

Learning RGB-D Salient Object Detection using background enclosure, depth contrast, and top-down features

Riku Shigematsu David Feng
 Australian National University
 riku.research@gmail.com
 david.feng@data61.csiro.au

Shaodi You Nick Barnes
 Australian National University
 Data61-CSIRO
 shaodi.you@data61.csiro.au
 nick.barnes@data61.csiro.au

Abstract

Recently, deep Convolutional Neural Networks (CNN) have demonstrated strong performance on RGB salient object detection. Although, depth information can help improve detection results, the exploration of CNNs for RGB-D salient object detection remains limited. Here we propose a novel deep CNN architecture for RGB-D salient object detection that exploits high-level, mid-level, and low level features. Further, we present novel depth features that capture the ideas of background enclosure and depth contrast that are suitable for a learned approach. We show improved results compared to state-of-the-art RGB-D salient object detection methods. We also show that the low-level and mid-level depth features both contribute to improvements in the results. Especially, F-Score of our method is 0.848 on RGBD1000 dataset, which is 10.7% better than the second place.

1. Introduction

In computer vision, visual saliency attempts to predict which parts of an image attract human attention. Saliency can be used in the context of many computer vision problems such as compression [7], object detection [19], visual tracking [20], and retargeting images and videos [25]. Early work attempted to predict human gaze direction on images [10]. However, in recent years, the field has focused on salient object detection, finding salient objects or regions in an image (e.g., [1, 3]). Rather than attention, the main task for salient object detection is to produce a fine grained binary mask for salient objects, based on human annotation.

Most salient object detection methods are based on RGB images. However, depth plays a strong role in human perception, and it has been shown that human perception of salient objects is also influenced by depth [14]. Thus, RGB-D salient object detection methods have been pro-

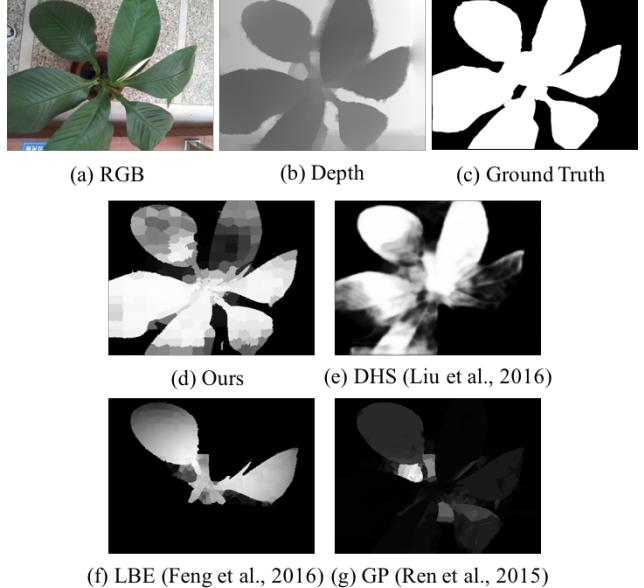


Figure 1. Comparing our RGB-D salient object detector output with other salient object detection methods.

posed [6, 13, 21, 22, 24] and have demonstrated superior performance in comparison to RGB-only methods.

While many salient object detection methods adopt a bottom-up strategy [6, 8, 13, 21, 24], recently, top-down methods through machine learning have demonstrated superior performance [15, 18, 22, 30]. Recent papers have tackled top-down learning for RGB salient object detection using deep CNN for their methods [15, 18, 30]. However, it is not yet clear whether deep CNNs are effective for RGB-D saliency detection.

The approach of this paper is premised on observations of the performance of state-of-the-art approaches in salient object detection. Top-down information has been shown to be effective in RGB salient object detection (top-down in-

formation also plays a role in human visual focus of attention [10]). Further, in RGB-D salient object detection, the effectiveness of background enclosure and of depth contrast have been demonstrated. Finally, deep CNNs have been shown to be effective for RGB salient object detection.

This paper makes three major contributions. (1) We propose a novel learning architecture that provides the first complete RGB-D salient object detection systems using a deep CNN, incorporating high level features, depth contrast and low-level features, and a novel mid-level feature. (2) We introduce the background enclosure distribution, BED, a novel mid-level depth feature that is suitable for learning based on the idea of background enclosure. (3) We introduce a set of low level features that are suitable for learning that incorporate the idea of depth contrast.

We show that our new approach produces state-of-the-art results for RGB-D salient object detection. Further, we evaluate the effectiveness of adding depth features, and of adding the mid-level feature in particular. In ablation studies, we show that incorporating low-level features that incorporate depth contrast performs better than RGB saliency alone, and that adding our new mid-level feature, BED, improves results further.

The rest of the paper is organized as follows: Section 2 presents the related work on salient object detection and deep CNN methods for the other computer vision task. In Section 3, we introduce the idea on RGB-D salient object detection with a deep CNN architecture and our algorithms for salient object detection. This is followed by the detailed neural network structure, technical details and the training approach in Section 4. We introduce systematic and extensive experimental results in Section 5, and conclude in Section 6.

2. Related Work

Saliency detection to model eye movements began with low-level hand-crafted features, with classic work by Itti *et al.* [10] being influential. A variety of salient object detection methods have been proposed in recent years, we focus on these as more relevant to our work.

RGB Salient object detection In RGB salient object detection, methods often measure contrast between features of a region versus its surrounds, either locally and/or globally [5, 10]. Contrast is mostly computed with respect to appearance-based features such as colour, texture, and intensity edges [4, 12].

RGB salient object detection using deep CNNs Recently, methods using deep CNNs have obtained strong results for RGB salient object detection. Wang *et al.* [28] combine local information and a global search. Often

the networks make use of deep CNN networks for object classification for a large number of classes, specifically VGG16 [26] or GoogleNet [27]. Some utilize these networks for extracting the low features [15, 16, 18]. Lee *et al.* incorporate high-level features based on these networks, along with low level features [15]. This approach to incorporating top-down semantic information about objects into salient object detection has been effective.

RGB-D Salient Object Detection Compared to RGB salient object detection, fewer methods use RGB-D values for computing saliency. Peng *et al.* calculate a saliency map by combining low, middle, and high level saliency information [21]. Ren *et al.* calculate region contrast and use background, depth, and orientation priors. They then produce a saliency map by applying PageRank and a MRF to the outputs [24]. Ju *et al.* calculate the saliency score using anisotropic center-surround difference and produce a saliency map by refining the score applying Grabcut segmentation and a 2D Gaussian filter [13]. Feng *et al.* improve RGB-D salient object detection results based on the idea that salient objects are more likely to be in front of their surroundings for a large number of directions [6]. All the existing RGB-D methods use hand-crafted parameters, such as for scale and weights between metrics. However, real world scenes contain unpredictable object arrangements for which fixed hand coded parameters may limit generalization. No published papers have yet presented a CNN architecture for RGB-D salient object detection. A preliminary paper (Arxiv only) uses only low-level color and depth features [22].

Datasets Two datasets are widely used for RGB-D salient object detection, RGBD1000 [21] and NJUDS2000 [13]. The RGBD1000 datasets contain 1000 RGB-D images captured by a standard Microsoft Kinect. The NJUDS2000 datasets contain around 2000 RGB-D images captured by a Fuji W3 stereo camera.

3. A novel deep CNN architecture for detecting salient objects in RGB-D images

In this section, we introduce our approach to RGB-D salient object detection. Our novel deep CNN learning architecture is depicted in Figure 2. We combine the strengths of previous approaches to high-level and low-level feature-based deep CNN RGB salient object detection [15], with a depth channel, incorporating raw depth, low level cues to capture depth contrast, and a novel BED feature to capture background enclosure.

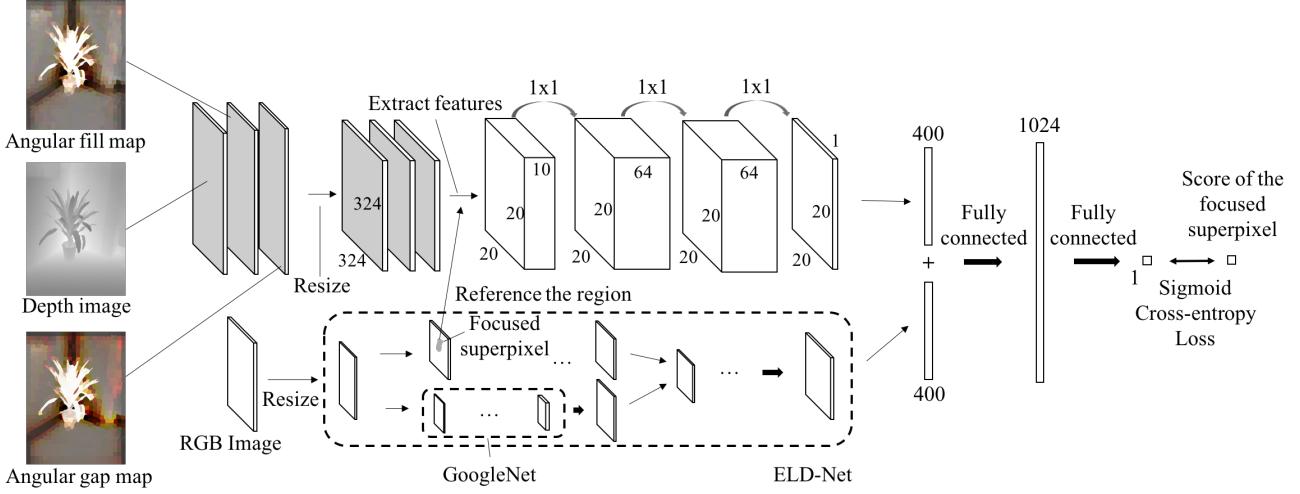


Figure 2. The whole architecture of our method. We extract ten superpixel-based handcrafted depth features for inputs (Section 3.1 and 3.2). Then we combine the depth features by concatenating the output with RGB low-level and high-level saliency features output (Section 3.3 and 3.4). Finally, we compute the saliency score with two fully connected layers.

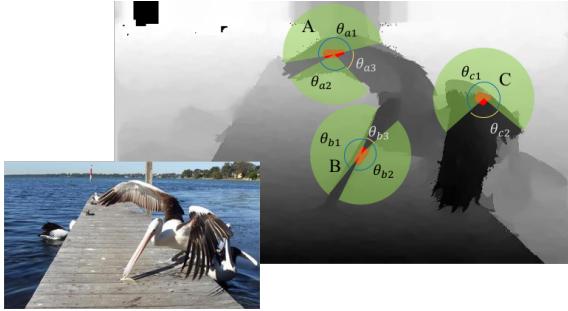


Figure 3. The concepts of the foreground function $f(P, t)$ and the opposing background function $g(P, t)$. For example, $f(P, t) = \frac{\theta_{a1} + \theta_{a2}}{2\pi}$ and $g(P, t) = \frac{\theta_{a3}}{2\pi}$ at point A.

3.1. BED Feature

High-level and low-level features have been shown to lead to high performance for detecting salient objects in RGB images in a deep CNN framework [15]. We also know that the effective encoding of depth input can improve convergence and final accuracy where training data is limited [9]. Here we add a novel mid-level feature that aims to represent the depth enclosure of salient regions for a learning approach, called the Background Enclosure Distribution (BED). BED relies on learning rather than hand-coded parameters that limit generalization.

Our proposed BED feature captures the enclosure distribution properties of a patch, that is, the spread of depth change in the surrounds, based on the idea that salient objects are more likely to be in front of their surroundings in a large number of directions. BED is inspired by LBE for

salient object detection, which has been shown to be an effective hand-crafted feature for non-learned salient object detection [6].

For each superpixel P , we define a foreground function $f(P, t)$ that measures the spread of directions (the integral over angle) in which P is in front of its background set, consisting of all patches with greater depth than $\text{depth}(P) + t$. That is, there is no superpixel in front of it in a particular direction, and at least one has greater depth. We also define an opposing background function $g(P, t)$ that measures the size of the largest angular region in which the superpixel is not in front of its background set.

We aim to measure the distribution of f and g over a range of distance to the background (i.e., t) to provide a stable representation of background enclosure. The distribution functions are given by:

$$F(P, a, b) = \int_a^b f(P, B(P, t)) dt \quad (1)$$

$$G(P, c, d) = \int_c^d 1 - g(P, B(P, t)) dt, \quad (2)$$

where (a, b) and (c, d) are some range of depth. We define a quantization factor q over the total range of depth of interest. Our BED feature consists of two distribution sets FF and GG :

$$FF(P, \sigma, q) = \{F(P, r, r - \sigma/q) | r \in \{\sigma/q, 2\sigma/q, \dots, \sigma\}\} \quad (3)$$

$$GG(P, \sigma, q) = \{G(P, r, r - \sigma/q) | r \in \{\sigma/q, 2\sigma/q, \dots, \sigma\}\}. \quad (4)$$

This provides a rich representation of image structure that is descriptive enough to provide strong discrimination between salient and non-salient structure.

Depth feature name	The number of the features
Depth of focused superpixel	1
Depth of the grid pixel	1
Depth contrast	1
Histogram distance	1
Angular density components	3
Angular gap components	3

Table 1. The depth features extracted from the focused superpixel and a grid cell.

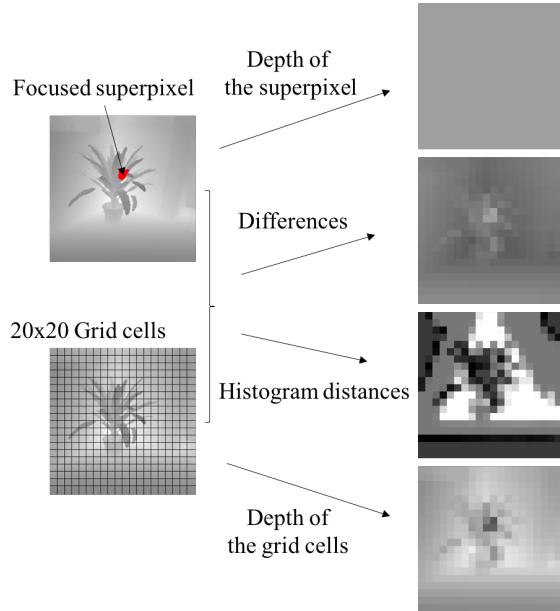


Figure 4. Our four 20×20 depth feature layers.

We construct a 20×20 feature layer for each of these distribution slices. This results in $2q$ feature layers for our BED feature.

3.2. Low-level Depth Features

In addition to background enclosure, we also capture the idea of depth contrast, that has been shown to be effective in previous work [13, 22, 24]. In addition to six BED features, we extract four low-level depth features from every superpixel. The extracted features are illustrated in Table 1 and Figure 4.

We use the SLIC algorithm [2] on the RGB image to segment it into superpixels (approximately 18×18 superpixels per image). In every learning step, we focus on one superpixel, calculate how salient the superpixel will be, compare it with ground truth, and perform back propagation.

For every focused superpixel, we calculate the average depth value to form a 20×20 layer of these values. We also subdivide the image into 20×20 grid cells and calculate

the average value for each to form a 20×20 layer. To capture depth contrast (local and global) that has been shown to be effective in RGB-D saliency, we create a 20×20 contrast layer between the depth of the superpixels and grid cells. We compute the contrast layer simply by subtracting the average depth value of each grid cell from the average depth value for each superpixel. Finally, we calculate the difference between the depth histogram of the focused superpixel and grid cells. We divide the entire range of depth values into 8 intervals and make the histogram of the distribution of the depth values of each superpixel and grid cell. To measure histogram contrast, we calculate the χ^2 distance between focused superpixel and the grid pixel features. The equation is illustrated in Equation (5):

$$f(x, y) = \frac{1}{2} \sum_{i=1}^{8} \frac{(x_i - y_i)^2}{(x_i + y_i)}, \quad (5)$$

where x_i is the number of depth values in quanta i for the superpixel, and y_i is the number of depth values in the range i for the grid cell. These features are also inspired by the RGB features that are shown to be effective in the original version of ELD-Net [15].

3.3. RGB low and high level saliency from ELD-Net

To represent high-level and low-level features for RGB, we make use of the extended version of ELD-Net [15]. We choose ELD-Net because this method is a state-of-the-art RGB saliency method and the network architecture is easy to extend to RGB-D saliency. From personal correspondence, Lee *et al.* published the source code for a better performing method in <https://github.com/gylee1103/ELDNet>. Rather than using VGG-Net as per the ELD-Net paper, this version uses GoogleNet [27] to extract high level features, and does not incorporate all low-level features.

3.4. Non-linear combination of depth features

The 20×20 depth images, the other low-level feature maps that capture depth contrast, and the BED feature map, as described in Section 3.1 and 3.2, need to be combined to capture background enclosure, depth contrast, and absolute depth. We also incorporate RGB information from the focused superpixel. In order to capture non-linear combinations of these, we use three convolutional layers followed by a fully convolutional layer to form our depth output.

3.5. Concatenation of Color and Depth Features

In order to effectively exploit color features, we make use of the pretrained caffemodel of ELD-Net [15] to initialize the weights of color features. The calculated $1 \times 20 \times 20$ color feature layers are concatenated with the depth feature output as shown in the Figure 2.

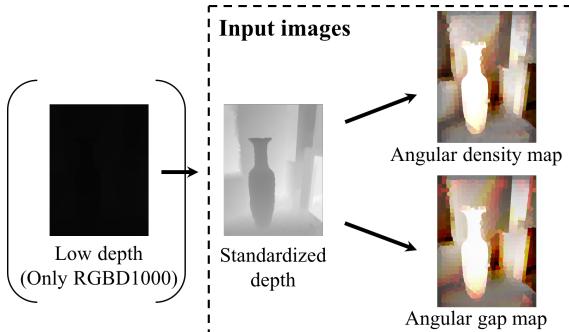


Figure 5. Developing input images from a depth image.

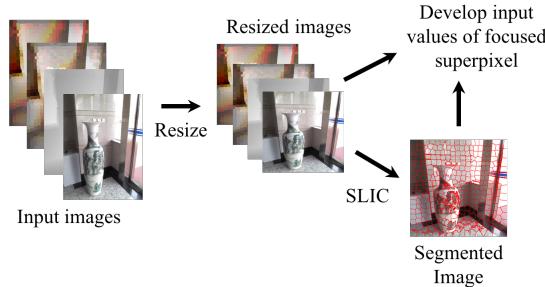


Figure 6. Extracting feature values of the focused superpixel from various input images.

We then connect the $1 \times 20 \times 20 + 1 \times 20 \times 20$ concatenated output features with a fully connected layer and calculate the saliency score for the focused superpixel. We calculate the cross entropy loss for a softmax classifier to evaluate the outputs. The cross entropy loss is calculated as follows:

$$E = -\{p \log \hat{p} + (1 - p) \log(1 - \hat{p})\}, \quad (6)$$

where p is the calculated saliency score of the focused superpixel and \hat{p} is the average saliency score for the ground truth image.

4. RGB-D saliency detection system

We develop our learning architecture for salient object detection based on the Caffe [11] deep learning framework. For faster learning, our training uses CUDA on a GPU.

4.1. Preprocessing on depth and color images

Since we concatenate the color and depth values, we want to synchronize the scale of depth values with color values. Hence, if required, we normalize the depth value to the same scale, i.e., 0 to 255, before extracting depth features. Depth values of RGBD1000 [21] are represented with greater bit depth and so require normalization. On NJUDS2000 [13] the scale of depth values are already 0

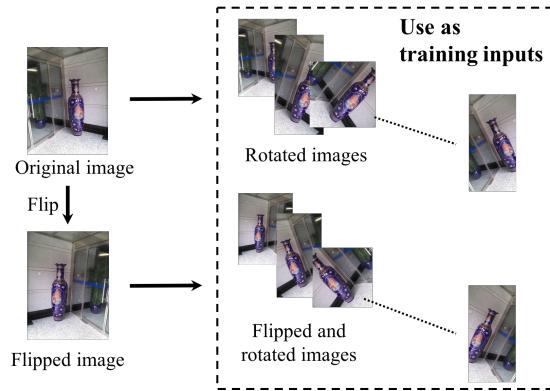


Figure 7. Increasing the number of training datasets by rotating and flipping.

- 255, and so are not modified. After normalization, we resize the color and depth images to 324×324 .

4.2. Superpixel Segmentation

We use gSLICr [23], the GPU version of SLIC, to segment the images into superpixels. We divide each image into approximately 18×18 superpixels, following Lee *et al.* [15]. Note that gSLICr may combine small superpixels with nearby superpixels [23].

4.3. Extracting low-level depth features

Following this, we create four 20×20 layers from each of the low-level depth features. The first consists of the average value of the spatially corresponding focused superpixel for each of the 20×20 inputs; the second is composed from the average depth values of 20×20 grid cells; the third layer consists of the difference of depth values between the mean depth of the focused superpixel and the mean depth of each of the grid cells; and the last layer consists of the histogram distance between the superpixel and grid cells. Figure 4 illustrates this process.

4.4. Extracting BED features

In order to calculate BED efficiently, we pre-compute the angular density components and angular fill components. Three channels are computed for each of equation (2) and (3), where $q = 3$ over the intervals between $0, \frac{\sigma}{3}, \frac{2\sigma}{3}, \sigma$ where σ is the standard deviation of the mean patch depths. The calculated values are connected to our architecture in the same way as loading color images. For each focused superpixel, we calculate each BED feature, for a total of six 20×20 feature maps. These are concatenated with depth to form a $(4 + 6) \times 20 \times 20$ feature input for each focused super pixel.

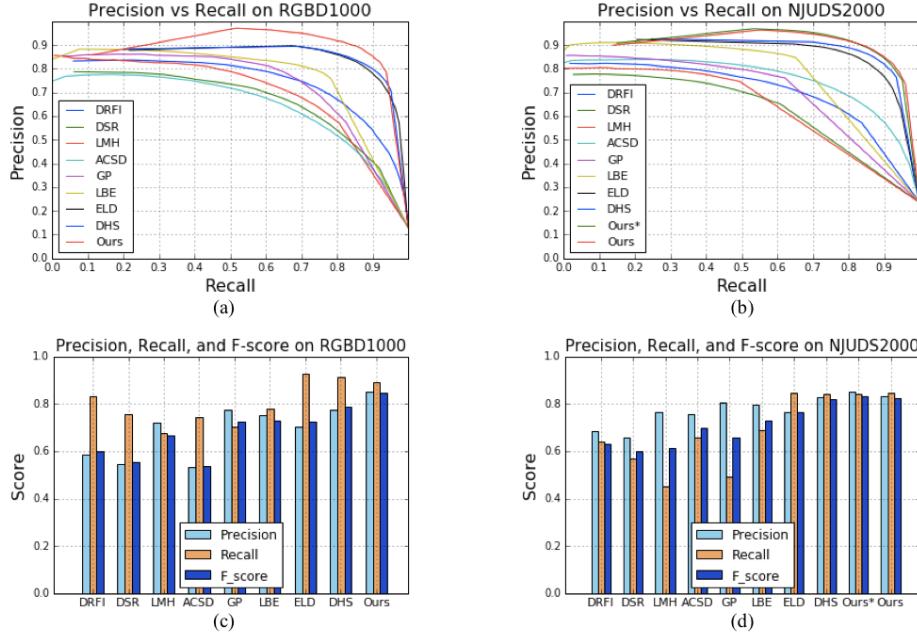


Figure 8. Comparing performance of our methods with other RGB-D saliency methods. The PR curve of our method and the other current RGB-D salient object detection methods on (a) RGBD1000 and (b) NJUDS2000. The F-score of our method and the other current methods on (c) RGBD1000 and (d) NJUDS2000.

	RGBD1000	NJUDS2000
DRFI [12]	0.597689	0.631111
DSR [17]	0.556269	0.599731
LMH [21]	0.667106	0.611340
ACSD [13]	0.535048	0.696379
GP [24]	0.723314	0.655915
LBE [6]	0.727232	0.729319
ELD [15]	0.724766	0.764561
DHS [18]	0.787486	0.817152
Ours	0.847635	0.821269

Table 2. Comparing average F-measure score with other state-of-the-art saliency methods on two datasets.

4.5. Improving the learning rate

To help address the scarcity of RGB-D salient object datasets, we enhance the training datasets by flipping and rotating images. We made 16 rotated images by rotating the image by 22.5 degree in each step, then each of these is also flipped. As a result, the enhanced training dataset has 32 times as many images as the original. For RGBD1000 [21], we make 19200 training images from 600 original images. On NJUDS2000 [13], we make 38400 training images from the original 1200 images.

The weights for ELD [15] can be initialized with a fine-tuned caffemodel. However, this is not suitable for depth, so

the weights for depth are initialized randomly. This means the weights for depth need a higher learning rate compared to weights of ELD. We set the learning rate for depth to be 10 times as much as for color. We set our initial base learning rate as 0.05. This means the initial learning rate of depth layers is 0.5. We reduce the learning rate by multiplying by 0.1 in every 10000 learning steps. 1000 superpixels are used for training in every step.

After training with the flipped and rotated datasets, we train our model using only the original images. This is because we assume that the most salient object may change for some images or their saliency maps may become incorrect when the images are flipped or rotated. We use two stage training to mitigate any possible negative effect. In the final original image only training stage, we set the learning rate as 0.01 and do not modify it. We train for 900 steps for RGBD1000 [21] and 1000 steps for NJUDS2000 [13]. 1000 superpixels are used for training in every step.

5. Experimental Evaluation

We evaluate our architecture's performance on two datasets: RGBD1000 [21] and NJUDS2000 [13]. On RGBD1000, we randomly divide the dataset into 600 images for a training set, 200 images for a validation set, and 200 images for test set. On NJUDS2000, we randomly divide the datasets into 1200 images for a training set, 385 images for a validation set, and 400 images for a test set.

The results are compared against other state-of-the-art RGB-D saliency detection methods: local background enclosure (LBE) [6]; and multi-scale depth-contrast (LMH) [21]; and saliency based on region contrast and background, depth, and an orientation prior (GP) [24]; anisotropic center-surround depth based saliency method (ACSD) [13]. We compare our results also with RGB saliency detection systems: DRFI [12] and DSR [17] which produce good scores [3]. Finally, we also add two state-of-the-art CNN-based RGB saliency detection approaches: saliency from low and high level features (ELD) [15]; and the Deep hierarchical saliency network (DHS) [18].

5.1. Evaluation Criteria

Like the other state-of-the-art RGB-D salient detection methods [6, 21, 24], we calculate the precision-recall curve and mean F-score for evaluating our results. The F-score is calculated as a following equation:

$$F_\beta = \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \quad (7)$$

where $\beta = 0.3$ in order to put more emphasis on precision than recall [1].

5.2. Experimental Setup

As mentioned, we perform training with the datasets augmented with rotated and flipped images, and then train with the original images only. In both cases, we use Adadelta optimizer [29] for updating weights. For training with the augmented datasets, we set the base learning rate as 0.05, a decay constant ρ as 0.9, and the constant ϵ as 1e-08. We decrease the base learning in every 10000 iterations by multiplying the base learning rate by 0.1. We perform 50000 training iterations on RGBD1000 [21] and NJUDS2000 [13]. Then for training with the original images only, we set the base learning rate to 0.01, a decay constant ρ to 0.9, and the constant ϵ to 1e-08. We perform 900 training iterations on RGBD1000 [21] and 1000 iterations on NJUDS2000 [13]. These parameter values were determined by performance on validation datasets.

5.3. Results

Our learning architecture outperforms the other RGB-D salient object detection methods (Figure 8a and 8b, Table 2). Our method is particularly effective for high recall rates with respect to other methods. Our approach outperforms the results of bottom-up approaches such as LBE [6] and LMH [21] (Figure 8a and 8b). In addition, compared to other top-down RGB salient object detection systems such as ELD-Net [15] and DHSNet [18], our approach performs better on the P-R curve and F-score.

On the NJUDS2000 [13], we perform training without using χ^2 distance of histogram difference of the depth of

	Precision	Recall	F-measure
Ours	0.834091	0.843668	0.821269
with mean depth	0.850724	0.840648	0.833303

Table 3. Replacing the superpixel histogram with mean depth improves results for NJUDS2000 [13] where depth data is noisy.

	Precision	Recall	F-measure
RGB only (ELD)	0.700276	0.927371	0.724766
Without BED	0.840953	0.891432	0.840704
Ours	0.84833	0.890801	0.847635

Table 4. Comparing scores with different input features on RGBD1000 [21].

	Precision	Recall	F-measure
RGB only (ELD)	0.766507	0.844895	0.764561
Without BED	0.830839	0.841835	0.816602
Ours	0.834091	0.843668	0.821269

Table 5. Comparing scores with different input features on NJUDS2000 [13].

the superpixel and grid cells, and using the average depth of the superpixel instead. This is because the quality of the depth images is not as good on NJUDS2000 datasets, as the depth images are captured by stereo camera. This change leads to an improvement in performance. (Figure 8b and 8d, Table 3) We name this method as Ours* in Figure 8. In general, this may be an effective approach if training data has noisy depth.

In order to evaluate the effect of the BED features, we measure the performance of our methods without using BED features. We perform training in the same architecture other than BED features, perform the same training, and use the same measures of performance. Tables 4 and 5 shows the results. The tables contain average precision, recall, and F-measure of three methods, the result of ELD-Net [15], our network without using six BED features, and our architecture. As can be seen BED contributes to an increase in the results. On the RGBD1000 dataset, precision increases well while holding the same recall. On NJUDS2000 datasets, precision increases and recall rate also increases slightly.

Figure 9 shows the output of our architecture with the other state-of-the-art methods.

6. Conclusion

In this paper, we proposed a novel architecture that provides the first complete RGB-D salient object detection systems using a deep CNN. We incorporate a novel mid-level feature, BED, to capture background enclosure, as well as low level depth cues that incorporate depth contrast, and high level features. Our results demonstrate that our novel

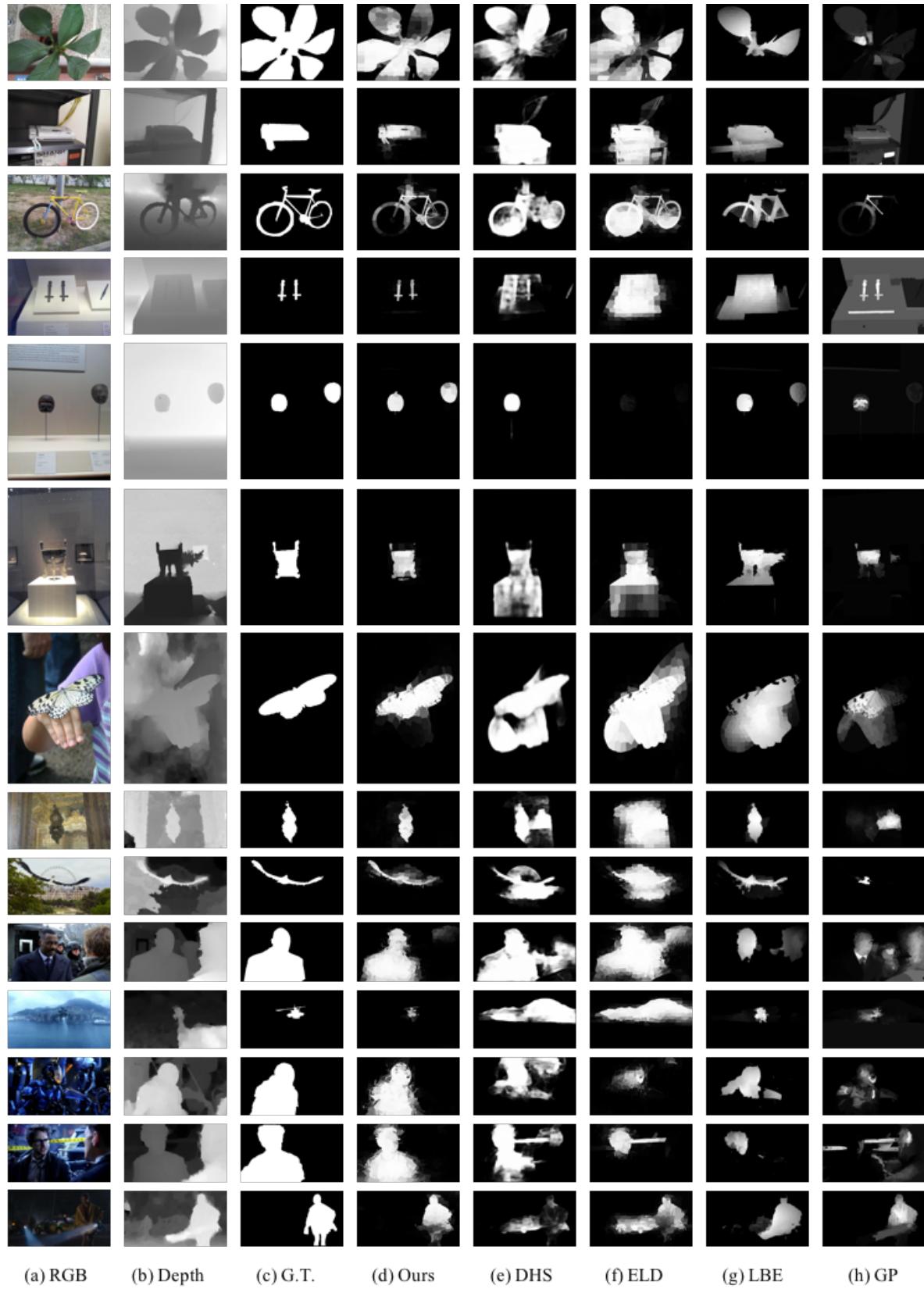


Figure 9. Comparing outputs of our architecture against DHS [18], ELD [15], LBE [6], GP [24]. Note that G.T. means Ground Truth.

architecture outperforms other RGB-D salient object detection methods. Further, we show that adding low-level depth and BED each yield an improvement to the detection results.

References

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Sussstrunk. Frequency-tuned salient region detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, June 2009.
- [2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sussstrunk. Slc superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.
- [3] A. Borji, M. M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, Dec 2015.
- [4] M. Cheng, J. Warrell, W. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *ICCV*, page 15291536, 2013.
- [5] M.-M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 409416, 2011.
- [6] D. Feng, N. Barnes, S. You, and C. McCarthy. Local background enclosure for rgb-d salient object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2343–2350, June 2016.
- [7] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, 19(1):185–198, Jan 2010.
- [8] J. Guo, T. Ren, J. Bei, and Y. Zhu. Salient object detection in rgb-d image based on saliency fusion and propagation. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service, ICIMCS ’15*, pages 59:1–59:5, New York, NY, USA, 2015. ACM.
- [9] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. *Learning Rich Features from RGB-D Images for Object Detection and Segmentation*, pages 345–360. Springer International Publishing, Cham, 2014.
- [10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.
- [11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia, MM ’14*, pages 675–678, New York, NY, USA, 2014. ACM.
- [12] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
- [13] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu. Depth saliency based on anisotropic center-surround difference. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1115–1119, Oct 2014.
- [14] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan. *Depth Matters: Influence of Depth Cues on Visual Saliency*, pages 101–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [15] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [16] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [17] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.
- [18] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [19] P. Luo, Y. Tian, X. Wang, and X. TangLuo. Switchable deep network for pedestrian detection. In *Computer Vision and Pattern Recognition*, pages 899–906. IEEE Press, 2014.
- [20] V. Mahadevan and N. Vasconcelos. Saliency-based discriminant tracking. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1007–1013, 2009.
- [21] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji. Rgbd salient object detection: a benchmark and algorithms. In *European Conference on Computer Vision*, pages 92–109. Springer, 2014.
- [22] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang. RGBD salient object detection via deep fusion. *CoRR*, abs/1607.03333, 2016.
- [23] C. Y. Ren, V. A. Prisacariu, and I. D. Reid. gSLICr: SLIC superpixels at over 250Hz. *ArXiv e-prints*, Sept. 2015.
- [24] J. Ren, X. Gong, L. Yu, W. Zhou, and M. Ying Yang. Exploiting global priors for rgb-d saliency detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2015.
- [25] V. Setlur, T. Lechner, M. Nienhaus, and B. Gooch. Retargeting images and video for preserving information saliency. *IEEE Computer Graphics and Applications*, 27(5):80–88, Sept 2007.
- [26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [28] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [29] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

- [30] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

7. Supplementary material

7.1. Introduction

The purpose of this supplementary material is to analyze the results of our method more closely. First, we examine cases where our method succeeds in reducing false positives compared to: the other state-of-the-art RGB learning based saliency methods, DHSNet [18] and ELDNet [15]; and RGB-D bottom-up saliency methods, LBE [6] and GP [24]. An increase of the precision rate is shown on the Figure 8c and 8d. This means our architecture is better able to reduce false positives compared to previous state-of-the-art methods. We also show cases where our method succeeds in reducing false negatives compared to DHS [18], ELD [15], LBE [6], and GP [24]. Then, in order to analyze the performance of the BED features we compare the output with that when BED features are not incorporated. Finally, we illustrate our failure cases and analyze the reasons.

Please find the attached all test outputs of our method, LBE [6], DHS [18], and GP [24]. The outputs of LBE [6], DHS [18], and GP [24] are obtained by running code from the authors' websites.

7.2. Comparing our results with other methods.

As can be seen in Figure 8c and 8d, a strong point of our method is that it increases precision while maintaining a high recall rate compared to other state-of-the-art RGB salient object detection methods [15, 18], and has a higher precision and recall rate than other RGB-D state-of-the-art salient object detection methods [6, 24]. This means our method is better able to exclude false positives. We provide such examples in Figure 10. In these examples, RGB top-down methods are able to detect the correct regions, however, these methods also detect many false positives. By combining color and depth data, our method solves this problem. In Figure 10 row 1, for example, the salient regions are difficult to detect without knowledge about the butterfly. Thus, the other state-of-the-art RGB and RGB-D methods fail to detect the region, while our method succeeds in detecting the insect utilizing both the color and depth images. A similar output is also shown in Figure 10 row 2. Though state-of-the-art RGB salient object detection methods fail and the salient regions are difficult to detect from the depth map, our method succeeds in detecting the salient regions combining color and depth features. In Figure 10 row 3-7, our method succeeds in detecting the salient objects. Though the other RGB top-down methods succeed in detecting the salient regions, they also find false posi-

tives. On these examples, other RGB-D bottom-up methods detect many false positives and negatives.

As mentioned above, our method is high capable of excluding false positives, but our method also helps to reduce false negatives. We illustrate such examples in Figure 11. For all examples, other state-of-the-art RGB and RGB-D methods fail to detect the entire salient regions. For example, in Figure 11 row 1, the other methods fail to detect all leaves, while our method succeeds by utilizing color and depth features.

7.3. Comparing our outputs including and not including BED features.

For further analysis of the effectiveness of the BED features, we compare the outputs using our full system versus our system with BED features excluded. Figure 12 shows examples where the BED features lead to improved performance.

BED features help to reduce false positives in some cases, for example, see Figure 12 row 1-4. Figure 12 row 1 shows this effect clearly. By making use of the BED features, our method succeeds in excluding the chair at the back and reduces false positives. BED features also help to reduce false negatives in some images, for example, see Figure 12 row 5-7. In Figure 12 row 5, our method is more successful at detecting the salient objects thanks to the BED features. Similar effects can be seen in Figure 12 row 6 and 7.

7.4. Failure cases.

Finally, we include examples of failure cases of our method in Figure 13. At first sight of the image in Figure 13 row 1, we may consider that only the wheel is a salient object. However, actually the wheel is connected with the square pole in the back, which is also labeled as salient in this dataset. In such a case, it is difficult to detect the whole salient object without knowledge of the object. In Figure 13 row 2, we can see a similar situation. At first sight, we may think only the pink T-shirt is a salient region, but actually the skirt is also labeled as salient. Humans can detect both of them as salient objects because we know both T-shirts and skirts are clothes, but our algorithm does not. In Figure 13 row 3, humans label only the cat as a salient object because it is an animal and we usually pay more attention to animals than non-living objects such as shoes. However, for an algorithm it is difficult to detect only the cat without such knowledge. In Figure 13 row 4, saliency is strongly related with face direction. The man on the left attracts more attention because he is looking more towards the camera. Our algorithm seems not to consider this fact. Saliency in Figure 13 row 5 is related to the face direction of the statue and the fact that humans usually pay more attention to statues than trees. The center tree is also detected as a salient object by our algorithm, we believe, because it seems

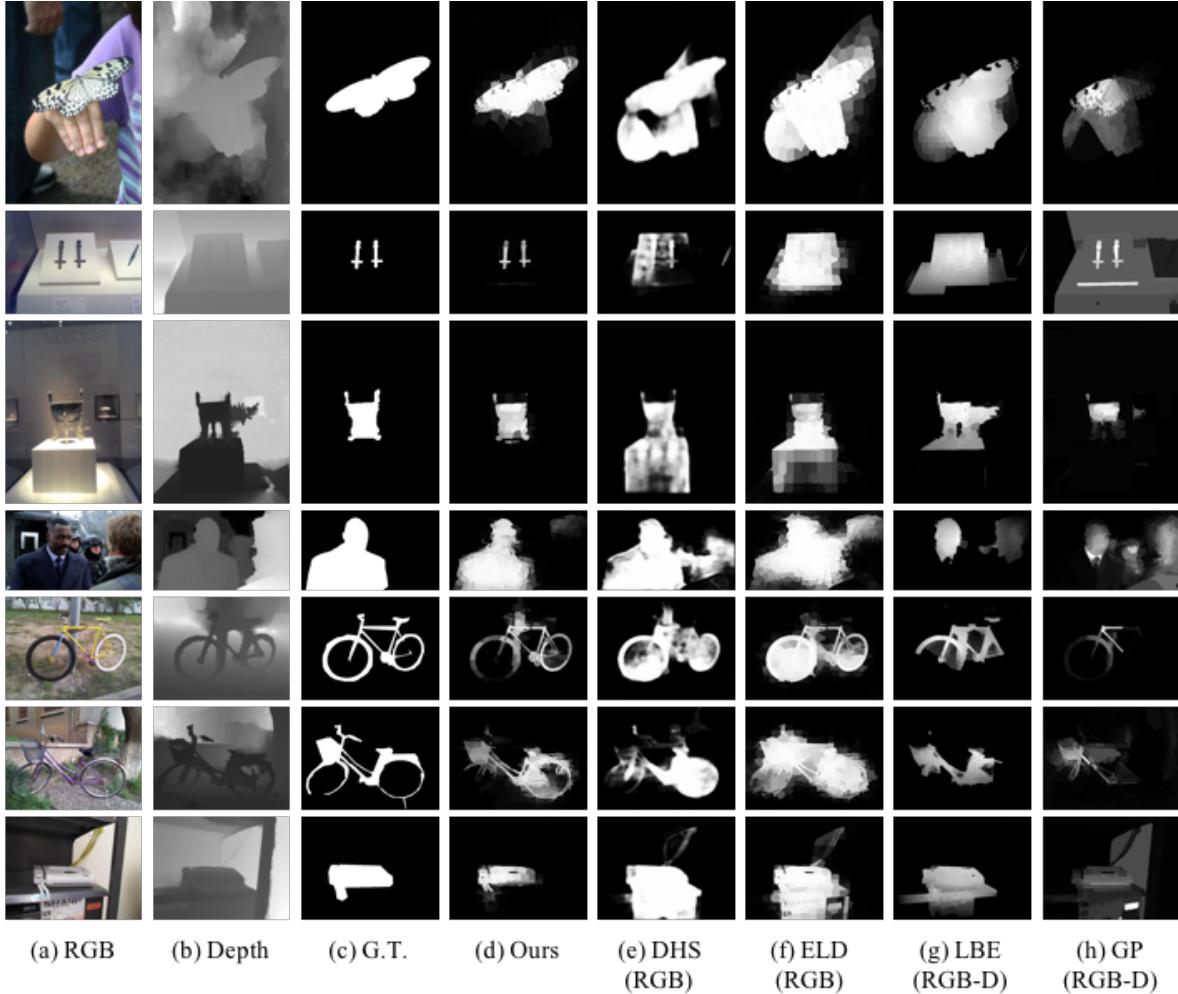


Figure 10. Cases where our method succeeds in reducing false positives compared to state-of-the-art methods, DHS [18], ELD [15], LBE [6], GP [24]. Note that RGB under the name represents RGB learning based methods, and RGB-D represents RGB-D bottom-up approach methods. G.T. means Ground Truth.

not to have such knowledge. Figure 13 row 6 is a difficult case because the salient object is very small and prerequisite knowledge that an object on the desk is more important than the desk is needed. Figure 13 row 7 is a case where both color and depth features do not seem to be sufficient detecting the salient regions. Saliency here is quite subtle. Our method can detect similar regions in this case but not clearly.

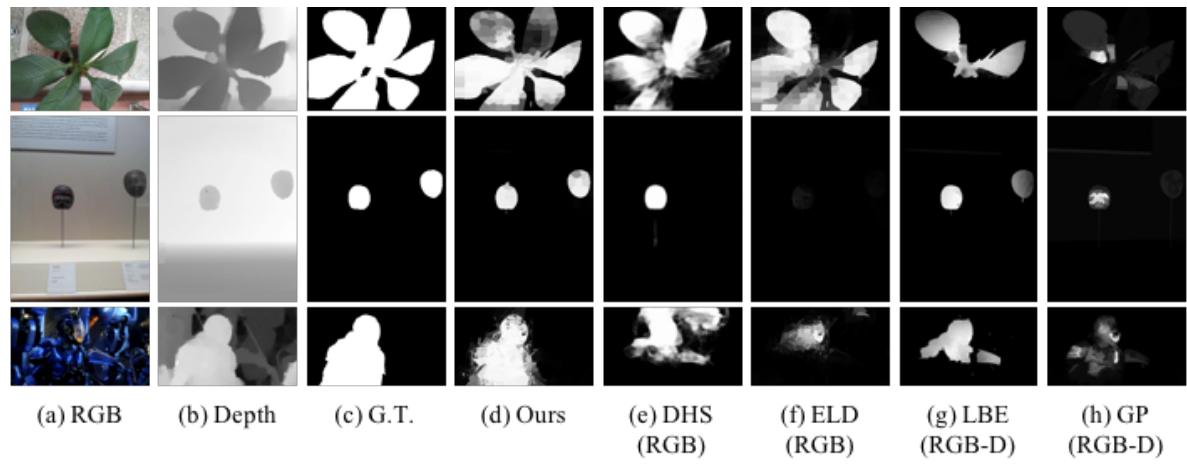
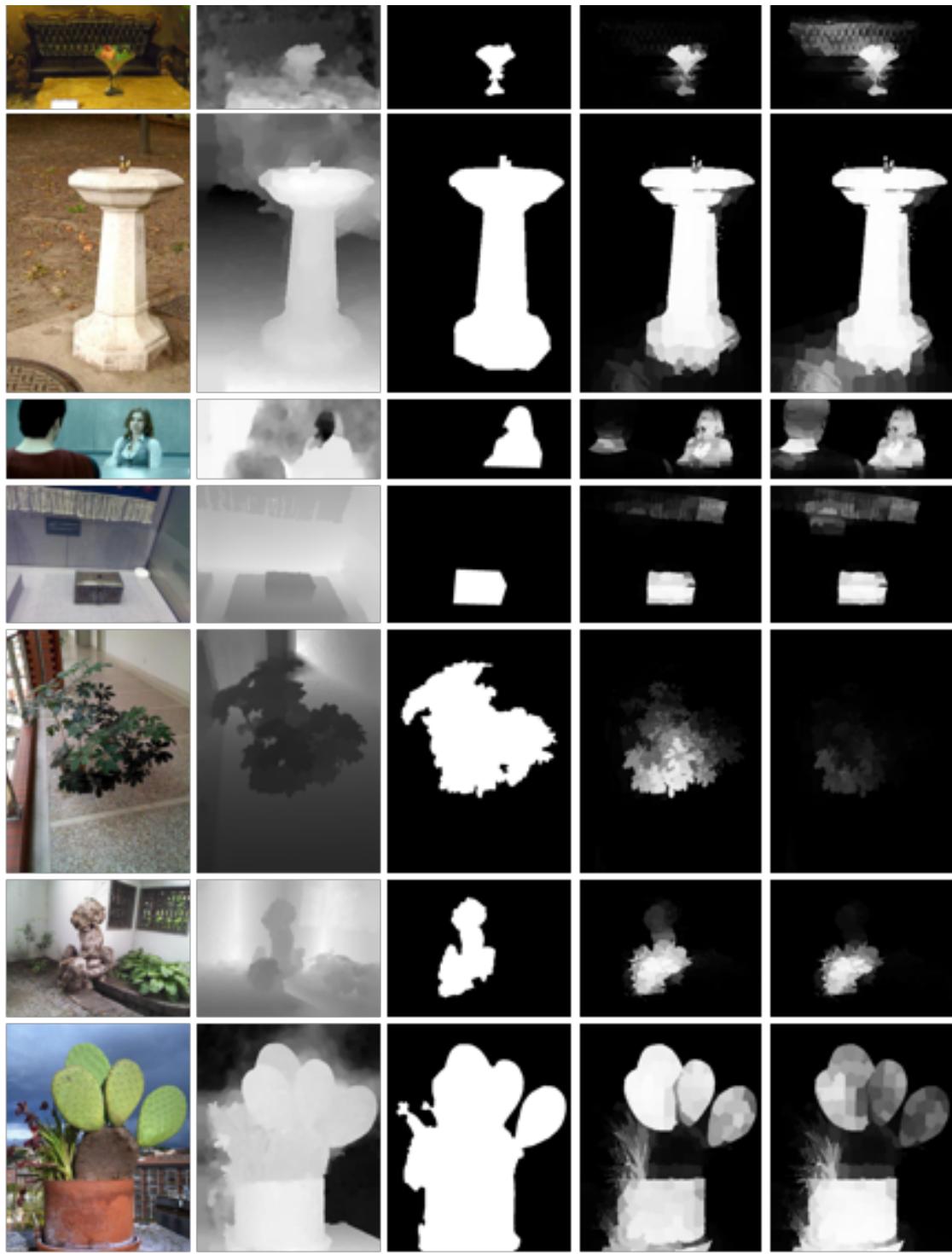


Figure 11. Cases where our method succeeds in reducing false negatives compared to other state-of-the-art methods, DHS [18], ELD [15], LBE [6], GP [24]. G.T. means Ground Truth.



(a) RGB

(b) Depth

(c) G.T.

(d) Ours

(e) Without BED

Figure 12. Comparing the results with BED features and without BED features. BED features help to reduce false positives and false negatives in some cases. G.T. means Ground Truth.

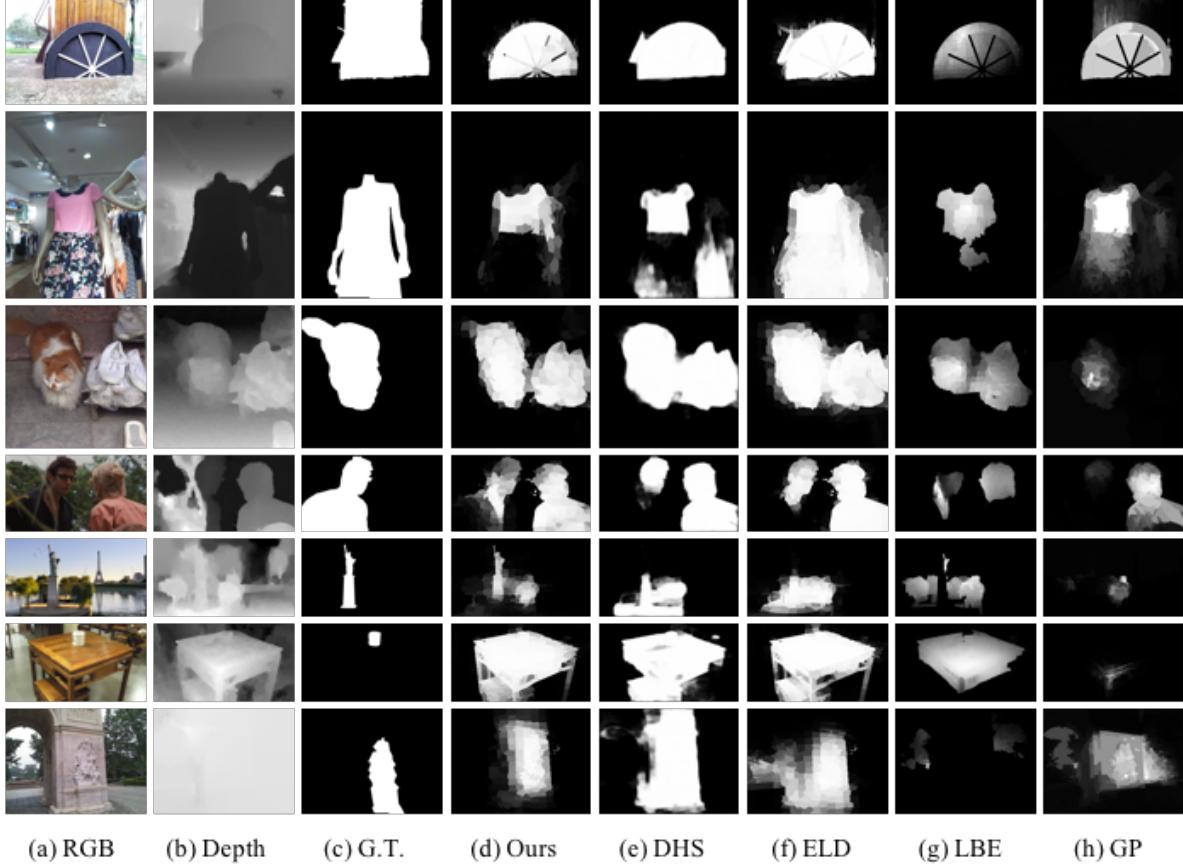


Figure 13. Examples of failure cases of our architecture. We also show the outputs of other state-of-the-art methods, DHS [18], ELD [15], LBE [6], GP [24]. Our method fails to detect salient regions when particular knowledge is needed, such as what is wheel or role of face direction in attention. G.T. means Ground Truth.