Salient Object Detection for Searched Web Images via Global Saliency

Peng Wang[1] Jingdong Wang[2] Gang Zeng[1] Jie Feng[1] Hongbin Zha[1] Shipeng Li[2]
[1]Key Laboratory on Machine Perception, Peking University [2]Microsoft Research Asia

Abstract

In this paper, we deal with the problem of detecting the existence and the location of salient objects for thumbnail images on which most search engines usually perform visual analysis in order to handle web-scale images. Different from previous techniques, such as sliding window-based or segmentation-based schemes for detecting salient objects, we propose to use a learning approach, random forest in our solution. Our algorithm exploits global features from multiple saliency information to directly predict the existence and the position of the salient object. To validate our algorithm, we constructed a large image database collected from Bing image search, that contains hundreds of thousands of manually labeled web images. The experimental results using this new database and the resized MSRA database [16] demonstrate that our algorithm outperforms previous state-of-the-art methods.

1. Introduction

Saliency detection on images has been studied for a long time. In recent years, many saliency detection methods [7, 11, 12, 15, 16, 18] have been designed because of its broad applications [6, 25, 30]. However, localizing salient objects is still a very challenging problem. In this paper, we address the problem of judging the existence and predicting the location of the salient object on thumbnail images. As discussed in [16], this problem is clearly different from predicting the regions where humans look, such as [14, 33].

Salient object detection has many practical applications, such as image cropping [25, 27], adaptive image display on mobile devices [6], extracting dominant colors on the object of interest for web image filter [32], removing the images that do not contain an object of interest in image search, and so on.

There are several challenges in detecting salient objects. On the one hand, objects have various visual characteristics, which makes it hard to differentiate salient objects from the background according to appearance only. On the other hand, thumbnail images have a low resolution (e.g., 130×130), which is enough for a human to recognize the salient object but makes it difficult to get a reliable segmentation that some previous salient object detection methods rely on.

1.1. Related work

Sliding window-based method. Sliding windows detect salient objects by combining local cues. Given a window on the image, the system evaluates the probability of the window containing an object. Heuristic methods that evaluate windows on a single saliency map are efficient [17]. The detection accuracy, however, is not guaranteed.

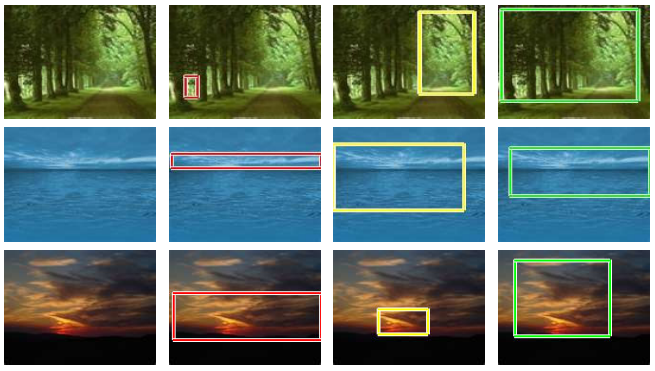
Alexe et al. [2] propose an "Objectness" [2] measure to localize objects in an image. They combine various "Objectness" cues, such as multi-scale saliency, edge density, color contrast and superpixel straddle, into a Bayesian framework. One later work [23] instead proposes a limited number of object bounding box candidates. Compared to "Objectness", this approach adopts more robust visual cues and uses Structured SVM [28] for ranking the candidates. A feature cascade scheme is then used for acceleration. Although the cues provided by the previous methods are effective, for the local characteristics around a single window, the produced bounding box might not be globally the best.

Feng et al. [9] compute the window saliency based on superpixels. They use all the superpixels outside the window to compose the the inside ones, thus the global image context is combined. Although higher precision is achieved compared with "Objectness" [2], the mono scale superpixel segmentation they use sometimes performs poorly on thumbnail images, which may make the composition fail.

Segmentation-based method. Alternative approaches generate a salient object bounding box through segmenting the salient object based on the saliency maps.

Marchesotti et al. [19] proposed to retrieve similar images, and they separately model the object and background based on those retrieved images. The final saliency regions are segmented via graph cut optimization. However, appearance-based retrieval depends highly on the database, which limits the generalization of the system.

The algorithm of Liu et al. [16] learns to optimally find weights by incorporating various saliency cues from the image. A binary segmentation step is then applied to find the salient object, but the procedure may suffer from noisy re-



(a) (b) (c) (d)

Figure 1. (a) Background images with needless salient object bounding boxes from (b) Alexe et al. [2] (c) Feng et al. [9] and (d) Liu et al. [16].

gions yielded from bounding box based training images. To avoid this defect and achieve global optimization, Chen et al. [7] apply Grab-cut [24] to iteratively refine the segmentation based on their proposed saliency maps. Wang et al. [31] integrate Auto-context [29] into the saliency cut for combining context information. Nevertheless, they train the classifier on the pixel level within each iteration, which slows down the progress.

Under our scenario, we focus on proposing an algorithm that efficiently generates one global optimal bounding box for object localization. This is under the consideration that most thumbnail images on the web contain a single salient object. Also, a single box is good to use for web image applications, such as thumbnail cropping [25, 27]. Nevertheless, sliding window-based schemes always propose too many candidate bounding boxes, thus it is perplexing to choose the best one for mentioned applications. Segmentation-based methods generally propose one global salient object region, but iterative approaches like [7, 31] make the algorithms inefficient in practical usage.

More importantly, detecting the existence of salient objects has not been concerned before localization by previous arts. This may lead to unexpected results for background images with repeating pattern, as illustrated in Fig. 1.

1.2. The framework

To deal with the issues mentioned in Sec. 1.1, we developed a salient object detection system with the framework shown in Fig. 2.

Firstly, targeting at detection on web images, a web image database is constructed with each image manually labeled as a background image or an object image with a bounding box enclosing the salient object region. Then, the features capturing the object's salient information from multiple channels are extracted. After that, two phases followed: object existence verification and localization. For detecting the object existence, we apply a binary classification approach. For localization, rather than through time

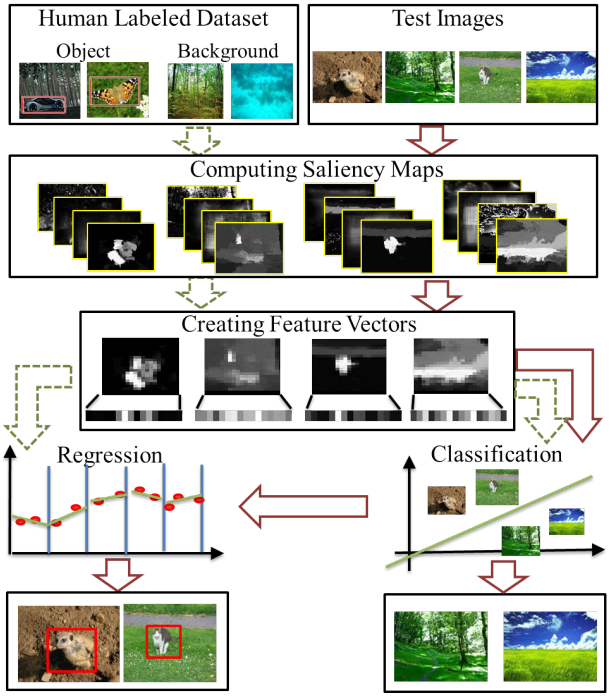


Figure 2. The framework of our salient object detection system. The pipeline of dotted arrows shows the training procedure and the pipeline of solid arrows shows the testing procedure.

costly segmentation or local sliding window, we learn a regression function through random forest [4] from the labeled database to directly predict the position of the salient object. Finally, given a test image, the extracted features could be fed into the trained classifier and regressor to get the final results.

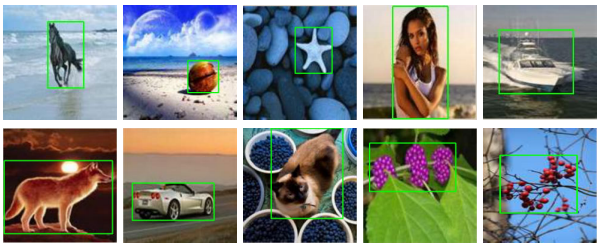
The remainder of the paper is organized as follows: Section 2 introduces the web image database we constructed. Section 3 presents the detail of the algorithm. In Section 4, we describe the evaluation experiments. Section 5 gives a conclusion of the work.

2. Web Image Database

We created a web image database by asking users to draw a rectangle to specify the location of a salient object region within an image. Unlike the MSRA object saliency image set [16], our database has several different aspects:

Image sources. Our web images are from real searching queries. We searched 1100 queries and downloaded 400 thumbnail images for each query. In total, more than 400k images are collected. In the selection process, for object images, we pruned the image without a single salient object region such as the image composed by many sub-images or including several objects in a cluttered layout, leaving approximate 300k images. All the images are around 130 $\times$

(a) Background Images



(b) Labeled Object Images

Figure 3. Example images in web image database. The background images (a) and the object images (b). The green box in each image in (b) is the manually labeled ground truth.

130. Furthermore, we consider not only the object images containing a salient object region, but also the background images containing no object from queries, such as "desert", "ocean", "forest" and so on.

Labeling scheme. Formally, each image $I(\mathbf{x})$ is assigned a corresponding label vector $\mathbf{y} = (o, t, l, b, r)$, where $\mathbf{x}$ represents a pixel, o indicates wether a salient object region is presented in the image or not, and (t, l, b, r) represents the top, left, bottom and right positions of the bounding box within the image.

Due to the large size of the database, all images are separated into disjoint parts and labeled by different users. Each image is labeled once. In terms of our observation, the labeling distinctions between different users are subtle, especially when labeling on thumbnail images.

Moreover, to further ensure the labeled bounding boxes are consistent between different users, we set two rules for users to follow. First, the labeled rectangle should enclose the entire object respecting the object boundaries. Second, when multiple objects exist, the rectangle should cover the salient objects which are overlapped or very close to each other. The labeled examples are shown in Fig. 3.

Bounding box distribution. Inspired by [23], we also computed the distributions of the labeled bounding boxes on our web image database. From Fig. 4, a bias of the labeled boxes is revealed indicating the bounding boxes' size is relatively large and the centroid position is generally not far from the image center.

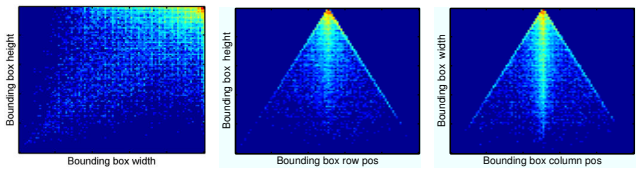


Figure 4. The learned distributions (100×100) of labeled object bounding boxes (redder is higher): height versus width, height versus row location, and width versus column location.

3. Proposed Algorithm

Input & output space. Given a set of training features $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\} \subset \mathcal{X}$ and their associated output labels $\{\mathbf{y}_1, ..., \mathbf{y}_n\} \subset \mathcal{Y}$, where n is the number of images and $\mathbf{f}_i$ is the global feature of image $I_i(\mathbf{x})$, we wish to learn a mapping $\mathbf{g}$: $\mathcal{X} \rightarrow \mathcal{Y}$. Here the output space $\mathcal{Y} \triangleq \{(o, t, l, b, r) | o \in \{+1, -1\}, (t, l, b, r) \in \mathbb{R}^4 \; s.t. \; t < b, \; l < r\}$, in which the output vector (o, t, l, b, r) is defined in Sec. 2.

In our scenario, as illustrated in Fig. 2, we separate such a problem into a classification problem and a localization problem because under the large amount of data, handling both problems together leads to high computational cost in training, such as [3] utilizing Structured SVM. So the whole output space $\mathcal{Y}$ is split into a binary classification space defined as $\mathcal{O} \triangleq \{+1, -1\}$, and a localization space defined as $\mathcal{W} \triangleq \{(t, l, b, r)\}$. Thus, the mapping function is now $\mathbf{g} = (g_c, \mathbf{g}_l)$, where g_c represents the mapping: $\mathcal{X} \rightarrow \mathcal{O}$ and $\mathbf{g}_l$ represents the mapping: $\mathcal{X} \rightarrow \mathcal{W}$.

3.1. Creating features

Fig.5 shows the computed object saliency maps from several image examples. Clearly, for background images, the salient contents in images are always scattered, or there's no obvious salient regions. While images containing an object generally produce a saliency map with a compact and closed salient region delivering the object boundaries. This attributes to the dissimilar appearance between the objects and the surrounding background. According to the observation, our feature vector $\mathbf{f}$ is constructed based on saliency information.

As Alexe et al. [2] indicated, the saliency of an object could be represented by multiple cues including silhouette, appearance contrast etc. Similarly, for our model, different object saliency cues as showed in Fig. 6, are taken into account for the final feature vector. Here we applied pixel, regional and global level object cues: the pixel level information is from Multi-scale Contrast (MC). The regional level information is from Center-Surrounding Histogram (CSH) and spatial weighted Region-based Contrast (RC). The global level information is from Color Spatial-Distribution (CSD). RC is proposed by [13] and MC, CSH, CSD are proposed by [16]. For the space limitation, we refer readers to the original papers for details.

With a number of saliency maps $S_k(\mathbf{x})$ in which $k = 1, \ldots, K$ and $K = 4$, we normalized all the saliency maps into $[0, 1]$ and fuse them into a single feature vector via two strategies:

1) Stack:

Partition each $S_k(\mathbf{x})$ into $N = p \times p$ blocks in a grid layout, and the average value in each block is extracted, which results in a feature vector $\mathbf{f}_k$. Then stack all $\mathbf{f}_k$ into a final feature vector of length $K \times p \times p$. We set $p = 30$ in our experiments. The feature vector of in our case is written as:

$$\mathbf{f}_{all} = [\, \mathbf{f}_{RC}^T, \mathbf{f}_{MC}^T, \, \mathbf{f}_{CSH}^T, \, \mathbf{f}_{CSD}^T]^T. \tag{1}$$

2) SumUp:

Combine all the saliency maps into one single map $S_{all}(\mathbf{x})$ similar with [16], then partition $S_{all}(\mathbf{x})$ to extract $\mathbf{f}_{all}$. In this work, we apply a non-linear combination of multiple saliency values as:

$$S_{all}(\mathbf{x}) = (\sum_{k=1}^{K} \lambda_k S_k(\mathbf{x}))^2, \tag{2}$$

where λ_k is the weight of the kth saliency map. Respecting the object boundaries, we learn the combination coefficients λ_k through the Conditional Random Field (CRF) scheme [16] from the salient object image set with accurate object-contour ground truth provided by [1]. This is because the database is similar to our web image database with images containing a single salient object. We found that the corresponding weights for RC, MC, CSH and CSD are $0.44, 0.17, 0.18$ and 0.21.

The square in Eqn.(2) aims to relatively enhance the salient region and suppress the weak salient part for regression. In our experiments, we separately tested the "Stack" and "SumUp" combination strategy and found that the "Stack" is good at classification while "SumUp" achieves higher localization scores (See Sec.4 for details).

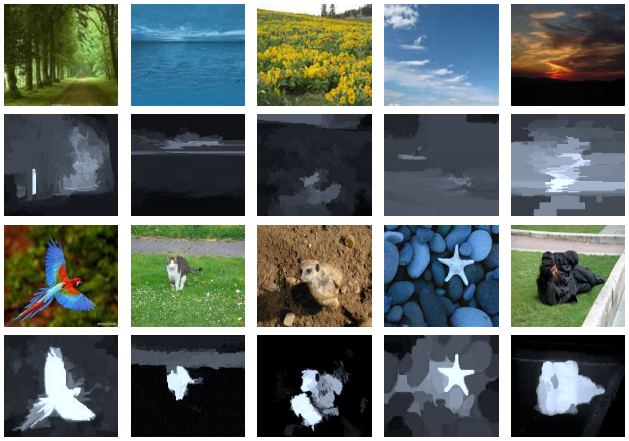


Figure 5. Background images (the 1_{st} row), object images (the 3_{rd} row) and their corresponding RC [13] saliency maps (the 2_{nd} row) & (the 4_{th} row). Obvious distinctions exist between two types of images.

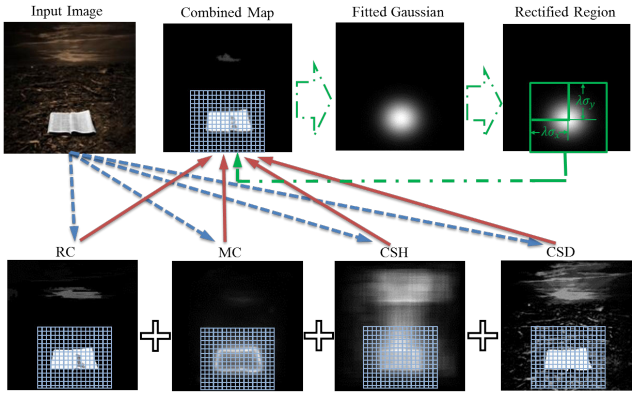


Figure 6. Creating features for regression. The dash arrows indicate the map computation. The solid arrows represent the map combination. The dash dot arrows tell the rectification and the grids mean the feature extraction

3.2. Detecting object existence via classification

Given the training feature, we feed the features into the simple and powerful random forest classifier [4] to learn the mapping g_c, as the highly variational and non-linear separable property of our feature points.

In the web image database, the ratio between object images and background images is nearly $10 : 1$. The highly imbalanced training data negatively affects the accuracy of classifier, since random forest tends to be biased towards the majority class. Normally this problem has two solutions: one is to adjust the weights between different classes through cost sensitive learning and the other one is down-sampling the majority class or up-sampling the minority class [5]. In our problem, many web images have an object covered the center part with similar shapes and sizes, which means that many redundant feature points exist for training the classifier. Thus we apply the down-sampling scheme to train a balanced random forest classifier. We give the details on how we do this in Sec.4.1.

3.3. Translation and scale invariance feature

According to the obvious bias of the bounding boxes' position and size reported in Sec. 2, the random forest regression prefers to generate a relatively large rectangle around the image center area. Thus, for images with a small object or with an object shifting away from the image center, sometimes the regressor lacks supporting data for prediction. To deal with this translation and scale variation problems, we perform a rectification on the saliency map $S_{all}(\mathbf{x})$ combined through Eqn.(2).

Fig. 6 shows our procedure of the rectification for better regression features. In the first step, we fit a single two dimensional un-normalized Gaussian model to the combined salient map. The Gaussian function takes the form:

$G(\mathbf{x}) = Ae^{-(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$ where the $\Sigma = \begin{pmatrix}\sigma_x^2 & 0\\ 0 & \sigma_y^2\end{pmatrix}$. For $G(\mathbf{x})$, we regard the 2D position of the pixel $\mathbf{x}$ as its input and the saliency value $S_{all}(\mathbf{x})$ as the output. Through least square estimation, we find A, μ, Σ by minimizing the objective $\sum_{\mathbf{x}}(G(\mathbf{x}) - S_{all}(\mathbf{x}))^2$.

After the estimation, we translate the image center to the position $\mu = (\mu_x, \mu_y)^T$ and crop the image on coordinate x based on the estimated σ_x. Particularly, we define the range of coordinate x on the image to be $[\mu_x - \lambda\sigma_x, \mu_x + \lambda\sigma_x]$. A similar operation is conducted on coordinate y. Because that some noisy regions affect the fitting, the procedure is repeated within the rectangle area 2∼3 times, resulting in a stable rectified region. In our experiments, we set $\lambda = 3$ by validating the regression results on a small constructed training set containing 500 web images. Through the rectification, the feature vector is then extracted by "Stack" or "SumUp" as in Sec. 3.1.

3.4. Localizing object via regression

Here we model the mapping $\mathbf{g}_l$ by learning the posterior distribution $p(\mathbf{w}|\mathbf{f})$ through regression, with our training set $\{\mathbf{f}^{(n)}, \mathbf{w}^{(n)}\}_{n=1}^N$ in which $\mathbf{f} \in \mathcal{X}$ and $\mathbf{w} \in \mathcal{W}$.

As indicated by Pauly et al. [22], for the high-dimensional feature space, which is our case, partitioning the input space into an ensemble of cells can reduce the complexity, and modeling within each cell could be simple. Formally, the partition is defined as: $\mathcal{P} = \{\mathcal{C}_t\}_{t=1}^T$. Based on the training data in each cell, the posterior of variable $\mathbf{w}$ could be modeled as a multivariate Gaussian distribution, i.e. $p(\mathbf{w}|\mathcal{C}_t, \mathcal{P}) = \mathcal{N}_t(\mu_t, \Sigma_t)$. In [4], Breiman demonstrates that replacing a single partition $\mathcal{P}$ with an ensemble of independent random partitions $\{\mathcal{P}_z\}_{z=1}^Z$ leads to an ensemble regressor achieving better performance.

In this paper, we apply random forest [4] to construct the multiple partition $\{\mathcal{P}_z\}_{z=1}^Z$ and train the Gaussian distribution in each cell. This technique has been formerly used to localize organs in medical images [22] and to estimate the poses in depth images [10] for its efficiency. Then given a feature point $\mathbf{f}$, $\mathbf{f}$ would fall into a specific cell $\mathcal{C}_z$ in each partition $\mathcal{P}_z$. The posterior probability estimated from different partitions in Random Forests are then combined through averaging, i.e. $p(\mathbf{w}|\mathbf{f}) = \frac{1}{Z}\sum_z P(\mathbf{w}|\mathcal{C}_z, \mathcal{P}_z)$. Finally, we can estimate in one shot the position of the object of interest contained in the bounding box $\hat{\mathbf{w}}$ using the mathematical expectation: $\hat{\mathbf{w}} = \int_{\mathbf{w}} \mathbf{w}p(\mathbf{w}|\mathbf{f})d\mathbf{w}$.

4. Experiments

We conducted our evaluation based on two databases. The first one is the MSRA image set $\mathcal{B}$ with images resized into 130×130 by the bi-cubic method, thus to simulate the thumbnail images. The MSRA $\mathcal{B}$ database contains 10 folders and each folder includes 500 images. All images contain one single salient object. The characteristic of this database is that all the images selected have high human label consistency, which is suitable under our scenario. The second one is the web image database which is described in Sec. 2.

Technique	RC	MC	CSH	CSD	SumUp	Stack
SVM	74.9 ±7.4	74.2 ±5.8	76.9 ±3.4	66.9 ±1.9	67.9 ±2.9	81.4 ±6.3
RF	75.8 ±5.7	76.4 ±1.4	77.6 ±2.3	67.5 ±1	69.1 ±1.6	**82.8** ±**3.5**

Table 1. Classification accuracy of various features and techniques in classifying object images and background images.

Our experiment includes both evaluations of classification and localization. Unless otherwise specified, our parameters for random forest were set as: 200 trees, the minimum node size is set to 15. All experiments run on a quad-core 3.2GHz computer.

4.1. Classification evaluation

In the MSRA $\mathcal{B}$ image set, there is no background image. For evaluating the classification performance, we added background image samples from the web image database. In detail, as stated in Sec. 3.2, we down-sampled the object images of the web image database randomly, and 3k images in the MSRA $\mathcal{B}$ database and 5k in web image set are proposed. The background images include randomly selected 5k images.

By applying the features as stated in Sec. 3.1, we got an acceptable accuracy in Table. 1. We also show the average accuracy of random forest (RF) classifier is superior, compared with SVM with RBF-kernel by 5-cross validation. The failure examples, as showed in Fig. 7, are often the cases that the background image has high contrast in the center area or the object's appearance inside the image is similar to the background.

4.2. Regression evaluation

To test the regressor's ability, the localization evaluation is conducted under the object images. For the MSRA $\mathcal{B}$ database, we randomly selected 9 folders (4500 images) for training and use the rest one (500 images) for testing. For the web image database, we took out 20k images for training and 5k images for testing.

(a) (b) (c) (d)

Figure 7. Wrongly classified background image (a) and object image (c) with their corresponding RC saliency maps (b)&(d).

4.2.1 Training parameters

As stated by previous methods [10, 26], random forest is sensitive to it's parameters. Here we investigate two most influential parameters affecting the regression accuracy through testing over the images from the web image database.

Number of training image. In Fig. 8(a), we show how boundary displacement error (BDE) which is presented by Eqn.(5) decreases with the increasing of the number of randomly generated training images. The graph illustrates that the error seems to tail off at 10k images. This situation is mostly due to the limited correlation between bounding box in the database and the saliency maps.

Minimum size of each node. Minimum node-size controls the size of each tree. Smaller node-size would lead to deeper trees, but cost more time to generate and vice versa. We also tested the effect of this parameter in Fig. 8(b). In our experiment, we showed that the random forest regressor over-fits the data when the node-size is smaller than 15.

4.2.2 Effectiveness of features

To test the effectiveness of each salient feature, we incrementally add different saliency maps into the feature vector. Fig. 8(c) shows that the BDE error decreases when we add different features sequentially. This implies our saliency features are complementary. Note that our method is not limited to just these types of saliency maps, if we relax the consideration of time cost, better results can be achieved by

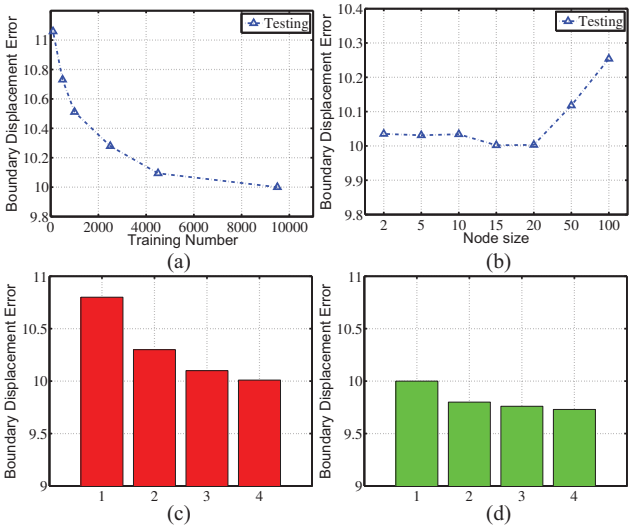
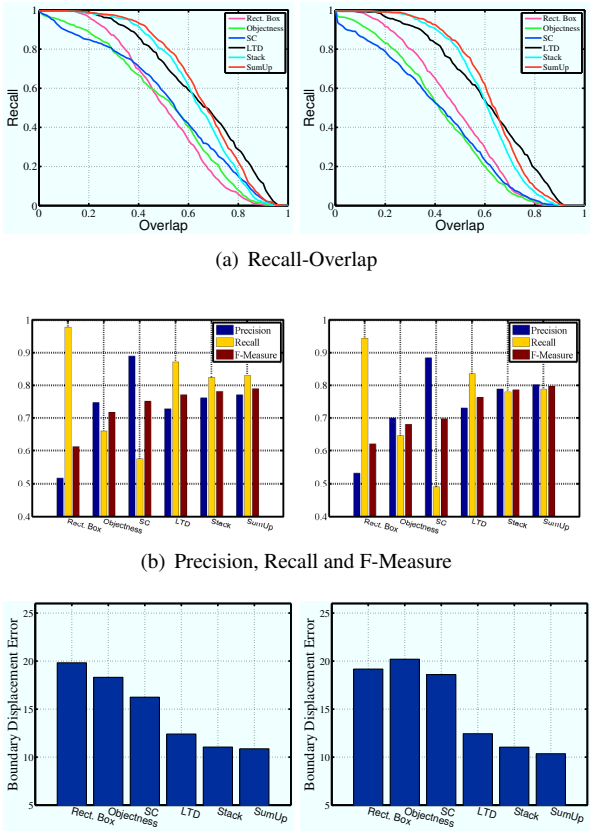


Figure 8. The training conditions vs. Regression error (a) Number of training images. (b) Minimum data number of leaf node. (c) Number of Saliency map (RC,CSD,MC,CSH are added sequentially). (d) Additionally add other global features (Original features, GIST of the saliency map, HOG of the image, GIST of the image are added sequentially).



(a) Recall-Overlap

(b) Precision, Recall and F-Measure

(c) Boundary Displacement Error

Figure 9. Quantitative comparisons on MSRA $\mathcal{B}$ image set (Left column) and the web image database (Right column).

including other features. To demonstrate this, we integrate the GIST [21] and HOG [8] image descriptor to see whether the error can be further reduced. Fig. 8(d) shows the result, where we show that by adding GIST descriptor computed from combined saliency map, the BDE drops further, however, adding the GIST and HOG descriptor computed from the image shows little influence. This is perhaps because the GIST descriptor on the saliency map also captures the shape of salient region.

4.2.3 Comparison with other approaches

We compared our localization method with three leading salient object localization approaches. The first is the retrained "Objectness" [2], and the second is the superpixel composition [9] (SC). Because these methods propose multiple ranked detections, we use the returned bounding box with the highest score in our setting to make the comparison valid. The third method is segmentation based method [16] (LTD). We further add a baseline of our rectifying bounding box (Rect. BBox). All the comparison are conducted on the two datasets.

Quantitative comparison. With the labeled ground truth bounding box $\mathbf{B}_{gt}$ and the detected bounding box $\mathbf{B}_d$ in

an image, we use the region-based and edge-based measurements similar with the one proposed by "LTD". We further induce the Recall-Overlap measurement from [23]. The region-based measurement includes Precision, Recall, F-measure, and Recall-Overlap.

Precision and Recall are mathematically defined as:

$$\begin{aligned} Precision &= area(\mathbf{B}_{gt} \cap \mathbf{B}_d)/area(\mathbf{B}_d), \\ Recall &= area(\mathbf{B}_{gt} \cap \mathbf{B}_d)/area(\mathbf{B}_{gt}). \end{aligned} \quad (3)$$

The F-measure is the weighted harmonic of Precision and Recall with the parameter α:

$$F_\alpha = \frac{(1+\alpha) \times Precision \times Recall}{\alpha \times Precision + Recall}. \quad (4)$$

We set $\alpha = 0.5$, following the previous work [20].

The Overlap-Recall curve is the recall rate of ground truth bounding boxes from the database by changing the threshold of the overlap score. The overlap score for experiments is based on the PASCAL VOC criterion defined as: $\frac{area(\mathbf{B}_{gt} \cap \mathbf{B}_d)}{area(\mathbf{B}_{gt} \cup \mathbf{B}_d)}$.

For edge-based measurement, the Bounding box Boundary Displacement Error (BDE) in the pixel level described by [13] is tested, which is defined with the L_1 norm as:

$$BDE = \frac{\|\mathbf{B}_{gt} - \mathbf{B}_d\|_1}{4}. \quad (5)$$

We measured these the criteria by averaging over all test images. Fig. 9 shows the results on the MSRA $\mathcal{B}$ image set and the results on our web image set.

Under the web applications mentioned in Sec. 1, the expected result of one image prefers a bounding box close to the human labeled result. This means that the high recall rate is preferred with the overlap upper 0.5 as presented by [23]. As can be seen in Fig. 9(a), we achieve the highest in this scenario. Moreover, as showed in Fig. 9(b) and Fig. 9(c), our method also outperforms others.

Qualitative results. To better understand the quantitative results, Fig. 10 gives several examples with ground truth to visually compare the results from our method and the others. The bounding boxes produced by Objectness [2] tend to enclose a local salient region, which performs relatively poor on both precision and recall. SC [9] combines global context, but the generated bounding boxes often just cover a part of the salient object, which achieves very high precision but very low recall. The segmentation results given by LTD [16] tends to segment out the connected homogeneous region which can be easily affected by noisy saliency. Our algorithm produces more precise bounding boxes than previous methods, albeit the edges of the bounding boxes are not perfectly aligned with the object boundaries. Our results' superiority is mostly due to the reason that the random forest automatically exploit the information from the global image saliency among the whole images set. More results are shown in Fig. 11.

(a) (b) (c) (d) (e)

Figure 10. Qualitative comparisons between (a) the Objectness [2] (b) SC [9] (c) LTD [16] (d) Our approach and (e) Ground Truth.

Time cost. At last, Table 2 compares the average computing time of all the methods. We use the author's implementation for Objectness, SC and LTD. Comparing ours and LTD, the time cost on computing saliency maps (Sal. Map) is close , but the Localization (Loc.) time of ours significantly outperforms the LTD's CRF inference and sufficiently efficient for real-time applications.

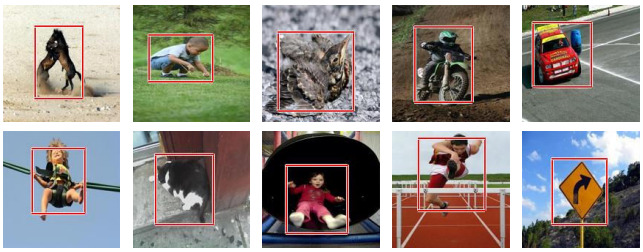
5. Conclusion and Discussion

We presented a large labeled web image database and a supervised scheme that judges the existence of and predicts the location of the salient object in thumbnail images. Our algorithm exploits random forest and global saliency with features created from the saliency maps combining information of multiple channels. We test the system and show promising results compared to several state-of-the-art algorithms.

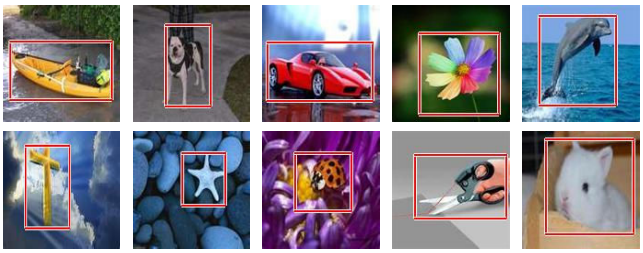
Fig. 12 also shows some failure cases. By looking at the saliency map, we can still see that the bounding box contains the salient information well. Unfortunately, the saliency map provides poor guidance for the regression function. These cases also indicate that people detect the salient object along with their cognitive process, which

Method	Objectness [2]	SC [9]	LTD [16]	Ours
Time(s)	3.4	0.3	Sal. Map: 3.7 Loc.: 11	Sal. Map: 4.4 Loc.: 0.02
Code	Matlab+C	C++	Matlab+C	Matlab+C

Table 2. Average time comparison to localize a web image of size (130× 130).

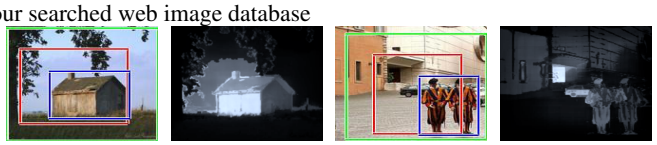


(a)



(b)

Figure 11. Detection results on (a) the MSRA $\mathcal{B}$ database and (b) our searched web image database



(a) (b) (c) (d)

Figure 12. Failure cases. Object image (a)&(c) and the corresponding saliency map (b)&(d). Blue, Green, Red box is the Ground Truth, rectified region and our result respectively.

leads to another challenging issue of detecting saliency by combining semantics.

Acknowledgements

The research work of Peng Wang, Gang Zeng and Hongbin Zha is supported by National Nature Science Foundation of China (NSFC Grant) 61005037, National Basic Research Program of China (973 Program) 2011CB302202, and Beijing Natural Science Foundation (BJNSF Grant) 4113071.

References

[1] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009. 4

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010. 1, 2, 3, 6, 7

[3] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In D. A. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *ECCV (1)*, volume 5302 of *Lecture Notes in Computer Science*, pages 2–15. Springer, 2008. 3

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 2, 4, 5

[5] C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. *Discovery*, (1999):1–12, 2004. 4

[6] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H. Zhang, and H.-Q. Zhou. A visual attention model for adapting images on small displays. *Multimedia Syst*, 9(4):353–364, 2003. 1

[7] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416. IEEE, 2011. 1, 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893. IEEE Computer Society, 2005. 6

[9] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *ICCV*, pages 1028–1035, 2011. 1, 2, 6, 7

[10] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *ICCV*, 2011. 5, 6

[11] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010. 1

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998. 1

[13] 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. 3, 4, 7

[14] T. Judd, K. A. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2009. 1

[15] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001. 1

[16] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell*, 33(2):353–367, 2011. 1, 2, 3, 4, 6, 7

[17] Y. Luo, J. Yuan, P. Xue, and Q. Tian. Saliency density maximization for object detection and localization. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *ACCV (3)*, volume 6494 of *Lecture Notes in Computer Science*, pages 396–408. Springer, 2010. 1

[18] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the 11th ACM International Conference on Multimedia (MM-03)*, pages 374–381. 1

[19] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *ICCV*, pages 2232–2239. IEEE, 2009. 1

[20] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004. 7

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 6

[22] O. Pauly, B. Glocker, A. Criminisi, D. Mateus, A. Martinez-Möller, S. G. Nekolla, and N. Navab. Fast multiple organ detection and localization in whole-body MR dixon sequences. In G. Fichtinger, A. L. Martel, and T. M. Peters, editors, *MICCAI (3)*, volume 6893 of *Lecture Notes in Computer Science*, pages 239–247. Springer, 2011. 5

[23] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, pages 1052–1059, 2011. 1, 3, 7

[24] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph*, 23(3):309–314, 2004. 2

[25] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems*, volume 1 of *Collecting and editing photos*, pages 771–780, 2006. 1, 2

[26] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304. IEEE, 2011. 6

[27] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 95–104, New York, NY, USA, 2003. ACM. 1, 2

[28] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 1

[29] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(10):1744–1757, 2010. 2

[30] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *CVPR (1)*, pages 347–354. IEEE Computer Society, 2006. 1

[31] L. Wang, J. Xue, N. Zheng, and G. Hua. Automatic salient object extraction with contextual cue. In *ICCV*, pages 105–112, 2011. 2

[32] P. Wang, D. Zhang, G. Zeng, and J. Wang. Contextual dominant color name extraction for web image search. In *ICME Workshop on Social Multimedia Computing*, 2012. 1

[33] W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao. Simulating human saccadic scanpaths on natural images. In *CVPR*, pages 441–448. IEEE, 2011. 1