Saliency Pattern Detection by Ranking Structured Trees

Lei Zhu[1,2], Haibin Ling[2,3,*], Jin Wu[1], Huiping Deng[1], Jin Liu[1]
[1]School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[3]Meitu HiScene Lab, HiScene Information Technologies, Shanghai, China
zhulei@wust.edu.cn, hbling@temple.edu, {wujin, denghuiping, liujin}@wust.edu.cn

Abstract

In this paper we propose a new salient object detection method via structured label prediction. By learning appearance features in rectangular regions, our structural region representation encodes the local saliency distribution with a matrix of binary labels. We show that the linear combination of structured labels can well model the saliency distribution in local regions. Representing region saliency with structured labels has two advantages: 1) it connects the label assignment of all enclosed pixels, which produces a smooth saliency prediction; and 2) regular-shaped nature of structured labels enables well definition of traditional cues such as regional properties and center surround contrast, and these cues help to build meaningful and informative saliency measures. To measure the consistency between a structured label and the corresponding saliency distribution, we further propose an adaptive label ranking algorithm using proposals that are generated by a CNN model. Finally, we introduce a K-NN enhanced graph representation for saliency propagation, which is more favorable for our task than the widely-used adjacent-graph-based ones. Experimental results demonstrate the effectiveness of our proposed method on six popular benchmarks compared with state-of-the-art approaches.

1. Introduction

Saliency detection aims to annotate the most attractive regions in a scene. An accurate saliency detection method is able to recommend regions that are informative, attentive and above all, it implies the presence of prototype of objects [12]. Compared with the research focusing on eye fixation and eye movement tracking, the work on *Saliency Object Detection* is more popular in the computer vision community as it is designed for general salient object discovery from an image [33]. Saliency object detection has been introduced in many tasks such as segmentation [54], image retrieval [6], and object recognition [46].

Since the first computational attention model [17] was published, the interest in saliency detection related research has increased rapidly. Intuitively, a salient object should visually stands out from its surroundings [23]. Following this idea, it is natural to compute the saliency of a pixel/region by the center-surround contrast [13, 14, 17, 23]. Features for describing pixels/regions can be empirically defined, or more comprehensively, can be learned by regressors such as Random Forest, SVM, and Boosting-based approaches [18, 21, 31, 50]. Very recently, deep-learning-based approaches have stepped into this field and demonstrated the broad prospects of such data driven models [26, 29, 55, 59].

Although the main stream in this field works on contrast definition and feature selection for improvement, there still are problems in existing pixel-based and segment-based region representations. Pixel-based approaches compute the saliency of a pixel as the contrast between a center and a surrounding regions. As the pixel-wise prediction ignores the relationship between pixels, inner regions of a proto-object may take very different salient values. Segment-based regions usually refer to superpixels, which are helpful for smoothing the predictions of nearby pixels with similar appearance. However, features that are extracted from superpixels may result in less informative saliency measures [53].

In this paper, we propose a new saliency detection method that takes the structural representation of rigid grids as receptive fields. Inspired by the work of edge pattern [9], we learn a *Structured Saliency Pattern* (SSP) that parses the saliency assignment of a local rectangular region. More specifically, SSP captures the spatial distribution of binary labels of a local region (patch) and indicates the foreground/background attribute in a region. During training, we learn SSP by saliency cues such as regional properties and center-surround contrast. During prediction, we combine a small number of SSPs to vote for pixel-level saliency in a region with arbitrary appearance.

*Corresponding author.

Previous local contrast-based model typically annotates all the regions that fit the center-surrounding structure, including isolated background regions. To address this issue, we propose using SSP for regularization by integrating the global semantic information derived from a CNN model. Finally, a K-NN enhanced graph-based saliency propagation is developed to refine the saliency map.

In a nutshell, the main contributions of this paper are

1) We explore a new structural representation of pixel-level saliency over a local rectangular region. The representation maps appearance features to matrices of binary labels. On the one hand, it captures more informative features from non-homogeneous regions than superpixels. On the other hand, it considers the connection among adjacent pixels and thus smooths the prediction.
2) Region saliency is computed as a weighted combination of structured labels. We propose an adaptive ranking method to decide the most appropriate labels that representing the local saliency distribution.
3) We propose a K-NN enhanced graph for saliency propagation, which considers neighboring relationships in both spatial and feature spaces.

2. Previous work

Saliency detection usually utilizes one or both of two human visual attention processes [42]. Bottom-up attention refers to detect salient objects from the perceptual data that only comes from images itself. By contrast, top-down attention is influenced by the experience, the goal, or the current mental state of the agent [60]. Although top-down attention is very important for perception, it is not always possible to obtain prior knowledge. This makes the bottom-up saliency more popular recently. Related applications mainly work on two aspects: eye fixation prediction and object segmentation, and the latter is concerned in this paper.

Among those bottom-up saliency methods, the Feature Integration Theory (FIT) [51] serves as the basis for many biologically motivated models. It suggests a pixel-based, multi-channel, parallel conspicuous maps computation and a fusion system. Follow this idea, Itti *et al*. [17] first propose a saliency model for computing the center-surround contrast and searching the local maximum response in the multi-scale DoG space. Harel *et al*. [15] extend it and compute the saliency in a graph-based way. The performance of the FIT model is further improved by integrating different feature channels and different contrast metrics [24, 37, 47]. There are also some variants and simplified versions of the FIT model, for example, using single scale or adopting different surround definition [3, 34, 48, 53]. Those methods usually suffer from two problems: first, since every pixel is evaluated independently, nearby pixels with the same semantics may take very different salient values due to the variation of local context. Second, it is very hard to achieve precise saliency assignment around the foreground/background boundary when edge information is absent.

To address the above issues, research on segment-based saliency becomes very popular by using superpixel extraction [2, 10]. Since hundreds of superpixels are usually sufficient, the local and global segment-based contrast can be computed efficiently [7,11,41,44]. The small number of superpixels benefits the application of graph-based techniques on saliency detection [4, 38, 43, 57, 58]. There are however two problems of segment-based methods: first, features from homologous region may result in less informative saliency measures. Some research addresses the problem by learning the effective features via traditional regressors [18, 21, 31, 50], or Deep Convolutional Neuronal Network (DCNN) [26, 29, 30, 59]. Note that, due to the input requirement of such DCNN models, deep features are actually extracted from a larger rectangular region centered at each segment rather than itself. Second, it is difficult for segment-based saliency models to construct a rigid center-surround structure for contrast computation. Existing approaches usually build such structure using segments and their immediate neighbors [28, 45, 58], or by grouping segments according to their visual properties [39, 60]. However, the shape and size of edge-preserving segments are very sensitive to the image content, which always produces irregular center-surround structure with different scales.

3. The proposed saliency model

Fig. 1 shows the flowchart of our proposed model. It generates candidate proposals that may enclose the salient objects as shown in Fig. 1a. Then, as shown in Fig. 1b, pixel-wise saliency is voted by predicting and ranking SSP (see Section 3.1) in each proposal. Finally, a K-NN enhanced graph-based saliency diffusion method is used to refine the saliency map as shown in Fig. 1c.

3.1. Salient region representation

From the perspective of perceptual psychology, people seeks interesting regions in an image by analyzing high-level features in a relatively large receptive field [12]. In this sense, structural information is very appropriate in describing local appearance on the semantic level [25].

Let $\mathcal{R}$ refer to a rectangular region of size $d \times d$ and $\mathbf{x}$ refer to the appearance features defined on $\mathcal{R}$. We determine the probability of saliency assignment $P(\mathbf{s}|\mathbf{x})$ of all enclosed $n = d^2$ pixels: $p_i \in \mathcal{R}, i = 1, 2, ..., n$, where $\mathbf{s} = [s_1, s_2, ..., s_n]$ and s_i refers to the salient value of p_i. We assume that $P(\mathbf{s}|\mathbf{x})$ can be represented by a linear combination of T binary vectors as:

$$P(\mathbf{s}|\mathbf{x}) = \sum_j w_j \cdot \boldsymbol{l}_j, \quad j = 1, 2, ..., T \tag{1}$$

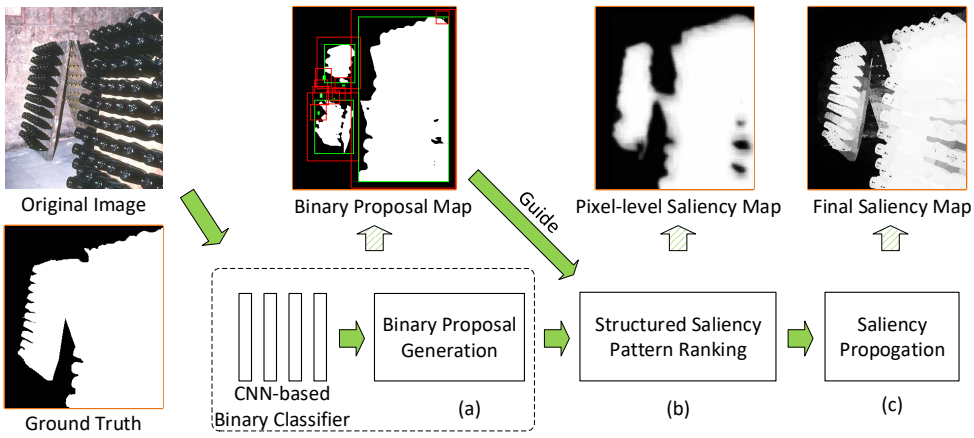


Figure 1: Flowchart of the proposed algorithm. (a) A CNN model is trained to generate a binary proposal map. The bounding box of each connected component is taken as the proposal. (b) SSPs are predicted in each proposal and are further combined to vote for the pixel-level saliency. (c) The saliency map is refined by K-NN enhanced graph-based saliency propagation.

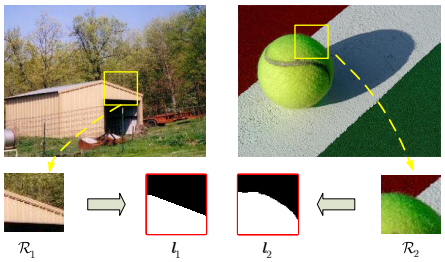


Figure 2: Regions with different appearance patterns share similar saliency patterns.

where $\boldsymbol{l}_j = [l_j^1, l_j^2, ..., l_j^n], l_j^i \in \{0, 1\}$ and w_j refers to the weight of binary vector $\boldsymbol{l}_j$. Since $\mathcal{R}$ is rectangular, $\boldsymbol{l}_j$ can be reshaped to a matrix of size $d \times d$. This representation enables an illustration of the spatial distribution of binary elements in $\boldsymbol{l}_j$. Therefore $\boldsymbol{l}_j$ is a kind of *structured label* and two examples $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are shown in Fig. 2.

Intuitively, a certain $\boldsymbol{l}_j$ can be a representative for the saliency structure of many different image patches if considering their pixel-level salient values as binary. Taking Fig. 2 for example, $\mathcal{R}_1$ and $\mathcal{R}_2$ have very different appearance but their spatial saliency distributions $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are similar. Therefore, we take $\boldsymbol{l}_j$ as a common *saliency pattern* which is capable of describing saliency assignment for a large number of image regions. The explanation may be that, the binary segmentation in local region depends on the crossing edge, which exhibits some forms of local structure.

Since $\boldsymbol{l}_j$ encodes the saliency structure of image patches, we call it a *Structured Saliency Pattern* (SSP). In this paper, we employ the structured trees [9] for learning SSPs.

3.1.1 Learning SSP by structured decision trees

As shown in Fig. 3, we learn SSP from the images with human-labeled ground truths. Let $\boldsymbol{z}_i$ refer to a ground-truth patch of size $d \times d$ and $\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_M\}$. Our goal is to learn several decision trees for extracting SSPs from $\mathcal{Z}$. In a standard decision tree, samples with a single label are routed by the split function in non-leaf nodes. However, traditional node splitting optimization is impractical for $\boldsymbol{z}_i$, since it is very expensive to investigate all possible label distributions when the output space is high dimensional.

As suggested by [9], we employ an efficient PCA-based method to map $\boldsymbol{z}_i$ into a binary variable b_i. Let $Z \subset \mathcal{Z}, Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m\}$ refer to m ground-truth patches that reach a node. We first represent Z as a matrix where each row of Z is a vectorized $\boldsymbol{z}_i$, then compute the mapping as:

$$Z_s = W * \mathrm{PCA}(W^{\mathsf{T}}), \quad W = Z - \frac{1}{m}\sum_{i=1}^{m} Z(i,:)$$
$$b_i = \begin{cases} 1, & \text{if } Z_s(i,1) > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $Z(i,:)$ refers to the i^{th} row of Z. At each leaf node, we define an SSP as the most representative one $\boldsymbol{z}_{\text{SSP}}$ that summarizes the spatial distributions of all $\boldsymbol{z}_i$ as

$$\boldsymbol{z}_{\text{SSP}} = Z(p,:), \quad p = \underset{i}{\operatorname{argmin}} \Big(\sum_{j=1}^{n} [W]^2(i,j) \Big), \tag{3}$$

where $[\cdot]^2$ refers to the element-wise square. At every split node, a split function is optimized to split a subset of $\mathcal{Z}$ depending on each $\boldsymbol{z}_i$ and its corresponding features.

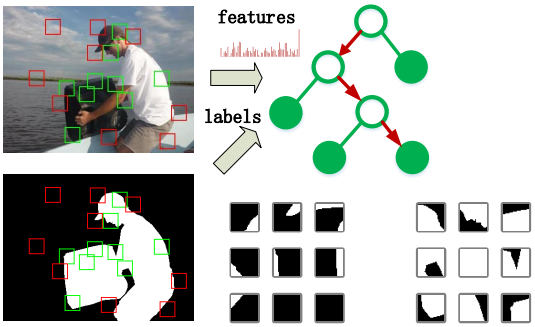


Figure 3: Rectangular samples are randomly extracted from an image and its ground truth. We selected features from samples for learning structured trees. During the test stage, SSPs are collected from the leave of these trees.

3.1.2 Feature selection for SSP

For feature selection, we first include following inner-region attributes: 1) color-based features, including the mean and variance of components in four color spaces (RGB, LAB, HSV and OPPONENTS), the hue and saturation histograms, 2) gradient-based features, including the average gradient magnitude in X and Y directions, the histograms of orientated gradient and difference of Gaussian descriptor, 3) the average edge intensity and focusness [19], and 4) the normalized center coordinates of regions.

We further use contrast-based features to measure the regional prominence. As shown in Fig. 4, those features are extracted from the structure comprising a region R_s that surrounds a center region R_c. Specifically, the contrast-based features are: 1) all features except the center coordinates, the hue and saturation histograms that are used for measuring inner-region attributes, 2) histogram-based features in the aforementioned four color spaces, 3) texture features, including the LM filter [27], Gabor filter and LBP. Two metrics are used for computing feature contrast: the χ^2 distance for histogram-based features, and the dimensional-wise absolute difference for other features.

At last, we introduce the pseudo-background assumption that is proved to be effective recently [18,21,50]. As shown in Fig. 4, we take four boundaries of an image as the background region R_b and use above regional contrast features to measure the differences between R_c and R_b.

In our method, multi-scale features are used in both training and prediction. This is achieved by resizing an image to several scales rather than changing the size of SSP.

3.2. SSP ranking by binary proposals

Since an SSP detects local prominence, predicting an SSP for all pixels in an image may highlight locally pop-out background regions. Additionally, millions of predictions are required at every pixel in a sliding window manner,

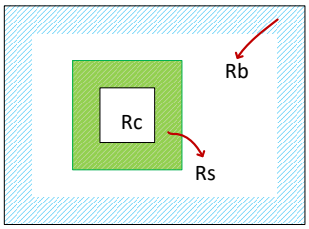


Figure 4: Illustration of three different types of regions.

which is very time consuming. In practice, it is fortunately unnecessary to predict all pixels. We observe that salient regions roughly occupy 10% to 40% of pixels in an image on average. Therefore, we only search several regions rather than the whole image if high quality proto-object proposals are available. One possible solution is to use the object proposal [8, 52, 61]. However, not all objects in an image are salient. Moreover, those approaches tend to enclose large object or cluster of objects rather than small ones.

To generate high quality proposals, we first learn a two-class classifier to obtain pixel-level foreground/background labels. It produces a binary map that indicates the rough locations of salient objects. We call this map as a *Binary Proposal Map* (Fig. 5a). Then a proposal is defined as an enlarged box from the bounding box of each connected component in the binary proposal map. The gap between two boxes is fixed to the size of SSP. On one hand, it ensures that every pixel in the box receives predictions from a receptive field with same size. On the other hand, it allows us to extract SSPs from the components that are smaller than a SSP even they contain only one pixel. In our model, we employ RefineNet [35] to generate the binary proposal map. RefineNet proposes a refinement network based on ResidualNet [16]. It fuses multi-resolution feature maps at each down-sampling layer recursively, thus avoids large memory cost in generating the high-dimensional and high-resolution feature maps [5]. In our method, we change the output representation of RefineNet to fit the binary prediction.

As introduced in Eq. 1, the salient values of all pixels in each sliding window of a proposal are the weighted combination of selected SSPs. A higher w_j in Eq. 1 implies that the corresponding SSP is in more accordance with the real saliency distribution than others in the window. We compute w_j adaptively by considering both the output of all structured trees and the segmentation result of the CNN classifier. As shown in Fig. 5b, let s refer a patch cropped from the segmentation result and $\boldsymbol{l}_j, j = 1, 2, ..., T$ refer to the SSP given by the j^{th} tree of total T trees at the same location with s. We compute w_j as follows:

$$w_j = \frac{1}{N} \exp\Big(- \frac{1}{k} \cdot L_j^2 / \operatorname{Var}(L_j) \Big), \tag{4}$$

where $\operatorname{Var}(\cdot)$ refers to the variance, N is the normalization

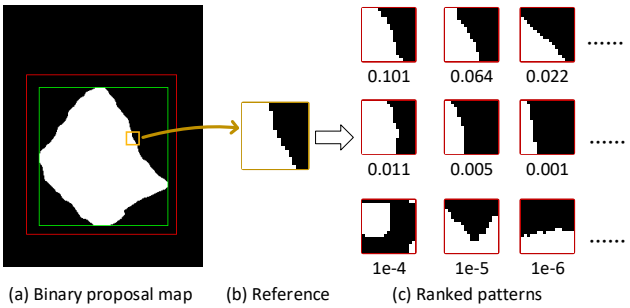


Figure 5: SSP ranking. (a) A binary proposal map generated by the CNN model. (b) A sample patch cropped from (a). (c) SSPs ranked according to the reference in (b), with the calculated weights shown under each SSP.

factor ensuring $\sum_j w_j = 1$, k is a damping constant and we set $k = 0.6$ in our experiments, and L_j is defined as the loss of each SSP, which contains two terms:

$$L_j = \alpha \left| \boldsymbol{l}_j - \frac{1}{T} \sum_j \boldsymbol{l}_j \right| + (1-\alpha) \left| \boldsymbol{l}_j - s \right|. \quad (5)$$

The first term refers to the loss towards the general averaged labels predicted by all trees. Since each tree is trained independently, the diversity of predictions makes the averaged labels more likely to the correct one. The second term refers to the loss towards the segmentation of CNN model. It gives large weight to the predicted label that is in accordance with the CNN's output. For computation efficiency, the absolute difference is adopted to calculate both losses. α balances two losses and we set it as 0.4 in our experiments. Fig. 5c shows several SSPs with descending weights.

3.3. Saliency propagation

Taking some high confidential salient regions as 'seed', most saliency propagation methods require an adjacent similarities to distribute saliency mass to similar nearby regions along graph edges [32, 38, 49, 58]. In our method, we propose a new graph representation by combining adjacent and Nearest Neighbor (NN) similarities. Introducing NN connection enables the exchange of saliency mass between similar regions regardless their spatial connectivities. It helps to obtain a balance saliency assignment on separated objects.

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ indicate segments of an image, which are obtained by superpixel extraction [2]. We construct a weighted graph model $G = \{\mathcal{X}, \mathbf{E}\}$ where $\mathbf{E} = \{e_{ij}\}$ and e_{ij} is the edge between x_i and x_j. We define $\mathbf{E}$ as a fusion of two graphs with different types:

$$\mathbf{E} = (1-\beta) \cdot W_a + \beta \cdot W_k, \quad (6)$$

where β balances two graphs W_a and W_k. We set it to 0.8 in the experiments. W_a refers to the adjacent similarities as $W_a(i,j) = \exp\left(-\left(|c_i - c_j|_2/\sigma_1\right)^2\right) \cdot a_{i,j}$, where c_i is the feature of x_i in CIE-LAB color space and σ_1 controls the fall-off rate of color distance. $a_{i,j}$ is a binary matrix and $a_{i,j} = 1$ if x_i is spatially connected to x_j. We model this distance as a normal distribution and adaptively set $\sigma_1 = \max_{i,j}\left(|c_i - c_j|_2\right)/3$ following the *Three Sigma Rule*.

W_k refers to the nearest neighbor similarities. Here we represent it as a K-NN similarity matrix:

$$W_k(i,j) = \begin{cases} \exp\left(-\left(|c_i - c_j|_2/\sigma_2\right)^2\right), & j \in \mathcal{N}_i, \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $\mathcal{N}_i$ refers to the K-neighbor system of segment x_i in the CIE-LAB color space. σ_2 controls the descending rate of each color channel in similarity estimation. To ensure that saliency mass can be smoothly transferred along graph edges based on the similarities, σ_2 can be identified by minimizing the following reconstruction error [20]:

$$\underset{\sigma_2}{\operatorname{argmin}} \sum_{i=1}^{n} ||c_i - \frac{1}{d_{ii}} \cdot \sum_{j \in \mathcal{N}_i} W_k(i,j) \cdot c_j||^2, \quad (8)$$

where $d_{ii} = \sum_j W_k(i,j)$ refers to the elements on the diagonal of the degree matrix of $W_k(i,j)$.

Saliency propagation can be formulated as a standard semi-supervise learning task once the graph $\mathbf{E}$ is specified. Following [4], we solve the following quadratic energy model to obtain the salient value y_i for segment x_i:

$$\underset{y_i}{\operatorname{argmin}} \sum_i k_{ii}(y_i - v_i)^2 + \frac{1}{2} \sum_{i,j} \mathbf{E}(i,j)(y_i - y_j)^2, \quad (9)$$

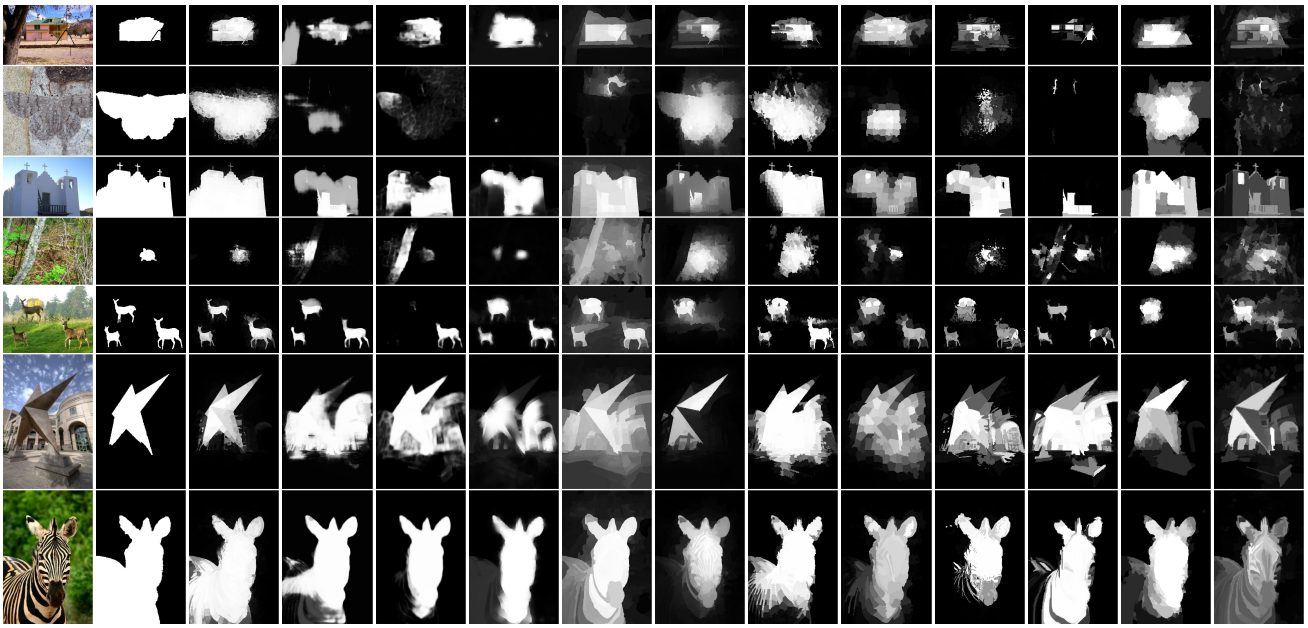
where $v_i \in \mathbf{v}$ is the averaged salient value of all pixels in segment x_i. Similar to d_{ii}, $k_{ii} = \sum_j \mathbf{E}(i,j)$ refers to the elements on the diagonal of degree matrix $\mathbf{K} = \operatorname{diag}\{k_{11}, k_{22}, ..., k_{nn}\}$. Eq. 9 has a close form solution:

$$y_i = (2 \cdot \mathbf{K} - \mathbf{E})^{-1} * (\mathbf{K} * \mathbf{v}), \quad (10)$$

which can be effectively computed under the condition of only hundreds of segments. After y_i is obtained, we upsample the saliency map using bilateral filtering [44] to get the finial pixel-level saliency map.

4. Experiment results

We evaluate the proposed method on six popular benchmarks that are widely used for saliency object detection: **THUR15K** [7], **DUT-OMRON** [58], **ECSSD** [49], **PASCAL-S** [33], **SOD** [18] and **TCD** [53]. Our proposed method is compared with 11 recently published approaches, including: DRFI [18], GRAB [57] and other nine deep-learning-based methods: MDF [29], MCDL [59],



(a) image (b) GT (c) ours (d) DCL (e) DHS (f) RFCN (g) SSD (h) GRAB (i) ELD (j) MTD (k) MCDL (l) MDF (m) LEGS (n) DRFI

Figure 6: Visual comparison of all evaluated method. From left to right: original image, ground truth, our approach, DCL, DHS, RFCN, SSD, GRAB, ELD, MTD, MCDL, MDF, LEGS and DRFI.

LEGS [55], ELD [26], MTD [32], DCL [30], DHS [36], RFCN [56] and SSD [22]. Note that, some approaches are not evaluated on certain benchmarks due to two reasons: 1) neither saliency maps nor codes are available, including: evaluations of SSD on THUR15K and TCD; evaluations of MTD on TCD. Additionally, evaluations of GRAB are only available on ECSSD. 2) Benchmarks are included in training, such as DHS and LEGS use partial images of DUTOMRON and PASCAL-S in their training sets, respectively. Our source code is made publicly available at: `https://github.com/zhulei2016/RST-saliency/`.

4.1. The setup of evaluations

In this section, we explain the experimental setup and the selection of the parameters in our system. For a fair comparison, we select only **THUS10K** [7] as the training set. The pretrained ResNet-101 layers are chosen as the basis of the RefineNet and we transfer its representations to fit the binary classification task. Two learning rates are adopted in our training process, that is, 5×10^{-5} for the first 160 epochs and 5×10^{-6} for another 100 epochs. We did not fine-tune RefineNet specifically and other parameters are kept as the default values.

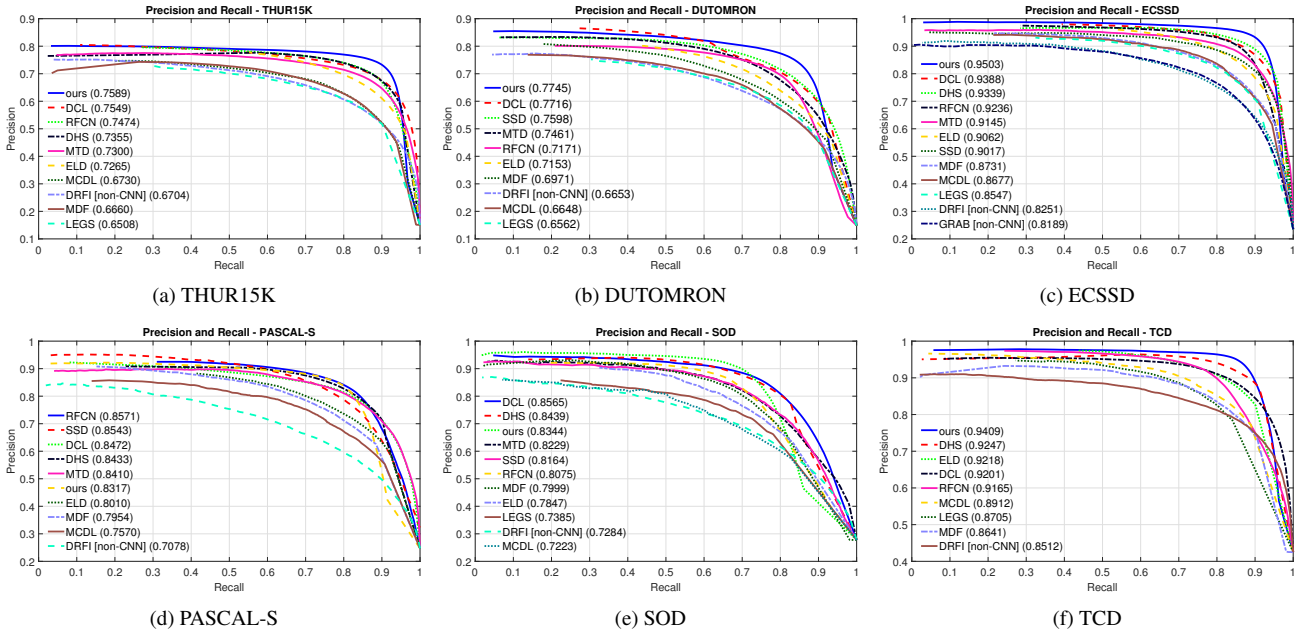
For learning SSP, totally 200 structured trees are assembled in parallel. Each tree has a maximum depth of 64 and a minimum 8 children in each node. Gini impurity is chosen as the measure in node split function. The training samples are image patches of size 17×17 and three kinds of training samples are collected: 1) *positive patches*, at least 50% of pixels in it are salient; 2) *weak positive patches*, the percentage of salient pixels is in the range of $(0, 50\%)$; 3) *negative patches*, all pixels belong to the background. We randomly select 750k positive, 750k weak positive and 1500k negative patches from **THUS10K**. To enclose multi-scale information, every image is rescaled according to factors $\{1, 0.75, 0.5\}$ before sample collection.

4.2. Visual comparison of salient maps

We first compare the results of all evaluated methods qualitatively. As shown in Fig. 6, we select several typical examples for demonstrating the robustness of our approach. Form top to bottom, those examples include images with: 1) clutter background, 2) salient object with monotonous color, 3) large salient objects, 4) small salient objects, 5) multiple salient objects, 6) salient object with low contrast, 7) salient region touches image boundaries. The results illustrate that our approach achieves higher visual consistency with the ground truth compared with other methods.

4.3. Segmentation by fixed thresholds

Precision versus Recall (PR) curve measurement is a straightforward way for evaluating saliency models via testing the segmentation precision over all possible thresholds. For every test image, we first normalize each saliency map



(a) THUR15K (b) DUTOMRON (c) ECSSD

(d) PASCAL-S (e) SOD (f) TCD

Figure 7: PR curves for all evaluated methods using fixed thresholds on six popular saliency datasets. All methods are sorted by descending Area Under Curve (AUC) (this figure is best viewed in color).

to the range of $[0, 255]$ then test all available integer to get the binary map. The PR values are calculated by comparing the binary map with the human-labeled ground truth. Finally, we average PR values over all test images to get an overall evaluation. Figure 7 shows PR curves measurement for all evaluated methods on six popular saliency datasets.

The result shows that our method outperforms others with clear margins on THUR15K, DUTOMRON, ECSSD and TCD. Our method is still superior to others in the 70% recall range on SOD. While on PASCAL-S, our method is less effective than several methods in this evaluation.

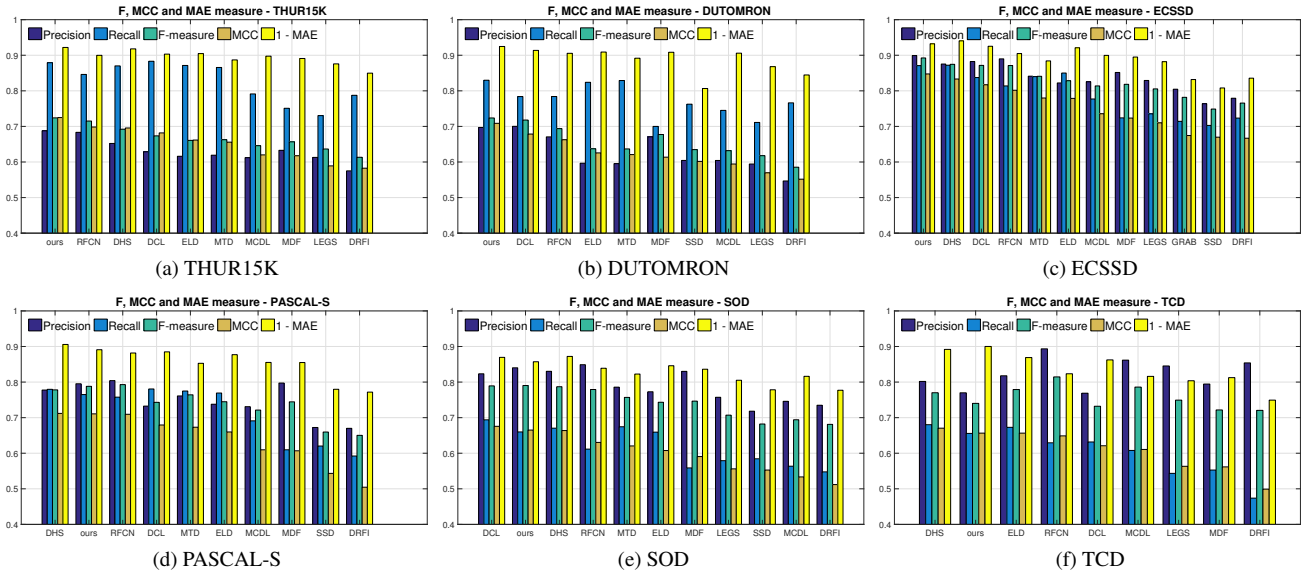
4.4. Segmentation by adaptive threshold

Quantitative comparison can also be achieved by comparing the segmented saliency map with the corresponding ground truth. A simple and effective way of segmentation is to compute the threshold according to the image statistics adaptively. As introduced in [1], this threshold T for a saliency map s can be obtained as $T = 2/(W \cdot H) \sum_i \sum_j s(i,j)$, where W and H refer to the width and height of saliency map s, respectively. We can calculate so called F-measure by comparing this binary map to the human-labeled ground truth as $F_\beta = (1+\beta^2) \cdot p \cdot r / \left(\beta^2 \cdot p + r\right)$, where β is a trade-off parameter that controls the weight between precision p and recall r. A small β means precision values are more counted in F-measure computation than recall values. Following the setting in [1], we set β^2 to 0.3 in the evaluation. As shown in Figure 8, the 1^{st} to 3^{rd} bins of each bar group respectively show the precision, recall and F-measure of all evaluated methods using adaptive thresholding on six saliency datasets.

We observe that the precision, recall and F-measure do not consider the true negative saliency predictions. In other word, these measures care more about the performance of methods on detecting the salient regions than their ability of avoiding highlighting non-salient regions. Therefore, we additionally include Matthews Correlation Coefficient (MCC) [40] in the comparison, which is defined as $\text{MCC} = (tp \cdot tn - fp \cdot fn) \,/\, \left((tp+fp)(tp+fn)(tn+fp)(tn+fn)\right)^{1/2}$, where tp, tn, fp, fn refer to the true positive, true negative, false positive and false negative, respectively. MCC can be taken as a fair measurement for evaluating binary classification task even if two classes are very unbalanced. It is suitable for our evaluation as foreground pixels usually take 10% to 40% of all pixels in a test image on average[1]. As shown in Figure 8, the 4^{th} bin of each bar group shows the the MCC measure of all evaluated methods using adaptive thresholding on six saliency datasets. Taking the evaluations of RFCN and our method on PASCAL-S for example, RFCN achieves recognizable higher AUC score and F_β than our method does, however, our method is slightly better than RFCN in the evaluation of MCC score.

Finally, we employ the Mean Absolute Error

[1]This can be inferred from Figure 7. As all pixels are binarized to foreground when recall equals to 1, the corresponding precision value equals to the percentage of foreground pixels in a ground truth image.



(a) THUR15K (b) DUTOMRON (c) ECSSD

(d) PASCAL-S (e) SOD (f) TCD

Figure 8: The precision, recall, F-measure, MCC (using adaptive thresholding) and MAE of all evaluated methods on six saliency datasets. All methods are sorted by descending 'MCC' measure (the bin in gold color of each bar group).

(MAE) [44] for evaluation, which is defined as $\text{MAE} = 1/(W \cdot H) \sum_i \sum_j \left| s(i,j) - gt(i,j) \right|$, where gt is the ground truth associated with s. As shown in Figure 8, the 5^{th} bin of each bar group shows the the MAE measure[2] of all evaluated methods on six saliency datasets.

4.5. The effectiveness of saliency propagation

In this section, we evaluate the performance of our saliency propagation method. Fig. 9 compares the performance of SSP prediction and the saliency propagation with different values of β in Eq. 6 on DUTOMRON. It shows that our refinement method improves the AUC score by 3% based on the result of SSP prediction. Fig. 9 also shows that, our proposed NN similarity contributes more than the commonly used adjacent similarity does to the improvement of AUC score, which proves the effectiveness of our saliency propagation method.

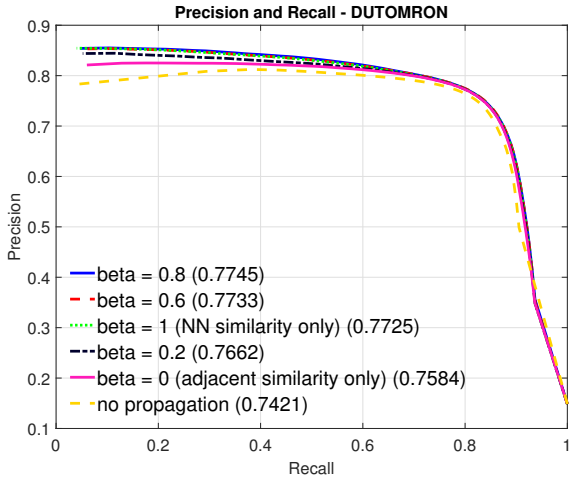


Figure 9: Evaluation of the proposed saliency propagation method. 'beta' refers to β in Eq. 6. The larger β indicates that larger weight is assigned to the NN similarity in our K-NN enhanced graph.

5. Conclusion and future work

In this paper we propose to use Structured Saliency Pattern (SSP) for describing local saliency distributions. We further introduce an adaptive ranking method for SSP refinement via learning a CNN model. Finally, a new K-NN enhanced graph model is proposed for saliency propagation. Our method is validated on six popular benchmarks and outperforms seven recently published state-of-the-art approaches with several evaluation measures. In future work, we plan to learn SSP from the feature maps of CNN models directly rather than from the hand-crafted features.

Acknowledgements This work was supported by the Natural Science Foundation of China (Grant No. 61502358, 61502357 and 61501336), by the National Key Research and Development Plan (Grant No. 2016YFB1001200), and by US National Science Foundation (Grants No. 1350521 and No. 1407156).

[2]For better illustration, we plot '$1 - \text{MAE}$' instead of MAE.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov. 2012.

[3] R. Achanta and S. Süsstrunk. Saliency detection using maximum symmetric surround. In *IEEE International Conference on Image Processing*, pages 2653–2656, 2010.

[4] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *IEEE International Conference on Computer Vision*, pages 914–921, 2011.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015.

[6] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM Transactions on Graphics*, 28(5):124, Dec. 2009.

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, Aug. 2015.

[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014.

[9] P. Dollar and C. L. Zitnick. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, Aug. 2015.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sep. 2004.

[11] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *IEEE International Conference on Computer Vision*, pages 1028–1035, 2011.

[12] S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundations. *ACM Transactions on Applied Perception*, 7(1):1–39, Jan. 2010.

[13] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vision Computing*, 30(1):51–64, Jan. 2012.

[14] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, Oct. 2012.

[15] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov. 1998.

[18] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2013.

[19] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision*, pages 1976–1983, 2013.

[20] M. Karasuyama and H. Mamitsuka. Manifold-based similarity adaptation for label propagation. In *Advances in Neural Information Processing Systems*, pages 1547–1555, 2013.

[21] J. Kim, D. Han, Y. W. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2014.

[22] J. Kim and V. Pavlovic. A shape-based approach for salient object detection using deep learning. In *European Conference on Computer Vision*, pages 455–470, 2016.

[23] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *IEEE International Conference on Computer Vision*, pages 2214–2219, 2011.

[24] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *IEEE International Conference on Computer Vision*, pages 2214–2219, 2011.

[25] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *IEEE International Conference on Computer Vision*, pages 2190–2197, 2011.

[26] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–668, 2016.

[27] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, Jun. 2001.

[28] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. Dagan Feng. Robust saliency detection via regularized random walks ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2710–2717, 2015.

[29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5455–5463, 2015.

[30] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 478–487, 2016.

[31] S. Li, H. Lu, Z. Lin, X. Shen, and B. Price. Adaptive metric learning for saliency detection. *IEEE Transactions on Image Processing*, 24(11):3321–3331, Jun. 2015.

[32] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural

network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930, Jun. 2016.

[33] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[34] Y. Li, Y. Zhou, J. Yan, Z. Niu, and J. Yang. Visual saliency based on conditional entropy. In *Asian Conference on Computer Vision*, pages 246–257, 2009.

[35] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017.

[36] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.

[37] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, Mar. 2011.

[38] S. Lu, V. Mahadevan, and N. Vasconcelos. Learning Optimal Seeds for Diffusion-Based Salient Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2014.

[39] R. Mairon and O. Ben-Shahar. A closer look at context: From coxels to the contextual emergence of object saliency. In *European Conference on Computer Vision*, pages 708–724, 2014.

[40] B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, Oct. 1975.

[41] M. Park, M. Kumar, and A. C. Loui. Saliency detection using region-based incremental center-surround distance. In *IEEE International Symposium on Multimedia*, pages 249–256, 2011.

[42] H. E. Pashler and S. Sutherland. *The psychology of attention*, volume 15. MIT press Cambridge, MA, 1998.

[43] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank. Salient object detection via structured matrix decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):818–832, 2017.

[44] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

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

[46] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang. Region-based saliency detection and its application in object recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(5):769–779, Aug. 2014.

[47] N. Riche, M. Mancas, B. Gosselin, and T. Dutoit. Rare: A new bottom-up saliency model. In *IEEE International Conference on Image Processing*, pages 641–644, 2012.

[48] H. J. Seo and P. Milanfar. Nonparametric bottom-up saliency detection by self-resemblance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 45–52, 2009.

[49] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical Image Saliency Detection on Extended CSSD. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, Apr. 2016.

[50] N. Tong, H. Lu, X. Ruan, and M.-h. Yang. Salient object detection via bootstrap learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1884–1892, 2015.

[51] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, Jan. 1980.

[52] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, Sep. 2013.

[53] K. Wang, L. Lin, J. Lu, C. Li, and K. Shi. Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence. *IEEE Transactions on Image Processing*, 24(10):3019–3033, Jun. 2015.

[54] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng. Video Object Discovery and Co-Segmentation with Extremely Weak Supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2016.

[55] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.

[56] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*, pages 825–841, 2016.

[57] Q. Wang, W. Zheng, and R. Piramuthu. Grab: Visual saliency via novel graph model and background priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 535–543, 2016.

[58] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2013.

[59] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274, 2015.

[60] L. Zhu, D. A. Klein, S. Frintrop, Z. Cao, and A. B. Cremers. A multi-size superpixel approach for salient object detection based on multivariate normal distribution estimation. *IEEE Transactions on Image Processing*, 23(12):5094–107, Dec. 2014.

[61] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405, 2014.