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

In human visual saliency, top-down and bottom-up information are combined as a basis of visual attention. Recently, deep Convolutional Neural Networks (CNN) have demonstrated strong performance on RGB salient object detection, providing an effective mechanism for combining top-down semantic information with low level features. Although depth information has been shown to be important for human perception of salient objects, the use of top-down information and 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 utilizes both top-down and bottom-up cues. In order to produce such an architecture, we present novel depth features that capture the ideas of background enclosure, depth contrast and histogram distance in a manner that is 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 results. In particular, the F-Score of our method is 0.848 on RGBD1000, which is 10.7% better than the current best.

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]. In recent years, research has focused on salient object detection, finding salient objects or regions in an image (*e.g.*, [1, 3]).

Most existing 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].

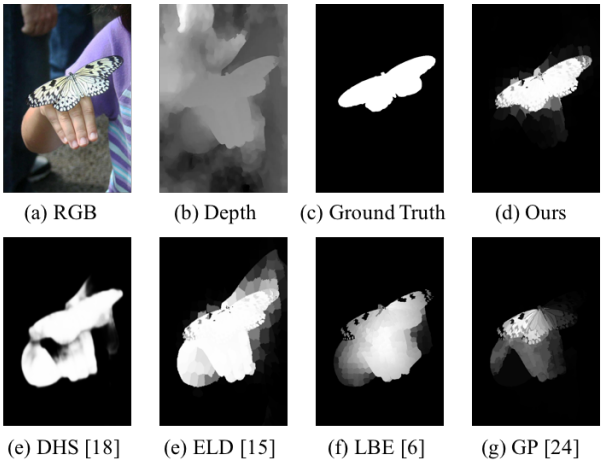

Figure 1. Comparing our RGB-D salient object detector output with other salient object detection methods. An example of which both low-level feature from color and depth, and high level semantic information are important.

Thus, RGB-D salient object detection methods have been proposed [6, 13, 21, 22, 24] and have demonstrated superior performance in comparison to RGB-only methods.

In theory, humans adopt both bottom-up and top-down strategies for saliency [10]. 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 [15, 18, 30].

However, it is not yet clear whether combining top-down information using deep CNNs is 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 plays an important role in human attention [10], and

has been shown to be effective in RGB salient object detection. Further, in RGB-D salient object detection, the effectiveness of background enclosure and of depth contrast have been demonstrated [6]. Finally, deep CNNs have been shown to be effective for RGB salient object detection [15, 18, 30] particularly in introducing top-down information.

This paper makes three major contributions. (1) We propose a novel learning architecture that provides the first complete RGB-D salient object detection system utilizing both top-down and bottom-up methods. (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 and depth histogram distance.

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 our low-level features based on depth contrast lead to better performance than RGB saliency alone, and that adding our new mid-level feature, BED, improves results further.

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 a region versus its surrounds, locally and/or globally [5, 10]. Contrast is mostly computed with respect to appearance-based features (e.g., color, 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 an 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].

Most 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. A preliminary paper 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 dataset contains 1000 RGB-D images captured by a standard Microsoft Kinect. The NJUDS2000 dataset contains 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.

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].

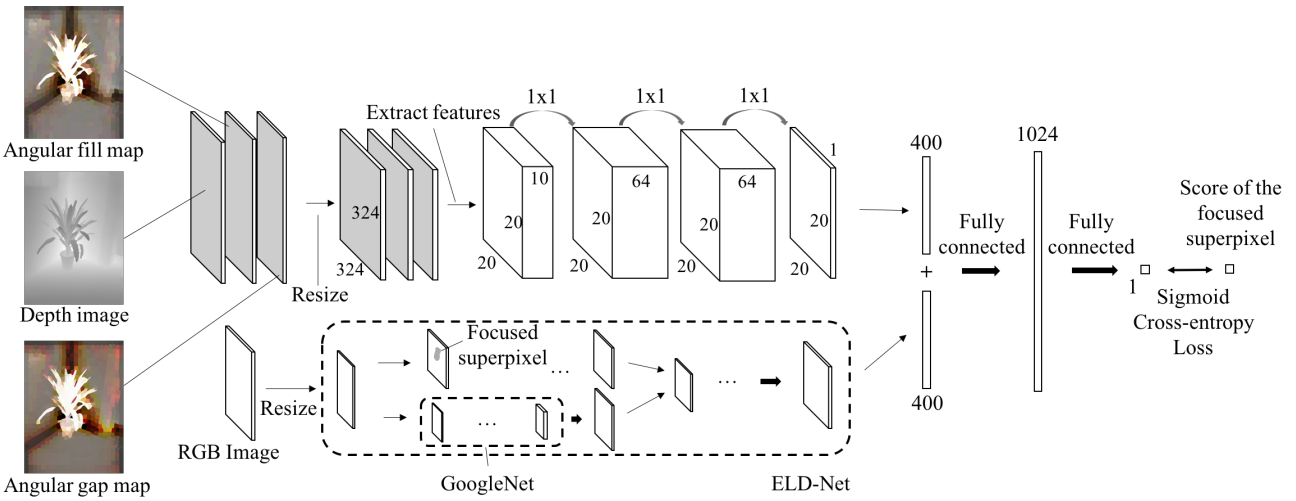

Figure 2. The whole architecture of our method. We extract ten superpixel-based handcrafted depth features for inputs (Section 3.1 and 3.2). Then we combine the depth features by concatenating the output with RGB low-level and high-level saliency features output (Section 3.3 and 3.4). Finally, we compute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

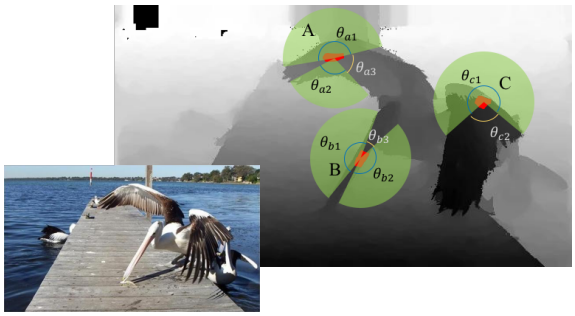

Figure 3. The concepts of the foreground function $f(P,t)$ and the opposing background function $g(P,t)$. For example, $f(P,t) = \frac{\theta_{a1}+\theta_{a2}}{2\pi}$ and $g(P,t) = \frac{\theta_{a3}}{2\pi}$ at point A.

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 defined by the threshold t, consisting of all patches with greater depth than $depth(P) + t$. Specifically, f computes the portion of angles $\theta \in [0, 2\pi)$ for which the line emanating from P with angle θ intersects this background set. We also define an opposing background function g 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 background thresholds (i.e., t) to provide a stable representation of background enclosure. The distribution functions are given by:

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

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

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 denoted by σ. Our BED feature consists of two distribution sets $\mathcal{F}$ and $\mathcal{G}$:

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

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

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

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]. Moreover, we utilize the depth histogram distance which is inspired by a color histogram distance in ELD-Net [15]. 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.

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
BED features	6

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

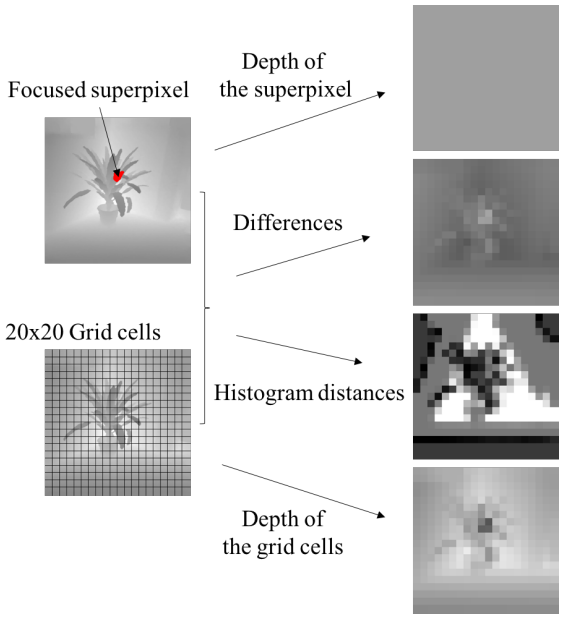

Figure 4. Our four 20×20 depth feature layers.

For every focused superpixel, we calculate the average depth value to form a 20×20 layer of these values. This layer contains the same 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. This is a new depth feature inspired by the difference of color histogram in ELD-Net [15]. 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. This is captured in Equation (5):

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

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 one of the state-of-the-art RGB saliency methods and, as can be seen, the network architecture can be extended 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 [26] 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 low-level feature maps and the BED feature maps, as described in Section 3.1 and 3.2, need to be combined for detecting salient objects. In order to combine these features well, we use three convolutional layers to form depth feature outputs.

3.5. Concatenation of Color and Depth Features

In order to effectively extract 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 layer is concatenated with the depth feature outputs as shown in the Figure 2.

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

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

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 above-mentioned 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.

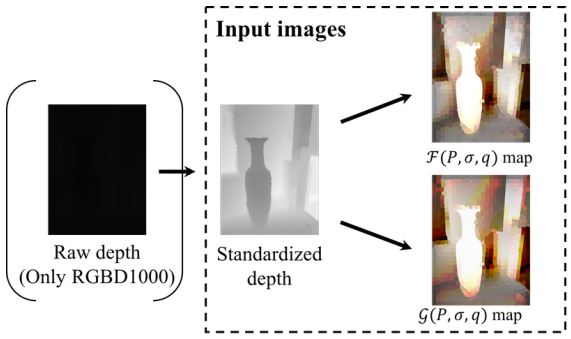

Figure 5. Developing input images from a depth image. The raw depth seems very dark because this map illustrates actual distances.

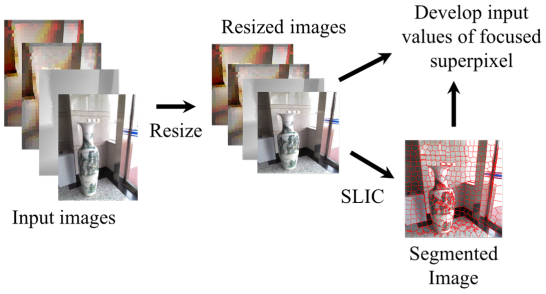

Figure 6. Extracting feature values of the focused superpixel from various input images.

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 - 255, and so are not modified.

After normalization, we resize the color and depth images to 324×324. Figure 5 and 6 represent these processes.

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 these processes.

4.4. Extracting BED features

In order to calculate BED features efficiently, we precompute them. Three channels are computed for each of equation (3) and (4), 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 5 illustrates these processes.

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 a 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]; multi-scale depth-contrast (LMH) [21]; saliency based on region contrast and background, depth, and an orientation prior (GP) [24]; and anisotropic center-surround depth based saliency (ACSD) [13]. We compare our results also with RGB saliency detection systems: DRFI [12] and DSR [17] which produce good scores [3]. 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]. For evaluating all of the above methods, we use the same our test split. Finally, we compare our results with a CNN-based RGB-D salient object detection method (DF) [22]. As DF is learning based and uses randomly sampled train and test splits, we refer their reported score.

5.1. Evaluation Criteria

Like the other state-of-the-art RGB-D salient detection methods [6, 21, 22, 24], we calculate the precision-recall curve and mean F-score for evaluating our results. The F-

score is calculated as a following equation:

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

where $\beta = 0.3$ to place more emphasis on precision [1].

5.2. Experimental Setup

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. 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 and for [13], 38400 training from 1200 original images.

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. The weights for ELD [15] can be initialized with a fine-tuned caffemodel. However, this is not suitable for depth, because 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 base learning rate for depth as 0.5. We decrease the base learning rate in every 10000 iterations by multiplying the base learning rate by 0.1. We perform 50000 training iterations on RGBD1000 [21] and NJUDS2000 [13]. 1000 superpixels are used for training in every step. Next we train with the original images only. 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 set the all 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]. 1000 superpixels are used for training in every step. These parameter values were determined by performance on validation datasets.

5.3. Results

Our learning architecture outperforms 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. Our model also gives a better score than other top-down RGB-D salient object detection system such as DF [22].

	RGBD1000	NJUDS2000
DRFI [12]	0.6017	0.6291
DSR [17]	0.5529	0.6000
LMH [21]	0.6756	0.6010
ACSD [13]	0.5618	0.6859
GP [24]	0.7232	0.6418
LBE [6]	0.7306	0.7419
ELD [15]	0.7248	0.7646
DHS [18]	0.7875	0.8172
DF [22]	0.7823	0.7874
Ours	**0.8476**	**0.8213**

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

	Precision	Recall	F-measure
Ours	0.8341	0.8437	0.8213
with mean depth (Ours)	0.8507	0.8406	0.8333

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)	0.7003	0.9274	0.7248
RGB+LD (Ours)	0.8410	0.8914	0.8407
RGB+LD+BED (Ours)	0.8483	0.8908	0.8476

Table 4. Comparing scores with different input features on RGBD1000 [21]. Note that LD means Low-level Depth Features.

	Precision	Recall	F-measure
RGB only (ELD)	0.7665	0.8449	0.7646
RGB+LD (Ours)	0.8308	0.8418	0.8166
RGB+LD+BED (Ours)	0.8341	0.8437	0.8213

Table 5. Comparing scores with different input features on NJUDS2000 [13]. Note that LD means Low-level Depth Features.

On the NJUDS2000 [13], we perform training without using χ^2 distance of histogram difference of the depth of 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 7. In general, this may be an effective approach if training data has noisy depth.

Our model is fast. Using Intel Core i7-6700 and GPU TITAN X, our model takes around 0.1 second per one image to calculate salient regions after BED features are obtained. Calculating BED features takes around 1 second per image with an unoptimized single threaded CPU implementation.

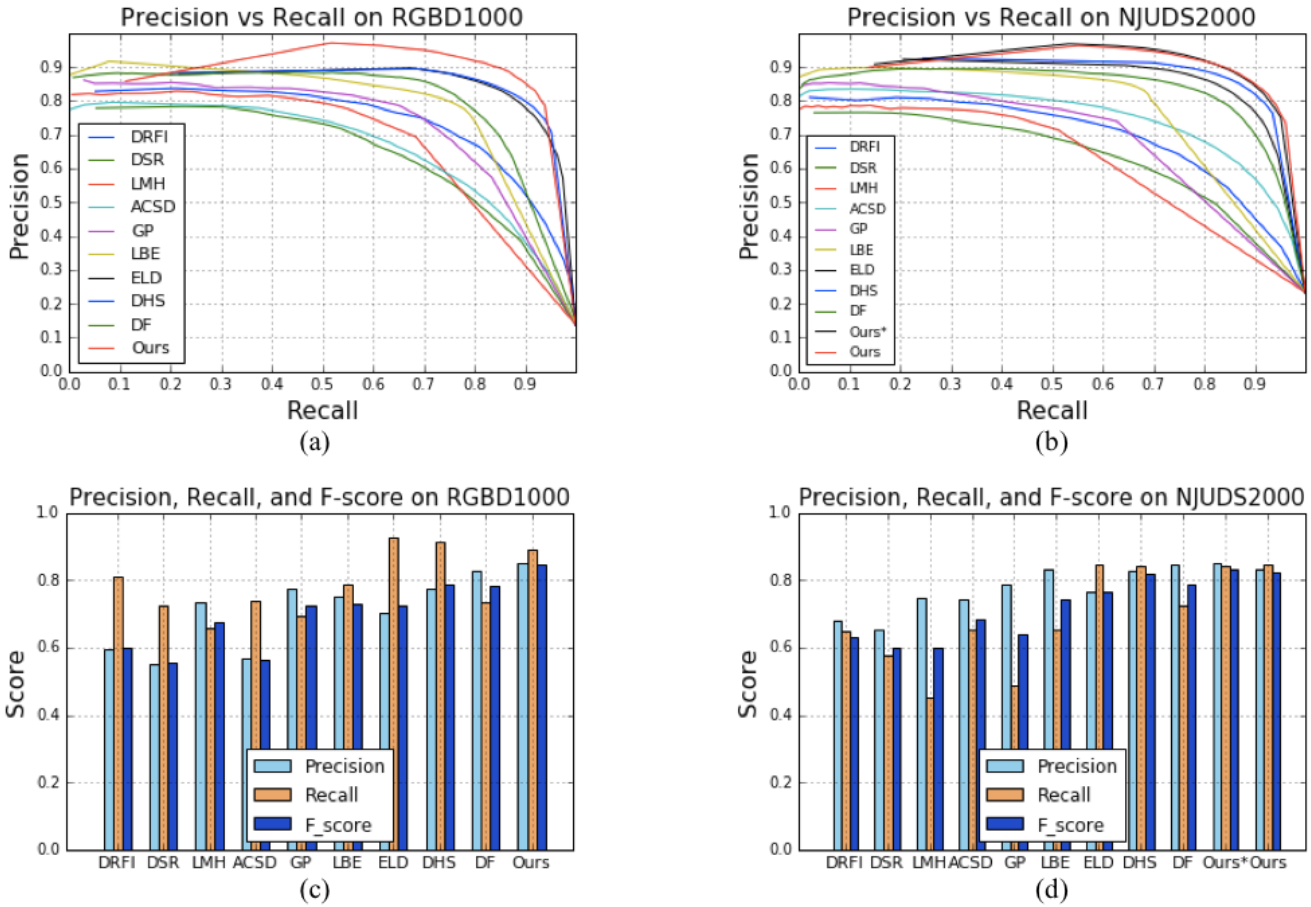

Figure 7. Comparing performance of our methods with other RGB-D saliency methods. The PR curve of our method and the other current RGB-D salient object detection methods on (a) RGBD1000 and (b) NJUDS2000. The F-score of our method and the other current methods on (c) RGBD1000 and (d) NJUDS2000.

We evaluate the contribution of the separate components of our method, the low level depth features including the novel depth histogram comparison, and the BED features. We perform training in the same architecture other than these 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 ELD-Net [15], our network using the low level depth features with ELD-Net, and our full architecture. As can be seen, the contribution of the low level depth features and BED are strong, and BED further contributes to an increase in the already high scores. On the RGBD1000 dataset, precision increases well while holding the same recall. On NJUDS2000 datasets, precision increases and recall rate also increases slightly. Figure 8 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. Human visual attention is mediated by top-down and bottom-up information, and it has been shown that depth influences attention. This paper uses a CNN to incorporate top-down and bottom-up information for detecting RGB-D salient objects. We incorporate a novel mid-level feature, BED, to capture background enclosure, as well as low level depth cues that incorporate depth contrast and depth histogram distance, and color features. Our results demonstrate that our novel 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.

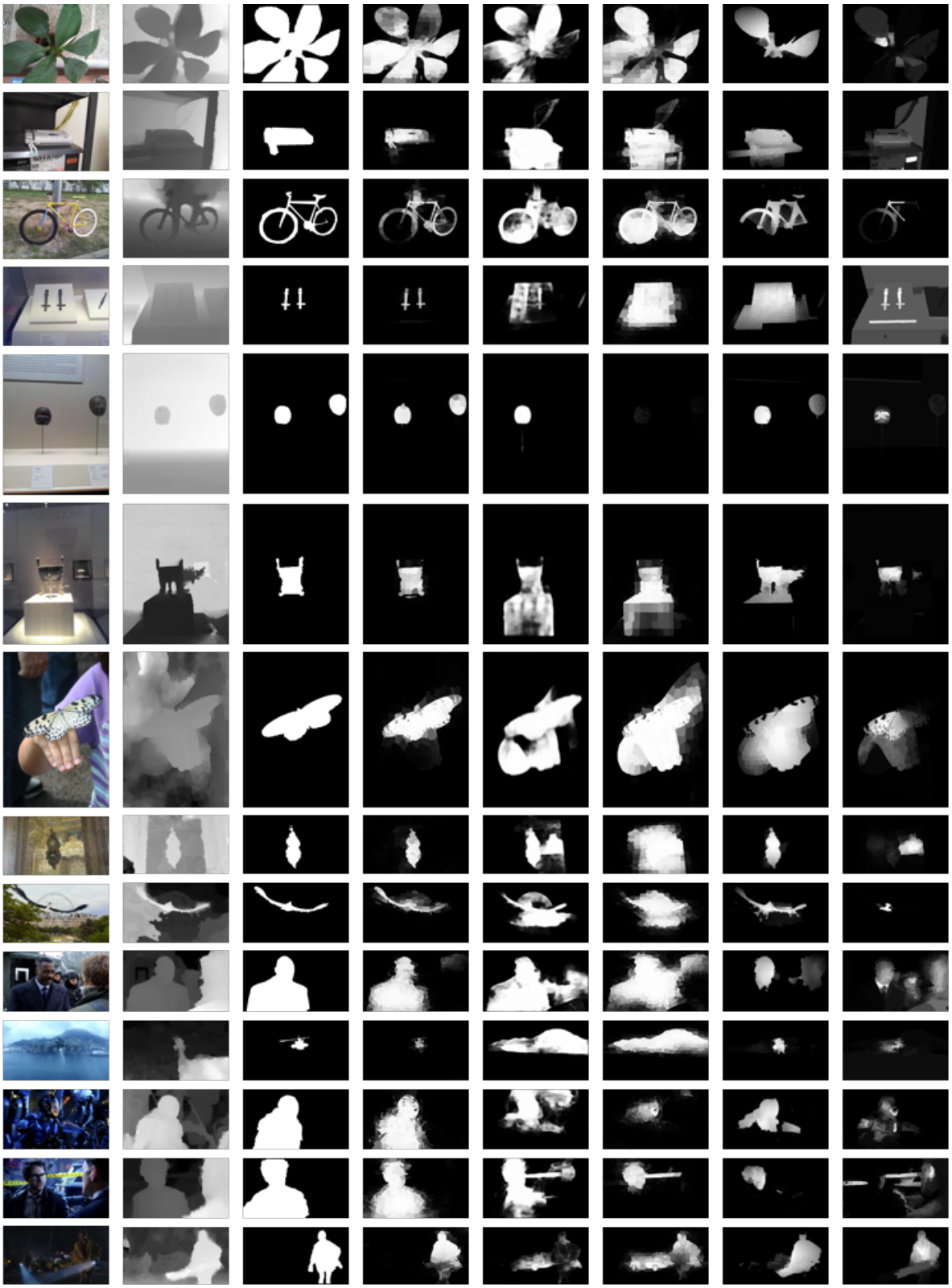

Figure 8. Comparing outputs of our architecture against DHS [18], ELD [15], LBE [6], GP [24]. Note that G.T. means Ground Truth.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. 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. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[3] 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 Recogntion*, 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, June 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. *IEEE Transactions on Image Processing*, 26(5):2274–2285, May 2017.

[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.