

Deep Saliency with Encoded Low level Distance Map and High Level Features

Gayoung Lee
KAIST

gylee1103@gmail.com

Yu-Wing Tai
SenseTime Group Limited

yuwing@gmail.com

Junmo Kim
KAIST

junmo.kim@kaist.ac.kr

Abstract

Recent advances in saliency detection have utilized deep learning to obtain high level features to detect salient regions in a scene. These advances have demonstrated superior results over previous works that utilize hand-crafted low level features for saliency detection. In this paper, we demonstrate that hand-crafted features can provide complementary information to enhance performance of saliency detection that utilizes only high level features. Our method utilizes both high level and low level features for saliency detection under a unified deep learning framework. The high level features are extracted using the VGG-net, and the low level features are compared with other parts of an image to form a low level distance map. The low level distance map is then encoded using a convolutional neural network (CNN) with multiple 1×1 convolutional and ReLU layers. We concatenate the encoded low level distance map and the high level features, and connect them to a fully connected neural network classifier to evaluate the saliency of a query region. Our experiments show that our method can further improve the performance of state-of-the-art deep learning-based saliency detection methods.

1. Introduction

Saliency detection aims to detect distinctive regions in an image that draw human attention. This topic has received a great deal of attention in computer vision and cognitive science because of its wide range of applications such as content-aware image cropping [22] and resizing [3], video summarization [24], object detection [20], and person re-identification [31]. Various papers such as DRFI [13], GMR [30], DSR [17], RBD [32], HDCT [15], HS [29] and GC [7] utilize low level features such as color, texture and location information to investigate characteristics of salient regions including objectness, boundary convexity, spatial distribution, and global contrast. The recent success of deep learning in object recognition and classification [23] brought to a revolution in computer vision. Inspired by the

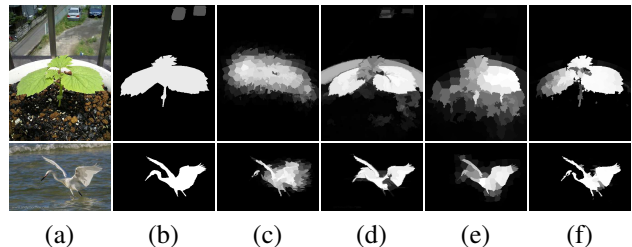


Figure 1: (a) Input images, (b) Ground truth masks, (c) Fuzzy saliency masks from VGG16 features (HF setting, described in Section 3.3), (d-f) Results of (d) MDF [16], (e) MCDL [21], and (f) our method.

human visual system, deep learning builds hierarchical layers of visual representation to extract the high level features of an image. Using extracted high level features, several recent works [27, 16, 21] have demonstrated state-of-the-art performance in saliency detection that significantly outperform previous works that utilized only low level features.

As discussed in [9], while high level features are good to evaluate objectness in an image, they are relatively weak in for determining precise localization. This is because multiple levels of convolutional and pooling layers “blur” the object boundaries, and high level features from the output of the last layer are too coarse spatially for the saliency detection task. This problem is illustrated in Figure 1(c). To generate a precise saliency mask, previous studies utilized various methods including object proposal [27] and super-pixel classification [16, 21]. Yet, it was still very hard to differentiate salient regions from their adjacent non-salient regions because their feature distances were not directly encoded.

In this paper, we introduce the encoded low level distance map (ELD-map), which directly encodes the feature distance between each pair of superpixels in an image. Our ELD-map encodes feature distance for various low level features including colors, color distributions, Gabor filter responses, and locations. Our ELD-map is unique in that it uses deep learning as an auto-encoder to encode these low level feature distances by multiple convolutional layers with 1×1 kernels. The encoded feature distance map has strong

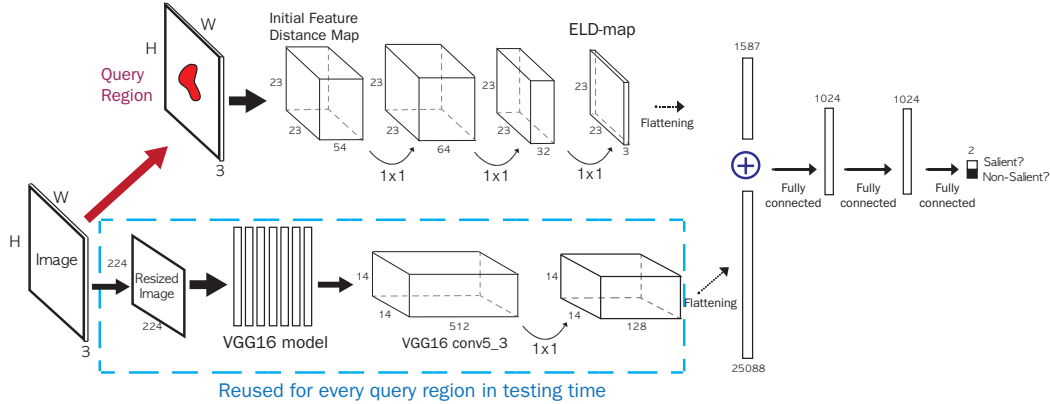


Figure 2: Overall pipeline of our method. We compute the ELD-map from the initial feature distance map for each query region and concatenate the high level feature from the output of the conv5_3 layer of the VGG16 model.

discriminative power to evaluate similarities between different parts of an image with precise boundaries among superpixels. We concatenate our ELD-map and the output of the last convolutional layer from the VGG-net (VGG16) [25] to form a new feature vector which is a composite of both high level and low level information. Using our new feature vector, we can precisely estimate saliency of superpixels. Without any post-processing, this method generates an accurate saliency map with precise boundaries.

In summary, our paper offers the following contributions:

- We introduce the ELD-map which shows that low level features can play complementary roles to assist high level features with the precise detection of salient regions.
- Compared with previous works that utilized either high level or low level features, but not both, our work demonstrates consistent improvements across different benchmark datasets.
- Because high level features can be reused for different query regions in an image, our method runs fast. The testing time in the ECSSD dataset [29] takes only around 0.5 seconds per an image.

2. Related Works

In this section, representative works in salient region detection are reviewed. We refer readers to [4] and [5] for a survey and a benchmark comparison of the state-of-the-art salient region detection algorithms.

Recent trends in salient region detection utilize learning-based approaches, which were first introduced by Liu *et al.* [19]. Liu *et al.* were also the first group to released a benchmark dataset (MSRA10K) with ground truth evaluation. Following this work, several representative benchmarks with ground truth evaluation were released. These benchmarks include ECSSD [29], Judd [14],

THUR15K [6], DUTOMRON [30], PASCAL-S [18], and FT [1]. They cover rich variety of images containing different scenes and subjects. In addition, each one exhibits different characteristics. For example, the ground truth of the MSRA10K dataset are binary mask images which were manually segmented by human, while the ground truth of the FT [1] dataset were determined by human fixation.

Discriminative Regional Feature Integration (DRFI) [13], Robust Background Detection (RBD) [32], Dense and Sparse Reconstruction (DSR) [17], Markov Chain (MC) [12], High Dimensional Color Transform (HDCT) [15], and Hierarchical Saliency (HS) [29] are the top 6 models for salient region detection reported in the benchmark paper [5]. These algorithms consider various heuristic priors such as the global contrast prior [29] and the boundary prior [13] and often generate high-dimensional features to increase discriminative power [15, 13] to distinguish salient regions from non-salient regions. These methods are all based on hand-crafted low level features without deep learning.

Deep learning has emerged in the field of saliency detection last year. Several methods that utilize deep learnings for saliency detection were simultaneously proposed. This includes Multiscale Deep Feature (MDF) [16], Multi-Context Deep Learning (MCDL) [21], and Local Estimation and Global Search (LEGS) [27]. They utilized high level features from the deep convolutional neural network (CNN) and demonstrated superior results over previous works that utilized only low level features. MDF and MCDL utilize superpixel algorithms, and query each region individually to assign saliency to superpixels. For each query region, MDF generates three input images that cover different scopes of an input image, and MCDL uses sliding windows with deep CNN to compute the deep features of the center superpixel. LEGS first generates an initial rough saliency mask from deep CNN and refines the saliency map using an object proposal algorithm.

Compared to the aforementioned methods, our work uti-

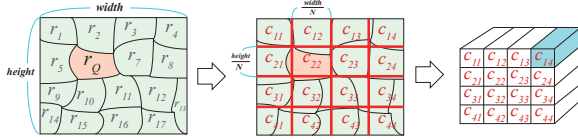


Figure 3: Visualization of the construction process for the initial low level feature distance map. Each grid cell, which represents uniformly divided area of an image, is described by the features of the superpixel that occupies the largest area of the grid cell. Using the features, we construct an $N \times N \times K$ feature distance map. The computed features and distances are summarized in Table 1 and Table 2

lizes high level and low level features simultaneously. The high level features evaluate the objectness in an image with coarse spatial location and the low level features evaluate similarities between the different superpixels in an image. Our high level and low level features are combined and evaluated by a multi-level fully connected neural network classifier, that seamlessly considers both high level and low level features to assign saliency to query superpixels. Experiments demonstrate that our method significantly outperforms previous methods that utilize either low level features or high level features, but not both.

3. Algorithms

The overall pipeline of our method is illustrated in Figure 2. First, the process for construction of the ELD-map is described. Then, we describe how the high level features were extracted and integrated with the ELD-map for salient region classification. At the end of this section, we report the results of our self evaluations to analyze the effects of the ELD-map and the high level features in our saliency detection framework.

3.1. Construction of the ELD-map

Our algorithm utilizes a superpixel-based approach for saliency detection. To segment an image into superpixels, the SLIC[2] algorithm is used. The major benefits of using the SLIC algorithm for superpixel segmentation are that the segmented superpixels are roughly regular and that it provides control on the number of superpixels.

After superpixel segmentation, the initial hand-crafted low level features of each superpixel are calculated, and the superpixel representation is converted into a regular grid representation as illustrated in Figure 3. To be more specific, we assign superpixels to grid cells according to their occupying area in each cell. This regular grid representation is efficient for CNN architecture because we can convert images with different resolutions and aspect ratios into a fixed size distance map without resizing and cropping.

In our implementation, the size of the regular grid was set to 23×23 . We index the superpixels as $S =$

Features of a superpixel ($f(r_c)$)	Feature Index
Average RGB value	1-3
Average LAB value	4-6
Average HSV value	7-9
Gabor filter response	10-33
Maximum Gabor response	34
Center location	35-36
RGB color histogram	37-61
LAB color histogram	62-86
HSV color histogram	87-110

Table 1: The list of extracted features of a superpixel.

Distance map features	$\#f(\cdot)$	Feature Index
$f(c_{ij}) - f(r_q)$	1-36	1-36
χ^2 distance($f(c_{ij}), f(r_q)$)	37-110	37-45
$f(c_{ij})$	1-9	46-54

Table 2: The list of feature distances used for computing the initial low level feature distance map. $f(r_q)$ is the extracted features of a query superpixel, r_q , and $f(c_{ij})$ is the extracted features of a grid cell c_{ij} , where $f(c_{ij}) := f(r_c^*)$. Details are described in Section 3.1.

$\{r_1, \dots, r_M\}$, and the grid cells of the regular grid as $G = \{c_{11}, c_{12}, \dots, c_{NN}\}$, $N = 23$. We denote the computed feature descriptor of each superpixel region as $f(r_c)$. The collected features for each superpixel are summarized in Table 1. Our hand-crafted features are all low level features related to colors (average colors in RGB, LAB, and HSV spaces, and their local color histograms), textures (Gabor filter responses [28] averaged over pixels in each region), and locations (center location of a superpixel). We normalize the pixel coordinates so that the range of coordinates was within $[0, 1]$ and include the maximum over 24 values for the Gabor filter response in each region. Each grid cell descriptor is equal to the descriptor of the superpixel which occupies the largest area inside that grid cell, *i.e.*, $f(c_{ij}) := f(r_c^*)$, where $r_c^* = \arg \max_{r_c} \#pixels(r_c \cap c_{ij})$.

Similar to MCDL[21] and MDF[16], we query the saliency score of each region individually. For each query region, we compute a low level feature distance map that modelled the feature distances between the queried superpixel $f(r_q)$ and grid cells $f(c_{ij})$ in the regular grid. For the mean color value and Gabor response, we simply compute the differences within them where negative values are allowed, and use the Chi-square (χ^2) distance for color histograms between r_c^* and r_q . We attach the average colors of $f(c_{ij})$ at the end of the distance measurements as a reference point, and find that this improved the performance. Table 2 summarizes the computed feature distances of the initial feature distance map where the number of the initial features (K) is 54. After computing the distances, the size of the initial feature distance map becomes $23 \times 23 \times 54$.

The initial feature distance map is then encoded to a compact but accurate feature distance map using the multiple 1×1 convolutional and ReLU layers, as illustrated in Figure 2. The multiple 1×1 convolutional and ReLU layers work as a fully connected layer across channels to find the best nonlinear combination of feature distances that better describe the similarities and dissimilarities between a query superpixel and the other regions of an image. Because the dimension of the initial map is reduced, we call this distance map as an encoded low level distance map (ELD-map). In our implementation, the size of the ELD-map was $23 \times 23 \times 3$. In the self-evaluation experiment in Table 3, we find that encoding the low level feature distance map with the deep CNN with 1×1 kernel enhances the performance of our method. The effects of the encoding will be discussed in Section 3.3.

3.2. Integration with High Level Features

We extract the high level features using the VGG16 model pretrained by the ImageNet Dataset [23]. The VGG16 [25] won the ImageNet2014 CLS-LOC task. We used the VGG16 model distributed by Caffe Model Zoo [11] without finetuning. We resize the input images to 224×224 to fit to the fixed input size of the VGG16 model and extract a “conv5_3” feature map, which is generated after passing the last convolutional layer. The extracted features has 512 channels and 14×14 resolution. To fit the features to our GPU memory, we attach an additional convolutional layer with a 1×1 kernel for feature selection and dimensionality reduction as in GoogleNet [26].

For each input image, we process it with the pre-trained deep CNN only once and reuse the extracted high level feature map for all queried regions. Therefore, our computational cost is small even when we use a very deep and powerful model such as the VGG16 model. Although other parts of our algorithm, including generating the ELD-map and applying fully-connected layers, should be repeated each time, the cost from these parts is much smaller than running the VGG16 model.

Before applying the fully-connected layers to classify the queried region, we concatenate the ELD-map and “conv5_3” feature map after flattening each map. Afterwards, two fully-connected layers with 1024 nodes generate a saliency score for the queried region using the concatenated features. We use the cross entropy loss for softmax classifier to evaluate the outputs:

$$L = - \sum_{j=0}^1 1_{(y=j)} \log\left(\frac{e^{z_j}}{e^{z_0} + e^{z_1}}\right) \quad (1)$$

where 0 and 1 denote non-salient and salient region labels respectively, and z_0 and z_1 are the score of each label of training data. Since the ELD-map features and the high

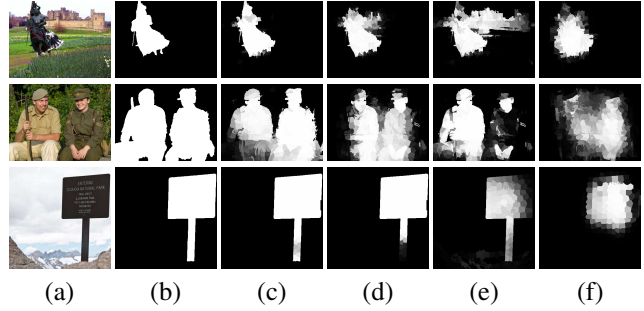


Figure 4: Visual comparisons of results in our self-evaluation experiments. (a) Input images, (b) Ground truth masks, (c-f) the results of our algorithm (c) using both ELD-map and high level features (ELD-HF) (d) using both non-encoded low level distance map and high level features (LD-HF) (e) using only encoded low level distance map (ELD) (f) using only high level features (HF). Details of each experiment are described in Table 3.

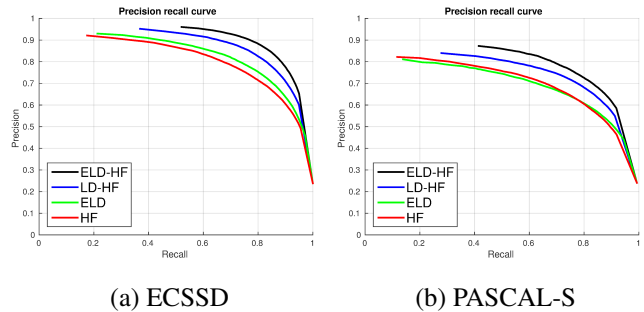


Figure 5: Precision-Recall graphs of the controlled experiments described in Table 3

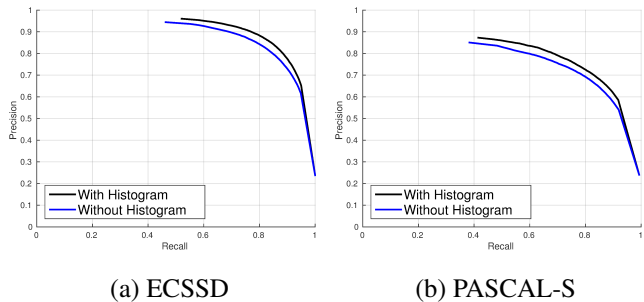


Figure 6: Precision-Recall graphs of the controlled experiments to show the effect of the statistical features.

level features are fixed in length, their spatial correlation can be learnt from training data automatically in the fully connected layers.

3.3. Analysis of the Effects of the Encoded Low level Distance map

Although theoretically neural networks can model any

Setting Description	Encoded Low level Distance map	Non-encoded Low level Distance map	High level features from VGG16	f-measure on ECSSD	f-measure on PASCAL-S
ELD-HF	Use	Not Use	Use	0.867	0.770
LD-HF	Not Use	Use	Use	0.835	0.735
ELD	Use	Not Use	Not Use	0.790	0.682
HF	Not Use	Location Only	Use	0.768	0.693

Table 3: The detail of settings of the controlled experiments. Using both ELD-map and high level features from VGG16 shows the best performance.

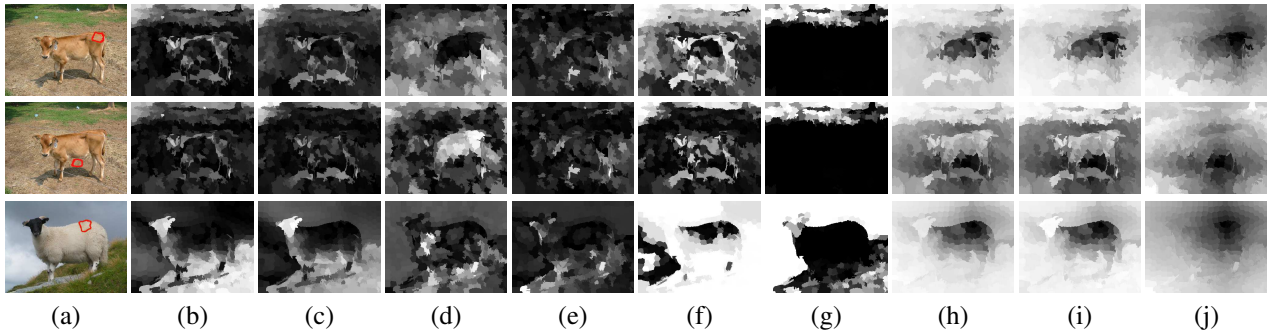


Figure 7: Comparisons of the discriminative power of different features and our ELD-map feature space. (a) Input images, the query superpixels are highlighted. (b)-(g) are the distance maps of the different features between the query superpixel and other superpixels in an image. (b)-(c) Distance maps of average color of (b) R-channel (RGB color space), and (c) L-channel (LAB color space). (d) Differences of the first Gabor filter responses. (e) Differences of the maximum gabor filter responses. (f)-(g) Chi-square distance maps of (f) L-channel histogram (LAB color space), and (g) H-channel histogram (HSV color space). (h)-(j) our Encoded Low level Distance map (ELD-map).

complex function[10], practically they may suffer from limited training data and limited computational resources. For instance, overfitting and underfitting frequently occur because of a small dataset and the complexity of desired features. It is also common for CNN to generate feature maps with much lower resolution than original input images. By providing strongly relevant information, the encoded low level distance map(ELD-map) complements the features from deep CNN and guides the classifier to learn properly. ELD-map has two main advantages: (1) it can easily generate the fine-grained dense saliency mask, and (2) it provides additional low level feature distances, which can be hard to learn for CNN, such as Chi-square distance between histograms.

We performed multiple controlled experiments to demonstrate the effects of the ELD-map in our algorithm. We conducted the experiments using four different settings: The **ELD-HF** setting uses both the ELD-map and the high level feature map from the VGG16 model. The **LD-HF** setting utilizes both the low level feature distances and the high level feature map, but does not encode the low level distances with the 1×1 convolutional network. The **ELD** setting uses only ELD-map without high level features from the VGG16 model. The **HF** setting uses the high level feature map from VGG16 model and the location distance between the query region and other regions to notify which

region is queried. We ran all models until the training data loss converged.

The results of the controlled experiments are shown in Figure 4. The model using only the high level feature map from the deep CNN detected the approximate location of the salient objects but was unable to capture detailed location because the high level feature maps had lower resolution than the resolution of the original input images. On the other hand, the model with only the low level features failed to distinguish the salient object in the first row. With both the low level feature distances and the high level feature map, the models could correctly capture salient objects and their exact boundaries. Also, we found that the ELD-map often helps to find salient objects that are difficult to detect using only CNN as shown in the second row. We speculate that the ELD-map can provide additional information which is hard to be accurately modeled by the convolutional layers. Some of the hand-crafted features of our method are statistical features, *e.g.* histogram, and we use χ^2 distance to measure the distance between histograms that would be difficult to learn by CNN. To demonstrate the effects of the statistical features, we re-train our network with the histogram features removed from our network. The comparisons are shown in Fig. 6. Clearly, the histogram features improve the performance of our work. Similarly, for features in other color space, *e.g.* LAB and HSV, it may require more layers

	ASD	PASCAL-S	ECSSD	DUT-OMRON	THUR15K
Ours	0.924	0.771	0.867	0.719	0.731
MCDL	0.928	0.737	0.837	0.703	0.686
MDF	0.931	0.759	0.831	0.694	0.670
LEGS	0.905	0.749	0.831	0.669	0.664
DRFI	0.919	0.692	0.787	0.665	0.670
DSR	0.886	0.645	0.737	0.626	0.611
GMR	0.909	0.664	0.740	0.610	0.597
HDCT	0.884	0.604	0.705	0.609	0.602
HS	0.902	0.637	0.731	0.616	0.585

Table 4: The F-measure scores of salient region detection algorithms on five popular datasets. The best score is marked in bold.

to model such transformation, but we can easily adopt them from hand-crafted features.

Table 3 summarizes the controlled experiments for the self-evaluation of our method. It also shows the quantitative comparisons in terms of f-measure on the ECSSD and the PASCAL-S datasets. The corresponding quantitative comparisons in terms of the Precision-Recall graphs are presented in Figure 5. The model utilizing both ELD-map and high level features exhibits the best performance. By comparing ELD-HF and LD-HF settings, we found that it is useful to apply 1×1 kernels among the low level features.

Figure 7 shows the initial hand-crafted distance features and ELD-map. For the ELD-map, which is originally the 23×23 size grid, we visualized each superpixel using the feature value of the closest grid cell according to the location of the center pixel. Each hand-crafted feature has its own weakness but it captures different aspects of similarities or dissimilarities between superpixels. Our 1×1 kernels work as fully-connected layers among low level feature distances and generate a powerful feature distance map by combining all of the original feature distances non-linearly. This nonlinear mapping is data-driven which is directly learnt from training data automatically. We can see the strong discriminative power of feature distances in ELD-map. While the third channel (j) is related to the position of the query region, the other two channels (h-i) seem to indicate the differences of appearance such as color and texture. Therefore, the ELD-map helps to group regions that belong to the same object, because regions which have the similar color and texture have similar values in the two channels of the ELD-map regardless of their position.

4. Experiment and Discussion

We evaluated the performance of our algorithm using various datasets. The **MSRA10K** [19] is a dataset with 10,000 images which includes the **ASD** dataset [1]. Most images in this dataset contains single object. The **PASCAL-S** [18] is generated from the PASCAL VOC dataset [8] and contains 850 natural images. The **ECSSD** [29] contains 1,000 images which have semantic meaning in their ground

	ASD	PASCAL-S	ECSSD	DUT-OMRON	THUR15K
Ours	0.035	0.121	0.080	0.091	0.095
MCDL	0.035	0.142	0.102	0.089	0.102
MDF	0.051	0.142	0.108	0.092	0.127
LEGS	0.063	0.155	0.119	0.133	0.125
DRFI	0.085	0.196	0.166	0.155	0.150
DSR	0.080	0.205	0.173	0.139	0.142
GMR	0.075	0.217	0.189	0.189	0.181
HDCT	0.119	0.229	0.199	0.164	0.177
HS	0.111	0.262	0.228	0.227	0.218

Table 5: The Mean Absolute Error(MAE) of salient region detection algorithms on five popular datasets. The best score is marked in bold.

truth segmentation. It also contains images with complex structures. The **DUT-OMRON** [30] has 5,168 high quality images and the **THUR15K** [6] contains 6,232 images of specific classes.

We trained our model using 9,000 images from the **MSRA10K** dataset after excluding the same images in **ASD** dataset. We did not use validation set and trained the model until its training data loss converges. From each image, we use about 30 salient superpixels and 70 non-salient superpixels; around 0.9 million input data are generated. The layers of VGG16 model are fixed by setting the learning rate equal to zero. For other layers, we initialize the weights by the ‘‘xavier’’ (caffe parameter), and we set the base learning rate equal to 0.001. We use stochastic gradient descent method with momentum 0.9 and decrease running rate 90% when training loss does not decrease. Training our model takes 3 hours for 100,000 iterations with mini-batch size 128.

Our results were compared with MCDL [21], MDF [16], LEGS [27], DRFI [13], DSR [17], GMR [30], HDCT [15], and HS [29], which are the state-of-the-art algorithms. DRFI, DSR, GMR, HDCT and HS use low level features and MCDL, MDF and LEGS utilize deep CNN for high level context. We obtained the result images from the project site of each algorithm or the benchmark evaluation [5]. The results which were not provided were generated from the authors’ source codes published in the web. The comparisons on Precision-Recall(PR) graph and Mean Absolute Error(MAE) graph are presented in Figure 8. Maximum F-measure scores and MAE values are also described in Table 4 and Table 5. We used the evaluation codes used in the benchmark paper [5]. The PR graph and f-measure score tend to be more informative than ROC curve because salient pixels are usually less than non-salient [5]. Following the criteria by Achanta et. al. [1], we moved the threshold from 0 to 255 to generate binary masks(M). Using the ground truth(G), the precision and recall is calculated as follows:

$$Precision = \frac{|M \cap G|}{|M|}, \quad Recall = \frac{|M \cap G|}{|G|} \quad (2)$$

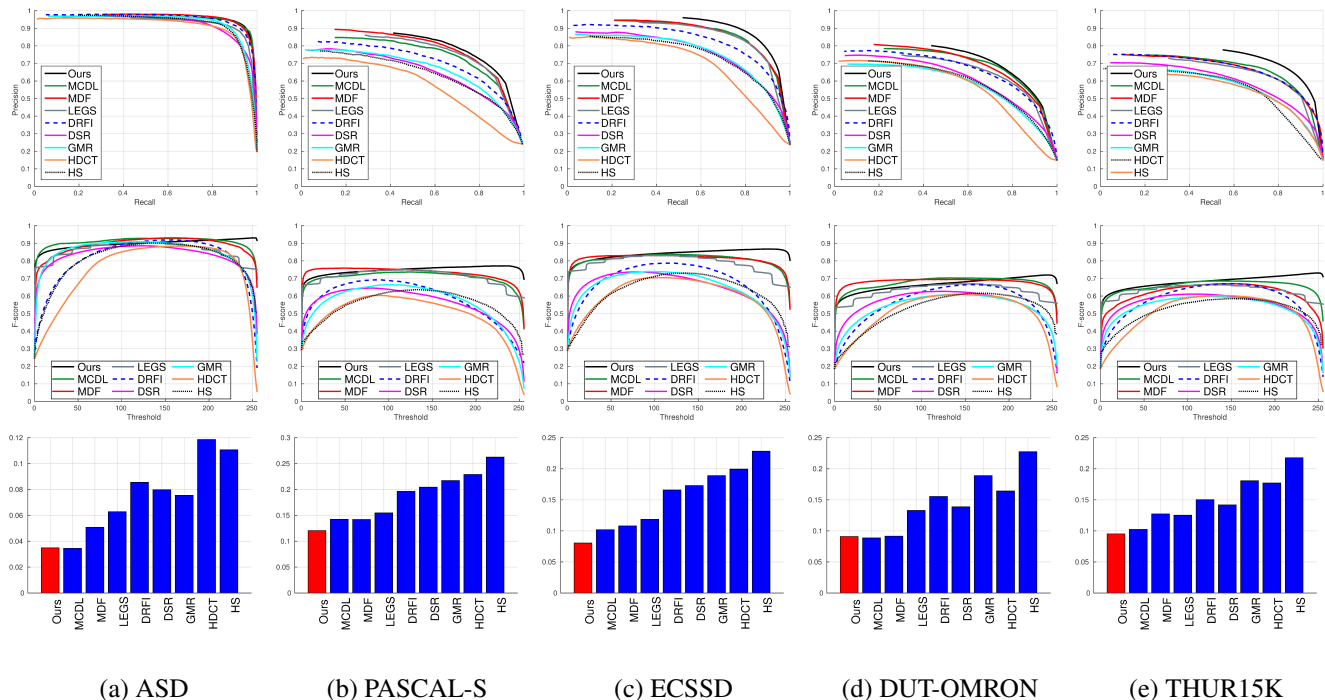


Figure 8: From top to bottom, Precision-Recall (PR) graph, F-measure score with different thresholds and Mean Absolute Error (MAE) of various algorithms on five popular datasets.

We also reported the F-Measure score which is a balanced measurement between precision and recall as follows:

$$F_{\beta} = \frac{(1 + \beta^2) Precision \times Recall}{\beta^2 \times Precision + Recall} \quad (3)$$

where β^2 is typically set to 0.3. We visualized f-measure score for the different thresholds and reported the maximum f-measure score which well describes the overall detection performance [5]. In our algorithm, making binary masks using the high threshold around 240 generated good f-measure score.

The overlapping-based evaluations give higher score to methods which assign high saliency score to salient pixel correctly. However, the evaluation on non-salient regions can be unfair especially for the methods which successfully detect non-salient regions, but missed the detection of salient regions [5]. Therefore, we also calculated the mean absolute error(MAE) for fair comparisons as suggested by [5]. The MAE evaluates the detection accuracy as follow:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |S(x, y) - G(x, y)| \quad (4)$$

where W and H are width and height of an image, S is

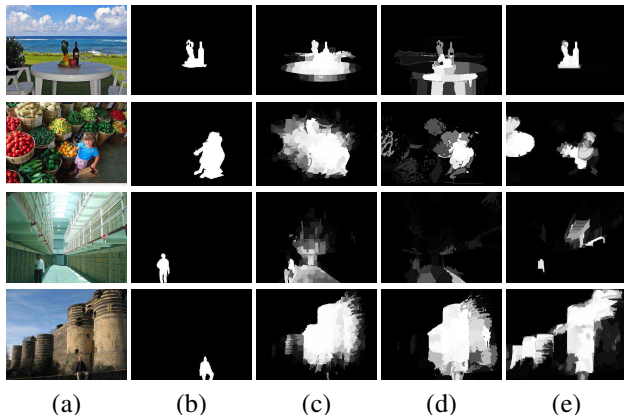
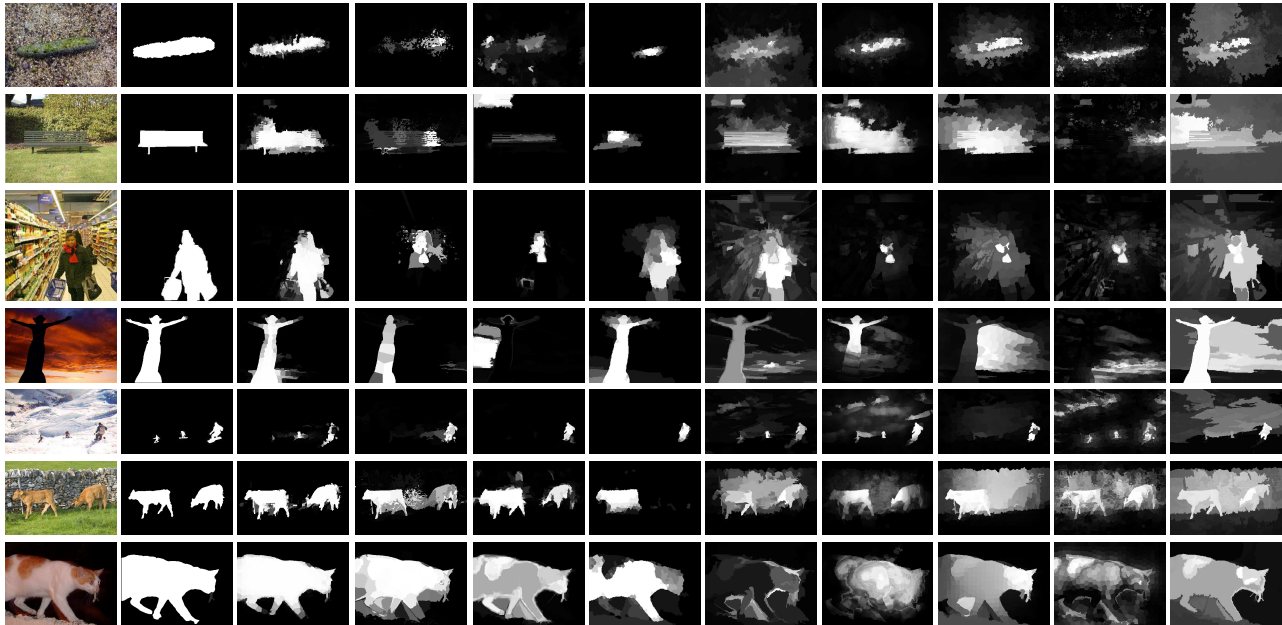


Figure 10: Failure cases of our algorithm. (a) Input images, (b) Ground truths, Results of (c) our method, (d) MCDL [21], (e) MDF [16].

the estimated saliency map and G is the ground truth binary mask.

In Figure 8, the PR-graph indicates our algorithm achieves the better performance than the previous works including MDF and MCDL which also utilize CNN models. Our algorithm shows the lowest MAE and the highest maximum F-measure score on most of the datasets. Visual comparisons of various methods are shown in Figure 9.



(a) Inputs (b) GT (c) Ours (d) MCDL (e) MDF (f) LEGS (g) DRFI (h) DSR (i) GMR (j) HDCT (k) HS
 Figure 9: Visual comparisons of our results and the state-of-the-art methods on difficult scenes. (a) original image, (b) ground truth, (c) Ours (d) MCDL [21] (e) MDF [16] (f) LEGS [27] (g) DRFI [13], (h) DSR [17], (i) GMR [30], (j) HDCT [15] (k) HS [29]. From the top to the bottom, row 1-2 are the images with a low-contrast salient object, row 3-4 are with complicated background, row 5-6 are with multiple salient objects and row 7 is with a salient object touching the image boundaries.

We visualize the results from various difficult cases including low-contrast objects (row 1-3), complicate backgrounds (row 4-6), small salient objects (row 7-8), multiple salient objects (row 9-10) and touching boundary examples (row 11-12). Our algorithm shows especially good performance on images with low-contrast salient objects and complicated backgrounds, and also works well on other difficult scenes.

In Figure 10, we reported some failure cases. The first and the second results contain correct salient objects but also highlight non-salient regions. The third and fourth examples have the extremely difficult scenes with a small, low-contrast and boundary touching the salient object. Because these kinds of data are not provided much by the training data, MSRA10K, we may further improve the performance with richer training data. For these difficult scenes, MCDL [21] and MDF [16] also fail to find the salient objects precisely.

The running time of our algorithm was measured from the ECSSD dataset, where tested images were of size 400×300 . We used a server machine with intel i7 CPU, 8GB RAM and GTX Titan-Black for testing. Our model, developed by C++ and based on Caffe [11] library, took around 0.5 seconds per image. The training of our deep CNN took around 3 hours under the same environment. The short training time and testing time is also an advantage of our method. This is due to the sharing of our high level features which only need to be computed once for a whole image.

5. Conclusion

In this paper, we have introduced a new method to integrate the low-level and the high-level features for saliency detection. The Encoded Low-level Distance map (ELD-map) has stronger discriminative power than the original low-level feature distances to measure similarities or dissimilarities among superpixels. When concatenated with the high-level features from the deep CNN model (VGG16), our method shows the state-of-the-art performance in terms of both visual qualities and quantitative comparisons. As a future work, we are planning to explore more various CNN architectures to further improve the performance of our work.

Acknowledgement

This work was partially supported by HRHRP(High Risk High Return Project of KAIST) and the MOTIE(The Ministry of Trade, industry & Energy), Korea, under the Technology Innovation Program supervised by KEIT(Korea Evaluation Institute of Industrial Technology), 10045252, Development of robot task intelligence technology. Furthermore, this research was also supported by the National Research Foundation of Korea (NRF) under Grant NRF-2014R1A2A2A01003140.

References

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 6
- [2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 34(11):2274–2282, 2012. 3
- [3] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. 26(3):10, 2007. 1
- [4] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *ArXiv e-prints*, 2014. 2
- [5] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *ArXiv e-prints*, 2015. 2, 6, 7
- [6] M.-M. Cheng, N. Mitra, X. Huang, and S.-M. Hu. Salientshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014. 2, 6
- [7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 37(3):569–582, 2015. 1
- [8] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2014. 6
- [9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1
- [10] K. Hornik, M. Stinchcombe, and H. White. Multilayer feed-forward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 5
- [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. 2014. 4, 8
- [12] B. Jiang, L. Zhang, H. Lu, C. Yang, and M. Yang. Saliency detection via dense and sparse reconstruction. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2013. 2
- [13] H. Jiang, J. Wang, Z. Yuan, N. Z. Y. Wu, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 6, 8
- [14] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2009. 2
- [15] J. Kim, D. Han, Y. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2, 6, 8
- [16] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 6, 7, 8
- [17] X. Li, H. Lu, L. Zhang, X. Ruan, and M. Yang. Saliency detection via dense and sparse reconstruction. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2013. 1, 2, 6, 8
- [18] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 280–287. IEEE, 2014. 2, 6
- [19] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2007. 2, 6
- [20] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 899–906. IEEE, 2014. 1
- [21] W. O. R. Zhao, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 6, 7, 8
- [22] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. Autocol-lage. 25(3):847–852, 2006. 1
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015. 1, 4
- [24] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 1
- [25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 4
- [26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 4
- [27] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 3183–3192, 2015. 1, 2, 6, 8
- [28] T. P. Weldon, W. E. Higgins, and D. F. Dunn. Efficient gabor filter design for texture segmentation. *Pattern Recognition*, 29(12):2005–2015, 1996. 3
- [29] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 6, 8
- [30] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang. Saliency detection via graph-based manifold ranking. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 6, 8
- [31] R. Zhao, W. Ouyang, and X. Wang. Unsupervised saliency learning for person re-identification. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 3586–3593. IEEE, 2013. 1
- [32] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2