Salient Object Detection via Bootstrap Learning

Na Tong[1], Huchuan Lu[1], Xiang Ruan[2] and Ming-Hsuan Yang[3]
[1]Dalian University of Technology [2]OMRON Corporation [3]University of California at Merced

Abstract

We propose a bootstrap learning algorithm for salient object detection in which both weak and strong models are exploited. First, a weak saliency map is constructed based on image priors to generate training samples for a strong model. Second, a strong classifier based on samples directly from an input image is learned to detect salient pixels. Results from multiscale saliency maps are integrated to further improve the detection performance. Extensive experiments on six benchmark datasets demonstrate that the proposed bootstrap learning algorithm performs favorably against the state-of-the-art saliency detection methods. Furthermore, we show that the proposed bootstrap learning approach can be easily applied to other bottom-up saliency models for significant improvement.

1. Introduction

As an important preprocessing step in computer vision problems to reduce computational complexity, saliency detection has attracted much attention in recent years. Although significant progress has been made, it remains a challenging task to develop effective and efficient algorithms for salient object detection.

Saliency models include two main research areas: visual attention which is extensively studied in neuroscience and cognitive modeling, and salient object detection which is of great interest in computer vision. Salient object detection methods can be categorized as bottom-up stimuli-driven [1, 8–12, 15–18, 20, 23, 28–38, 41, 43] and top-down task-driven [19, 40, 42] approaches. Bottom-up methods are usually based on low-level visual information and are more effective in detecting fine details rather than global shape information. In contrast, top-down saliency models are able to detect objects of certain sizes and categories based on more representative features from training samples. However, the detection results from top-down methods tend to be coarse with fewer details. In terms of computational complexity, bottom-up methods are often more efficient than top-down approaches.

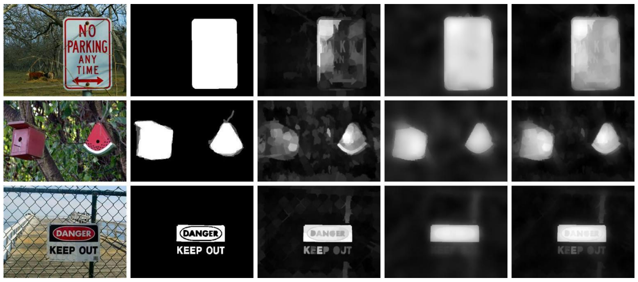


Figure 1. Saliency maps generated by the proposed method. Brighter pixels indicate higher saliency values. Left to right: input, ground truth, weak saliency map, strong saliency map, and final saliency map.

In this paper, we propose a novel algorithm for salient object detection via bootstrap learning [22]. To address the problems of noisy detection results and limited representations from bottom-up methods, we present a learning approach to exploit multiple features. However, unlike existing top-down learning-based methods, the proposed algorithm is bootstrapped with samples from a bottom-up model, thereby alleviating the time-consuming off-line training process or labeling positive samples manually.

2. Related Work and Problem Context

Both weak and strong learning models are exploited in the proposed bootstrap learning algorithm. First, we compute a weak contrast-based saliency map based on superpixels of an input image. This coarse saliency map is smoothed by a graph cut method, where a set of training samples is collected, where positive samples are pertaining to the salient objects while negative samples are from the background in this image. Next, a strong classifier based on Multiple Kernel Boosting (MKB) [39] is learned to measure saliency where three feature descriptors (RGB, CIELab color pixels, and the Local Binary Pattern histograms) are extracted and four kernels (linear, polynomial, RBF, and sigmoid functions) are used to exploit rich feature representations. Furthermore, we use multiscale superpixels to detect salient objects of varying sizes. As the weak saliency model tends to detect fine details and the strong saliency model focuses on global shapes, these two are combined to generate the final saliency map. Experiments on six benchmark

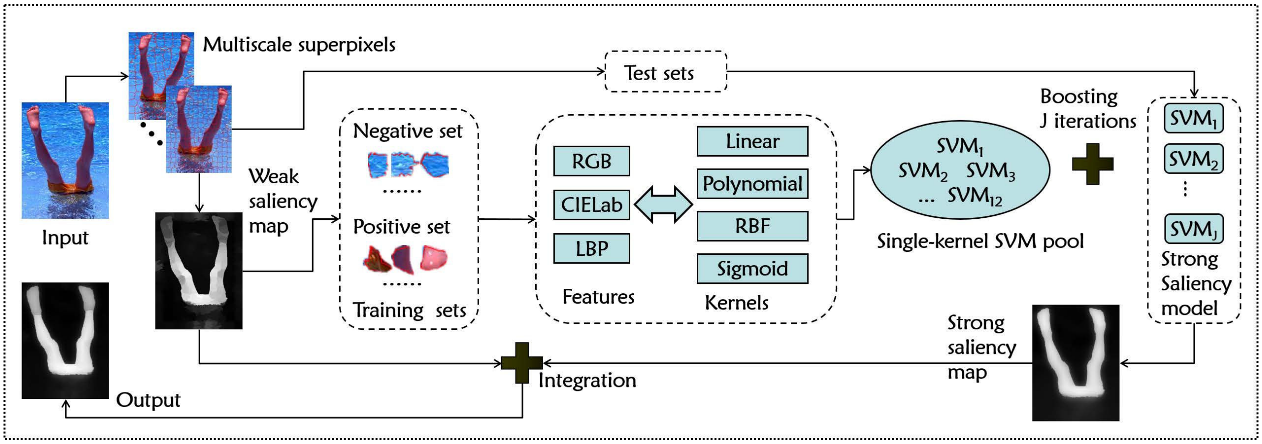


Figure 2. Bootstrap learning for salient object detection. A weak saliency map is constructed based on image priors to generate training samples for a strong model. A strong classifier based on multiple kernel boosting is learned to measure saliency where three feature descriptors are extracted and four kernels are used to exploit rich feature representations. Detection results at multiple scales are integrated. The weak and strong saliency maps are weighted combined to generate the final saliency map.

datasets show that the proposed bootstrap learning algorithm performs favorably against the state-of-the-art saliency detection methods. In addition, we incorporate the proposed bootstrap learning algorithm with existing bottom-up saliency methods, and achieve significant improvement in salient object detection. Figure 1 shows some saliency maps generated by the proposed method where brighter pixels indicate higher saliency values.

Numerous bottom-up saliency detection methods have been proposed in recent years. Itti *et al.* [16] propose a saliency model based on a neural network that integrates three feature channels over multiple scales for rapid scene analysis. While it is able to identify salient pixels, the results contain a significant amount of false detections. A graph-based saliency measure is proposed by Harel *et al.* [12]. However, this method focuses on eye fixation prediction and generates a low resolution saliency map similar to [16]. Saliency models based on Bayesian inference have been proposed in [29, 35, 36]. In [18], the low-level saliency stimuli and the shape prior are integrated using an iterative energy minimization measure. In [28], Perrazzi *et al.* present a contrast-based saliency filter and measure saliency by the uniqueness and spatial distribution of regions over an image. While the above-mentioned contrast-based methods are simple and effective, pixels within the salient objects are not always highlighted well. Shen and Wu [30] construct a unified model combining lower-level features and higher-level priors for saliency detection based on the theory of low rank matrix recovery. In [34], Wei *et al.* focus on the background instead of the foreground and build a saliency detection model based on two background priors, i.e., boundary and connectivity. Cheng *et al.* [10] utilize a soft abstraction method to remove unnecessary image details and produce perceptually accurate salient regions. In [37], Yan *et al.* formulate a multiscale method using a tree model to deal with the scale problem. A graph-based bottom-up method is proposed using manifold ranking [38]. Recently, Zhu *et al.* [43] construct a salient object detection method based on boundary connectivity.

Compared to bottom-up approaches, considerable efforts have been made on top-down saliency models. In [42], Zhang *et al.* construct a Bayesian-based top-down model by integrating both the top-down and bottom-up information where saliency is computed locally. A saliency model based on the Conditional Random Field is formulated with latent variables and a discriminative dictionary in [40]. Jiang *et al.* [19] propose a learning-based method by regarding saliency detection as a regression problem where the saliency detection model is constructed based on the integration of numerous descriptors extracted from training samples with ground truth labels.

As these two categories bring forth different properties of efficient and effective salient detection algorithms, we propose a bootstrap learning approach which exploits the strength of both bottom-up contrast-based saliency models and top-down learning methods.

3. Bootstrap Saliency Model

Figure 2 shows the main steps of the proposed salient object detection algorithm. We first construct a weak saliency map from which the training samples are collected. For each image, we learn a strong classifier based on superpixels. To deal with the scale problem, multiscale detection results are generated and merged to construct a strong saliency map. The final saliency map is the weighted integration of the weak and strong maps for accurate detection results.

3.1. Image Features

Superpixels have been used extensively in vision tasks as the basic units to capture the local structural information. In this paper, we compute a fixed number of superpixels from an input image using the Simple Linear Iterative Clustering (SLIC) method [2]. Three descriptors including the RGB, CIELab and Local Binary Pattern (LBP) features are used to describe each superpixel. The rationale to use two different color representations is based on empirical results where better detection performance is achieved when both are used, which can be found in the supplementary document. We consider the LBP features in a 3×3 neighborhood of each pixel. Next, each pixel is assigned to a value between 0 and 58 in the uniform pattern [27]. We construct an LBP histogram for each superpixel, i.e., a vector of 59 dimensions ($\{h_i\}, i = 1, 2, ...59$, where h_i is the value of the i-th bin in an LBP histogram).

3.2. Weak Saliency Model

The center-bias prior has been shown to be effective in salient object detection [5, 25]. Based on this assumption, we develop a method to construct a weak saliency model by exploiting the contrast between each region and the regions along the image border. However, existing contrast-based methods usually generate noisy results since low-level visual cues are limited. In this paper, we exploit the center-bias and dark channel priors to better estimate saliency maps.

The dark channel prior is proposed for the image haze removal task [14]. The main observation is that, for regions that do not cover the sky (e.g., ground or buildings), there exist some pixels with low intensity values in one of the RGB color channels. Thus, the minimum pixel intensity in any such region is low. The dark channel of image patches is mainly generated by colored or dark objects and shadows, which usually appear in the salient regions as shown in Figure 3. The sky region of an image usually belongs to the background, which is just consistent with the dark channel property for the sky region. Therefore, we exploit the dark channel property to estimate saliency of pixels. In addition, for situations where the input image has dark background or bright foreground, we use an adaptive weight computed based on the average value on the edge of dark channel map.

In the proposed method, we define the dark channel prior of an image on the pixel level. For a pixel p, the dark channel prior $S_d(p)$ is computed by

$$S_d(p) = 1 - \min_{q \in patch(p)} \left(\min_{ch \in \{r,g,b\}} \left(I^{ch}(q) \right) \right), \quad (1)$$

where $patch(p)$ is the 5×5 image patch centered at p and $I^{ch}(q)$ is the color value of pixel q on the corresponding color channel ch. Note that all the color values are normalized into $[0, 1]$. We achieve pixel-level accuracy instead of the patch-level counterpart in [14]. We also show the effect of dark channel prior quantitatively in Figure 7(b).

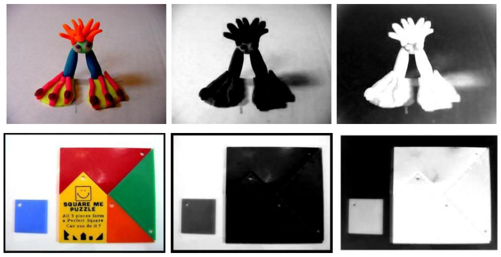


Figure 3. Examples of dark channel prior. Left to right: input, dark channel map and dark channel prior (the opposite of dark channel map and brighter pixels indicate higher saliency values).

An input image is segmented into N superpixels, $\{c_i\}, i = 1, \ldots, N$. The regions along the image border are represented as $\{n_j\}, j = 1, \ldots, N_B$, where N_B is the number of regions along the image border. We compute the dark channel prior for each region c_i using $S_d(c_i) = \frac{1}{N_{c_i}} \sum_{p \in c_i} S_d(p)$, where N_{c_i} is the number of pixels within the region c_i. The coarse saliency value for the region c_i is constructed by

$$f_0(c_i) = g(c_i) \times S_d(c_i) \times \sum_{\kappa \in \{F_1, F_2, F_3\}} \left(\frac{1}{N_B} \sum_{j=1}^{N_B} d_\kappa(c_i, n_j) \right), \quad (2)$$

where $d_\kappa(c_i, n_j)$ is the Euclidean distance between region c_i and n_j in the feature space that κ represents, i.e., the RGB (F_1), CIELab (F_2) and LBP (F_3) texture features respectively. Note that all the distance values in each feature space are normalized into $[0, 1]$. In addition, $g(c_i)$ is computed based on the center prior using the normalized spatial distance between the center of the superpixel c_i and the image center [18]. Thus the saliency value of the region closer to the image center is assigned a higher weight. We generate a pixel-wise saliency map $\mathcal{M}_0$ using (2), where the saliency value of each superpixel is assigned to the contained pixels.

Most existing methods usually use Gaussian filtering to generate smoothed saliency maps at the expense of accuracy. In this paper, we use a simple yet effective algorithm based on the Graph Cut method [7, 21], to determine the foreground and background regions in $\mathcal{M}_0$. Given an input image, we construct an undirected graph $G = (V, E, T)$, where E is a set of undirected edges that connect the nodes V (pixels) while T is the set of the weights of nodes connected to the background and foreground terminals. The weight of each node (pixel) p connected to the foreground terminal is assigned with the saliency value in the pixel-wise map $\mathcal{M}_0$. Thus for each pixel p, the set T consists of two components, defined as $\{T^f(p)\}$ and $\{T^b(p)\}$, and is computed by

$$T^f(p) = \mathcal{M}_0(p), \quad T^b(p) = 1 - \mathcal{M}_0(p), \quad (3)$$

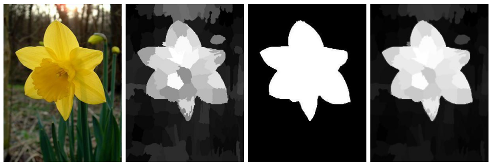


Figure 4. Performance of Graph Cut. Left to right: input, saliency maps without Graph Cut, binary results using Graph Cut, saliency maps after summing up the previous two maps.

where $T^f(p)$ is the weight of pixel p connected to the foreground while $T^b(p)$ is the weight to the background. The minimum cost cut generates a foreground mask $\mathcal{M}_1$ using the Max-Flow [6] method to measure the probability of each pixel being foreground.

As shown in Figure 4, $\mathcal{M}_1$ is a binary map which may contain noise in both foreground and background. Thus we consider both the binary map $\mathcal{M}_1$ and the map $\mathcal{M}_0$ to construct the continuous and smoothed weak saliency map $\check{\mathcal{M}}_w$ by

$$\check{\mathcal{M}}_w = \frac{\mathcal{M}_0 + \mathcal{M}_1}{2}. \quad (4)$$

We show the performance of the Graph Cut method quantitatively in Figure 7(b). The training set for the strong classifier is selected from the weak saliency map. We compute the average saliency value for each superpixel and set two thresholds to generate the training set containing both positive and negative samples. The superpixels with saliency values larger than the high threshold are labeled as the positive samples with $+1$ while those with saliency values smaller than the low threshold as the negative samples labeled with -1. More details about the threshold setting can be found in the supplementary document.

3.3. Strong Saliency Model

One of the main difficulties using a Support Vector Machine (SVM) is to determine the appropriate kernel for the given dataset. This problem is more complicated when the dataset contains thousands of diverse images with different properties. While numerous saliency detection methods based on various features have been proposed, it is still not clear how these features can be well integrated. To cope with these problems, we present a method similar to the Multiple Kernel Boosting (MKB) [39] method to include multiple kernels of different features. We treat SVMs with different kernels as weak classifiers and then learn a strong classifier using the boosting method. Note that we restrict the learning process to each input image to avoid the heavy computational load of extracting features and learning kernels for a large amount of training data (as required in several discriminative methods [19] in the literature for saliency detection).

The MKB algorithm is a boosted Multiple Kernel Learning (MKL) method [4], which combines several SVMs of different kernels. For each image, we have the training samples $\{r_i, l_i\}_{i=1}^H$ from the weak saliency map $\check{\mathcal{M}}_w$ (See Section 3.2) where r_i is the i-th sample, l_i represents the binary label of the sample and H indicates the number of the samples. The linear combination of kernels $\{k_m\}_{m=1}^M$ is defined by

$$k(r, r_i) = \sum_{m=1}^{M} \beta_m k_m(r, r_i), \sum_{m=1}^{M} \beta_m = 1, \beta_m \in \mathbb{R}_+, \quad (5)$$

where β_m is the kernel weight and M denotes the number of the weak classifiers, and $M = N_f \times N_k$. Here, N_f is the number of the features and N_k indicates the number of the kernels (e.g., $N_f = 3, N_k = 4$ in this work). For different feature sets, the decision function is defined as a convex combination,

$$Y(r) = \sum_{m=1}^{M} \beta_m \sum_{i=1}^{H} \alpha_i l_i k_m(r, r_i) + \bar{b}, \quad (6)$$

where α_i is the Lagrange multiplier while $\bar{b}$ is the bias in the standard SVM algorithm. The parameters $\{\alpha_i\}$, $\{\beta_m\}$ and $\bar{b}$ can be learned from a joint optimization process.

We note that (6) is a conventional function for the MKL method. In this paper we use the boosting algorithm instead of the simple combination of single-kernel SVMs in the MKL method. We rewrite (6) as

$$Y(r) = \sum_{m=1}^{M} \beta_m (\boldsymbol{\alpha}^\top \mathbf{k}_m(r) + \bar{b}_m), \quad (7)$$

where $\boldsymbol{\alpha} = [\alpha_1 l_1, \alpha_2 l_2, \ldots, \alpha_H l_H]^\top$, $\mathbf{k}_m(r) = [k_m(r, r_1), k_m(r, r_2), \ldots, k_m(r, r_H)]^\top$ and $\bar{b} = \sum_{m=1}^M \bar{b}_m$. By setting the decision function of a single-kernel SVM as $z_m(r) = \boldsymbol{\alpha}^\top \mathbf{k}_m(r) + \bar{b}_m$, the parameters can be learned straightforwardly. Thus, (7) can be rewritten as

$$Y(r) = \sum_{j=1}^{J} \beta_j z_j(r). \quad (8)$$

In order to compute the parameters β_j, we use the Adaboost method and the parameter J in (8) denotes the number of iterations of the boosting process. We consider each SVM as a weak classifier and the final strong classifier $Y(r)$ is the weighted combination of all the weak classifiers. Starting with uniform weights, $\omega_1(i) = 1/H, i = 1, 2, \ldots, H$, for the SVM classifiers, we obtain a set of decision functions $\{z_m(r)\}, m = 1, 2, \ldots, M$. At the j-th iteration, we compute the classification error for each of the weak classifiers,

$$\epsilon_m = \frac{\sum_{i=1}^{H} \omega(i)|z_m(r_i)|(\text{sgn}(-l_i z_m(r_i)) + 1)/2}{\sum_{i=1}^{H} \omega(i)|z_m(r_i)|}, \quad (9)$$

where $\text{sgn}(x)$ is the sign function, which equals to 1 when $x > 0$ and -1 otherwise. We locate the decision function

$z_j(r)$ with the minimum error ϵ_j, i.e., $\epsilon_j = \min_{1 \leq m \leq M} \epsilon_m$. Then the combination coefficient β_j is computed by $\beta_j = \frac{1}{2} \log \frac{1-\epsilon_j}{\epsilon_j} \cdot \frac{1}{2}(\text{sgn}(\log \frac{1-\epsilon_j}{\epsilon_j}) + 1)$. Note that β_j must be larger than 0, indicating $\epsilon_j < 0.5$, which accords with the basic hypothesis that the boosting method could make the weak classifiers into a strong one. In addition, we update the weight using the following equation,

$$\omega_{j+1}(i) = \frac{\omega_j(i) e^{-\beta_j l_i z_j(r_i)}}{2\sqrt{\epsilon_j(\epsilon_j - 1)}}. \tag{10}$$

After J iterations, all the β_j and $z_j(r)$ are computed and we have a boosted classifier (8) as the saliency model learned directly from an input image. We apply this strong saliency model to the test samples (based on all the superpixels of an input image), and a pixel-wise saliency map is thus generated.

To improve the accuracy of the map, we first use the Graph Cut method to smooth the saliency detection results. Next, we obtain the strong saliency map $\check{\mathcal{M}}_s$ by further enhancing the saliency map with the guided filter [13] as it has been shown to perform well as an edge-preserving smoothing operator.

3.4. Multiscale Saliency Maps

The accuracy of the saliency map is sensitive to the number of superpixels as salient objects are likely to appear at different scales. To deal with the scale problem, we generate four layers of superpixels with different granularities, where $N = 100, 150, 200, 250$ respectively. We represent the weak saliency map (See Section 3.2) at each scale as $\{\check{\mathcal{M}}_{w_i}\}$ and the multiscale weak saliency map is computed by $\mathcal{M}_w = \frac{1}{4} \sum_{i=1}^{4} \check{\mathcal{M}}_{w_i}$. Next, the training sets from the four scales are used to train one strong saliency model and the test sets (based on all the superpixels from four scales) are tested by the learned model simultaneously. Four strong saliency maps from four scales are constructed (See Section 3.3), denoted as $\{\check{\mathcal{M}}_{s_i}\}, i = 1, 2, 3, 4$. Finally, we obtain the final strong saliency map as $\mathcal{M}_s = \frac{1}{4} \sum_{i=1}^{4} \check{\mathcal{M}}_{s_i}$. As such, the proposed method is robust to scale variation.

3.5. Integration

The proposed weak and strong saliency maps have complementary properties. The weak map is likely to detect fine details and to capture local structural information due to the contrast-based measure. In contrast, the strong map works well by focusing on global shapes for most images except the case when the test background samples have similarity with the positive training set or large differences compared to the negative training set, or vice versa for the test foreground sample. In this case, the strong map may misclassify the test regions as shown in the bottom row of Figure 1. Thus we integrate these two maps by a weighted combination,

$$\mathcal{M} = \sigma \mathcal{M}_s + (1 - \sigma) \mathcal{M}_w, \tag{11}$$

where σ is a balance factor for the combination, and $\sigma = 0.7$ to weigh the strong map more than the weak map, and $\mathcal{M}$ is the final saliency map via bootstrap learning. More discussions about the values of σ can be found in the supplementary document.

4. Experimental Results

We present experimental results of 22 saliency detection methods including the proposed algorithms on six benchmark datasets. The ASD dataset, selected from a bigger image database [25], contains 1,000 images, and is labeled with pixel-wise ground truth [1]. The THUS dataset [9] consists of $10,000$ images where all images are labeled with pixel-wise ground truth. The SOD dataset [26] is composed of 300 images from the Berkeley segmentation dataset where each one is labeled with salient object boundaries, based on which the pixel level ground truth [34] is built. Some of the images in the SOD dataset include more than one salient object. The SED2 dataset [3] contains 100 images which are labeled with pixel-wise ground truth annotations. It is challenging due to the fact that every image has two salient objects. The Pascal-S dataset [24] contains 850 images which are also labeled with pixel-wise ground truth. For comprehensive evaluation, we use all the images in the Pascal-S dataset for test instead of using 40% for training and the rest for test as [24]. The DUT-OMRON dataset [38] contains 5168 challenging images with pixel-wise ground truth annotations. All the experiments are carried out using MATLAB on a desktop computer with an Intel i7-3770 CPU (3.4 GHz) and 32GB RAM. For fair comparison, we use the original source code or the provided saliency detection results in the literature. The MATLAB source code is available on our project site.

We first evaluate the proposed algorithms and other 19 state-of-the-art methods including the IT98 [16], SF [28], LRMR [30], wCO [43], GS_SP [34], XL13 [36], RA10 [29], GB [12], LC [41], SR [15], FT [1], CA [11], SVO [8], CBsal [18], GMR [38], GC [10], HS [37], RC-J [9] and DSR [23] methods on the ASD, SOD, SED2, THUS and DUT-OMRON datasets. In addition, the DRFI [19] method uses images and ground truth for training, which contains part of the ASD, THUS and SOD datasets, and the results on the Pascal-S dataset are not provided. Accordingly, we only compare our method with the DRFI model on the SED2 dataset. Therefore, our methods are evaluated with 20 methods on the SED2 datasets. The MSRA [25] dataset consists of 5,000 images. Since more than 3,700 images in the MSRA dataset are included in the THUS dataset, we do not present the evaluation results on this dataset due to space limitations.

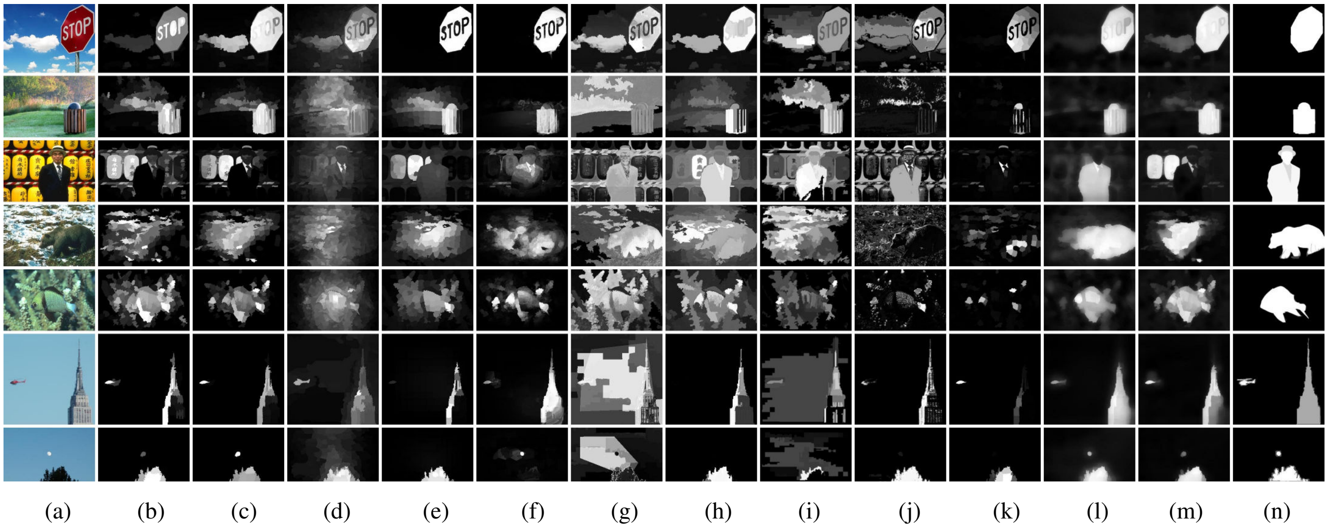


Figure 5. Comparison of our saliency maps with ten state-of-the-art methods. Left to right: (a) input (b) GS_SP [34] (c) wCO [43] (d) LRMR [30] (e) GMR [38] (f) DSR [23] (g) XL13 [36] (h) HS [37] (i) RC-J [9] (j) GC [10] (k) SF [28] (l) Ours (m) wCO-bootstrapped (n) ground truth. Our model is able to detect both the foreground and background uniformly.

4.1. Qualitative Results

We present some results of saliency maps generated by twelve methods for qualitative comparison in Figure 5, where "wCO-bootstrapped" means the wCO model bootstrapped by the proposed learning approach. The saliency maps generated by the proposed algorithms highlight the salient objects well with fewer noisy results. We note that these salient objects appear at different image locations although the center-bias is used in the proposed algorithm. The detected foreground and background in our maps are smooth due to the using of the Graph Cut and guided filtering methods. As a result of using both weak (effective for picking up details) and strong (effective for discriminating boundaries) saliency maps, the proposed bootstrap learning algorithm performs well for images containing multiple objects as shown in the bottom two rows of Figure 5. Furthermore, due to the contribution of the LBP features (effective for texture classification), the proposed method is able to detect salient objects accurately despite similar appearance to the background regions as shown in the fourth and fifth rows of Figure 5. More results can be found in the supplementary document.

4.2. Quantitative Results

We use the Precision and Recall (P-R) curve to evaluate all the methods. We set the fixed threshold from 0 to 255 with an increment of 5 for a saliency map with consistent gray value, thus producing 52 binary masks. Using the pixel-wise ground truth data, 52 pairs of average P-R values of all the images included in the test datasets are computed. Figure 6 shows the P-R curves where several state-of-the-art methods and the proposed algorithms perform well. To better assess these methods, we compute the Area Under ROC Curve (AUC) for the best performing methods. Table 1 shows that the proposed algorithms perform favorably against other state-of-the-art methods in terms of AUC on all the six datasets that contain both single and multiple salient objects.

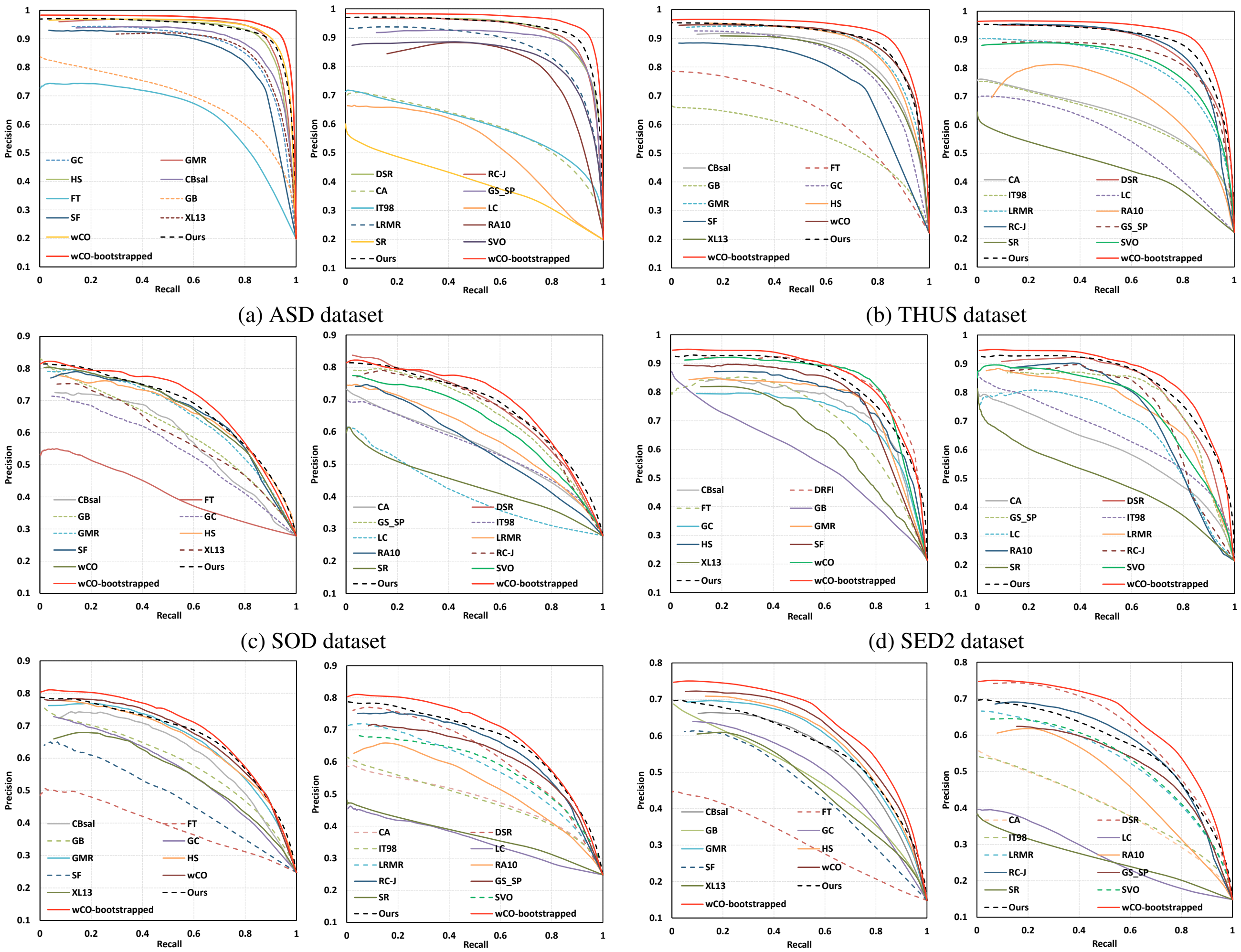
In addition, we measure the quality of the saliency maps using the F-Measure by adaptively setting a segmentation threshold for binary segmentation [1]. The adaptive threshold is twice the average value of the whole saliency map. Each image is segmented with superpixels and masked out if the mean saliency values are lower than the adaptive threshold. The average precision and recall values are computed based on the generated binary masks and the ground truth while the F-Measure is computed by

$$F_\eta = \frac{(1+\eta^2) \times Precision \times Recall}{\eta^2 \times Precision + Recall}, \tag{12}$$

and η^2 is set to 0.3 to weigh precision more than recall. Figure 7(a) shows the F-Measure values of the evaluated methods on the six datasets. Overall, the proposed algorithms perform well (with top or second values) against the state-of-the-art methods.

4.3. Analysis of the Bootstrap Saliency Model

Every component in the proposed algorithm contributes to the final saliency map. Figure 7(b) shows the performance of each step in the proposed method, i.e., the dark channel prior, graph cut, weak saliency map, and strong saliency map, among which the dark channel prior appears to contribute least but is still indispensable for the overall performance. The proposed weak saliency model may generate less accurate results than several state-of-the-art methods, but it is efficient with less computational complexity.



(a) ASD dataset

(b) THUS dataset

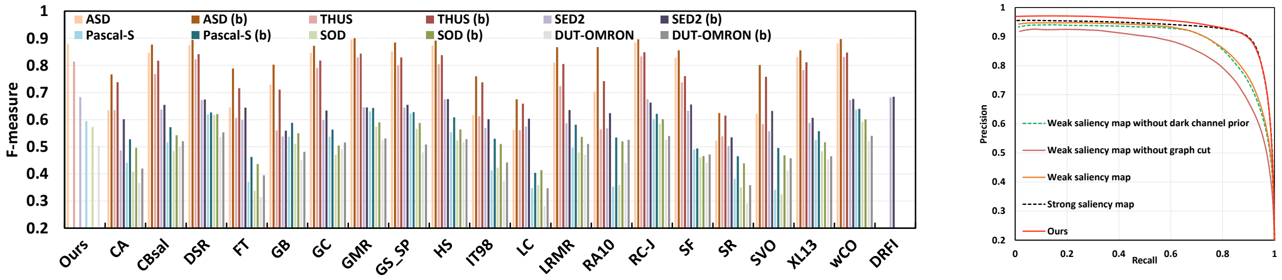
(c) SOD dataset

(d) SED2 dataset

(e) Pascal-S dataset

(f) DUT-OMRON dataset

Figure 6. P-R curve results on six datasets.



(a) F-measure

(b) P-R curve

Figure 7. (a) is the F-measure values of 21 methods on six datasets. Note that “ * (b)” shows improvement of state-of-the-art methods by the bootstrap learning approach on the corresponding dataset as stated in Section 4.4. (b) shows performance of each component in the proposed method on the ASD dataset.

4.4. Bootstrapping State-of-the-Art Methods

The performance of the proposed bootstrap learning method hinges on the quality of the weak saliency model. If a weak saliency model does not perform well, the proposed algorithm is likely to fail as an insufficient number of good training samples can be collected for constructing the strong model for a specific image. Figure 8 shows examples where the weak saliency model does not perform well, thereby affecting the overall performance of the proposed algorithm. This motivates us that the proposed algorithm can be used to bootstrap the performance of the state-

Table 1. AUC (Area Under ROC Curve) on the ASD, SED2, SOD, THUS, Pascal-S and DUT-OMRON datasets. The best two results are shown in **red** and **blue** fonts respectively. The colomn named "wCO-b" denotes the wCO model after bootstrapping using the proposed approach. The proposed methods rank first and second on the six datasets. The two rows named "*ASD (b)*" show the AUC of the saliency results by taking other state-of-the-art saliency maps as the weak saliency maps in the proposed approach on the ASD dataset. All the evaluation results of the state-of-the-art methods are largely improved over the original results as shown in the two rows named "*ASD*".

	wCO-b	Ours	HS	RC-J	GC	DSR	GS_SP	GMR	SF	XL13	CBsal
ASD	**.9904**	**.9828**	.9683	.9735	.9456	.9774	.9754	.9700	.9233	.9609	.9628
SED2	**.9399**	**.9363**	.8387	.8606	.8618	.9136	.8999	.8620	.8501	.8470	.8728
SOD	**.8525**	**.8477**	.8169	.8238	.7181	.8380	.7982	.7982	.8238	.7868	.7409
THUS	**.9723**	**.9635**	.9322	.9364	.9032	.9504	.9462	.9390	.8510	.9353	.9270
Pascal-S	**.8774**	**.8682**	.8368	.8379	.7479	.8299	.8553	.8315	.6830	.7983	.8087
DUT-OMRON	**.9063**	.8794	.8604	.8592	.7931	.8922	.8786	.8500	.7628	.8160	.8419
ASD (b)	-	-	.9876	.9869	.9773	.9872	**.9888**	.9844	.9723	.9791	.9811
	DRFI	wCO	LRMR	RA10	SVO	GB	FT	CA	SR	LC	IT98
ASD	-	.9805	.9593	.9326	.9530	.9146	.8375	.8736	.6973	.7772	.8738
SED2	.9349	.9062	.8886	.8500	.8773	.8448	.8185	.8585	.7593	.8366	.8904
SOD	-	.8217	.7810	.7710	.8043	.8191	.6006	.7868	.6695	.6168	.7862
THUS	-	.9525	.9199	.8810	.9280	.8132	.7890	.8712	.7149	.7673	.8655
Pascal-S	-	.8597	.8121	.7836	.8226	.8380	.6220	.7829	.6585	.6191	.7797
DUT-OMRON	-	**.8927**	.8566	.8264	.8662	.8565	.6758	.8137	.6799	.6549	.8218
ASD (b)	-	**.9904**	.9825	.9817	.9722	.9619	.9506	.9531	.8530	.8988	.9479

of-the-art methods (i.e., with better weak saliency maps). Thus, we generate different weak saliency maps by applying the graph cut method on the results generated by the state-of-the-art methods. Note that we only use two scales instead of four scales for efficiency and use equal weights in (11) (to better use these "weak" saliency maps) in the experiments. Figure 9 shows the P-R curves on the ASD dataset and Figure 7(a) shows the F-measure on six tested datasets. In addition, the AUC measures are shown on the two rows named *"ASD (b)"* of Table 1. These results show that the performance of all state-of-the-art methods can be significantly improved by the proposed bootstrap learning algorithm. For example, the performance improvement of the SR method over the original model for the AUC value is 22.3%. Four methods, the wCO, DSR, HS and GS_SP, achieve over 0.987 using the proposed bootstrap learning method and nine methods achieve higher than 0.98 in terms of AUC. Four methods, the RC-J, GMR, wCO and DSR, achieve over 0.98 for the highest precision value in the P-R curve on the ASD dataset. The average F-measure performance gains of 19 methods on the ASD, SOD, SED2, Pascal-S, THUS and DUT-OMRON datasets are 10.5%, 14.0%, 5.2%, 14.4%, 11.5% and 9.9% respectively.

5. Conclusion

In this paper, we propose a bootstrap learning model for salient object detection in which both weak and strong saliency models are constructed and integrated. Our learning process is restricted within multiple scales of the input image and is unsupervised since the training examples for the strong model are determined by a weak saliency map based on contrast and image priors. The strong saliency model is constructed based on the MKB algorithm which combines all the weak classifiers into a strong one using the Adaboost algorithm. Extensive experimental results demonstrate that the proposed approach performs favorably against 20 state-of-the-art methods on six benchmark datasets. In addition, the proposed bootstrap learning algorithm can be applied to other saliency models for significant improvement.

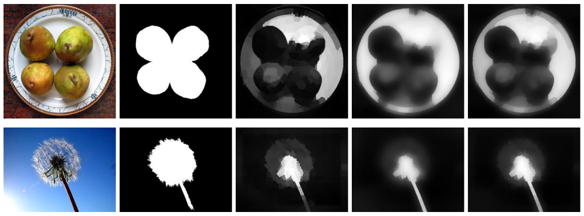


Figure 8. Failure cases of the proposed algorithm as the weak saliency maps do not perform well. Left to right: input, ground truth, weak saliency map, strong saliency map and the bootstrap saliency map generated by the proposed algorithm.

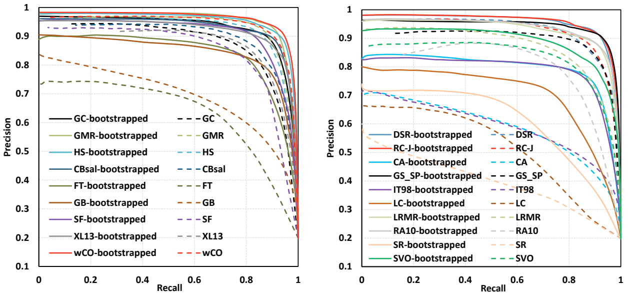


Figure 9. P-R curve results show improvement of state-of-the-art methods by the bootstrap learning approach on the ASD dataset.

Acknowledgements. N. Tong and H. Lu are supported by the Natural Science Foundation of China #61472060 and the Fundamental Research Funds for the Central Universities under Grant DUT14YQ101. M.-H. Yang is supported in part by NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009. 1, 5, 6

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. *EPFL*, 2010. 3

[3] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, 2007. 5

[4] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, 2004. 4

[5] A. Borji, D. N. Sihite, and L. Itti. Salient object detection: A benchmark. In *ECCV*, 2012. 3

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 4

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 3

[8] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *ICCV*, 2011. 1, 5

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

[10] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *ICCV*, 2013. 2, 5, 6

[11] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, 2010. 5

[12] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006. 1, 2, 5

[13] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010. 5

[14] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *PAMI*, 33(12):2341–2353, 2011. 3

[15] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007. 1, 5

[16] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20:1254–1259, 1998. 2, 5

[17] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang. Saliency detection via absorbing markov chain. In *ICCV*, 2013.

[18] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li. Automatic salient object segmentation based on context and shape prior. In *BMVC*, 2011. 1, 2, 3, 5

[19] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013. 1, 2, 4, 5

[20] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *ICCV*, 2011. 1

[21] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 3

[22] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *AAAI*, 2002. 1

[23] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, 2013. 1, 5, 6

[24] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 5

[25] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, 2007. 3, 5

[26] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *POCV*, 2010. 5

[27] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002. 3

[28] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012. 1, 2, 5, 6

[29] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, 2010. 2, 5

[30] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, 2012. 2, 5, 6

[31] J. Sun, H. Lu, and S. Li. Saliency detection based on integration of boundary and soft-segmentation. In *ICIP*, 2012.

[32] N. Tong, H. Lu, L. Zhang, and X. Ruan. Saliency detection with multi-scale superpixels. *SPL*, 21(9):1035–1039, 2014.

[33] N. Tong, H. Lu, Y. Zhang, and X. Ruan. Salient object detection via global and local cues. *Pattern Recognition*, doi:10.1016/j.patcog.2014.12.005, 2014.

[34] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, 2012. 2, 5, 6

[35] Y. Xie and H. Lu. Visual saliency detection based on Bayesian model. In *ICIP*, 2011. 2

[36] Y. Xie, H. Lu, and M.-H. Yang. Bayesian saliency via low and mid level cues. *TIP*, 22(5):1689–1698, 2013. 2, 5, 6

[37] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013. 2, 5, 6

[38] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 1, 2, 5, 6

[39] F. Yang, H. Lu, and Y.-W. Chen. Human tracking by multiple kernel boosting with locality affinity constraints. In *ACCV*, 2010. 1, 4

[40] J. Yang and M.-H. Yang. Top-down visual saliency via joint CRF and dictionary learning. In *CVPR*, 2012. 1, 2

[41] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *ACM MM*, 2006. 1, 5

[42] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7), 2008. 1, 2

[43] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, 2014. 1, 2, 5, 6