

$$\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{i=1}^{N}\Bigg[\sum_{k\in \boldsymbol{l}_i}\log(p(k|\boldsymbol{X}_i;\boldsymbol{\theta})) + \sum_{k\notin \boldsymbol{l}_i}\log(1-p(k|\boldsymbol{X}_i;\boldsymbol{\theta})) - \lambda\big\|\boldsymbol{f}(\boldsymbol{X}_i;\boldsymbol{\theta})\big\|_1\Bigg] + \eta\|\boldsymbol{\theta}\|_2^2,$$

where $\boldsymbol{\theta}$ denotes network parameters; The first and second terms are cross-entropy loss to measure prediction accuracy; The third term is the L1 regularization on the predicted saliency map $\boldsymbol{f}$; the last term represents weight decay; λ and η are empirically set to be 5e-4 and 1e-4, respectively. μ in (3) is set to 10. The weight parameters of shared layers are initialized with the pre-trained VGG model [43], while weights in the other layers are randomly initialized using the method of [13]. All input images are down-sampled to a fixed resolution of 256×256. The FIN has a stride of 16 pixels, leading to output saliency maps of 16×16. We minimize the above objective function using mini-batch Stochastic Gradient Descent (SGD), with a batch size of 64, and momentum of 0.9. The learning rate is initialized as 0.01 and decreased by a factor of 0.1 for every 20 epochs.

3.4. Self-training with Estimated Pixel-level Labels

After pre-training, the coarse saliency maps generated by FIN can already capture the foreground regions. In the second training stage, we refine the prediction by iterating between two steps: a) estimating ground truth saliency maps using the trained FIN, and b) finetuning FIN with the estimated ground truth. To improve the output resolution, we extend the architecture of FIN with two modifications (See Figure 4). Firstly, we build three additional deconvolutional layers on top of the 14-th convolutional layer, where the first two layers have $\times 2$ upsampling factors and the last layer has a $\times 4$ upsampling factor. Secondly, inspired by [33], two skip connections from the 7-th and 10-th convolutional layers are added after the first two deconvolutional layers, respectively, to combine high-level features with semantic meaning and low-level features with finer details. Meanwhile, two techniques are adopted to guarantee the quality of the estimated ground truth: i) refinement with the proposed CRF, and ii) training with a bootstrapping loss [41] that is robust to label noise.

Refinement with the proposed CRF The input image is first over-segmented into a set of superpixels $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_m\}$ using the method in [8]. Each superpixel $\boldsymbol{z}_i$ is characterized by its mean RGB and LAB feature. According to the estimated saliency map $\boldsymbol{F}$ by FIN, we label superpixel $\boldsymbol{z}_i$ as foreground ($\alpha_i = 1$) if the mean saliency value of its pixels is larger than 0.5, or background ($\alpha_i = 0$) otherwise. Two Gaussian Mixture Models (GMMs) are learned to model the foreground and background appearance, respectively, with each GMM containing $K = 5$ components. To refine the saliency labels $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$, a binary fully connected CRF is defined over these labels with the following energy function

$$E(\boldsymbol{\alpha}; \boldsymbol{Z}, \boldsymbol{\omega}) = \sum_i \psi_u(\alpha_i; \boldsymbol{z}_i, \boldsymbol{\omega}) + \sum_{i<j} \psi_p(\alpha_i, \alpha_j; \boldsymbol{z}_i, \boldsymbol{z}_j), \tag{7}$$

Algorithm 1 Iterative CRF refinement.

```
Input: A set of superpixel $\boldsymbol{Z}$ and initial label set $\boldsymbol{\alpha}^0$
Output: Refined label set $\boldsymbol{\alpha}^\star$.
Initialize $\boldsymbol{\alpha}^\star \leftarrow \boldsymbol{\alpha}^0$ .
repeat
    Learn GMM parameters $\boldsymbol{\omega}$ based on label $\boldsymbol{\alpha}^\star$.
    Initialize $P_i^0(\alpha_i) \propto \exp\{-\psi_u(\alpha_i; \boldsymbol{z}_i, \boldsymbol{\omega})\}$.
    for $t = 1, 2, \dots, T$ do
        $P_i^t(\alpha_i) \leftarrow \exp\Big\{ -\psi_u(\alpha_i; \boldsymbol{z}_i, \boldsymbol{\omega})$
                $- \sum_{j \neq i} \psi_p(\alpha_i, \alpha_j; \boldsymbol{z}_i, \boldsymbol{z}_j) P_j^{t-1}(\alpha_j) \Big\}$
        Normalize $P_i^t$
    end for
    Update $\boldsymbol{\alpha}^\star \leftarrow \arg\max_{\boldsymbol{\alpha}} \prod_i P_i^T(\alpha_i)$.
until converge.
```

where $\boldsymbol{\omega} = \{\boldsymbol{\omega}_0, \boldsymbol{\omega}_1\}$ indicates the parameters of GMM models. The unary term is independently computed for each superpixel based on the GMM models and defined as $\psi_u(\alpha_i; \boldsymbol{z}_i, \boldsymbol{\omega}) = -\log\left(\frac{p(\boldsymbol{z}_i | \boldsymbol{\omega}_{\alpha_i})}{p(\boldsymbol{z}_i|\boldsymbol{\omega}_0) + p(\boldsymbol{z}_i|\boldsymbol{\omega}_1)}\right)$, where $p(\boldsymbol{z}_i|\boldsymbol{\omega}_c)$ denotes the probability density of superpixel $\boldsymbol{z}_i$ belonging to foreground ($c = 1$) or background ($c = 0$). The pairwise term enforces label consistency and has the form

$$\psi_p(\alpha_i, \alpha_j; \boldsymbol{z}_i, \boldsymbol{z}_j) = \mathbf{1}(\alpha_i \neq \alpha_j)(\rho_1 g_1(\boldsymbol{z}_i, \boldsymbol{z}_j) + \rho_2 g_2(\boldsymbol{z}_i, \boldsymbol{z}_j)),$$

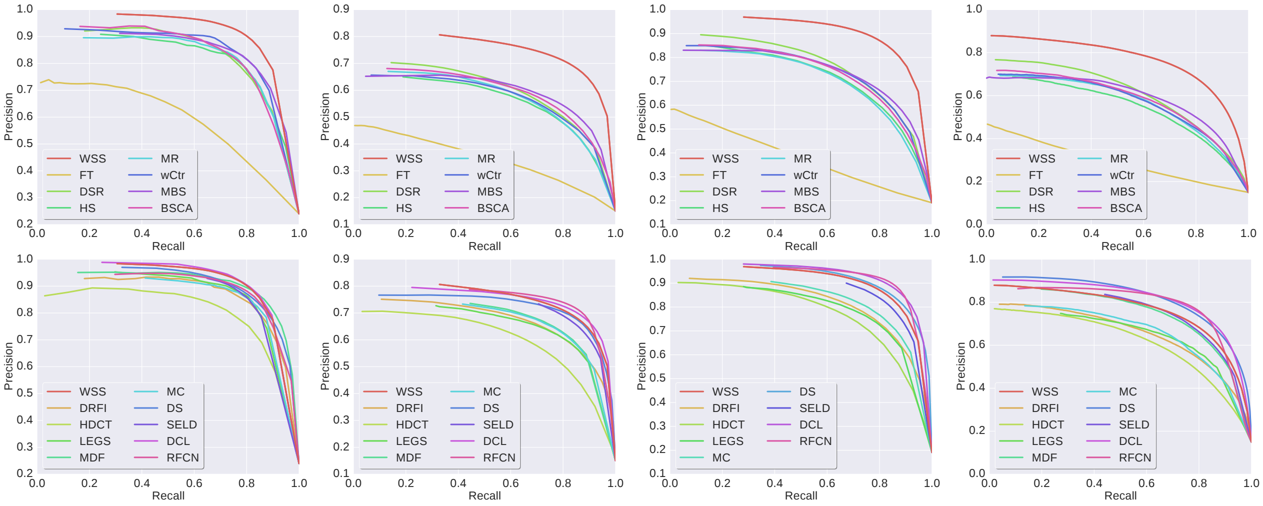
where $\mathbf{1}(\cdot)$ is the indicator function; g_1 and g_2 are Gaussian kernels measuring similarity with weights ρ_1 and ρ_2, respectively. Following [21], we define the kernel functions considering both appearance and position information

$$g_1(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\left(- \frac{|\boldsymbol{p}_i - \boldsymbol{p}_j|^2}{2\gamma_1^2} - \frac{|\boldsymbol{I}_i - \boldsymbol{I}_j|^2}{2\gamma_2^2} \right), \tag{8}$$

$$g_2(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\left(- \frac{|\boldsymbol{p}_i - \boldsymbol{p}_j|^2}{2\gamma_3^2} \right), \tag{9}$$

where $\boldsymbol{I}_i$ and $\boldsymbol{p}_i$ are color feature and position of superpixel $\boldsymbol{z}_i$. All hyper-parameters in ψ_p are set following [21].

Conventional CRFs find the optimal label set $\boldsymbol{\alpha}^\star$ to solve the energy function. In comparison, we propose an EM-like procedure by iteratively updating the GMM parameter $\boldsymbol{\omega}$ and the optimal label set $\boldsymbol{\alpha}$. Given the current optimal $\boldsymbol{\alpha}^\star$, we minimize (7) by learning parameter $\boldsymbol{\omega}$ of foreground and background GMMs; when $\boldsymbol{\omega}$ is fixed, we optimize (7) via mean field based message passing to obtain $\boldsymbol{\alpha}^\star$. Detailed procedure is presented in Algorithm 1. The number of iteration is set to 5 in all experiments. In each iteration, message passing is conducted for $T = 5$ times. By jointly updating GMMs and labels, our algorithm is more robust to initial label noise, yielding more accurate refinement. After refinement, we assign the label of each superpixel to all its pixels to obtain a refined saliency map $\boldsymbol{R}$.



SED THUR HKU-IS DUTS

Figure 5. PR curves of unsupervised methods (first row) and fully supervised methods (second row). The proposed WSS significantly outperforms unsupervised methods and compares favorably against fully supervised methods

Fine-tuning with the robust loss. We use the refined saliency map $\boldsymbol{R}$ as the estimated ground truth to fine-tune the extended FIN. To further reduce the impact of noisy labels, we adopt the bootstrapping loss [41] for training:

$$L_B(\boldsymbol{r},\boldsymbol{f}) = -\sum_i \left[\delta r_i + (1-\delta)a_i\right]\log(f_i) + \left[\delta(1-r_i) + (1-\delta)(1-a_i)\right]\log(1-f_i),$$

where $\boldsymbol{r}$ and $\boldsymbol{f}$ are vectorized version of estimated ground truth $\boldsymbol{R}$ and the output saliency map $\boldsymbol{F}$ of the extended FIN; $a_i = \mathbf{1}(f_i > 0.5)$; i is pixel index; δ is a weight parameter and fixed to 0.95 following [41]. The bootstrapping loss is derived from the cross-entropy loss, and enforces label consistency by treating a convex combination of i) the noisy label r_i, and ii) the current prediction a_i of the FIN, as the target. We solve the loss function using mini-batch SGD, with a batch size of 64. The learning rates of the pre-trained and newly-added layers of FIN are initialized as 1e-3 and 1e-2, respectively, and decreased by 0.1 for every 10 epochs. In practice, the self-training starts to converge after two iterations of "ground truth estimation"–"fine-tuning". At test time, the extended FIN directly generates the saliency maps and no post-processing is required.

4. Experiments

Existing DNN-based methods adopt public saliency data sets for both training and evaluation without a well-established protocol for choosing training/test sets. The usage of different training sets in different methods leads to inconsistent and unfair comparisons. In addition, most existing data sets are originally built for the purpose of model evaluation rather than training, with inadequate amounts of samples for training very complex DNNs. To facilitate fair comparison and effective model training, we contribute a large scale data set named DUTS, containing 10,553 training images and 5,019 test images. All training images are collected from the ImageNet DET training/val sets [7], while test images are collected from the ImageNet DET test set and the SUN data set [50]. Accurate pixel-level ground truths are manually annotated by 50 subjects. The data set can be found at our webpage[2]. To our knowledge, DUTS is currently the largest saliency detection benchmark with the explicit training/test evaluation protocol. For fair comparison in the future research, the training set of DUTS serves as a good candidate for learning DNNs, while the test set and other existing public data sets can be used for evaluation.

We evaluate our **Weakly Supervised Saliency (WSS)** method on the test set of DUTS and 5 public data sets: SED [2], ECSSD [52], THUR [5], PASCAL-S [30] and HKU-IS [26]. We compare WSS with 16 existing methods, including 7 unsupervised ones: FT [1], DSR [28], HS [53], MR [53], wCtr [59], MBS [56], BSCA [40]; and 9 fully supervised ones: DRFI [17], HDCT [19], LEGS [46], MC [57], MDF [26], DS [29], SELD [25], DCL [27], RFCN [49]. Except DRFI and HDCT, all supervised methods are based on DNNs pre-trained on ImageNet [7] classification tasks. Following existing works [53, 46], we evaluate all methods using Precision-Recall (PR) curves, F_β measure, and Mean Absolute Error (MAE).

4.1. Performance Comparison

It is unfair to directly compare supervised unsupervised ones. Therefore, we compare methods within each setting. The proposed WSS is compared with methods of both settings to provide a more comprehensive understanding of them. Both the PR curves in Figure 5 and the F_β measure in

[2] http://saliencydetection.net/duts
http://ice.dlut.edu.cn/lu

Table 1. The F_β measure of our method (WSS), the top 4 unsupervised methods, and top 7 fully supervised methods. All 7 supervised methods use DNNs supervised by pixel-level labels. The **bold fonts** denote the best methods in each setting. The speeds are in the last row.

	Unsupervised				Weakly	Fully						
	MR	wCtr	MBS	BSCA	WSS	LEGS	MDF	MC	DS	SELD	DCL	RFCN
ECSSD	0.690	0.676	0.673	**0.705**	0.823	0.785	0.807	0.796	0.826	0.810	0.829	**0.834**
SED	0.782	**0.786**	0.776	0.756	0.838	0.800	0.795	0.817	0.794	0.815	**0.825**	0.813
PASAL-S	0.583	0.597	**0.604**	0.597	0.720	–	0.705	0.687	0.655	0.714	0.710	**0.747**
THUS	0.542	0.528	**0.547**	0.536	0.663	0.607	0.636	0.610	0.626	0.634	0.657	**0.694**
HKU-IS	0.655	**0.677**	0.663	0.654	0.821	0.723	–	0.743	0.788	0.769	0.853	**0.856**
DUTS	0.510	0.506	**0.511**	0.500	0.657	0.585	0.673	0.594	0.632	0.628	**0.714**	0.712
FPS	6.71	6.76	**76.9**	0.67	62.5	0.52	0.04	0.44	**8.33**	1.80	2.18	0.60

Table 1 show that WSS consistently outperforms unsupervised methods with a considerable margin and compares favorably against fully supervised counterparts. Meanwhile, WSS is also highly efficient with a real-time speed of 62.5 FPS, which is 8 times faster than supervised methods. Note that most of the saliency detection data sets contain huge amounts of objects not belonging to the 200 training categories. The superior performance of WSS confirms that WSS can well generalize to these unseen categories. We also perform additional evaluations to verify the generalization ability of our method. We provide the quantitative and qualitative results on unseen categories, the MAE results, and the PR curves on PASCAL-S and ECSSD in the supplementary material due to limited space.

4.2. Ablative Study

To further verify our main contributions, we compare different variants of our methods. Denote FIN1 as the saliency prediction of the FIN after the first stage pretraining (Section 3.3), WSS1 and WSS1-CRF as the results of FIN1 refined by the proposed iterative CRF and baseline CRF with a fixed unary term, respectively. WSS1-GAP and WSS1-GMP represent the variants of WSS1 by replacing the proposed GSP with GAP and GMP, respectively. WSS1-AVE denotes the variant of WSS1 by replacing the FIN output with the average of score maps across all channels. The F_β on 5 data sets are demonstrated in Figure 6. Besides, we also re-implement the pooling methods in [39, 20] and compare them wit GSP. Detailed results can be found in the supplementary material.

Iterative CRF. WSS1 significantly outperforms FIN1 across all data sets, indicating the critical role of saliency map refinement in the second training stage. Meanwhile, WSS1 also improves the performance of WSS1-CRF with a large margin in all the 5 data sets, which verifies the effectiveness of the proposed iterative CRF over baseline CRFs.

GSP vs. GAP & GMP. The performance of WSS1-GMP is higher than WSS1-GAP in most data sets, while WSS1 with the proposed GSP consistently outperforms both WSS1-GMP and WSS1-GAP, suggesting GSP is more suitable for weakly supervised learning than GMP and GAP.

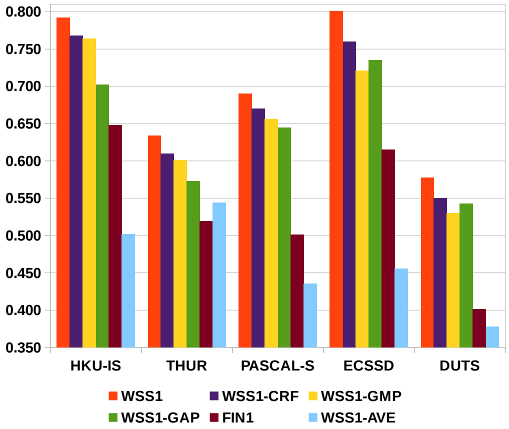


Figure 6. F_β measure of different variants of WSS.

FIN vs. Average Score Maps The performance of WSS1-AVE is inferior than other variants. Even FIN1 without any refinement significantly beats WSS1-AVE in 4 data sets. This confirms the previously discussed disadvantage (Section 3.2) of using the average score map as saliency estimation, and further proves the contribution of the FIN.

4.3. Failure Cases

Since our method is trained on image-level tags only, it sometimes fails to uniformly delineate the object regions in very complex scenarios. We hope to mitigate this issue by exploring various forms of weak supervision in the future.

5. Conclusions

This paper proposes a two-stage training method for saliency detection with image-level weak supervision. In the first stage, two novel network designs, *i.e.*, GSP and FIN, are proposed to estimate saliency maps through learning to predict image-level category labels. In the second stage, the FIN is further fine-tuned using the estimated saliency maps as ground truth. An iterative CRF is developed to refine the estimated ground truth and further improve performance. Extensive evaluations on benchmark data sets verify the effectiveness of our method.

Acknowledgment. This paper is supported by the Natural Science Foundation of China #61472060, #61528101 and #61632006.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009. 7

[2] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *PAMI*, 34(2):1–8, 2012. 7

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *arXiv*, 2014. 5

[4] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li. Disc: Deep image saliency computing via progressive representation learning. *TIP*, 27(6):1135–1149, 2016. 2

[5] 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. 7

[6] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 4

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 7

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 6

[9] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *ICML*, 2008. 4

[10] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *TIP*, 22(10):3766–3778, 2013. 2

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 5

[12] B. Hariharan and R. Girshick. Low-shot visual object recognition. *arXiv*, 2016. 5

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

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[16] H. Jiang. Weakly supervised learning for salient object detection. *arXiv*, 2015. 2

[17] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013. 2, 7

[18] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele. Weakly supervised object boundaries. *arXiv*, 2015. 2

[19] J. Kim, D. Han, Y.-W. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In *CVPR*, 2014. 2, 7

[20] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 2, 8

[21] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011. 6

[22] Y. Kong, L. Wang, X. Liu, H. Lu, and X. Ruan. Pattern mining saliency. In *ECCV*, 2016. 1

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[24] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, 2016. 2

[25] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016. 2, 7

[26] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, 2015. 1, 2, 7

[27] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016. 7

[28] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, 2013. 7

[29] 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. *TIP*, 25(8):3919 – 3930, 2016. 2, 7

[30] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 2, 7

[31] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016. 2

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

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 6

[34] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014. 1

[35] S. Lu, V. Mahadevan, and N. Vasconcelos. Learning optimal seeds for diffusion-based salient object detection. In *CVPR*, 2014. 2

[36] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005. 3

[37] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2

[38] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv*, 2014. 2

[39] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 2, 8

[40] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *CVPR*, 2015. 7

[41] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv*, 2014. 6, 7

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

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014. 5, 6

[44] H. O. Song, R. B. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, T. Darrell, et al. On learning to localize objects with minimal supervision. In *ICML*, 2014. 2

[45] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 4

[46] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, 2015. 2, 7

[47] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, 2015. 1

[48] L. Wang, W. Ouyang, X. Wang, and H. Lu. STCT: sequentially training convolutional networks for visual tracking. In *CVPR*, 2016. 1

[49] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016. 1, 2, 7

[50] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[51] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 4

[52] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013. 7

[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. 7

[54] D. Zhang, D. Meng, and J. Han. Co-saliency detection via a self-paced multiple-instance learning framework. *PAMI*, 39(5):865–878, 2017. 2

[55] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 2

[56] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, 2015. 1, 7

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

[58] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 2

[59] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, 2014. 7

[60] W. Zou and N. Komodakis. Harf: Hierarchy-associated rich features for salient object detection. In *ICCV*, 2015. 2