# Structured Modeling of Joint Deep Feature and Prediction Refinement for Salient Object Detection

Yingyue Xu[1], Dan Xu[2], Xiaopeng Hong[3,7,1], Wanli Ouyang[4], Rongrong Ji[5,7], Min Xu[6], Guoying Zhao[1*]

[1]University of Oulu[†] [2]University of Oxford [3]Xi'an Jiaotong University
[4]SenseTime Computer Vision Group, The University of Sydney
[5]Xiamen University [6]University of Technology Sydney [7]Peng Cheng Laborotory

## Abstract

*Recent saliency models extensively explore to incorporate multi-scale contextual information from Convolutional Neural Networks (CNNs). Besides direct fusion strategies, many approaches introduce message-passing to enhance CNN features or predictions. However, the messages are mainly transmitted in two ways, by feature-to-feature passing, and by prediction-to-prediction passing. In this paper, we add message-passing between features and predictions and propose a deep unified CRF saliency model . We design a novel cascade CRFs architecture with CNN to jointly refine deep features and predictions at each scale and progressively compute a final refined saliency map. We formulate the CRF graphical model that involves message-passing of feature-feature, feature-prediction, and prediction-prediction, from the coarse scale to the finer scale, to update the features and the corresponding predictions. Also, we formulate the mean-field updates for joint end-to-end model training with CNN through back propagation. The proposed deep unified CRF saliency model is evaluated over six datasets and shows highly competitive performance among the state of the arts.*

## 1. Introduction

Visual saliency, born from psychology [18], refers to the attentional selection process on the scenes by the human visual system (HVS). At its early stage, saliency detection models focus on highlighting the most conspicuous regions or eye fixations on a scene [12, 10, 16, 1, 47]. Later, the connotation of saliency is extended to object-level prediction by emphasizing the most outstanding objects. As a result, many salient object detection models are proposed [41, 49, 32, 15, 29, 39, 26, 11, 24, 46], which may have a broad range of potential computer vision applications, such as segmentation [34], image cropping [36], image fusion [9], image classification [38], crowd counting [31], video compression [8], *etc*.

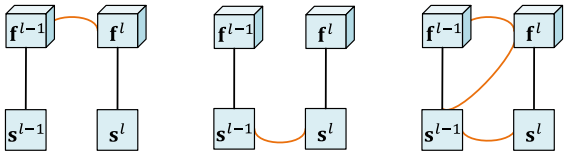


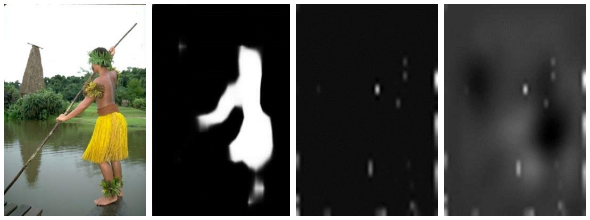

Figure 1. Message-passing of (a) feature and feature, (b) prediction and prediction, and (c) *joint* feature and prediction. $\mathbf{f}^l$ and $\mathbf{s}^l$ refer to the features and the corresponding prediction map at the $l$-th scale respectively. The orange curves indicate there are message-passing between two nodes.

Recently, CNN based saliency models extensively explore to incorporate multi-scale contextual information. Besides directly fusing feature representations, recent approaches introduce message-passing to refine the multi-scale CNN contexts. Zhang *et al*. [50] propose a gated bi-directional message-passing module to pass messages among *features*, and thus to reinforce the multi-scale CNN features for saliency detection (Figure 1-a). Other models pass messages among *predictions* based on the conditional random field (CRF). For instance, recent saliency models [23, 11, 21] tend to adopt Dense-CRF [19] that models highly structured message-passing between pixels on prediction maps, as a post-processing method for saliency refinement. Further, Xu *et al*. [44] introduce the multi-scale CRF to model message-passing between multi-scale prediction maps for depth estimation (Figure 1-b).

After close inspection, we notice that there are clear influences between features and predictions, and thus propose

*Guoying Zhao is the corresponding author.

[†]This work is supported by the Academy of Finland ICT 2023 project (313600), Tekes Fidipro program (Grant No.1849/31/2015), Business Finland project (Grant No.3116/31/2017), Infotech Oulu, and National Natural Science Foundation of China (Grant No.61772419). Computational resources are supported by CSC-IT Center for Science, Finland and Nvidia.



i ii iii iv

Figure 2. From left to right columns are (i) input image; (ii) prediction map $\mathbf{s}^2$; (iii) selected feature map from $\mathbf{f}^3$; and (iv) the corresponding estimated feature map with message-passing between features and predictions. The backbone is detailed in Section 4.

a joint feature and prediction refinement model by performing an extra message-passing between features and predictions. As in Figure 1-c, messages are passing between features and features, predictions and predictions, and features and predictions. The motivations are two-fold:

Firstly, predictions may provide necessary contextual information to features. As the quality of the multi-channel feature maps may vary, the prediction map at the lower level can provide more spatial details to the features. As in Figure 2, the selected feature map in the third column shows inferior quality. Via message-passing between features and predictions, the inferior feature map is enhanced with spatial and shape information from the lower level prediction map. Then, the reinforced deep features will further improve the corresponding prediction map. Secondly, building message-passing between features and predictions facilitates efficient model training. During back propagation, the loss error may be transmitted slowly to impact the lower level features. Modeling message-passing between features and predictions is crucial to strengthen the connection of the model for efficient model training.

Thus, in this paper, we propose a deep unified CRF saliency model that formulates messages passing with CRFs for joint feature and prediction refinement. We design a novel deep cascade CRFs architecture that is seamlessly incorporated with CNN to integrate and refine multi-scale deep features and predictions for a refined saliency map. At each scale, a CRF block is embedded that takes the features and predictions from the lower scale as observed variables to estimate the hidden features and predictions at the current scale. Within each CRF inference, feature-feature, feature-prediction and prediction-prediction messages passing are built. Then, the output refined features and the prediction map are incorporated into the CRF block at the next scale. Thus, a series of CRFs are constructed in a cascade flow and progressively learn a unified saliency map from the coarse scale to the finer scale. Moreover, the CRF inference is formulated with mean-field updates that allow jointly end-to-end training with CNN by back-propagation. The framework of the model is presented in Figure 3.

The contributions of this paper are two-fold:

- We propose a cascade CRFs architecture that is seamlessly incorporated with the backbone CNN to progressively integrate and refine multi-scale contexts from CNNs for salient object detection.
- We model structural information of deep features and deep predictions into a unified CRF model for joint refinement and develop the mean-field approximation inference that supports end-to-end model training through back propagation.

## 2. Related Works

**Saliency Models Based on Multi-scale CNNs** In the past few years, a broad range of saliency models based on CNNs has been proposed. Early deep saliency models benefit from adjusting the inputs to VGG [37], of which the inputs are either multi-scale resized images [28] or global and local image segments [24, 39, 54, 17]. Recently, deep saliency models extensively take the advantages of the multi-scale contexts from CNNs and adopt variable fusion strategies to produce the saliency map. A hierarchical architecture can effectively refine the CNN side outputs from coarse to fine scales [26, 52, 27]. PiCANet [27] hierarchically embeds global and local contexts. Moreover, some saliency models adopt recurrent or cascade structures to progressively learn saliency maps from coarse to fine scales [40, 20, 53, 4]. Zhang *et al*. [53] introduce a multi-path recurrent feedback scheme to progressively enhance the saliency prediction map. RA [4] introduces reverse attention with side-output residual learning to refine the saliency map in a top-down manner. Also, skip connections are widely applied to integrate prediction maps from CNNs [51, 11]. DSS [11] adopts short connections to the side output layers of CNNs to fuse multiple prediction maps. In this work, we aim at integrating multi-scale deep features and deep predictions to boost performance.

**Fully Connected CRFs** As the conditional random field (CRF) is a flexible graphical model in incorporating label agreement assumptions into inference functions, it has been widely adopted for labeling refinement tasks. Several deep saliency models [23, 11, 21] take the advantages of CRF inference and apply a fully connected Dense-CRF [19] to CNN as a post-processing method for refinement. Dense-CRF works on the *discrete* semantic segmentation, which yields an effective iterative message-passing algorithm using mean-field theory. The mean-field approximation can be performed using highly efficient Gaussian filtering in feature space, reducing the complexity from quadratic to linear. However, Dense-CRF parameters are pre-selected by cross validations from a large number of trials and thus is disconnected from the training of CNNs. Zheng *et al*. [55]

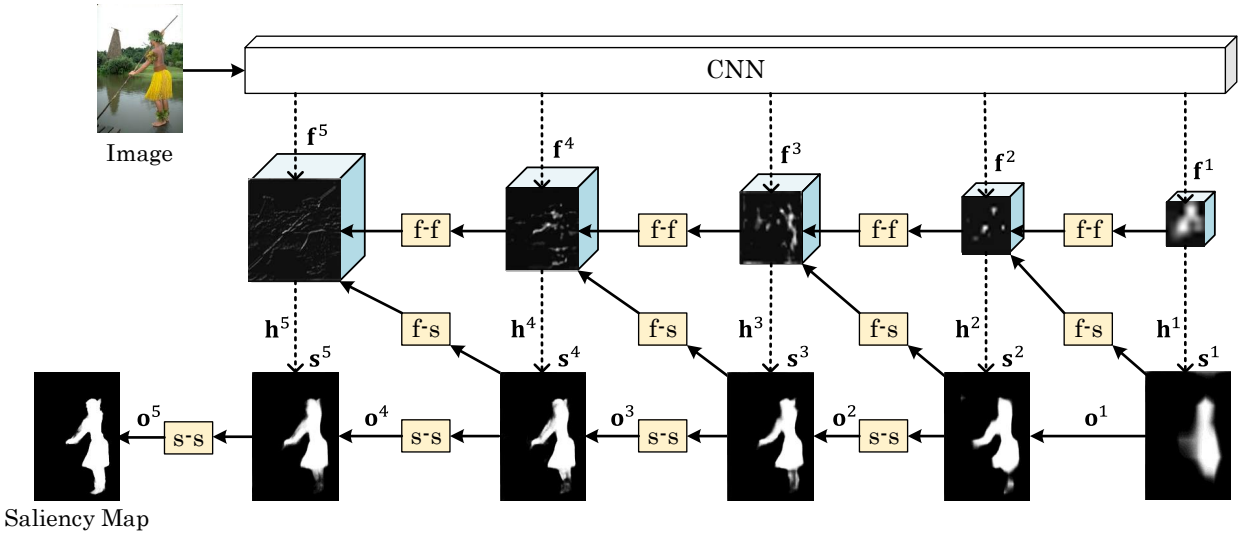



Figure 3. Framework of the proposed deep unified CRF saliency model for jointly modeling structural deep features and predictions. Multi-scale features ($\mathbf{f}^1 \cdots \mathbf{f}^5$) and the corresponding prediction maps ($\mathbf{s}^1 \cdots \mathbf{s}^5$) are extracted from the backbone CNN. At each scale, a CRF block is embedded to jointly refine features and prediction maps with message-passing between features and features (f-f), features and predictions (f-s), and predictions and predictions (s-s). "$\mathbf{h}^1 \cdots \mathbf{h}^5$" and "$\mathbf{o}^1 \cdots \mathbf{o}^5$" are the estimated features and predictions at each scale respectively. "$\mathbf{f}^1 \cdots \mathbf{f}^5$" correspond to "pool5a", "conv5_3", "conv4_3", "conv3_3" and "conv2_2" in the enhanced HED [42] structure, while "$\mathbf{s}^1 \cdots \mathbf{s}^5$" are "upscore-dsn6", "upscore-dsn5", "upscore-dsn4", "upscore-dsn3", and "upscore-dsn2". The dashed arrows omit the details within the backbone CNN. Figure 4 details the implementation within each CRF block.

firstly formulate the CRF inference on top of CNN predictions that enables joint model training via back propagation for semantic segmentation. To solve depth estimation in the *continuous* domain, Xu *et al*. [44, 45] introduce the continuous CRF that incorporates multi-scale CNN prediction maps. Later, Xu *et al*. [43] also propose the attention gated CRF that allows message-passing among the *continuous* features for contour prediction. Chu *et al*. [5] pass messages among features for pose estimation.

All these models formulate CRF with message-passing only among features or among predictions, while we first formulate the *continuous* feature variables and the *discrete* prediction variables into a deep unified CRF model. The new CRF formulation provides explainable solutions for the features, the predictions and the interactions among them, leading to distinct model formulation, inference, and neural network implementation.

## 3. The Deep Unified CRF Saliency Model

**Formulation.** Given an input image $\mathbf{I}$ of $N$ pixels, suppose that a backbone CNN network computes $L$ scales of deep feature maps $\mathbf{F} = \{\mathbf{f}^l\}_{l=1}^L$, where $\mathbf{f}^l = \{\mathrm{f}_{i,m}^l\}_{i=1,m=1}^{N,M}$ consists a set of $M$ feature vectors. Accordingly, $L$ scales of prediction maps $\mathbf{S} = \{\mathbf{s}^l\}_{l=1}^L$ can be computed, where $\mathbf{s}^l = \{\mathrm{s}_i^l\}_{i=1}^N$. The ground truth saliency map corresponding to the input image is denoted as $\mathbf{g} = \{\mathrm{g}_i\}_{i=1}^N$, and each element $\mathrm{g}_i$ takes binary values of 1 or 0.

We formulate the CRF inference to jointly refine multi-scale features and predictions. The objective is to approximate the hidden multi-scale deep feature maps $\mathbf{H} = \{\mathbf{h}^l\}_{l=1}^L$ and the hidden multi-scale prediction maps $\mathbf{O} = \{\mathbf{o}^l\}_{l=1}^L$. In particular, at the $l$-th scale, the observed variables are the features $\mathbf{f}^{l-1}$, $\mathbf{f}^l$ and the prediction $\mathbf{s}^{l-1}$, and the objective is to estimate the corresponding $\mathbf{h}^l$ and $\mathbf{o}^l$. With a cascade flow of a series of CRFs, the side outputs are progressively refined from coarse ($l = 1$) to fine ($l = L$). The refined prediction map $\mathbf{o}^L$ is the final saliency map.

The conditional distribution of the CRF at the $l$-th scale is defined as follow:

$$P(\mathbf{h}^l, \mathbf{o}^l | \mathbf{I}, \Theta) = \frac{1}{Z(\mathbf{I}, \Theta)} \exp\Big\{ - E\left(\mathbf{h}^l, \mathbf{o}^l, \mathbf{I}, \Theta\right) \Big\}, \quad (1)$$

where $\Theta$ refers to the relative parameters. The energy function $E = E(\mathbf{h}^l, \mathbf{o}^l, \mathbf{I}, \Theta)$ is formulated as follow:

$$E = \sum_i \phi_h(\mathrm{h}_i^l, \mathrm{f}_i^l) + \sum_i \phi_o(\mathrm{s}_i^l, \mathrm{o}_i^l) + \sum_{i \neq j} \psi_h(\mathrm{h}_i^l, \mathrm{h}_j^{l-1}) + \sum_i \psi_{hs}(\mathrm{h}_i^l, \mathrm{o}_i^{l-1}) + \sum_{i \neq j} \psi_o(\mathrm{o}_i^l, \mathrm{o}_j^l). \quad (2)$$

The first term of Eq. 2 is a feature level unary term corresponding to an isotropic Gaussian: $\phi_h(\mathrm{h}_i^l, \mathrm{f}_i^l) = -\frac{\alpha_i^l}{2} ||\mathrm{h}_i^l - \mathrm{f}_i^l||^2$ where $\alpha_i^l > 0$ is a weighting factor. The second term is a prediction level unary term, $\phi_o(\mathrm{s}_i^l, \mathrm{o}_i^l) = ||\mathrm{s}_i^l - \mathrm{o}_i^l||^2$.

The third term is a feature level pairwise term describing the potential between features, where

$$\psi_h(\mathrm{h}_i^l, \mathrm{h}_j^{l-1}) = \mathrm{h}_i^l \mathrm{W}_{i,j}^{l,l-1} \mathrm{h}_j^{l-1}, \tag{3}$$

where $\mathrm{W}_{i,j}^{l,l-1} \in \mathbb{R}^{M \times M}$ is a bilinear kernel. The fourth term is a feature level pairwise term defining the potential between features and predictions, where

$$\psi_{hs}(\mathrm{h}_i^l, \mathrm{o}_i^{l-1}) = \mathrm{h}_i^l \mathrm{V}_{i,j}^{l,l-1} \mathrm{o}_j'^{l-1}, \tag{4}$$

where $\mathrm{V}_{i,j}^{l,l-1} \in \mathbb{R}^{M \times M}$ is also a bilinear kernel to couple the features and the predictions. $\mathrm{o}'^{l-1}$ denotes a concatenation of $M$ prediction maps $\mathrm{o}^{l-1}$. The fifth term is a prediction level pairwise term defining the potential between the predictions as follows:

$$\psi_o(\mathrm{o}_i^l, \mathrm{o}_j^l) = \beta_1 K_{i,j}^1 ||\mathrm{o}_i^l - \mathrm{o}_j^l||^2 + \beta_2 K_{i,j}^2 ||\mathrm{o}_i^l - \mathrm{o}_j^l||^2. \tag{5}$$

$K_{i,j}^1$ and $K_{i,j}^2$ are Gaussian kernels that measure the relationship between two pixels. Specifically, $K_{ij}^1$ is the similarity kernel measuring the appearance similarity between two pixels as $K_{ij}^1 = \nu_1 \exp(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2})$ and $K_{ij}^2$ is the proximity kernel that measures the spatial relationship between two pixels as $K_{ij}^2 = \nu_2 \exp(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2})$. $\nu_1$ and $\nu_2$ are the contributions of each Gaussian kernel, respectively.

**Inference.** We perform the mean-field approximation to estimate a distribution $q(\mathbf{h}^l, \mathbf{o}^l | \mathbf{I}, \Theta) = \prod_{i=1}^N q_{i,l}(\mathbf{I}, \Theta | \mathrm{h}_i^l, \mathrm{o}_i^l)$ that is an approximation to $P(\mathbf{h}^l, \mathbf{o}^l | \mathbf{I}, \Theta)$ by minimizing the Kullback-Leiber divergence [35]. By considering $J_{i,l} = \log q_{i,l}(\mathbf{h}^l, \mathbf{o}^l | \mathbf{I}, \Theta)$ and rearranging its expression into an exponential form, the mean-field updates can be derived as:

$$\begin{aligned} \bar{\mathrm{h}}_i^l = & \frac{1}{\alpha_i^l} \Big( \alpha_i^l \mathrm{f}_i^l + \sum_{l \neq l-1} \sum_{i \neq j} \mathrm{W}_{i,j}^{l,l-1} \mathrm{h}_j^{l-1} \\ & + \sum_{l \neq l-1} \sum_{i \neq j} \mathrm{V}_{i,j}^{l,l-1} \mathrm{o}_j'^{l-1} \Big). \end{aligned} \tag{6}$$

$$\rho_i^l = 1 + 2\Big( \beta_1 \sum_{j \neq i} K_{ij}^1 + \beta_2 \sum_{j \neq i} K_{ij}^2 \Big), \tag{7}$$

$$\mu_i^l = \frac{o_i^l}{\rho_i^l} + \frac{2}{\rho_i^l} \Big( \beta_1 \sum_{j \neq i} K_{ij}^1 \mu_j^l + \beta_2 \sum_{j \neq i} K_{ij}^2 \mu_j^l \Big). \tag{8}$$

Where Eq. 6 and 8 represent the mean-field inference of the estimated latent feature and prediction variables, respectively. Eq. 7 is the variance of the mean-field approximated distribution used as the normalization factor in Eq. 8. At the $l$-th scale, the optimal $\mathbf{o}^l$ can be approximated by mean-field updates of $T$ iterations on the prediction level. At each time $t$ of the mean-field iteration, an estimated saliency map $\mu_t^l$ can be approximated. After $T$ mean-field iterations, the estimated prediction map $\mu_T^l$ is regraded as the estimation of $\mathbf{o}^l$ from the CRF at the $l$-th scale. The details of the mean-field updates are presented in Figure 4.

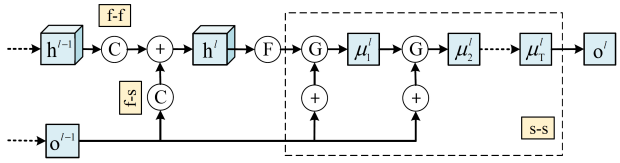



Figure 4. Details of the mean-field updates within CRF. The circled symbols indicate message-passing operations within the CRF block. (i) Message-passing to estimate $\mathbf{h}^l$ by convolutions (Eq. 6): "C" indicates a convolutional layer followed by the corresponding deconvolutional layer, crop layer and a scale layer. (ii) Message-passing to estimate $\mathbf{o}^l$ with Gaussian pairwise kernels in $T$ iterations (Eq. 8): "G" means the Gaussian filtering. "F" is the process of computing a prediction map by a convolutional layer with kernel size 1 followed with the corresponding deconvolutional layer and a crop layer. "+" refers to element-wise sum.

In the cascade flow, the observation $\mathbf{s}^l$ is obtained via integrating the prediction map $\mathbf{s}^l$ and the estimated map $\mathbf{o}^{l-1}$ from the CRF at the previous scale, *i.e.*, $\mathrm{s}_i^l = \mathrm{s}_i^l + \mathrm{o}_i^{l-1}$.

**Mean-field Iteration with Neural Networks.** The inference of the CRF block is based on the mean-field approximation, which can be implemented as a stack of CNN layers to facilitate jointly training as in Figure 4. The mean-field updates for Eq. 6 is implemented with convolutions. The similarity kernel $K^1$ and the proximity kernel $K^2$ in Eq. 7 and 8 are computed based on permutohedral lattice [2] to reduce the computational cost from quadratic to linear [35]. The weighting of $\beta_1$ and $\beta_2$ is convolved with an $1 \times 1$ kernel. By combining the outputs, the normalization matrix $\rho^l$ and the corresponding $\mu^l$ can be computed. The weights $\beta_1$ and $\beta_2$ are obtained by back propagation.

## 4. Implementation

**Training Data.** As many state-of-the-art saliency models use the MSRA-B dataset [30] as the training data [11, 23, 40, 26], we also follow the same training protocol as in [11, 23] to optimize the deep unified CRF model, for fair comparisons. The MSRA-B dataset consists of 2,500 training images, 500 validation images, and 2000 testing images. The images are resized to $240 \times 320$ as the input to the data layer. Horizontal flipping is used for data augmentation such that the number of training samples is twice as large as the original number.

**Baseline Model.** In order to learn high quality feature maps and to fairly compare the proposed deep unified CRF model with the state of the arts, the front-end CNN is based on the implementation of DSS [11] with the enhanced HED [42] structure. Latest state-of-the-art saliency models, *e.g.*, DSS [11] and RA [4] both adopt such front-end

network to extract multi-scale side outputs. The only difference is that we discard the side output prediction maps computed from the layer "conv1_2", which is used as the sixth side output map by DSS [11] and RA [4]. Thus, totally five scales of side outputs are extracted. Specifically, the multi-scale features "$\mathbf{f}^1 \cdots \mathbf{f}^5$" correspond to "pool5a", "conv5_3", "conv4_3", "conv3_3" and "conv2_2" in the enhanced HED [42] structure, while the corresponding prediction maps "$\mathbf{s}^1 \cdots \mathbf{s}^5$" are "upscore-dsn6", "upscore-dsn5", "upscore-dsn4", "upscore-dsn3", and "upscore-dsn2" respectively.

**Optimization.** To reduce training time, the proposed deep unified CRF model is optimized with two stages, including a pre-training and an overall optimization.

In the pre-training stage, we firstly optimize the model by adding feature-feature and feature-prediction messages passing to the front-end CNN. The parameters $\Delta$ of the networks and the scale-specific parameters $\varepsilon$=$\{\epsilon_l\}_{l=1}^{L}$ are trained by minimizing the standard sigmoid cross-entropy loss. The output score maps at each scale are optimized respectively. Thus, the loss function is as follow:

$$\mathcal{L}^l_{\text{Stage1}}(\Delta, \epsilon_l) = -\sum_{i=1}^{\mathcal{N}} \Big( \mathrm{g}_i \log(\mathrm{s}_i^l) + (1-\mathrm{g}_i)\log(1-\mathrm{s}_i^l) \Big),$$

where $\mathrm{g}_i$ is the $i$-th ground-truth label and $\mathrm{s}_i^l$ is the $i$-th pixel on the prediction map at the $l$-th scale. $\mathcal{N}$ refers to the total number of image pixels over the training set.

In the second stage, overall parameter optimization is performed to the whole network. Still, the parameters $\{\Delta, \varepsilon\}$ trained from the pre-training stage will be jointly optimized with the parameters $\beta = \{\beta_1^l, \beta_2^l\}_{l=1}^{L}$ for prediction-prediction message-passing at each scale. The sigmoid cross-entropy loss function is computed for the final scale $L$ as follow:

$$\mathcal{L}_{\text{Stage2}}(\Delta, \varepsilon, \beta) = -\sum_{i=1}^{\mathcal{N}} \Big( \mathrm{g}_i \log(\mathrm{o}_i^L) + (1-\mathrm{g}_i)\log(1-\mathrm{o}_i^L) \Big).$$

**Parameters.** In this work, the VGG-16 [37] is adopted to initialize the parameters for the pre-training stage, and the front-end CNN is finetuned. The parameters for the pre-training stage are set as: batch_size (1), learning rate (1e-9), max_iter (14000), weight decay (0.0005), momentum (0.9), iter_size (10). The learning rate is decreased by 10% when the training loss reaches a flat.

In the second training stage, the parameters learned from the pre-training stage are optimized with a learning rate of 1e-12, while the parameters for prediction-prediction message-passing are learned with the learning rate as 1e-8. Another 10 epochs are trained for the overall optimization.

All the implementation is based on the public Caffe library [13]. The Gaussian pairwise kernels are implemented based on continuous CRF [44]. The GPU for training acceleration is the Nvidia Tesla P100. The pre-training takes about 6 hours and the overall training takes about 14 hours.

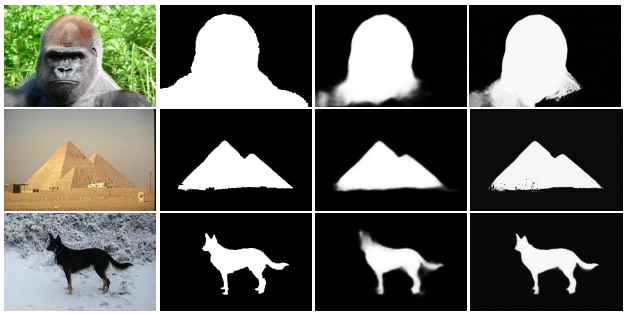



Figure 5. Prediction maps $\mathbf{o}^5$ from the deep unified CRF model with only message-passing bewteen predictions with emphasis on (1) proximity ($T = 6$, $\sigma_\alpha = 1$, $\sigma_\beta = 10$, $\sigma_\gamma = 10$) and (2) similarity ($T = 6$, $\sigma_\alpha = 10$, $\sigma_\beta = 10$, $\sigma_\gamma = 1$). Example images are selected from the ECSSD dataset with the ground truth.

# 5. Experiments

## 5.1. Datasets

For comprehensive comparisons, the proposed deep unified CRF saliency model is evaluated over six datasets, including: MSRA-B [30], ECSSD [48], PASCAL-S [25], DUT-OMRON [49], HKU-IS [24] and iCoseg [3]. MSRA-B is the training dataset consisting of 5000 images. ECSSD contains a pool of 1000 images with even more complex salient objects on the scenes. PASCAL-S is a dataset for salient object detection consisting of a set of 850 images from PASCAL VOC 2010 [6] with multiple salient objects on the scenes. DUT-OMRON dataset contains a large number of 5168 more difficult and challenging images. HUK-IS consists of 4447 challenging images and pixel-wise saliency annotation. ICoseg contains 643 images and each image may consist of multiple salient objects.

## 5.2. Evaluation Metrics

We employ two types of evaluation metrics to evaluate the performance of the saliency maps: F-measure and mean absolute error (MAE). When a given saliency map is slidingly thresholded from 0 to 255, a precision-recall (PR) curve can be computed based on the ground truth. F-measure is computed to count for the saliency maps with both high precision and recall:

$$F = \frac{\left(1+b^2\right) \cdot \texttt{precision} \cdot \texttt{recall}}{b^2 \cdot \texttt{precision} + \texttt{recall}}, \qquad (9)$$

where $b^2$ is set as 0.3 [1] to emphasize the precision. In this paper, the Max F-measure is evaluated. MAE [33] measures

| $T$ | $\sigma_\alpha$ | $\sigma_\beta$ | $\sigma_\gamma$ | $\nu_1$ | $\nu_2$ | F-measure | MAE |
|---|---|---|---|---|---|---|---|
| 3 | 1 | 1 | 1 | 1 | 1 | 0.892 | 0.071 |
| 6 | 1 | 1 | 1 | 1 | 1 | 0.892 | 0.071 |
| 6 | 10 | 10 | 10 | 1 | 1 | <u>0.909</u> | 0.071 |
| 6 | 10 | 1 | 1 | 1 | 1 | 0.896 | 0.084 |
| 6 | 1 | 10 | 1 | 1 | 1 | 0.893 | <u>0.070</u> |
| 6 | 1 | 1 | 10 | 1 | 1 | 0.892 | <u>0.070</u> |
| 6 | 10 | 10 | 1 | 1 | 1 | **0.910** | 0.094 |
| 6 | 1 | 10 | 10 | 1 | 1 | 0.894 | **0.069** |
| 6 | 10 | 1 | 10 | 1 | 1 | 0.896 | 0.095 |
| 6 | 1 | 1 | 1 | 3 | 5 | 0.892 | 0.071 |
| 6 | 1 | 1 | 1 | 5 | 3 | 0.892 | 0.071 |
| 6 | 1 | 1 | 1 | 1 | 1 | 0.892 | 0.071 |

Table 1. One CRF block with only message-passing between predictions at scale 5 is jointly trained with the backbone CNN for 10 epochs. The model is tested on ECSSD dataset. $T$ refers to the number of mean-field iterations. The similarity kernel is controlled by $\sigma_\alpha$ and $\sigma_\beta$ and the weight $\nu_1$, while the proximity kernel is controlled by $\sigma_\gamma$ and the weight $\nu_2$. The best performances are in bold while the second best results are underlined.

the overall pixel-wise difference between the saliency map `sal` and the ground truth `gt`:

$$\texttt{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\texttt{sal}(i) - \texttt{gt}(i)|. \tag{10}$$

### 5.3. Model Analysis

**Gaussian Pairwise Kernels.** Recall that $\nu_1$, $\sigma_\alpha$ and $\sigma_\beta$ are pre-defined parameters to control the bandwidth of $K^1$, while $\nu_2$ and $\sigma_\gamma$ control the bandwidth of $K^2$ for prediction level message-passing. Various schemes of parameter settings are experimented as in Table 1, of which one CRF block at scale 5 is jointly trained with the front-end CNN for 10 epochs. It can be perceived that when the similarity $K^1$ is emphasized, the output map receives better F-measure; when the proximity $K^2$ is emphasized, the MAE sharply reduces. Meanwhile, different settings of $\nu_1$ and $\nu_2$ result in the same performance. This is because that the mean-field iterations learn the weights of the two Gaussian pairwise kernels $\beta_1$ and $\beta_2$ respectively, such that we can initialize $\nu_1$ and $\nu_2$ as 1[1]. Thus, compared to Dense-CRF [19], our proposed CRF releases two pre-defined parameters.

Figure 5 presents three examples when the similarity $K^1$ or the proximity $K^2$ are emphasized respectively. When the proximity counts more, the output saliency maps are smoother. However, as the *dog* example shows, the saliency objects possess more ambiguous object boundaries. The similarity kernel, however, emphasizes more on the image feature similarity, such that it is more sensitive to boundary division. But as the similarity may be too sensitive to the details on the image, this also results in some defects shown in the *pyramids* example where the left corner of the pyramids contains many flaws. For evaluation, F-measure is based on thresholded segmentation to evaluate region similarity [7], while MAE calculates for pixel level accuracy. Thus, the emphasis on similarity gets better F-measure, while the emphasis on proximity gets better MAE.

[1]We set $\nu_1$ and $\nu_2$ as 1 in the following experimental descriptions.

| Scale ($l$) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Predictions ($\mathbf{s}^l$) | 0.824 | 0.864 | 0.882 | 0.883 | 0.884 |
| Estimations ($\mathbf{o}^l$) | - | 0.894 | 0.915 | 0.921 | 0.921 |

Table 2. F-measure of prediction maps $\mathbf{s}^l$ from each scale of the pre-trained backbone CNN and the estimated prediction maps $\mathbf{o}^l$ at each scale of the deep unified CRF model, on ECSSD dataset.

**Scale-specific Gaussian Kernels.** We evaluate the F-measure of the estimated prediction maps $\mathbf{o}^l$ at each scale of the deep unified CRF model with only message-passing between predictions. With Gaussian kernels $\sigma_\alpha = 10$, $\sigma_\beta = 10$, $\sigma_\gamma = 1$, it results in the highest F-measure in Table 1. According to Table 2, the quality of the estimated prediction maps $\mathbf{o}^l$ from the CRFs continuously improves remarkably at scale 2, 3 and 4. However, the prediction maps $\mathbf{s}^4$ and $\mathbf{s}^5$ from the front-end CNN possess similar F-measure and the estimated map $\mathbf{o}^5$ from the CRFs has almost no improvements. Thus, effective F-measure enhancement is performed only at scales from 2 to 4. Hence, it is sufficient for the cascade CRFs structure to integrate five side outputs from CNN rather than integrating six scales of maps by DSS [11] and RA [4] models.

In practice, the parameters of the first three CRFs at the scales from 2 to 4 are set to emphasize similarity to enhance F-measure, while the parameters of the last CRF at the last scale 5 are set to emphasize proximity to further reduce MAE. Thus, the kernels of the CRFs at scale 2 to 4 are set as $\sigma_\alpha = 60$, $\sigma_\beta = 5$,$\sigma_\gamma = 3$ to emphasize more on the similarity, while the kernels of the CRFs at scale 5 are defined as $\sigma_\alpha = 1$, $\sigma_\beta = 10$, $\sigma_\gamma = 10$. $T$ is set as 3 for efficient computation.

**Message-passing.** We train the deep unified CRF model by involving variable combinations of message-passing within the inference. As in Table 4, the baseline is the backbone CNN based on the enhance HED structure and we evaluate the F-measure of the output prediction map $\mathbf{o}^5$. By adding a Dense-CRF for post processing, the F-measure is 0.902. Then, the message-passing comparisons are conducted by implementing pairwise terms in Eq. 2 to the cascade CRFs architecture for joint model training. 1) By adding message-passing between predictions, the F-measure rises to 0.921. Also, by joint training CRF through back propagation, the prediction map from the baseline framework improves from 0.884 to 0.899. 2) By adding message-passing between

| Dataset | Metric | DRFI | MDF | RFCN | DHS | Amulet | UCF | DCL$^+$ | MSR$^+$ | DSS | DSS$^+$ | RA | Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSRA-B | maxF | 0.851 | 0.885 | - | - | - | - | 0.916 | 0.930 | 0.920 | 0.928 | 0.931 | **0.935** |
| | MAE | 0.123 | 0.066 | - | - | - | - | 0.047 | 0.042 | 0.043 | 0.035 | 0.036 | **0.029** |
| PASCAL-S | maxF | 0.690 | 0.759 | 0.829 | 0.824 | 0.832 | 0.818 | 0.822 | 0.852 | 0.826 | 0.831 | 0.829 | **0.858** |
| | MAE | 0.210 | 0.142 | 0.118 | 0.094 | 0.100 | 0.116 | 0.108 | **0.081** | 0.102 | 0.093 | 0.101 | 0.089 |
| DUT-OMRON | maxF | 0.664 | 0.694 | 0.747 | - | 0.743 | 0.730 | 0.757 | 0.785 | 0.764 | 0.781 | 0.786 | **0.802** |
| | MAE | 0.150 | 0.092 | 0.095 | - | 0.098 | 0.120 | 0.080 | 0.069 | 0.072 | 0.063 | 0.062 | **0.057** |
| HKU-IS | maxF | 0.775 | 0.860 | 0.894 | 0.892 | 0.897 | 0.888 | 0.904 | 0.916 | 0.900 | 0.916 | 0.913 | **0.920** |
| | MAE | 0.146 | 0.129 | 0.088 | 0.052 | 0.051 | 0.061 | 0.049 | **0.039** | 0.050 | 0.040 | 0.045 | **0.039** |
| ECSSD | maxF | 0.784 | 0.847 | 0.899 | 0.907 | 0.914 | 0.902 | 0.901 | 0.913 | 0.908 | 0.921 | 0.921 | **0.928** |
| | MAE | 0.172 | 0.106 | 0.091 | 0.059 | 0.061 | 0.071 | 0.068 | 0.054 | 0.062 | 0.052 | 0.056 | **0.049** |
| ICoseg | maxF | 0.812 | 0.838 | 0.846 | 0.851 | **0.899** | 0.884 | 0.875 | 0.871 | 0.860 | 0.872 | 0.868 | 0.890 |
| | MAE | 0.145 | 0.101 | 0.097 | 0.070 | 0.070 | 0.068 | 0.066 | 0.147 | 0.075 | 0.068 | 0.082 | **0.062** |

Table 3. Evaluation results on six dataset and with models DRFI [14], MDF [22], RFCN [40], DHS [26], Amulet [51], UCF [52], DCL [23], MSR [21], DSS [11], RA [4] and the deep unified CRF model. "+" marks the models utilizing Dense-CRF [19] for post-processing. "-" means that the corresponding dataset is used as the training data. The evaluation on MSRA-B is performed on the testing set.

| Method | F-measure |
|---|---|
| Baseline | 0.884 |
| Baseline + Dense-CRF (post-processing) [19] | 0.902 |
| Baseline + CRF (/w P) (backbone output) | 0.899 |
| Baseline + CRF (/w P) (CRF output) | 0.921 |
| Baseline + CRF (/w F) (CRF output) | 0.904 |
| Baseline + CRF (/w P & F) (CRF output) | 0.928 |

Table 4. F-measure of the estimated prediction map $\mathbf{o}^5$ by implementing pairwise terms in Eq. 2 to the deep unified CRF model for message-passing comparisons. "P" refers to message-passing between predictions, "F" means message-passing between features, and "/w P & F" means CRF with feature-feature, feature-prediction and prediction-prediction messages passing.

features, the F-measure is 0.904, with 2% increase to the baseline output. Finally, by adding feature-feature, feature-prediction and prediction-prediction messages passing, the F-measure further improves to 0.928.

Further, Figure 6 plots the training loss by the cascade CRFs architecture with and without feature-prediction message-passing respectively. Clearly, building connections between features and predictions facilitates more efficient model training. The running time of the cascade CRFs architecture is similar to DSS model with Dense-CRF, with the same parameter settings for Gaussian kernels, taking approximately 0.48s per image.

### 5.4. Cross Dataset Evaluation

For comprehensive analysis, the proposed deep unified CRF model is compared with ten state-of-the-art saliency models including DRFI [14], MDF [22], RFCN [40], DHS [26], Amulet [51], UCF [52], DCL [23], MSR [21], DSS [11], RA [4]. All the models are CNN-based approaches except the DRFI model. All the implementations are based on public codes and suggested settings by the corresponding authors. Table 3 lists the max F-measure and MAE of the ten saliency models and the proposed deep unified CRF model over six datasets. It is observed that the deep unified CRF model results in better F-measure and significantly reduced MAE. Compared to DCL$^+$ [23], MSR [21] and DSS$^+$ [11] that apply Dense-CRF [19] as a post-processing method, the proposed jointly trained cascade CRFs effectively improve the performance. Figure 7 presents saliency maps from the compared models and the proposed deep unified CRF model.

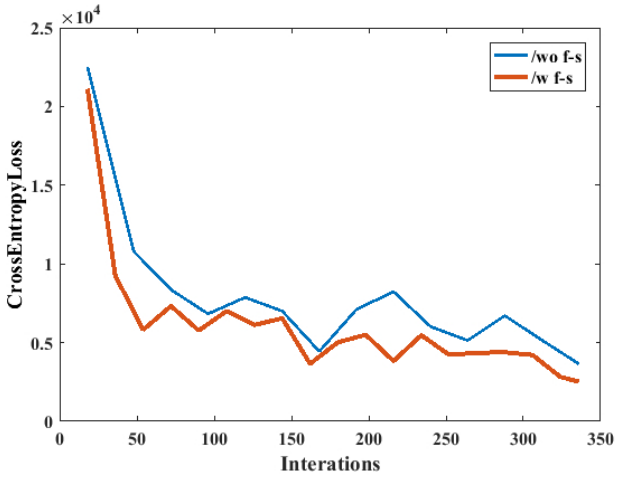



Figure 6. Training loss with ("/w f-s") and without ("/wo f-s") feature-prediction message-passing.

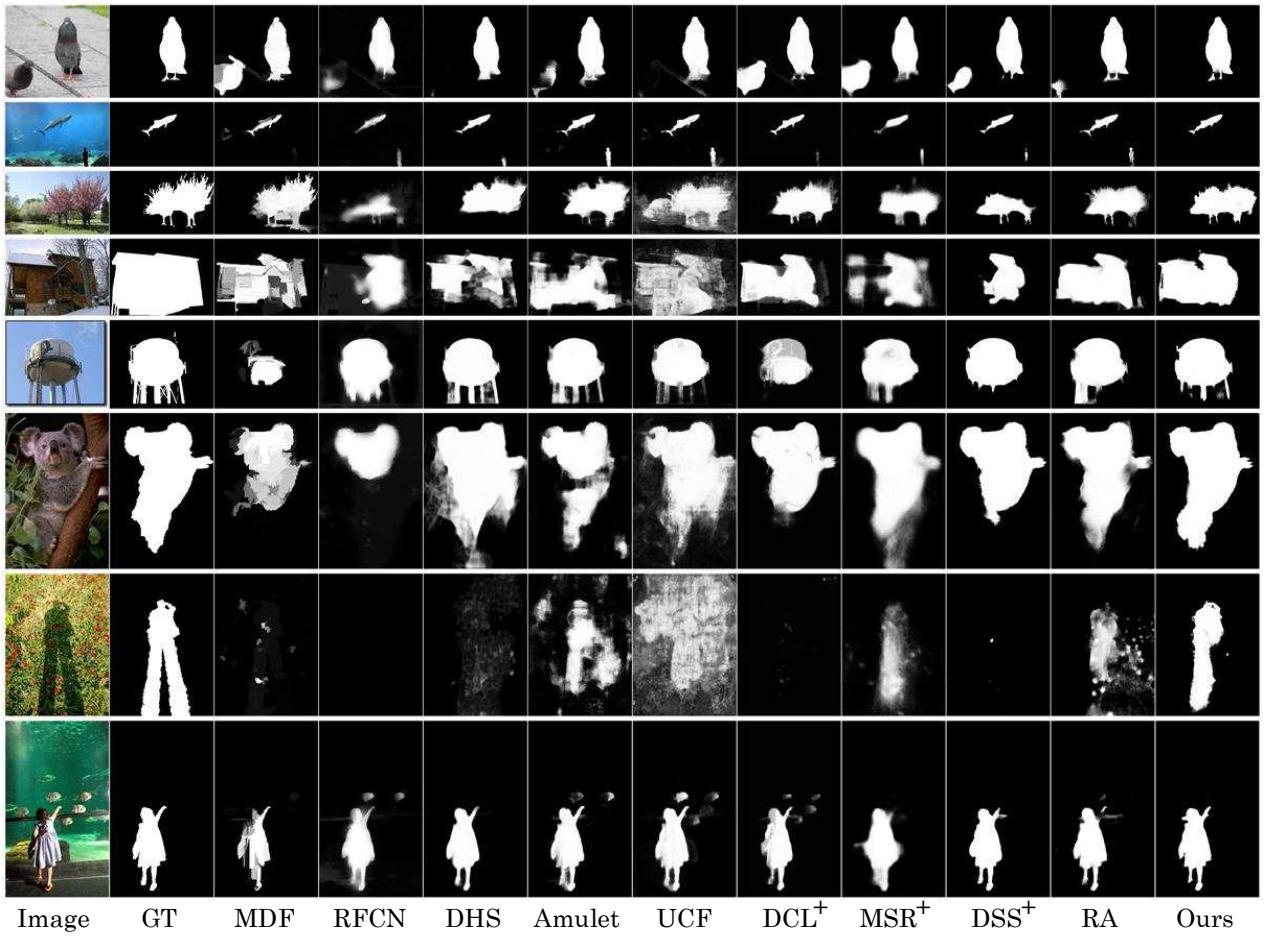



Figure 7. Examples of saliency maps from MDF [22], RFCN [40], DHS [26], Amulet [51], UCF [52], DCL [23], MSR [21], DSS [11], RA [4] and the proposed deep unified CRF model. "+" marks the models utilizing Dense-CRF [19] for post-processing.

### 5.5. Noise

Besides horizontal flipping for data augmentation, we explore other noise adding methods for robust model training. We add one CRF block with only prediction-prediction message-passing at scale 5 and jointly train with CNN for 10 epochs, and test the F-measure of the prediction map $\mathbf{o}^5$ with various noise adding methods, on ECSSD dataset. Data augmentation with horizontal flipping results in F-measure as 0.910.

Firstly, we enlarge the training sets with both the horizontal flipping and the vertical flipping, the F-measure slightly decreases to 0.905. This may because that the symmetry properties for salient objects mostly apply to horizontal directions. Also, we add noises to the images, *i.e.*, blurring, sharpening, Gaussian noise, and inversion, for data augmentation. The training takes much longer time and the F-measure is 0.896. Different from the detection and recognition tasks, the salient object detection tasks attach more importance to the smoothness of the resulted saliency maps. This may be the reason why additional augmentation does not result in a better model.

## 6. Conclusion

This paper presents a novel deep unified CRF model. Firstly, we jointly formulate the continuous features and the discrete predictions into a unified CRF, which provides explainable solutions for the features, the predictions, and the interactions among them. Secondly, we infer mean-field approximations for highly efficient message-passing, leading to distinct model formulation, inference, and neural network implementation. Experiments show that the proposed cascade CRFs architecture results in highly competitive performance and facilitates more efficient model training.

## References


[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Proc. CVPR*, pages 1597–1604. IEEE, 2009.

[2] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[3] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Proc. CVPR*, pages 3169–3176. IEEE, 2010.

[4] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *Proc. ECCV*, pages 234–250. Springer, 2018.

[5] Xiao Chu, Wanli Ouyang, Xiaogang Wang, et al. Crf-cnn: Modeling structured information in human pose estimation. In *NIPS*, pages 316–324, 2016.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[7] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *Proc. ECCV*, pages 186–202. Springer, 2018.

[8] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE TIP*, 19(1):185–198, 2010.

[9] Jungong Han, Eric J Pauwels, and Paul De Zeeuw. Fast saliency-aware multi-modality image fusion. *Neurocomputing*, 111:70–80, 2013.

[10] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2007.

[11] Qibin Hou, Ming-Ming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019.

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

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. Multimedia*, pages 675–678. ACM, 2014.

[14] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. CVPR*, pages 2083–2090. IEEE, 2013.

[15] Peng Jiang, Nuno Vasconcelos, and Jingliang Peng. Generic promotion of diffusion-based salient object detection. In *Proc. ICCV*, pages 217–225. IEEE, 2015.

[16] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Proc. ICCV*, pages 2106–2113. IEEE, 2009.

[17] Jongpil Kim and Vladimir Pavlovic. A shape-based approach for salient object detection using deep learning. In *Proc. ECCV*, pages 455–470. Springer, 2016.

[18] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[19] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[20] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *Proc. CVPR*, pages 3668–3677. IEEE, 2016.

[21] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *Proc. CVPR*, pages 247–256. IEEE, 2017.

[22] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proc. CVPR*, pages 5455–5463. IEEE, 2015.

[23] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *Proc. CVPR*, pages 478–487. IEEE, 2016.

[24] Guanbin Li and Yizhou Yu. Visual saliency detection based on multiscale deep cnn features. *IEEE TIP*, 25(11):5012–5024, 2016.

[25] Yin Li, Xiaodi Hou, Christian Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proc. CVPR*, pages 280–287. IEEE, 2014.

[26] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proc. CVPR*, pages 678–686. IEEE, 2016.

[27] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proc. CVPR*, pages 3089–3098. IEEE, 2018.

[28] Nian Liu, Junwei Han, Dingwen Zhang, Shifeng Wen, and Tianming Liu. Predicting eye fixations using convolutional neural networks. In *Proc. CVPR*, pages 362–370. IEEE, 2015.

[29] Qing Liu, Xiaopeng Hong, Beiji Zou, Jie Chen, Zailiang Chen, and Guoying Zhao. Hierarchical contour closure-based holistic salient object detection. *IEEE TIP*, 26(9):4537–4552, 2017.

[30] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2011.

[31] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proc. ICCV*. IEEE, 2019.

[32] Rotem Mairon and Ohad Ben-Shahar. A closer look at context: From coxels to the contextual emergence of object saliency. In *Proc. ECCV*, pages 708–724. Springer, 2014.

[33] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proc. CVPR*, pages 733–740. IEEE, 2012.

[34] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *Proc. ECCV*, pages 366–379. Springer, 2010.

[35] Kosta Ristovski, Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*, pages 840–846, 2013.

[36] Anthony Santella, Maneesh Agrawala, Doug DeCarlo, David Salesin, and Michael Cohen. Gaze-based interaction for semi-automatic photo cropping. In *Proc. CHI*, pages 771–780. ACM, 2006.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[38] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, pages 3360–3367. IEEE, 2010.

[39] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *Proc. CVPR*, pages 3183–3192. IEEE, 2015.

[40] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *Proc. ECCV*, pages 825–841. Springer, 2016.

[41] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *Proc. ECCV*, pages 29–42. Springer, 2012.

[42] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. ICCV*, pages 1395–1403. IEEE, 2015.

[43] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NIPS*, pages 3961–3970, 2017.

[44] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*, pages 5354–5362, 2017.

[45] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *IEEE TPAMI*, 41(6):1426–1440, June 2019.

[46] Yingyue Xu, Xiaopeng Hong, Xin Liu, and Guoying Zhao. Saliency detection via bi-directional propagation. *JVCI*, 53:113–121, 2018.

[47] Yingyue Xu, Xiaopeng Hong, Fatih Porikli, Xin Liu, Jie Chen, and Guoying Zhao. Saliency integration: An arbitrator model. *IEEE TMM*, 21(1):98–113, 2018.

[48] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proc. CVPR*, pages 1155–1162. IEEE, 2013.

[49] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proc. CVPR*, pages 3166–3173. IEEE, 2013.

[50] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *Proc. CVPR*, pages 1741–1750. IEEE, 2018.

[51] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proc. ICCV*, pages 202–211. IEEE, 2017.

[52] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *Proc. ICCV*, pages 212–221. IEEE, 2017.

[53] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *Proc. CVPR*, pages 714–722. IEEE, 2018.

[54] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *Proc. CVPR*, pages 1265–1274. IEEE, 2015.

[55] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proc. ICCV*, pages 1529–1537. IEEE, 2015.