Salient Object Detection: A Discriminative Regional Feature Integration Approach

Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Yihong Gong,
Nanning Zheng, and Jingdong Wang

Abstract—Salient object detection has been attracting a lot of interest, and recently various heuristic computational models have been designed. In this paper, we formulate saliency map computation as a regression problem. Our method, which is based on multi-level image segmentation, utilizes the supervised learning approach to map the regional feature vector to a saliency score. Saliency scores across multiple layers are finally fused to produce the saliency map. The contributions lie in two-fold. One is that we propose a discriminate regional feature integration approach for salient object detection. Compared with existing heuristic models, our proposed method is able to automatically integrate high-dimensional regional saliency features and choose discriminative ones. The other is that by investigating standard generic region properties as well as two widely studied concepts for salient object detection, *i.e.*, regional contrast and backgroundness, our approach significantly outperforms state-of-the-art methods on six benchmark datasets. Meanwhile, we demonstrate that our method runs as fast as most existing algorithms.

1 Introduction

Visual saliency has been a fundamental problem in neuroscience, psychology, neural systems, and computer vision for a long time. It is originally defined as a task of predicting the eye-fixations on images [2]. Recently it is extended to identifying a region [3], [4] containing the salient object, known as *salient object detection* or *salient region detection*. Applications of salient object detection include object detection and recognition [5], [6], image compression [7], image cropping [8], photo collage [9], [10], dominant color detection [11], [12] and so on.

The study on human visual systems suggests that the saliency is related to uniqueness, rarity and surprise of a scene, characterized by primitive features like color, texture, shape, etc. Recently a lot of efforts have been made to design various heuristic algorithms to compute the saliency [13]–[21]. Built upon the feature integration theory [2], [22], almost all the approaches compute conspicuity (feature) maps from different saliency cues and then combine them together to form the final saliency map. Hand-crafted integration rules, however, are fragile and poor to generalize. For instance, in a recent survey [23], none of the algorithms can consistently outperforms others on the benchmark data sets. Though some learning-based salient object detection algorithms are proposed [19], [24], [25], the potential of supervised learning is not deeply investigated.

- *H. Jiang, Z. Yuan, Y. Gong and N. Zheng, are with Xi'an Jiaotong University. M.M. Cheng is with Oxford University. J. Wang is with Microsoft Research Asia.*
- *A preliminary version of this work appeared at CVPR [1].*
- *Project website jianghz.com/drfi.*

In this paper, we formulate salient object detection as a regression problem, learning a regressor that directly maps the regional feature vector to a saliency score. Our approach consists of three main steps. The first one is multi-level segmentation, which decomposes the image to multiple segmentations. Second, we conduct a region saliency computation step with a Random Forest regressor that maps the regional features to a saliency score. Last, a saliency map is computed by fusing the saliency maps across multiple layers of segmentations.

The key contributions lie in the second step, region saliency computation. Firstly, unlike most existing algorithms that compute saliency maps heuristically from various features and combine them to get the saliency map, which we call saliency integration, we learn a Random Forest regressor that directly maps the feature vector of each region to a saliency score, which we call discriminative regional feature integration (DRFI). This is a principle way in image classification [26], but rarely studied in salient object detection. Secondly, by investigating standard generic region properties and two widely studied concepts in salient object detection, *i.e.*, regional contrast and backgroundness, our proposed approach consistently outperforms state-of-the-art algorithms on all six benchmark data sets with large margins. Rather than heuristically hand-crafting special features, it turns out that the learned regressor is able to automatically integrate features and pick up discriminative ones for saliency. Even though the regressor is trained on a small set of images, it demonstrates good generalization ability to other data sets.

The rest of this paper is organized as follows. Sec. 2 introduces related work and discusses their differences with our proposed method. The saliency com-

putation framework is presented in Sec. 3. Sec. 4 describes the regional saliency features adopted in this paper. Sec. 5 presents the learning framework of our approach. Empirical analysis of our proposed method and comparisons with other algorithms are demonstrated in Sec. 6. Finally, Sec. 7 discusses and concludes this paper.

2 Related work

Salient object detection, stemming from eye fixation prediction, aims to separate the entire salient object from the background. Since the pioneer work of Itti *et al.* [2], it attracts more and more research interests in computer vision, driven by applications such as content-aware image resizing [8], picture collage [10], etc. In the following, we focus on salient object detection (segmentation) and briefly review existing algorithms. A comprehensive survey can be found from a recent work [23]. A literature review of eye fixation prediction can be seen in [27], which also includes some analysis on salient object detection. We simply divide existing algorithms into two categories: unsupervised and supervised, according to if the groundtruth annotations of salient objects are adopted.

Unsupervised approaches. Most salient object detection algorithms characterize the uniqueness of a scene as salient regions following the center-surround contrast framework [2], where different kinds of features are combined according to the feature integration theory [22]. The multi-scale pixel contrast is studied in [19], [28]. The discriminant center-surround hypothesis is analyzed in [16], [29]. Color histograms, computed to represent the center and the surround, are used to evaluate the center-surround dissimilarity [19]. An information theory perspective is introduced to yield a sound mathematical formulation, computing the center-surround divergence based on feature statistics [18]. A cost-sensitive SVM is trained to measure the separability of a center region w.r.t. its surroundings [30]. The uniqueness can also be captured in a global scope by comparing a patch to its k nearest neighbors [17] or as its distance to the average patch over the image along the principal component axis coordinates [31].

The center-surround difference framework is also investigated to compute the saliency from region-based image representation. The multi-level image segmentation is adopted for salient object detection based on local regional contrast [32]. The global regional contrast is studied in [15], [21] as well. To further enhance the performance, saliency maps on hierarchical segmentations are computed and finally combined through a tree model via dynamic programming [33]. Both the color and textural global uniqueness are investigated in [34], [35].

Recently, Cheng *et al.* [36] propose a soft image abstraction using a Gaussian Mixture Model (GMM), where each pixel maintains a probability belonging to all the regions instead of a single hard region label, to better compute the saliency. The global uniqueness can also be captured with the low-rank matrix recovery framework [37]–[39]. The low-rank matrix corresponds to the background regions while sparse noises are indications of salient regions. A submodular salient object detection algorithm is presented in [40], where superpixels are gradually grouped to form potential salient regions by iteratively optimizing a submodular facility location problem. The Bayesian fraemwork is introduced for salient object detection in [41], [42]. A partial differential equation (PDE) is also introduced for salient object detection in a recent work [43].

In addition to capturing the uniqueness, many other priors are also proposed for saliency computation. Centeral prior, *i.e.*, the salient object usually lies in the center of an image, is investigated in [32], [44]. Object prior, such as connectivity prior [45], concavity context [20], and auto-context cue [46], backgroundness prior [47]–[50], generic objectness prior [51]–[53], and background connectivity prior [38], [54], [55] are also studied for saliency computation. Example-based approaches, searching for similar images of the input, are developed for salient object detection [8], [56]. The depth cue is leveraged for saliency analysis derived from stereopsic image pairs in [57] and a depth camera (*e.g.*, Kinect) in [58]. Li *et al.* [59] adopt the light field camera for salient object detection. Besides, spectral analysis in the frequency domain is used to detect salient regions [13].

Supervised approaches. Inspired by the feature integration theory, some approaches focus on learning the linear fusion weight of saliency features. Liu *et al.* [19] propose to learn the linear fusion weight of saliency features in a Conditional Random Field (CRF) framework. Recently, the large-margin framework was adopted to learn the weights in [60]. Due to the highly non-linear essence of the saliency mechanism, the linear mapping might not perfectly capture the characteristics of saliency. In [24], a mixture of linear Support Vector Machines (SVM) is adopted to partition the feature space into a set of sub-regions that were linearly separable using a divide-and-conquer strategy. Alternatively, a Boosted Decision Tree (BDT) is learned to get an initial saliency map, which will be further refined using a high dimensional color transform [61]. In [25], generic regional properties are investigated for salient object detection. Li *et al.* [62] propose to generate a saliency map by adaptively averaging the object proposals [63] with their foreground probabilities that are learned based on eye fixations features using the Random Forest regressor. Additionally, Wang *et al.* [64] learn

a Random Forest to directly localize the salient object on thumbnail images. In [65], a saliency map is used to guide the sampling of sliding windows for object category recognition, which is online learned during the classification process.

Our proposed discriminative regional feature integration (DRFI) approach is a supervised salient object detection algorithm. Compared with unsupervised methods, our approach extends the contrast value used in existing algorithms to the contrast vector to represent a region. More importantly, instead of designing heuristic integration rules, our approach is able to *automatically* combine the high-dimensional saliency features in a data-driven fashion and pick up the discriminative ones. Compared with existing supervised methods, our method learns a highly non-linear combination of saliency features and does not require any assumption of the feature space. The most similar approaches to ours might be [25], [61]. [25] is a light touch on the discriminative feature integration without presenting a deep investigation, which only considers the regional property descriptor. In [61], the learned saliency map is only used as a pre-processing step to provide a coarse estimation of salient and background regions while our approach directly output the saliency map.

It is noted that some supervised learning approaches exist to predict eye fixation [66], [67]. The features, *e.g.*, the local energy of the steerable pyramid lters [68] in [66] and the perceptual Gestalt grouping cues in [67], seem to be more suitable for eye fixation prediction, while our approach is specifically designed for salient object detection. We also note the discriminative feature fusion has also been studied in image classification [69], which learns the adaptive weights of features according to the classification task to better distinguish one class from others. Instead, our approach integrates three types of regional features in a discriminative strategy for the saliency regression on multiple segmentations.

3 Image Saliency Computation

The pipeline of our approach consists of three main steps: multi-level segmentation that decomposes an image into regions, regional saliency computation that maps the features extracted from each region to a saliency score, and multi-level saliency fusion that combines the saliency maps over all the layers of segmentations to get the final saliency map. The whole process is illustrated in Fig. 1.

Multi-level segmentation. Given an image I, we represent it by a set of M-level segmentations $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_M\}$, where each segmentation $\mathcal{S}_m$ is a decomposition of the image I. We apply the graph-based image segmentation approach [70] to generate multiple segmentations using M groups of different parameters.

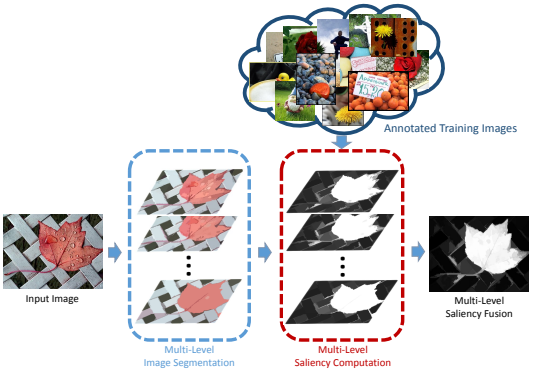


Fig. 1. The framework of our proposed discriminative regional feature integration (DRFI) approach.

Due to the limitation of low-level cues, none of the current segmentation algorithms can reliably segment the salient object. Therefore, we resort to the multi-level segmentation for robustness purpose. In Sec. 5.1, we will further demonstrate how to utilize multi-level segmentation to generate a large amount of training samples.

Regional saliency computation. In our approach, we predict saliency scores for each region that is jointly represented by three types of features: regional contrast, regional property, and regional backgroundness, which will be described in Sec. 4. At present, we denote the feature as a vector $\mathbf{x}$. Then the feature $\mathbf{x}$ is passed into a random forest regressor f, yielding a saliency score. The Random Forest regressor is learnt from the regions of the training images and integrates the features together in a discriminative strategy. The learning procedure will be given in Sec. 5.

Multi-level saliency fusion. After conducting region saliency computation, each region has a saliency value. For each level, we assign the saliency value of each region to its contained pixels. As a result, we generate M saliency maps $\{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_M\}$, and then fuse them together, $\mathbf{A} = g(\mathbf{A}_1, \cdots, \mathbf{A}_M)$, to get the final saliency map $\mathbf{A}$, where g is a combinator function introduced in Sec. 5.3.

4 Regional Saliency Features

In this section, we present three types of regional saliency features, leading to a 93-dimensional feature vector for each region.

4.1 Regional contrast descriptor

A region is likely thought to be salient if it is different from others. Unlike most existing approaches that compute the contrast values, *e.g.*, the distances of region features like color and texture, and then combine them together directly forming a saliency score, our approach computes a contrast descriptor, a vector representing the differences of feature vectors of regions.

Color and texture features			Differences of features		Contrast	Backgroundness
	features	dim	definition	dim		
$\mathbf{a}_1$	average RGB values	3	$d(\mathbf{a}_1^{R_i}, \mathbf{a}_1^S)$	3	$c_1 \sim c_3$	$b_1 \sim b_3$
$\mathbf{h}_1$	RGB histogram	256	$\chi^2(\mathbf{h}_1^{R_i}, \mathbf{h}_1^S)$	1	c_4	b_4
$\mathbf{a}_2$	average HSV values	3	$d(\mathbf{a}_2^{R_i}, \mathbf{a}_2^S)$	3	$c_5 \sim c_7$	$b_5 \sim b_7$
$\mathbf{h}_2$	HSV histogram	256	$\chi^2(\mathbf{h}_2^{R_i}, \mathbf{h}_2^S)$	1	c_8	b_8
$\mathbf{a}_3$	average L*a*b* values	3	$d(\mathbf{a}_3^{R_i}, \mathbf{a}_3^S)$	3	$c_9 \sim c_{11}$	$b_9 \sim b_{11}$
$\mathbf{h}_3$	L*a*b* histogram	256	$\chi^2(\mathbf{h}_3^{R_i}, \mathbf{h}_3^S)$	1	c_{12}	b_{12}
$\mathbf{r}$	absolute response of LM filters	15	$d(\mathbf{r}^{R_i}, \mathbf{r}^S)$	15	$c_{13} \sim c_{27}$	$b_{13} \sim b_{27}$
$\mathbf{h}_4$	max response histogram of the LM filters	15	$\chi^2(\mathbf{h}_4^{R_i}, \mathbf{h}_4^S)$	1	c_{28}	b_{28}
$\mathbf{h}_5$	histogram of the LBP feature	256	$\chi^2(\mathbf{h}_4^{R_i}, \mathbf{h}_5^S)$	1	c_{29}	b_{29}

Fig. 2. Color and texture features describing the visual characteristics of a region which are used to compute the regional feature vector. $d(\mathbf{x}_1, \mathbf{x}_2) = (|x_{11} - x_{21}|, \cdots, |x_{1d} - x_{2d}|)$ where d is the number of elements in the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. And $\chi^2(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{b} \frac{2(h_{1i}-h_{2i})^2}{h_{1i}+h_{2i}}$ with b being the number of histogram bins. The last two columns denote the symbols for regional contrast and backgroundness descriptors. (In the definition of a feature, S corresponds to R_j for the regional contrast descriptor and B for the regional backgroundness descriptor, respectively.)

To compute the contrast descriptor, we describe each region $R_i \in \mathcal{S}_m$ by a feature vector, including color and texture features, denoted by $\mathbf{v}^{R_i}$. The detailed description is given in Fig. 2. For color features, we consider RGB, HSV, and L*a*b* color spaces. For texture features, we adopt the LBP feature [71] and the responses of the LM filter bank [72].

As suggested in previous works [15], [21], the regional contrast value x_k^c derived from the k-th feature channel is computed by checking R_i against all other regions,

$$x_k^c(R_i) = \sum_{j=1}^{N_m} \alpha_j w_{ij} D_k(\mathbf{v}^{R_i}, \mathbf{v}^{R_j}), \quad (1)$$

where $D_k(\mathbf{v}^{R_i}, \mathbf{v}^{R_j})$ captures the difference of the k-th channel of the feature vectors $\mathbf{v}^{R_i}$ and $\mathbf{v}^{R_j}$. Specifically, the difference of the histogram feature is computed as the χ^2 distance and as their absolute differences for other features. $w_{ij} = e^{-\frac{||\mathbf{p}_i^m - \mathbf{p}_j^m||^2}{2\sigma_s^2}}$ is a spatial weighting term, where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the mean positions of R_i and R_j, respectively. σ_s controls the strength of the spatial weighting effect. We empirically set it as 1.0 in our implementation. α_j is introduced to account for the irregular shapes of regions, defined as the normalized area of the region R_j. N_m is the number of regions in $\mathcal{S}_m$. As a result, we get a 29-dimensional feature vector. The details of the regional contrast descriptor are given in Fig. 2.

4.2 Regional backgroundness descriptor

There exist a few algorithms attempting to make use of the characteristics of the background (*e.g.*, homogeneous color or textures) to heuristically determine if one region is background, *e.g.*, [47]. In contrast, our algorithm extracts a set of features and adopts the supervised learning approach to determine the background degree (accordingly the saliency degree) of a region.

It has been observed that the background identification depends on the whole image context. Image regions with similar appearances might belong to the background in one image but belong to the salient object in some other images. It is not enough to merely use the property features to check if one region is in the background or the salient object.

Therefore, we extract the pseudo-background region and compute the backgroundness descriptor for each region with the pseudo-background region as a reference. The pseudo-background region B is defined as the 15-pixel wide narrow border region of the image. To verify such a definition, we made a simple survey on the MSRA-B data set with 5000 images and found that 98% of pixels in the border area belongs to the background. The backgroundness value x_k^b of the region R_i on the k-th feature is then defined as

$$x_k^b(R_i) = D_k(\mathbf{v}^{R_i}, \mathbf{v}^B). \quad (2)$$

We get a 29-dimensional feature vector. See details in Fig. 2.

4.3 Regional property descriptor

Additionally, we consider the generic properties of a region, including appearance and geometric features. These two features are extracted independently from each region like the feature extraction algorithm in image labeling [73]. The appearance features attempt to describe the distribution of colors and textures in a region, which can characterize their common properties for the salient object and the background. For example, the background usually has homogeneous color distribution or similar texture pattern. The geometric features include the size and position of a region that may be useful to describe the spatial distribution of the salient object and the background. For example, the salient object tends to be placed near the center of the image while the background usually scatters over the entire image. Finally, we obtain a 35-dimensional regional property descriptor. The details are given in Fig. 3.

In summary, we obtain a 93-dimensional ($2 \times 29 + 35$) feature vector for each region. Fig. 4 demonstrates

description	notation	dim
average normalized x coordinates	p_1	1
average normalized y coordinates	p_2	1
$10th$ percentile of the normalized x coord.	p_3	1
$10th$ percentile of the normalized y coord.	p_4	1
$90th$ percentile of the normalized x coord.	p_5	1
$90th$ percentile of the normalized y coord.	p_6	1
normalized perimeter	p_7	1
aspect ratio of the bounding box	p_8	1
variances of the RGB values	$p_9 \sim p_{11}$	3
variances of the L*a*b* values	$p_{12} \sim p_{14}$	3
variances of the HSV values	$p_{15} \sim p_{17}$	3
variance of the response of the LM filters	$p_{18} \sim p_{32}$	15
variance of the LBP feature	p_{33}	1
normalized area	p_{34}	1
normalized area of the neighbor regions	p_{35}	1

Fig. 3. The regional property descriptor. (The abbreviation coord. indicates coordinates.)

visualizations of the most important features for each kind of regional feature descriptor.

5 Learning

In this section, we introduce how to learn a Random Forest to map the feature vector of each region to a saliency score. Learning the multi-level saliency fusion weight is also presented.

5.1 Generating training samples

We use supervised multi-level segmentation to generate training samples. We first learn the similarity score of each adjacent regions, to show the probability that the adjacent regions both belong to the salient region or the background. Similar regions will be grouped together in a hierarchical way. Training samples of the saliency regressor are those confident regions in the grouping hierarchy.

By learning the similarity score, we hope that those regions from the object (or background) are more likely to be grouped together. In specific, given an over-segmentation of an image, we connect each region and its spatially-neighboring regions forming a set of pairs $\mathcal{P} = \{(R_i, R_j)\}$ and learn the probability $p(a_i = a_j)$, where a_i is the saliency label of the region R_i. Such a set of pairs into two parts: a positive part $\mathcal{P}^+ = \{(R_i, R_j)|a_i = a_j\}$ and a negative part $\mathcal{P}^- = \{(R_i, R_j)|a_i \neq a_j\}$. Following [74], each region pair is described by a set of features including the regional saliency of two regions (2×93 features), the feature contrast of two regions (similar to the regional contrast descriptor, 29 features), and the geometry features of the superpixel boundary of two regions (similar to $p_1 \sim p_7$ in Fig. 3, 7 features). Given these 222-dimensional feature, we learn a boosted decision tree classifier to estimate the similarity score of each adjacent region pair.

Based on the learned similarity of two adjacent regions, we produce multi-level segmentation $\{\mathcal{S}_1^t, \mathcal{S}_2^t, \ldots, \mathcal{S}_M^t\}$ to gather a large mount of training samples. Specifically, denote $\mathcal{S}_0^t$ as the over-segmentation of the image generated using the graph-based image segmentation algorithm [70]. The regions in $\mathcal{S}_0^t$ are represented by a weighted graph, which connects the spatially neighboring regions. The weight of each edge is the learned similarity of two adjacent superpixels. Similar to the pixel-wise grouping in [70], pairs of regions are sequentially merged in the order of decreasing the weights of edges. We change the tolerance degree of small regions, *i.e.*, the parameter k of the approach [70] (see the details in [70]) to generate the segmentations from $\mathcal{S}_1^t$ to $\mathcal{S}_M^t$. To avoid too fine groupings, we discard $\mathcal{S}_i^t$ if $\frac{|\mathcal{S}_0^t|}{|\mathcal{S}_i^t|} > 0.6$, where $|\cdot|$ denotes the number of superpixels.

Fig. 4. Illustration of the most important features. From top to bottom: input images, the most important contrast feature (c_{12}), the most important backgroundness feature (b_{12}), the most important property feature (p_5), and the saliency map of our approach (DRFIs) produced on a single-level segmentation. Brighter area indicates larger feature value (thus larger saliency value according to c_{12}, b_{12} and the saliency map).

Given a set of training images with ground truth annotations and their multi-level segmentation, we can collect lots of confident regions $\mathcal{R} = \{R^{(1)}, R^{(2)}, \cdots, R^{(Q)}\}$ and the corresponding saliency scores $\mathcal{A} = \{a^{(1)}, a^{(2)}, \cdots, a^{(Q)}\}$ to learn a Random Forest saliency regressor. Only confident regions are kept for training since some regions may contain pixels from both the salient object and background. A region is considered to be confident if the number of pixels belonging to the salient object or the back-

ground exceeds 80% of the total number of pixels in the region. Its saliency score is set as 1 or 0 accordingly. In experiments we find that few regions of all the training examples, around 6%, are unconfident and we discard them from training.

One benefit to generate multi-level segmentation is that a large amount of training samples can be gathered. In the Sec. 6.3, we empirically analyze different settings of M_t and validate our motivation to generate training samples based on multi-level image segmentation. Additionally, with the guiding of learned similarity metric, there might be only a few large regions left in the high-level segmentation, which are helpful for the Random Forest regressor to learn object-level properties. However, the learned similarity is hard to generalize across datasets. This is why our approach did not perform the best on SED2 dataset in our pervious version [1]. To this end, we adopt the unsupervised multi-level segmentation in the testing phrase, which is also more efficient without learning the similarity score.

5.2 Learning the regional saliency regressor

Our aim is to learn the regional saliency estimator from a set of training examples. As aforementioned, each region is described by a feature vector $\mathbf{x} \in \mathbb{R}^d$, composed of the regional contrast, regional property, and regional backgroundness descriptors (*i.e.*, $d = 93$). From the training data $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(Q)}\}$ and the saliency scores $\mathcal{A} = \{a^{(1)}, a^{(1)}, \cdots, a^{(Q)}\}$, we learn a random forest regressor $f_r : \mathbb{R}^d \to \mathbb{R}$ which maps the feature vector of each region to a saliency score.

A Random Forest saliency regressor is an ensemble of T decision trees, where each tree consists of split and leaf nodes. Each split node stores a feature index f and a threshold τ. Given a feature vector $\mathbf{x}$, each split node in the tree makes a decision based on the feature index and threshold pair (f, τ). If $\mathbf{x}(f) < \tau$ it traverses to the left child, otherwise to the right child. When reaching a leaf node, its stored prediction value will be given. The final prediction of the forest is the average of the predictions over all the decision trees.

Training a Random Forest regressor is to independently build each decision tree. For each tree, the training samples are randomly drawn *with replacement*, $\mathcal{X}_t = \{\mathbf{x}^{(t_1)}, \mathbf{x}^{(t_2)}, \cdots, \mathbf{x}^{(t_Q)}\}$, $\mathcal{A}_t = \{a^{(t_1)}, a^{(t_2)}, \cdots, a^{(t_Q)}\}$, where $t_i \in [1, Q], i \in [1, Q]$. Constructing a tree is to find the pair (f, τ) for each split node and the prediction value for each leaf node. Starting from the root node, m features $\mathcal{F}_m$ are randomly chosen *without replacement* from the full feature vector. The best split will be found among these features $\mathcal{F}_m$ to maximize the splitting criterion

$$(f^*, \tau^*) = \max_{f \in \mathcal{F}_m, \tau} \left(\frac{\sum_{t_i \in \mathcal{D}_l} \left(a^{(t_i)}\right)^2}{|\mathcal{D}_l|} + \frac{\sum_{t_i \in \mathcal{D}_r} \left(a^{(t_i)}\right)^2}{|\mathcal{D}_r|} - \frac{\sum_{t_i \in \mathcal{D}} \left(a^{(t_i)}\right)^2}{|\mathcal{D}|} \right), \quad (3)$$

where $\mathcal{D}_l = \{(\mathbf{x}^{(t_i)}, a^{(t_i)}) | \mathbf{x}^{(t_i)}(f) < \tau\}$, $\mathcal{D}_r = \{(\mathbf{x}^{(t_i)}, a^{(t_i)}) | \mathbf{x}^{(t_i)}(f) \geq \tau\}$, and $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_r$. Such a spliting procedure is repeated until $|D| < 5$ and a leaf node is created. The prediction value of the leaf node is the average saliency scores of the training samples falling in it. We will empirically examine the settings of parameters T and m in Sec. 6.3.

Learning a saliency regressor can automatically integrate the features and discover the most discriminative ones. Additionally, in the training procedure of the random forest, the feature importance can be estimated simultaneously. Refer to the supplementary for more details. Fig. 6 presents the most important 60 features.

5.3 Learning the multi-level saliency fusor

Given the multi-level saliency maps $\{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_M\}$ for an image, our aim is to learn a combinator $g(\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_M)$ to fuse them together to form the final saliency map $\mathbf{A}$. Such a problem has been already addressed in existing methods, such as the conditional random field solution [19]. In our implementation, we find that a linear combinator, $\mathbf{A} = \sum_{m=1}^{M} w_m \mathbf{A}_m$, performs well by learning the weights using a least square estimator, *i.e.*, minimizing the sum of the losses ($\|\mathbf{A} - \sum_{m=1}^{M} w_m \mathbf{A}_m\|_F^2$) over all the training images.

In practice, we found that the average of multi-level saliency maps performs as nearly well as a weighted average.

6 Experiments

In this section, we empirically analyze our proposed approach. Comparisons with state-of-the-art methods on benchmark data sets are also demonstrated.

6.1 Data sets

We evaluate the performance over five data sets that are widely used in salient object detection and segmentation.

MSRA-B.[1] This data set [19] includes 5000 images, originally containing labeled rectangles from nine users drawing a bounding box around what they consider the most salient object. There is a large variation among images including natural scenes, animals, indoor, outdoor, etc. We manually segment the salient

1. http://research.microsoft.com/en-us/um/people/jiansun/

object (contour) within the user-drawn rectangle to obtain binary masks. The ASD data set [13] is a subset (with binary masks) of MSRA-B, and thus we no longer make evaluations on it.

iCoSeg.[2] This is a publicly available co-segmentation data set [75], including 38 groups of totally 643 images. Each image is along with a pixel-wise groundtruth annotation, which may contain one or multiple salient objects. In this paper, we use it to evaluate the performance of salient object detection.

SED.[3] This data set [76] contains two subsets: SED1 that has 100 images containing only one salient object and SED2 that has 100 images containing exactly two salient objects. Pixel-wise groundtruth annotations for the salient objects in both SED1 and SED2 are provided. We only make evaluations on SED2. Similar to the larger MSRA-B dataset, only single one salient object exists in each image in SED1, where state-of-the-art performance was reported in our previous version [1]. Additionally, evaluations on SED2 may help us check the adaptability of salient object detection algorithms on multiple-object cases.

ECSSD.[4] To overcome the weakness of existing data set such as ASD, in which background structures are primarily simple and smooth, a new data set denoted as Extended Complex Scene Saliency Dataset (ECSSD) is proposed recently in [77]. It contains 1000 images with diversified patterns in both foreground and background, where many semantically meaningful but structurally complex images are available. Binary masks for salient objects are produced by 5 subjects.

DUT-OMRON.[5] Similarly, this dataset is also introduced to evaluate salient object detection algorithms on images with more than a single salient object and relatively complex background. It contains 5,168 high quality natural images, where each image is resized to have a maximum side length of 400 pixels. Annotations are available in forms of both bounding boxes and pixel-wise binary object masks. Furthermore, eye fixation annotations are also provided makeing this dataset suitable for simultaneously evaluating salient object localization and detection models as well as fixation prediction models.

We randomly sample 3000 images from the MSRA-B data set to train our model. Five-fold cross validation is run to select the parameters. The remaining 2000 images are used for testing. Rather than training a model for each data set, we use the model trained from the MSRA-B data set and test it over others. This can help test the adaptability to other different data sets of the model trained from one data set and avoid the model overfitted to a specific one.

2. http://chenlab.ece.cornell.edu/projects/touch-coseg
3. http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/
4. http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency
5. http://ice.dlut.edu.cn/lu/dut-omron/homepage.htm

6.2 Evaluation Metrics

We evaluate the performance using the measures used in [23] based on the overlapping area between groundtruth annotation and saliency prediction, including the PR (precision-recall) curve, the ROC (receiver operating characteristic) curve and the AUC (Area Under ROC Curve) score. Precision corresponds to the percentage of salient pixels correctly assigned, and recall is the fraction of detected salient pixels belonging to the salient object in the ground truth.

For a grayscale saliency map, whose pixel values are in the range $[0, 255]$, we vary the threshold from 0 to 255 to obtain a series of salient object segmentations. The PR curve is created by computing the precision and recall values at each threshold. The ROC curve can also be generated based on true positive rates and false positive rates obtained during the calculation of the PR curve.

6.3 Parameters Analysis

In this section, we empirically analyze the performance of salient object detection against the settings of parameters during both training and testing phrases. Since we want to test the cross-data generalization ability of our approach, we run five-fold cross-validation on the training set. Settings of parameters are thus blind to other testing data and fair comparisons with other approaches can be conducted. Average AUC scores resulting from cross-validation under different parameter setting are plotted in Fig. 5.

Training parameters analysis.. There are three parameters during training, number of segmentations M_t to generate training samples, number of trees T and number of randomly chosen features m when training the Random Forest regressor.

Larger number of segmentations lead to larger amount of training data. As a classifier usually benefits more from greater quantity of training samples, we can observe from Fig. 5(a) that the performance steadily increase when M_t becomes larger. We finally set $M_t = 48$ to generate around 1.7 million samples to train our regional Random Forest saliency regressor.

As shown in Fig. 5(b), the performance of our approach with more trees in the Random Forest saliency regressor is higher. The more trees there are, the less variances are among the decision trees, and thus the better performance can be achieved. Though the performance keeps increasing as more trees adopted, we choose to set $T = 200$ trees to train the regressor to balance the efficiency and the effectiveness.

When splitting each node during the construction a decision tree, only m randomly chosen features can be observed . Intuitively, on one hand, increasing m will give the node greater chance to select more discriminative features. On the other hand, however, larger m will bring smaller variances between decision trees. For instance, suppose m is set to the

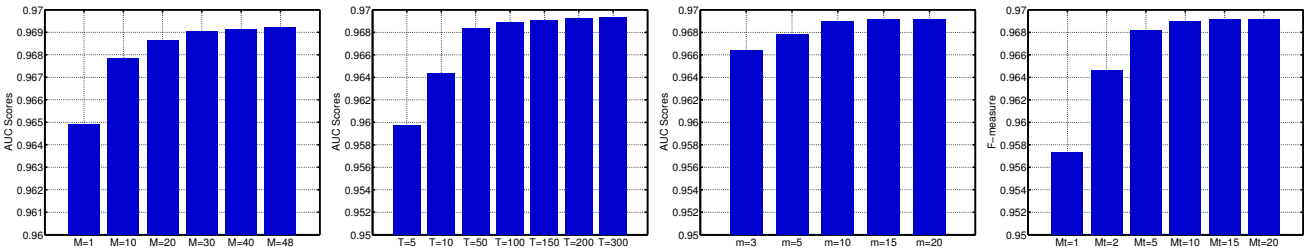


Fig. 5. Empirical analysis of parameters in terms of the AUC scores based on five-fold cross-validation of the training set. From left to right, (a) AUC scores versus the number of segmentations to generate training samples, (b)(c) AUC scores versus number of decision trees and number of randomly chosen features at each node in the Random Forest saliency regressor, and (d) number of segmentations in generating saliency maps.

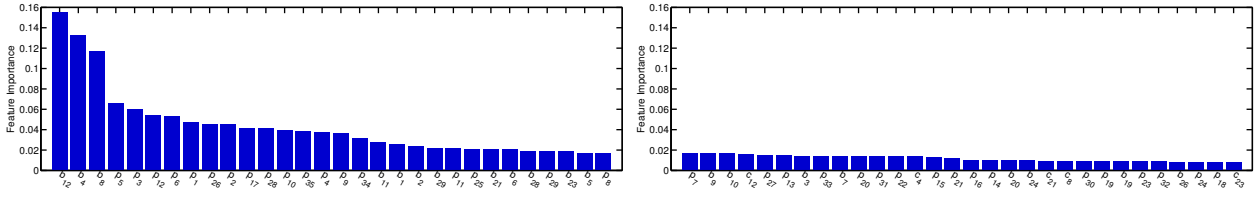


Fig. 6. The most important 60 regional saliency features given by Random Forest regressor, occupying around 90% of the energy of total features. There are 5 contrast features, 20 backgroundness features, and 35 property features. From left to right: the first and second 30 important features, respectively. See Fig. 2 and Fig. 3 for the description of the features.

dimension of the feature vector, *i.e.*, all of the features can be seen during the splitting, most likely the same most discriminative feature will be chosen at each split node. Consequently, nearly identical decision trees are built that can not complement each other well and thus may result in inferior performance. According to Fig. 5(c), we empirically set $m = 15$ since the performance is the best.

Testing parameters analysis. One can see in Fig. 5(d) that the AUC scores of the saliency maps increase when more layers of segmentations are adopted. The reason is that there may exist some confident regions that cover the most (even entire) part of an object in more layers of segmentations. However, a larger number of segmentations introduce more computational burden. Therefore, to balance the efficiency and the effectiveness, we set M to 15 segmentations in our experiments.

6.4 Feature Importance

Our approach uses a wide variety of features. In this section, we empirically analyze the usefulness of these regional saliency features. Fig. 6 shows the rank of the most important 60 regional features produced during the training of Random Forest regressor, which occupy around 90% of the energy of total features.

The feature rank indicates that the property descriptor is the most critical one in our feature set (occupies 35 out of top 60 features). The reason might be that salient objects share some common properties, validating our motivation to exploit the generic regional properties for salient object detection that are widely studied in other tasks such as image classification [26]. For example, high rank of geometric features p_5, p_3 and p_6 might correspond to the compositional bias of salient objects. The importance of variance features p_{12} and p_{26}, on the other hand, might be related with the background properties. Among the contrast-based descriptors, regional contrast descriptor is the least important. Since it might be affected by cluttered scenes and less important compared with the regional backgroundness descriptor which is in some sense more robust. Moreover, we also observe that color features are much more discriminative than texture features.

To further validate the importance of features across different data sets, we train classifiers by removing each kind of feature descriptor on each benchmark data set (testing set of MSRA-B). AUC scores of saliency maps are demonstrated in Fig. 7. As can be seen, removing some feature descriptors does not necessarily lead to performance decrease. Consistent with the feature rank given by the Random Forestregressor, regional contrast descriptor is the least important one on most of the benchmark data sets as least performance drop are observed with its removal on most of the data sets. Regional property descriptor still plays the most important role on MSRA-B, ECSSD and DUT-OMRON. Since there are multiple salient objects in an image in SED2 and iCoSeg, the common properties learned from the training data, where only a single salient object exists in most of the images,

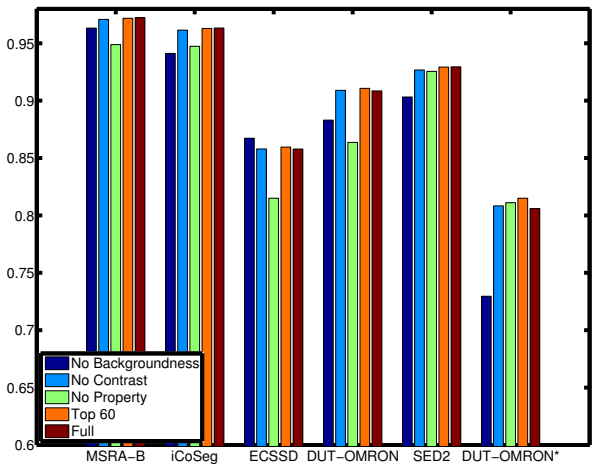


Fig. 7. Feature importance across different data sets. For each data set, we report the AUC scores of saliency maps by removing each kind of descriptor to see the performance drop. Additionally, we also demonstrate the performance exploiting only the top 60 features shown in Fig. 6.

	MSRA-B	iCoSeg	ECSSD	DUT-OMRON	SED2	DUT-OMRON*
SVO	0.899	0.861	0.799	0.866	0.834	0.793
CA	0.860	0.837	0.738	0.815	0.854	0.760
CB	0.930	0.852	0.819	0.831	0.825	0.624
RC	0.937	0.880	0.833	0.859	0.840	0.679
SF	0.917	0.911	0.777	0.803	0.872	0.715
LRK	0.925	0.908	0.810	0.859	0.881	0.758
HS	0.930	0.882	0.829	0.860	0.820	0.735
GMR	0.942	0.902	0.834	0.853	0.831	0.646
PCA	0.938	0.895	0.817	0.887	0.903	0.776
MC	0.951	0.898	0.849	0.887	0.863	0.715
DSR	0.956	0.921	0.856	0.899	0.895	0.776
RBD	0.945	0.941	0.840	0.894	0.873	0.779
DRFIs	0.954	0.944	0.858	0.910	0.902	0.804
DRFI	0.971	0.968	0.875	0.931	0.933	0.822

Fig. 8. AUC: area under ROC curve (larger is better). The best three results are highlighted with red, green, and blue fonts, respectively.

may not perform the best. The backgroundness descriptor performs well on MSRA-B, SED2, and iCoSeg. However, it plays the least important role on ECSSD data set. Its removal even leads to better performance, which indicates the pseudo-background assumption might not always hold well. Finally, instead of considering all of the 93 features, we also adopt only the top 60 features for training. Surprisingly, this feature vector performs as well as the entire feature descriptor, even slightly better on DUT-OMRON, implying some features contribute little.

We visualize the most important features of each descriptor in Fig. 4. As we can see, even the most powerful backgroundness feature provide far less accurate information of salient objects. By integrating all of the weak information, much better saliency maps can be achieved. Note that we do not adopt the multi-level fusion enhancement. Another advantages of our approach is the automatic fusion of features. For example, the rules to employ geometric features are discovered from the training data instead of heuristically defined as previous approaches [32], which might be poor to generalize.

6.5 Performance Comparison

We report both quantitative and qualitative comparisons of our approach with state-of-the-art methods. To save the space, we only consider the top four models ranked in the survey [23]: SVO [51], CA [17], CB [32], and RC [15] and recently-developed methods: SF [21], LRK [78], HS [33], GMR [48], PCA [31], MC [50], DSR [49], RBD [55] that are not covered in [23]. Note that we compare our approach with the extended version of RC. In total, we make comparisons with 12 approaches. Additionally, we also report the performance of our DRFI approach with a single layer (DRFIs).

Quantitative comparison.. Quantitative comparisons are shown in Fig. 8, Fig. 9 and Fig. 10. As can be seen, our approach (DRFI) consistently outperforms others on all benchmark data sets with large margins in terms of AUC scores, PR and ROC curves. In specific, it improves by 1.57%, 2.66%, 2.34%, 3.45% and 3.21% over the best-performing state-of-the-art algorithm according to the AUC scores on MSRA-B, iCoSeg, ECSSD, DUT-OMRON, and SED2, respectively.

Our single-level version (DRFIs) performs best on iCoSeg, ECSSD, and DUT-OMRON as well. It improves by 0.32%, 0.23%, and 1.22% over the best-performing state-of-the-art method in terms of AUC scores on these three data sets, respectively. It is slightly worse (but still one of the top 3 best models) on MSRA-B and SED2 data set. Such improvement is substantial by considering the already high performance of state-of-the-art algorithms. More importantly, though the Random Forest regressor is trained on MSRA-B, it performs best on other challenging data sets like ECSSD and DUT-OMRON.

With the multi-level enhancement, performance of our approach can be further improved. For instance, it improves by 1.22% on MSRA-B and 1.78% on DUT-OMRON.

Qualitative comparison. We also provide the qualitative comparisons of different methods in Fig. 11. As can be seen, our approach (shown in Fig. 11 (n)(o)) can deal well with the challenging cases where the background is cluttered. For example, in the first two rows, other approaches may be distracted by the textures on the background while our method almost successfully highlights the entire salient object. It is also worth pointing out that our approach performs

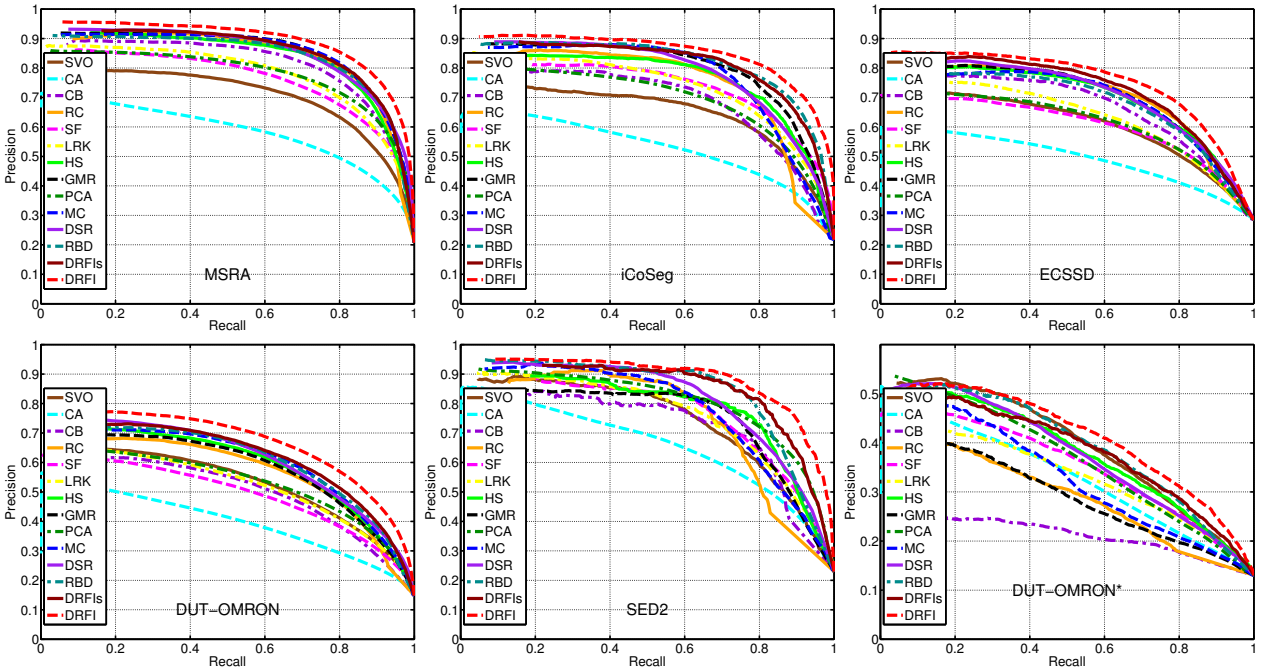


Fig. 9. Quantitative comparisons of saliency maps produced by different approaches on different data sets in terms of PR curves. See supplemental materials for more evaluations.

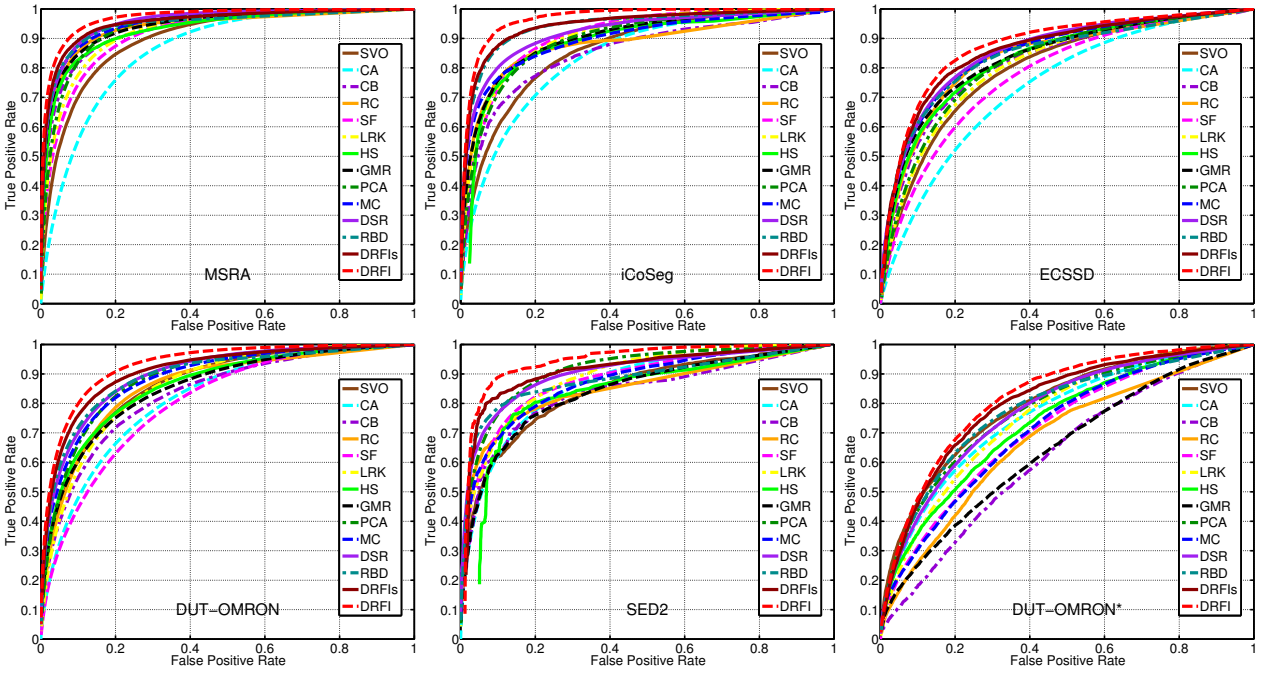
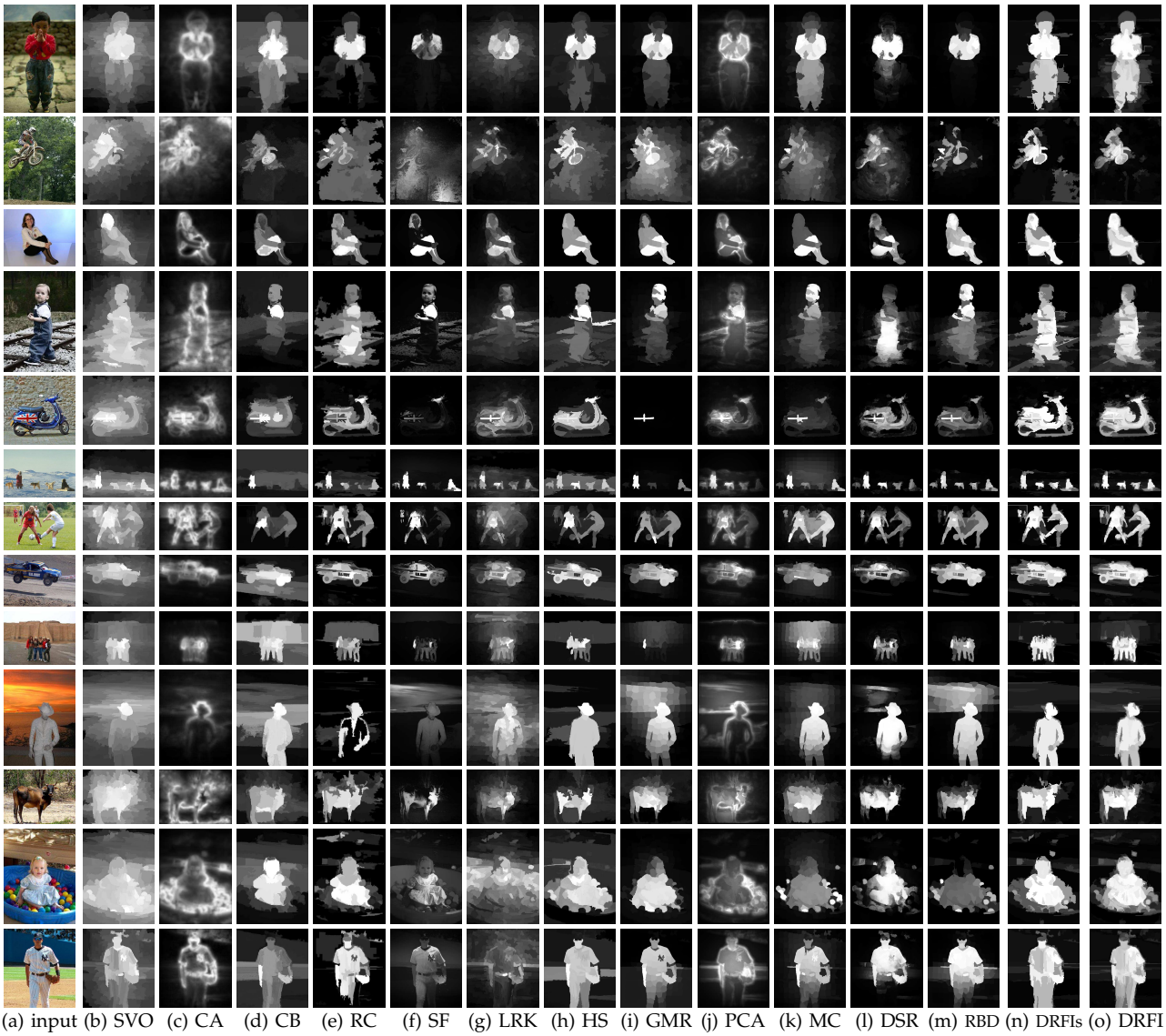


Fig. 10. Quantitative comparison of saliency maps produced by different approaches on different data sets in terms of ROC curves. See supplemental materials for more evaluations.

well when the object touches the image border, *e.g.*, the first and last third rows in Fig. 11, even though it violates the pseudo-background assumption. With the multi-level enhancement, more appealing results can be achieved.

6.6 Robustness Analysis

As suggested by Fig. 6 and Fig. 7, the region backgroundness and property descriptors, especially the geometric properties, play important roles in our approach. In natural images, the pseudo-background assumption may not be held well. Additionally, the distributions of salient objects may be different from our training set. It is natural to doubt that whether



(a) input (b) SVO (c) CA (d) CB (e) RC (f) SF (g) LRK (h) HS (i) GMR (j) PCA (k) MC (l) DSR (m) RBD (n) DRFIs (o) DRFI

Fig. 11. Visual comparison of the saliency maps. Our method (DRFI) consistently generates better saliency maps.

our approach can still perform well on these challenging cases. To this end, we select 635 images from DUT-OMRON dataset (we call it DUT-OMRON*), where salient objects touch the image border and are far from the image center. Quantitative comparisons with state-of-the-art approaches are presented in Fig. 8, Fig. 9, and Fig. 10. Check our project website jianghz.com/drfi and the supplementary material for more details.

Not surprisingly, performances of all approaches decline. But our approach DRFI still significantly outperforms other methods in terms of PR curve, ROC curve and AUC scores. Even with a single level, our approach DRFIs performs slightly better than others (ranked as the second best in terms of AUC scores). In specific, DRFIs and DRFI are better than the top-performing method by around 2.22% and 1.36% (compared with 2.20% and 1.26% on the whole DUT-OMRON data set) according to the AUC scores.

6.7 Efficiency

Since the computation on each level of the multiple segmentations is independent, we utilize the multi-thread technique to accelerate our C++ code. Fig. 12 summarizes the running time of different approaches, tested on the MSRA-B data set with a typical 400×300 image using a PC with an Intel i5 CPU of 2.50GHz and 8GB memory. 8 threads are utilized for acceleration. As we can see, our approach can run as fast as most existing approaches. If equipped as a pre-processing step for an application, *e.g.*, picture collage, our approach will not harm the user experiences.

For training, it takes around 24h with around 1.7 million training samples. As training each decision tree is also independent to each other, parallel computing techniques can also be utilized for acceleration.

Method	SVO	CA	CB	RC	SF	LRK	HS
Time(s)	56.5	52.3	1.40	0.138	0.210	11.5	0.365
Code	M+C	M+C	M+C	C	C	M+C	EXE
Method	**GMR**	**PCA**	**MC**	**DSR**	**RBD**	**DRFIs**	**DRFI***
Time(s)	1.16	2.07	0.129	4.19	0.267	0.183	0.418
Code	M+C	M+C	M+C	M+C	M	C	C

Fig. 12. Comparison of running time. M indicates the code is written in MATLAB and EXE is corresponding to the executable. (*8 threads are used for acceleration.)

Fig. 13. Failure cases of our approach.

7 Discussions and Future Work

7.1 Unsupervised vs Supervised

As data-driven approaches, especially supervised learning methods, dominate other fields of computer vision, it is somewhat surprising that the potential of supervised salient object detection is relatively under exploited. The main research efforts of salient object detection still concentrate on developing heuristic rules to combine saliency features. Note we are not saying that heuristic models are useless. We instead favor the supervised approaches for following two advantages. On one hand, supervised learning approaches can *automatically* fuse different kinds of saliency features, which is valuable especially when facing high-dimensional feature vectors. It is nearly infeasible for humans, even domain experts, to design rules to integrate the 93 dimensional feature vector of this paper. For example, the sixth important feature p_{12} (variance of the L* values of a region) seems to be rather obscure for salient object detection. Yet integration rules discovered from training samples indicate it is highly discriminative, more than the traditional regional contrast features.

On the other hand, data-driven approaches always own much better generalization ability than heuristic methods. In a recent survey of salient object detection [23], none of existing unsupervised algorithms can consistently outperforms others on all benchmark data sets since different pre-defined heuristic rules favor different settings of the data set (*e.g.*, the number of objects, center bias, etc). Leveraging the large amount of training samples (nearly two milliion), our learned regional saliency regressor almost performs the best over all six benchmark data sets. Even though it is trained on a single data set, it performs better than others on challenging cases which are significantly different from the training set.

One potential reason that learning-based approaches is not favored for salient object detection might be related to the efficiency. Though a lot of training time is required to train a classifier, testing time is more likely to be the major concern of a system rather than the offline training time. Once trained, the classifier can be used as an off-shelf tool. In this paper, we demonstrate that a learning-based approach can run as fast as some heuristic methods.

7.2 Limitations of Our Approach

Since our approach mainly consider the regional contrast and backgroudness features, it may fail on cluttered scenes. See Fig. 13 for illustration. In the first column, high saliency values are assigned to the texture ares in the background as they are distinct in terms of either contrast or background features. The salient object in the second column have similar color with the background and occupies a large portion of the image, making it challenging to generate good detection result. For the third column, it is not fair to say that our approach completely fails. The flag is indeed salient. However, as the statue violates the pseudo-background assumption and occupying large portions as well, it is difficult to generate an appealing saliency map using our approach.

7.3 Conclusion and Future Work

In this paper, we address the salient object detection problem using a discriminative regional feature integration approach. The success of our approach stems from the utilization of supervised learning algorithm: we learn a Random Forest regressor to automatically integrate a high-dimensional regional feature descriptor to predict the saliency score and automatically discover the most discriminative features. Experimental results validate that compared with traditional approaches, which heuristically compute saliency maps from different types of features

Our approach is closely related to the image labeling method [26]. The goal is to assign predefined labels (geometric category in [26] and object or background in salient object detection) to the pixels. It needs further study to investigate the connection between image labeling and salient object, and if the two problems are essentially equivalent. Utilizing the data-driven image labeling approaches for salient object detection is also worth exploring in the future.

Additionally, there exist some obvious directions to further improve our approach.

- Incorporating more saliency features. In this paper, we consider only contrast, backgroundness, and generic property features of a region. By considering more saliency features, better detection

results can be expected. For example, background connectivity prior [55] can be incorporated to relax the pseudo-background assumption. Additionally, spatial distribution prior [19], [21], focusness prior [52], diverse density score [53] based on generic objectness, and graph-based manifold ranking score [48] can also be integrated.

- Better fusion strategy. We simply investigate the linear fusion of saliency maps with any post optimization step. As a future work, we can utilize the optimization step of other approaches to enhance the performance. For example, we can run saliency detection on hierarchical detections and fuse them as suggested in [77]. The optimization method proposed in [55] is also applicable.
- Integrating more cues. A recent trend on salient object detection is to integrate more cues in addition to traditional RGB data. Our approach is natural to be extedned to consider cues such as depth on RGB-D input, temporal consistency on video sequences, and saliency co-occurrence for co-salient object detection.

ACKNOWLEDGEMENTS

This work was supported in part by the National Basic Research Program of China under Grant No. 2015CB351703 and 2012CB316400, and the National Natural Science Foundation of China under Grant No. 91120006.

REFERENCES

[1] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE CVPR*, 2013, pp. 2083–2090.
[2] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE TPAMI*, 1998.
[3] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *ACM Multimedia*, 2003.
[4] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *CVPR*, vol. 0, pp. 1–8, 2007.
[5] C. Kanan and G. W. Cottrell, "Robust classification of objects, faces, and flowers using natural image statistics," in *CVPR*, 2010, pp. 2472–2479.
[6] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.
[7] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE TIP*, 2004.
[8] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *ICCV*, 2009, pp. 2232–2239.
[9] S. Goferman, A. Tal, and L. Zelnik-Manor, "Puzzle-like collage," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 459–468, 2010.
[10] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum, "Picture collage," in *CVPR (1)*, 2006, pp. 347–354.
[11] P. Wang, D. Zhang, J. Wang, Z. Wu, X.-S. Hua, and S. Li, "Color filter for image search," in *ACM Multimedia*, 2012.
[12] P. Wang, D. Zhang, G. Zeng, and J. Wang, "Contextual dominant color name extraction for web image search," in *ICME Workshops*, 2012, pp. 319–324.
[13] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009.
[14] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," in *CVPR*, 2012, pp. 478–485.
[15] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE TPAMI*, 2014.
[16] D. Gao, V. Mahadevan, and N. Vasconcelos, "The discriminant center-surround hypothesis for bottom-up saliency," in *NIPS*, 2007.
[17] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *CVPR*, 2010, pp. 2376–2383.
[18] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *ICCV*, 2011.
[19] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE TPAMI*, vol. 33, no. 2, pp. 353–367, 2011.
[20] Y. Lu, W. Zhang, H. Lu, and X. Xue, "Salient object detection using concavity context," in *ICCV*, 2011, pp. 233–240.
[21] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *CVPR*, 2012, pp. 733–740.
[22] A. Treisman and G. Gelad, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.
[23] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *ECCV (2)*, 2012, pp. 414–429.
[24] P. Khuwuthyakorn, A. Robles-Kelly, and J. Zhou, "Object of interest detection by saliency learning," in *ECCV*, 2010.
[25] P. Mehrani and O. Veksler, "Saliency segmentation based on learning and graph cut refinement," in *BMVC*, 2010.
[26] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *IJCV*, vol. 75, no. 1, pp. 151–172, 2007.
[27] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.
[28] F. Liu and M. Gleicher, "Region enhanced scale-invariant saliency detection," in *ICME*, 2006, pp. 1477–1480.
[29] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," in *ICCV*, 2007, pp. 1–6.
[30] X. Li, Y. Li, C. Shen, A. R. Dick, and A. van den Hengel, "Contextual hypergraph modeling for salient object detection," in *ICCV*, 2013, pp. 3328–3335.
[31] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *CVPR*, 2013.
[32] 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.
[33] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013.
[34] C. Scharfenberger, A. Wong, K. Fergani, J. S. Zelek, and D. A. Clausi, "Statistical textural distinctiveness for salient region detection in natural images," in *CVPR*, 2013, pp. 979–986.
[35] K. Shi, K. Wang, J. Lu, and L. Lin, "Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors," in *CVPR*, 2013, pp. 2115–2122.
[36] 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, pp. 1529–1536.
[37] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *CVPR*, 2012.
[38] W. Zou, K. Kpalma, Z. Liu, J. Ronsin *et al.*, "Segmentation driven low-rank matrix recovery for saliency detection," in *BMVC*, 2013, pp. 1–13.
[39] H. Peng, B. Li, R. Ji, W. Hu, W. Xiong, and C. Lang, "Salient object detection via low-rank and structured sparse matrix decomposition," in *AAAI*, 2013.
[40] Z. Jiang and L. S. Davis, "Submodular salient region detection," in *CVPR*, 2013, pp. 2043–2050.
[41] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *ECCV (5)*, 2010, pp. 366–379.
[42] Y. Xie, H. Lu, and M.-H. Yang, "Bayesian saliency via low and mid level cues," *IEEE TIP*, vol. 22, no. 5, pp. 1689–1698, 2013.
[43] R. Liu, J. Cao, G. Zhong, Z. Lin, S. Shan, and Z. Su, "Adaptive partial differential equation learning for visual saliency detection," in *CVPR*, 2014.
[44] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li, "Salient object detection for searched web images via global saliency," in *CVPR*, 2012, pp. 3194–3201.
[45] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *CVPR*, 2008.

[46] L. Wang, J. Xue, N. Zheng, and G. Hua, "Automatic salient object extraction with contextual cue," in *ICCV*, 2011.
[47] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *ECCV (3)*, 2012, pp. 29–42.
[48] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *CVPR*, 2013.
[49] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *ICCV*, 2013.
[50] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing markov chain," in *ICCV*, 2013.
[51] 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, pp. 914–921.
[52] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by ufo: Uniqueness, focusness and objectness," in *ICCV*, 2013.
[53] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *ICCV*, 2013.
[54] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *ICCV*, 2013, pp. 153–160.
[55] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *CVPR*, 2014.
[56] M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. A. Rowley, "Image saliency: From intrinsic to extrinsic context," in *CVPR*, 2011, pp. 417–424.
[57] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *CVPR*, 2012, pp. 454–461.
[58] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *BMVC*, 2013.
[59] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light fields," in *CVPR*, 2014.
[60] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," in *CVPR*, 2014.
[61] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform," in *CVPR*, 2014.
[62] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *CVPR*, 2014.
[63] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *CVPR*, 2010, pp. 3241–3248.
[64] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li, "Salient object detection for searched web images via global saliency," in *CVPR*, 2012, pp. 3194–3201.
[65] F. Moosmann, D. Larlus, and F. Jurie, "Learning saliency maps for object categorization," in *EECVW*, 2006.
[66] T. Judd, K. A. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *ICCV*, 2009, pp. 2106–2113.
[67] Y. Lu, W. Zhang, C. Jin, and X. Xue, "Learning attention map from images," in *CVPR*, 2012, pp. 1067–1074.
[68] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: a flexible architecture for multi-scale derivative computation," in *ICIP (3)*, 1995, pp. 444–447.
[69] B. Fernando, É. Fromont, D. Muselet, and M. Sebban, "Discriminative feature fusion for image classification," in *CVPR*, 2012, pp. 3434–3441.
[70] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.
[71] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.
[72] T. K. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *IJCV*, vol. 43, no. 1, pp. 29–44, 2001.
[73] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *ICCV*, 2005, pp. 654–661.
[74] H. Jiang, Y. Wu, and Z. Yuan, "Probabilistic salient object contour detection based on superpixels," in *ICIP*, 2013, pp. 3069–3072.
[75] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "iCoseg: Interactive co-segmentation with intelligent scribble guidance." in *IEEE CVPR*, 2010, pp. 3169–3176.
[76] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *CVPR*, 2007.
[77] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*. CVPR, 2013, pp. 1155–1162.
[78] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *CVPR*, 2012.