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 Laboratory

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.

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...
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 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]
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 I of N pixels, suppose that a backbone CNN network computes L scales of deep feature maps \( F = \{ f^l \}_{l=1}^L \), where \( f^l = \{ f^l_{1,m} \}_{i=1,m=1}^{N,M} \) consists a set of M feature vectors. Accordingly, L scales of prediction maps \( S = \{ s^l \}_{l=1}^L \) can be computed, where \( s^l = \{ s^l_i \}_{i=1}^N \). The ground truth saliency map corresponding to the input image is denoted as \( g = \{ g_i \}_{i=1}^N \), and each element \( g_i \) 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 \( H = \{ h^l \}_{l=1}^L \) and the hidden multi-scale prediction maps \( O = \{ o^l \}_{l=1}^L \). In particular, at the l-th scale, the observed variables are the features \( f^{l-1} \), \( f^l \) and the prediction \( s^{l-1} \), and the objective is to estimate the corresponding \( h^l \) and \( o^l \). 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 \( o^L \) is the final saliency map.

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

\[
P(h^l, o^l | I, \Theta) = \frac{1}{Z(I, \Theta)} \exp \left\{ -E(h^l, o^l, I, \Theta) \right\}, \tag{1}
\]

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

\[
E = \sum_i \phi_h(h^l_i, f^l_i) + \sum_i \phi_o(s^l_i, o^l_i) + \sum_{i \neq j} \psi_h(h^l_i, h^l_{j}^{l-1}) + \sum_i \psi_h(o^l_i, o^l_{j}^{l-1}) + \sum_{i \neq j} \psi_o(o^l_i, o^l_{j}). \tag{2}
\]

The first term of Eq. 2 is a feature level unary term corresponding to an isotropic Gaussian: \( \phi_h(h^l_i, f^l_i) = -\frac{\alpha^2}{2} ||h^l_i - f^l_i||^2 \) where \( \alpha^2 > 0 \) is a weighting factor. The second term is a prediction level unary term, \( \phi_o(s^l_i, o^l_i) = ||s^l_i - o^l_i||^2 \).
The third term is a feature level pairwise term describing the potential between features, where
\[ \psi_h(h^{l-1}_i, h^{l-1}_j) = h^{l-1}_i W_{ij} h^{l-1}_j, \]  
where \( W_{ij} \in \mathbb{R}^{M \times M} \) is a bilinear kernel. The fourth term is a feature level pairwise term defining the potential between features and predictions, where
\[ \psi_{hs}(h^{l-1}_i, o^{l-1}_i) = h^{l-1}_i V_{ij} o^{l-1}_j, \]  
where \( V_{ij} \in \mathbb{R}^{M \times M} \) is also a bilinear kernel to couple the features and the predictions, \( o^{l-1} \) denotes a concatenation of \( M \) prediction maps \( o^{l-1} \). The fifth term is a prediction level pairwise term defining the potential between the predictions as follows:
\[ \psi_o(o^{l-1}_i, o^{l-1}_j) = \beta_1 K_{ij}^1 ||o^{l-1}_i - o^{l-1}_j||^2 + \beta_2 K_{ij}^2 ||o^{l-1}_i - o^{l-1}_j||^2. \]  

\( K_{ij}^1 \) and \( K_{ij}^2 \) 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
\[ K_{ij}^1 = \nu_1 \exp (-\frac{||p_i - p_j||^2}{2\sigma^2_i}) \]  
and \( K_{ij}^2 \) is the proximity kernel that measures the spatial relationship between two pixels as
\[ K_{ij}^2 = \nu_2 \exp (-\frac{||p_i - p_j||^2}{2\sigma^2_i}). \]  
\( \nu_1 \) and \( \nu_2 \) are the contributions of each Gaussian kernel, respectively.

**Inference.** We perform the mean-field approximation to estimate a distribution \( q(h^l, o^l | l, \Theta) = \prod_{i=1}^{N} q_{i,l}(h^l_i, o^l_i) \) that is an approximation to \( P(h^l, o^l | l, \Theta) \) by minimizing the Kullback-Leiber divergence [35]. By considering \( J_{i,l} = \log q_{i,l}(h^l_i, o^l_i | l, \Theta) \) and rearranging its expression into an exponential form, the mean-field updates can be derived as:
\[ \tilde{h}^l_i = \frac{1}{\alpha^l_i} \left( o^l_i + \sum_{l \neq l-1} \sum_{i \neq j} W_{ij} h^{l-1}_j \right) + \sum_{i \neq l-1} \sum_{i \neq j} V_{ij} h^{l-1}_j o^{l-1}_j, \]  
\[ \rho^l_i = 1 + 2 \left( \beta_1 \sum_{j \neq i} K_{ij}^1 + \beta_2 \sum_{j \neq i} K_{ij}^2 \right), \]  
\[ \mu^l_i = \frac{o^l_i}{\rho^l_i} + 2 \left( \beta_1 \sum_{j \neq i} K_{ij}^1 \mu^l_j + \beta_2 \sum_{j \neq i} K_{ij}^2 \mu^l_j \right). \]  

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 \( o^l \) 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 \( \mu^l_i \) can be approximated. After \( T \) mean-field iterations, the estimated prediction map \( \mu^l_i \) is regraded as the estimation of \( o^l \) from the CRF at the \( l \)-th scale. The details of the mean-field updates are presented in Figure 4.

In the cascade flow, the observation \( s^l \) is obtained via integrating the prediction map \( s^l \) and the estimated map \( o^{l-1} \) from the CRF at the previous scale, i.e., \( s_i^l = s_i^l + o_i^{l-1} \).

**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 2,000 testing images. The images are resized to 240×320 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 “F_{l-6}^{l-1}” correspond to “pool5a”, “conv5_3”, “conv4_3”, “conv3_3” and “conv2_2” in the enhanced HED [42] structure, while the corresponding prediction maps “o_{l-6}^{l-1}” 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 $\varepsilon = \{\epsilon_i\}_{i=1}^5$ 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:

$$L_{\text{Stage1}}(\Delta, \varepsilon_i) = -\sum_{i=1}^{N} \left( g_i \log(s_i) + (1 - g_i) \log(1 - s_i) \right),$$

where $g_i$ is the $i$-th ground-truth label and $s_i$ is the $i$-th pixel on the prediction map at the $l$-th scale. $N$ 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:

$$L_{\text{Stage2}}(\Delta, \varepsilon, \beta) = -\sum_{i=1}^{N} \left( g_i \log(o_i) + (1 - g_i) \log(1 - o_i) \right).$$

**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.

**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 3000 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:

$$F = \frac{(1 + b^2) \cdot \text{precision} \cdot \text{recall}}{b^2 \cdot \text{precision} + \text{recall}},$$

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...
the overall pixel-wise difference between the saliency map $\text{sal}$ and the ground truth $\text{gt}$:

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

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 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 $\sigma_\beta$, while the proximity kernel is controlled by $\sigma_\gamma$ and the weight $\nu_1$, while the proximity kernel $K^2$ 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

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

<table>
<thead>
<tr>
<th>$T$</th>
<th>$\sigma_\alpha$</th>
<th>$\sigma_\beta$</th>
<th>$\sigma_\gamma$</th>
<th>$\nu_1$</th>
<th>$\nu_2$</th>
<th>F-measure</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.892</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.892</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.909</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.896</td>
<td>0.084</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.893</td>
<td>0.070</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>0.892</td>
<td>0.070</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.910</td>
<td>0.094</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0.894</td>
<td>0.069</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>0.896</td>
<td>0.095</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>0.892</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>0.892</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.892</td>
<td>0.071</td>
</tr>
</tbody>
</table>

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 $\sigma_\beta$, while the proximity kernel is controlled by $\sigma_\gamma$ and the weight $\nu_1$, while the proximity kernel $K^2$ 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

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>DRFI</th>
<th>MDF</th>
<th>RFCN</th>
<th>DHS</th>
<th>Amulet</th>
<th>UCF</th>
<th>DCL+</th>
<th>MSR+</th>
<th>DSS</th>
<th>DSS+</th>
<th>RA</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRA-B</td>
<td>maxF</td>
<td>0.851</td>
<td>0.885</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.916</td>
<td>0.930</td>
<td>0.920</td>
<td>0.928</td>
<td>0.931</td>
<td>0.935</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.123</td>
<td>0.066</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.047</td>
<td>0.042</td>
<td>0.043</td>
<td>0.035</td>
<td>0.036</td>
<td>0.029</td>
<td></td>
</tr>
<tr>
<td>PASCAL-S</td>
<td>maxF</td>
<td>0.690</td>
<td>0.759</td>
<td>0.829</td>
<td>0.824</td>
<td>0.832</td>
<td>0.818</td>
<td>0.822</td>
<td>0.852</td>
<td>0.826</td>
<td>0.831</td>
<td>0.829</td>
<td>0.858</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.210</td>
<td>0.142</td>
<td>0.118</td>
<td>0.094</td>
<td>0.100</td>
<td>0.116</td>
<td>0.108</td>
<td>0.081</td>
<td>0.102</td>
<td>0.093</td>
<td>0.101</td>
<td>0.089</td>
</tr>
<tr>
<td>DUT-OMRON</td>
<td>maxF</td>
<td>0.644</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
<tr>
<td>HKU-IS</td>
<td>maxF</td>
<td>0.775</td>
<td>0.860</td>
<td>0.894</td>
<td>0.892</td>
<td>0.897</td>
<td>0.888</td>
<td>0.904</td>
<td>0.916</td>
<td>0.900</td>
<td>0.916</td>
<td>0.913</td>
<td>0.920</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.146</td>
<td>0.129</td>
<td>0.088</td>
<td>0.052</td>
<td>0.051</td>
<td>0.061</td>
<td>0.049</td>
<td>0.039</td>
<td>0.050</td>
<td>0.040</td>
<td>0.045</td>
<td>0.039</td>
</tr>
<tr>
<td>ECSSD</td>
<td>maxF</td>
<td>0.784</td>
<td>0.847</td>
<td>0.899</td>
<td>0.907</td>
<td>0.914</td>
<td>0.902</td>
<td>0.901</td>
<td>0.913</td>
<td>0.908</td>
<td>0.921</td>
<td>0.921</td>
<td>0.928</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.172</td>
<td>0.106</td>
<td>0.091</td>
<td>0.059</td>
<td>0.061</td>
<td>0.071</td>
<td>0.068</td>
<td>0.054</td>
<td>0.062</td>
<td>0.052</td>
<td>0.056</td>
<td>0.049</td>
</tr>
<tr>
<td>ICoSseg</td>
<td>maxF</td>
<td>0.812</td>
<td>0.838</td>
<td>0.846</td>
<td>0.851</td>
<td>0.899</td>
<td>0.884</td>
<td>0.875</td>
<td>0.871</td>
<td>0.860</td>
<td>0.872</td>
<td>0.868</td>
<td>0.890</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.145</td>
<td>0.101</td>
<td>0.097</td>
<td>0.070</td>
<td>0.070</td>
<td>0.068</td>
<td>0.066</td>
<td>0.147</td>
<td>0.075</td>
<td>0.068</td>
<td>0.082</td>
<td>0.062</td>
</tr>
</tbody>
</table>

Table 4. F-measure of the estimated prediction map $\hat{y}^5$ 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.884</td>
</tr>
<tr>
<td>Baseline + Dense-CRF (post-processing) [19]</td>
<td>0.902</td>
</tr>
<tr>
<td>Baseline + CRF (/w P) (backbone output)</td>
<td>0.899</td>
</tr>
<tr>
<td>Baseline + CRF (/w P) (CRF output)</td>
<td>0.921</td>
</tr>
<tr>
<td>Baseline + CRF (/w F) (CRF output)</td>
<td>0.904</td>
</tr>
<tr>
<td>Baseline + CRF (/w P &amp; F) (CRF output)</td>
<td>0.928</td>
</tr>
</tbody>
</table>

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] 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.

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.
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 $o^5$ 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 be 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


