Deep Level Sets for Salient Object Detection

Ping Hu[†] phu005@ntu.edu.sg
Bing Shuai[†] bshuai001@ntu.edu.sg
Jun Liu[†] jliu029@ntu.edu.sg
Gang Wang[‡] wanggang@ntu.edu.sg

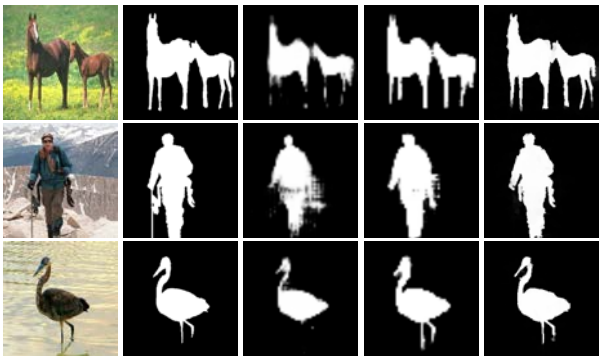
[†]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[‡]Alibaba Group, Hangzhou, China

Abstract

Deep learning has been applied to saliency detection in recent years. The superior performance has proved that deep networks can model the semantic properties of salient objects. Yet it is difficult for a deep network to discriminate pixels belonging to similar receptive fields around the object boundaries, thus deep networks may output maps with blurred saliency and inaccurate boundaries. To tackle such an issue, in this work, we propose a deep Level Set network to produce compact and uniform saliency maps. Our method drives the network to learn a Level Set function for salient objects so it can output more accurate boundaries and compact saliency. Besides, to propagate saliency information among pixels and recover full resolution saliency map, we extend a superpixel-based guided filter to be a layer in the network. The proposed network has a simple structure and is trained end-to-end. During testing, the network can produce saliency maps by efficiently feedforwarding testing images at a speed over 12FPS on GPUs. Evaluations on benchmark datasets show that the proposed method achieves state-of-the-art performance.

1. Introduction

With limited computational resource, the human vision system can effectively select important information from complex visual inputs for further processing. Inspired by this biological capability, visual saliency computation is introduced into the field of computer vision for its potential to enhance tasks like image processing [4, 12], image understanding [58, 63], video analysis and compression [13, 45]. Over the past years, many saliency detecting methods have been proposed. Early works focused on fixation-level saliency detection [19, 14, 17] that aims to predict human's attentional priority when viewing an image. Later, it was extended to object-level saliency detection [54, 55, 12, 64, 10, 34, 60] which targets at computing saliency maps to highlight the regions of salient objects accurately.



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

Figure 1. Examples of pixel-wise saliency prediction. (a) Input. (b) Groundtruth. (c) Saliency maps by deep network trained with Binary Cross Entropy (BCE) loss. (d) Saliency maps by deep network trained with Level Set method. (e) Final results by deep level set network inserted with a guided superpixel filtering layer.

Based on different mechanisms, these methods can be roughly divided into two classes: bottom-up methods which are stimulus-driven and top-down approaches that are task driven. Bottom-up methods use low-level features and cues, like contrast [10, 40], spatial property [12, 54], spectral information [17, 42], objectness [59, 22] etc. Because of the unawareness of image content, purely low-level cues are difficult to detect salient objects in complex scenes. Different from bottom-up methods, top-down approaches [12, 34, 20] incorporate high-level visual knowledge into detection. For this kind of models, it is critical to effectively learn the semantic relationship between salient objects and background from data. Recently, Convolutional Neural Networks(CNN) have shown superior performance in many vision tasks due to its superior ability to extract high-level and multi-scale features. In saliency detection, several recent works using CNNs [62, 29, 24, 33, 26, 39, 30, 51, 46] have significantly outperformed previous methods which only use low-level cues and hand-crafted features.

However, when applying CNNs to pixel-wise saliency

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.65

labeling, it may suffer from some limitations. It is difficult for a network to learn saliency at boundaries of salient regions. This is because pixels around the boundaries are centered at similar receptive fields, while the network is trained to discriminate binary labels. The network may produce maps with inaccurate boundaries and shape. Also as discussed in [5], for dense pixel-labeling tasks, the training of networks is always based on the assumption that pixels are independent. However, treating them independently may be not optimal because it fails to exploit the correlation between pixels. Although superpixels or object proposals can be utilized to refine the coarse result [62, 29, 26, 30, 31], a more accurate coarse map can further improve the result.

In this paper, to relive these limitations, we propose an end-to-end deep Level Set network for salient object detection. Level Set [38] method is widely used in image binary segmentation task [8, 28]. The value of level set function for a pixel is the signed distance between the pixel and the segmentation boundary. The signs indicate segmentation labels. With initial values, the level sets are iteratively updated to optimize an energy function that forces the boundary to evolve to accurately segment foreground from the background. When applying it to salient object detection, which is also a binary segmentation task, our target is to generate a level set function with an interface that accurately separates salient objects from the background. The signed distances for pixels inside and outside the interface should be positive and negative respectively, and their absolute values are allowed to gradually increase as pixels' positions gradually moving away from object contours. With the signed distance, the final saliency label can be generated easily by the Heaviside transformation that projects negative numbers to 0 and positive numbers to 1. Instead of directly learning a binary label for each pixel independently, our network is trained to learn the level sets for salient objects. There are at least two folds of advantages for combining level sets with deep networks: (i) The level set function can express segmentation labels by the signs. At the same time, the absolute values are allowed to change gradually so that it can help deep network to model the gradual change and correlation of pixels. This helps the network to learn the gradual change around boundaries more naturally and easily. (ii) With level set formulation, shape and area can be implicitly represented in the energy function, so the network can be aware of the salient object as a whole instead of learning saliency for every pixel independently. As shown in Fig. 1(c) and Fig. 1(d), a VGG16-based network trained with Level Set function can discriminate pixels around object boundaries more precisely and generate saliency maps more compact and accurate than network directly trained with binary groundtruth. To further refine the result, an extended guided filter [15] for superpixels is inserted into the network as a layer. With the guided super-pixel filters, saliency can be propagated among pixels. Finally, we combine this module with the VGG16-based network and train the network end-to-end. As shown in Fig. 1(e), the proposed network can produce saliency maps that are compact, uniform and accurate. In summary, this work has the following three contributions:

- We use Level Set formulation to help deep networks learn information about salient objects more easily and naturally. The trained network can detect salient objects precisely and output salient maps that are compact and uniform.
- We extended the guided filter to incorporate superpixels information and use it as a layer in the end-to-end network. This filtering module can further refine the saliency map.
- The proposed network can efficiently detect salient objects by performing a single feed-forward pass. It achieves state-of-the-art performance on benchmark datasets at a speed over 12FPS with a modern GPU.

2. Related Work

2.1. Level Set Segmentation

The Level Set method (LSM) [38] is widely applied in image segmentation with active contour [37], due to its ability to automatically handle various topological changes. The basic idea is to define an implicit function in a higher dimension to represent contours as the zero level set. The function is referred as Level Set Function and evolved according to a partial differential equation (PDE) derived from a Lagrangian formulation of active contour model [6, 35]. However, early PDE driven level set methods utilize edge information and are usually sensitive to noises. To solve this problem, the Variational Level Set Methods [8, 49, 28] are proposed to derive the evolutional PDE directly from a certain energy function. With new kind of methods, additional information like region [8, 27], shape [49, 52] can be conveniently and naturally formulated in the level set domain. Level-Set based segmentation problem can be solved by iteratively applying gradient descent to minimize the evolution energy. Although pointed out by Chan $et\ al$ [7] that variational level set segmentation problem is non-convex, a well-initialized level sets and momentum-based learning strategies used in training neural network can help to achieve an optimal result [3]. These properties of variational level set methods make it suitable to be combined with deep networks to solve binary segmentation problem.

2.2. Salient Object Detection

Early approaches treat saliency detection as an unsupervised problem and focus on low-level features and cues.

The most widely used one is $contrast$ prior, which believes that salient regions present high contrast over background in certain context [14, 12, 34, 21, 40, 17]. Cheng $et\ al$ [10] compute saliency of objects based on color uniqueness and spatial distribution. Zhang $et\ al$ [59] and Jiang $et\ al$ [22] detect salient objects from the perspective of objects' uniqueness and surroundness. Shen $et\ al$ [43] assume that background can be represented by a low-rank matrix, and salient objects are the sparse noise. Another useful assumption called $center\ bias$ prior assumes that salient objects tend to be located at the center of images [54, 64, 59, 56, 53]. Assuming image boundaries belong to the background, Zhang $et\ al$ [60] and Tu $et\ al$ [48] compute the difference between pixels and image boundaries to be saliency and achieve real-time testing speed. These methods tend to fail at complex situations because they don't aware the image content or can't effectively learn the interaction between salient objects and background.

2.3. Deep Saliency Networks

Recently, deep learning has achieved human-competitive performance in many computer vision tasks [18, 16]. Instead of defining hand-craft features, deep networks are able to extract semantic features of different levels. With this advantage, deep learning has been applied in saliency detection and achieved state-of-the-art performance. Methods proposed in [33, 30, 24] merge low-level, mid-level and high-level features learned by the VGG16 net [44] together to hierarchically detect salient objects. Wang $et\ al$ [51] utilize a recurrent fully convolutional network to incorporate saliency prior knowledge for more accurate saliency. Kuen $et\ al$ [25] utilize a convolutional network to generate a coarse map and then refine it with an attentional recurrent network. Instead of predicting saliency for every single pixel, superpixel and object region proposal are also combined with deep network [46, 26, 29, 62, 31, 50] to achieve accurate segmentation of salient object. To extract information from superpixels or object region proposals, these methods always add a new network into the model. However, this increases the size of the model and decreases the efficiency at testing phase. Different from these methods, our network achieves this via the guided superpixel filtering layer. This layer can efficiently perform feed-forward(or back-propagation) to further propagate saliency(or differential error) among pixels. Experiments on benchmarks show that our end-to-end deep level set network runs fast at testing and achieves state-of-the-art performance.

3. Proposed Method

The pipeline of the proposed network is illustrated in Fig. 3. In this section, we first introduce the proposed deep level set network in detail. Then, we describe how to extend guided filters as a layer in the network to process superpixels. At the end, we present implementation details of the proposed network.

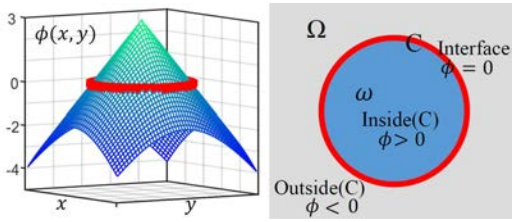


Figure 2. An example of level set segmentation. Left: visualization of a level set function $\phi(x, y)$ on the 2D space Ω. Right: respective segmentation in Ω. The zero level set and corresponding segmentation boundary C are marked in red.

3.1. Deep Level Sets for Salient Object Detection

3.1.1 Formulation of Level Sets

When applying level set methods [38, 61] for binary segmentation in 2D space Ω, the interface $C \subset \Omega$ is defined as the boundary of an open subset $\omega \subset \Omega$, and thus $C = \partial\omega$. The interface curve can be represented by the zero level set of a Lipschitz function: $\phi : \Omega \to \mathbb{R}$ that

$$\begin{cases} C = \{(x, y) \in \Omega : \phi(x, y) = 0\}, \\ inside(C) = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ outside(C) = \{(x, y) \in \Omega : \phi(x, y) < 0\}, \end{cases} \tag{1}$$

$inside(C)$ denotes the region ω, and $outside(C)$ denote the region outside ω in Ω. An example is shown in Fig. 2.

With level sets ϕ, the length of C is represented as,

$$\begin{aligned} Length\{C\} &= \int_{\Omega} |\nabla H(\phi(x, y))| dxdy \\ &= \int_{\Omega} \delta(\phi(x, y))|\nabla\phi(x, y)| dxdy \end{aligned} \tag{2}$$

Where(x, y) is the coordinate, $H(z)$ is the Heaviside function, and $\delta(z)$ is the Dirac delta function.

$$\begin{aligned} H(z) &= \begin{cases} 1, z \geq 0 \\ 0, z < 0 \end{cases}, \\ \delta(z) &= \frac{d}{dz} H(z) \end{aligned} \tag{3}$$

3.1.2 Level Sets for Deep Saliency

Traditionally, in level set methods for image segmentation [8, 28], an initial level sets ϕ_0 and an image are given as input. Then, gradient descent is applied to minimize an energy function and update the value of level set function ϕ. The energy function is always defined based on the difference of image features, such as color and texture, between foreground and background. With level sets, information

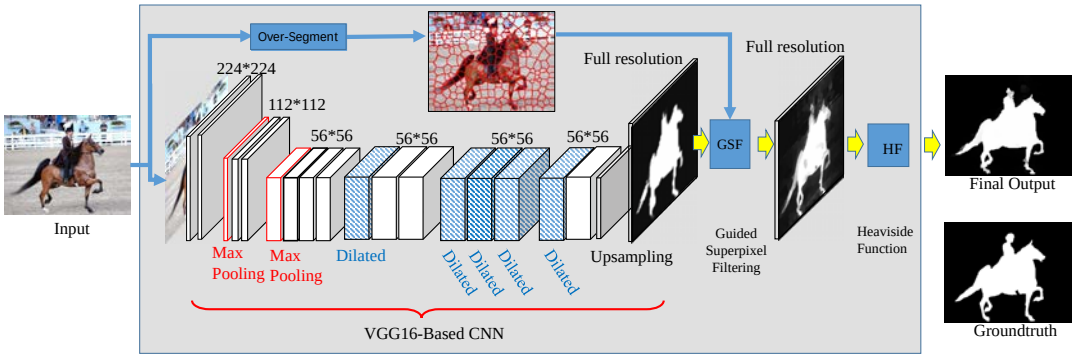


Figure 3. The architecture of the deep level set network. The CNN is built on the VGG16 net and generates coarse saliency level set maps at a resolution of 56*56. At the end of the CNN, an up-sampling layer is added to scale saliency level set map into full resolution. A Guided Superpixel Filtering(GSF) layer is followed and takes the scaled saliency level set map and superpixels as inputs. At last, the output of GSF layer was transformed by a Heaviside Function(HF) to be the final saliency map. The network can be trained end-to-end.

such as shape and regions can be integrated to enhance the performance. However, it is difficult to measure the difference for complex images like scene images using low-level features. This limited the application of level set segmentation.

Deep networks have superior ability to learn and encode useful high-level features. This makes it possible to apply level set to deal with complex scene images based on deep networks. The gradient descent based solution for level set segmentation also means that it can be combined with deep network seamlessly. With these in mind, we combine the level set method with deep networks to detect salient objects.

As shown in Fig. 3, we build a CNN based on VGG16 net and replaced the last three Max-pooling layers with dilated convolutional layers [57]. The last fully connected layer is changed into convolutional layers and a Sigmoid layer so that the network takes RGB images of 224*224 as input and produce maps of 56*56. And at the end, an Up-sampling layer without learnable parameters is added to scale the map to full resolution.

To incorporate the deep convolutional network with the level set method, we linearly shift the saliency values output by the CNN into [-0.5,0.5] and treat it as the level set ϕ. Pixel space of the input image is referred as Ω. C is the segmentation boundary with $\phi = 0$, and $H(\phi)$ is the final saliency value. To produce saliency maps that are compact and accurate, we train the network to learn a level sets ϕ that minimizes the following energy function,

$$\begin{aligned} L = \alpha \int_{\Omega} |H(\phi(x,y)) - gt(x,y)|^2 dxdy + \gamma\, Length(C) \\ + \lambda \Big[\int_{\Omega} |H(\phi(x,y)) - c_1|^2 H(\phi(x,y)) dxdy \\ + \int_{\Omega} |H(\phi(x,y)) - c_2|^2 (1 - H(\phi(x,y))) dxdy \Big] \end{aligned} \tag{4}$$

In the first term, $gt(x,y)$ is the groundtruth value of pixel at (x,y). Minimizing this term with $\alpha > 0$ supervises the network to learn what is saliency in images. The second term that is defined in Eq. 2 constraints on the length of the segmentation boundary. Different from traditional level set segmentation methods [8] that prefer minimal contours, we set $\gamma < 0$ that drives the level sets to have longer segmentation interfaces, so that it can express more details about shapes of salient objects. The last two terms with $\lambda > 0$ force the salient map to be uniform both inside and outside salient regions. Constants c_1 and c_2 play as average saliency values for $inside(C)$ and $outside(C)$ respectively. Keeping ϕ fixed and minimizing the energy function with respect to c_1 and c_2, these two constants can be expressed as,

$$\begin{aligned} c_1 &= \frac{\int_{\Omega} H(\phi(x,y)) H(\phi(x,y)) dxdy}{\int_{\Omega} H(\phi(x,y)) dxdy} \\ c_2 &= \frac{\int_{\Omega} H(\phi(x,y))(1 - H(\phi(x,y))) dxdy}{\int_{\Omega} (1 - H(\phi(x,y))) dxdy} \end{aligned} \tag{5}$$

It is easy to optimize the energy function defined in Eq. 4 with deep networks jointly. By calculus of variations [11], the derivative of energy function L upon ϕ can be written as,

$$\begin{aligned} \frac{\partial L}{\partial \phi} = \delta(\phi) \Big[2\alpha(H(\phi) - gt) - \gamma\, \mathrm{div}(\frac{\nabla \phi}{|\nabla \phi|}) \\ \lambda(H(\phi) - c_1)^2 + 2\lambda(H(\phi) - c_1) H(\phi) \\ - \lambda(H(\phi) - c_2)^2 + 2\lambda(H(\phi) - c_2)(1 - H(\phi)) \Big] \end{aligned} \tag{6}$$

As suggested in [8], this kind of energy functions is non-convex. If we use the simple Heaviside function H as in Eq. 3 which only acts on zero level set, we may get stuck in the local minima. To tackle this, we adopt the Approximated Heaviside Function(AHF) proposed in [8] that acts on all

level curves and tends to find a global minimizer. That is,

$$H_\varepsilon(z) = \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{z}{\varepsilon})),$$
$$\delta_\varepsilon = \frac{\partial H_\varepsilon(z)}{\partial z} = \frac{1}{\pi}\cdot\frac{\varepsilon}{\varepsilon^2 + z^2} \quad (7)$$

With AHF above, the deep level set network can back-propagate the error differential to previous layers to update the weights to minimize the energy. In practice, we set $\alpha = 0.75$, $\gamma = -0.005$, and $\lambda = 0.2$. There is also another parameter ε which controls the range of support for δ_ε. We will analyze these parameters in part of experiments.

3.2. Guided Super-pixel Filtering

Superpixels are regions of pixels that decompose an image into a simpler and compact representation and keep it semantic information intact at the same time, so it is applied to help to refine the segmentation result [46, 26, 29, 62, 31]. However, previous methods always adopt a framework that processes superpixels one by one. This may be inefficient due to the repetitive computation. In this section, to avoid dealing with single superpixel, we extend the guided filter to effectively and efficiently utilize superpixels to propagate saliency information locally and recover full resolution saliency map. Due to its mathematic property, the derivative of output upon input can be computed easily. Therefore, this filtering module can be added to the network as a layer (Fig. 3) and the whole network can be jointly optimized.

Guided image filter [15] is an explicit image filtering algorithm with a running time of $O(N)$. As shown in [15], guided filter is an edge-preserving and gradient-preserving filter, so that it can help us utilize the object boundary in the guidance image to further detect saliency within objects and suppress saliency outside the objects.

The original guided filter performs on a square pixel grid. It involves a guidance image I, an input map p, and the output map q. The filtering process can be expressed as a weighted average process,

$$q_i = \sum_j W_{ij}(I)p_j \quad (8)$$

and the weight is,

$$W_{ij}(I) = \frac{1}{|\omega|^2}\sum_{k:(i,j)\in w_k}(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}) \quad (9)$$

where q_i is the output at pixel i, ω_k is a window centered at pixel k, μ_k and σ_k^2 are the mean and variance of I in ω_k.

To further reduce the computational cost and achieve more accurate and uniform result, we extend the guided filter to deal with superpixels. At first, we oversegement an image into superpixels. These superpixels act as nodes and are connected to their neighbors to form an undirected graph. Average color and average ϕ are computed within every node. Then we perform filtering on this graph for each node. When computing the weight $W_{ij}(I)$ in Eq. 8, we encounter the problem. In Eq. 9, the computation of the weight is based on square windows ω_k centered at pixel k, however, the graph formed by superpixel is not in the form of grids. To solve this, we use a subgraph formed by nodes that are no more than D steps far away from node k to play as the window ω_k (as shown in Fig. 4). This can be implemented by Breath-First-Search(BFS) algorithm on graph efficiently. The radius D plays as the size of the window. An example of a processed saliency map is shown in Fig. 5.

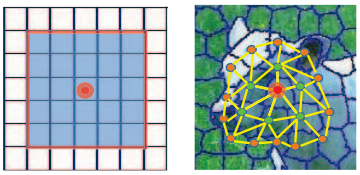
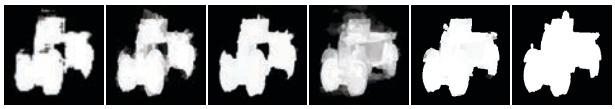


Figure 4. Window centered at a node k (denoted by red dot). Left: a window center at k with a 5-pixel width on the pixel grid. Right: a sub graph centered at k with a radius of 2 hoops (D = 2).



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

Figure 5. Example of superpixel filtering. (a)level set map output by the VGG16-based CNN. (b)Average (a) within every superpixel. (c) Saliency map generated using (b). (d) Guided superpixel filtering on (a) with radius of D = 3. (e) Saliency map generated using (d). (f)groundtruth.

Guided filter can be conveniently added into deep networks and fine-tuned end-to-end with,

$$\frac{\partial q_i}{\partial p_j} = Wij(I) \quad (10)$$
$$\delta_{p_j} = \sum_i W_{ij}(I)\delta_{q_i} \quad (11)$$

Where δ_{q_i} is the differential error at the output end, and δ_{p_i} is the differential error back propagated to the input end. Since $W_{ij}(I) = W_{ji}(I)$, the back propagation of this guided filter is achieved by simply filtering the differential errors reaching its output end.

3.3. Implementation Details

The structure of the proposed network is shown in Fig. 3. The final deep level set network is composed of a VGG16-based CNN, a Guided Superpixel Filtering(GSF) layer, and a Heaviside Function(HF) layer. ReLU layers are applied after each convolutional layers in the CNN. The CNN is followed by a Guided Superpixel Filtering layer with a hyper-parameter D. The GSF layer takes superpixels and a level

set map ϕ produced by the CNN as input. In this work, we utilize the fast gSLICr [41] to over-segment images into 400∼500 superpixels. What at last is an Approximated Heaviside Function layer with a hyper-parameter ε as in Eq. 7. This layer transforms the level set map into the final saliency map. We train the network with MSRA10K dataset [10], which contains 10000 scene images. During training, we scale both training images and groundtruths to 224*224. Input images are subtracted by mean pixel values. The final network is first trained with Binary Cross Entropy(BCE) loss for 15 epochs, then fine-tuned with the proposed level set method for 15 epochs, and finally the Guided Superpixel Filtering layer is added and finetuned. We use Adam [23] with an initial learning rate of 1e-4 to update the weights. The learning rate is reduced when validation performance stops improving. We implement the network with Torch framework. All experiments are performed with TESLA k40c GPU, 2.3GHz CPU, and 64G RAM.

4. Experiments

4.1. Datasets

We evaluate the performance of the proposed method on several benchmark datasets. The **SED2** [2] is composed of 100 images with two salient objects. The **PASCAL** [32] has 850 images with complex scene. Since groundtruths in this dataset are not binary, we threshold them at 0.5 as done in previous works. The **ECSSD** [55] contains 1000 images with structurally complex content. The **HKU-IS** [29] contains 4447 challenging images with multiple objects, objects on the boundary, or objects of low contrast. The **OMRON** [56] has 5168 challenging images with complex background and objects. The **THUR** [9] has 6232 images collected from Flickr with 5 topics:"Butterfly", "Coffee Mug", "Dog Jump", "Giraffe" and "Plane".

4.2. Evaluation Metrics

Four metrics are used for quantitative performance comparison and analysis, including Precision-Recall (PR) curve, adaptive-F_β, ω-F_β [36], and Mean Absolute Error (MAE). Let saliency value be in the range [0,255], PR curve is generated by computing $Precision = \frac{|M \cap G|}{|M|}$ and $Recall = \frac{|M \cap G|}{|G|}$ on the binary mask (M) and groundtruth (G) with threshold value varying from 0 to 255.

To compute the adaptive-F_β, we binarize a saliency map with twice of its mean saliency value as the threshold. Then Precision and Recall are computed on the binary map, and the adaptive-F_β is:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \qquad (12)$$

where β^2 is set as 0.3 typically.

As pointed out by [36], traditional evaluation metrics may suffer from interpolation flaw, dependency flaw, and equal-importance flaw. Therefore, the ω-F_β metric proposed in [36] is utilized for performance comparison. We use the code provided by authors with the default setting.

The Mean Absolute Error (MAE) is another widely used evaluation metric which is the average per-pixel difference between saliency map(S) and groundtruth(G). With saliency map value varying in [0,1] and groundtruth value varying in $\{0,1\}$,

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)| \qquad (13)$$

where W and H are width and height of the map respectively.

4.3. Performance Comparison

We compare the proposed method with several recent state-of-the-art methods on the aforementioned datasets. These methods include contrast based models FT [1], HC [10], DRFI [21], center-bias based algorithms GMR [56], BSCA [47], and recent deep learning based methods MTDS [31], MDF [29], MCDL [62], ELD [26], LEGS [50]. For fair comparison, we use the detection results or original codes provided by authors with default setting.

Fig. 6 shows the corresponding performance comparison with Precision-Recall Curve and adaptive-F_β. It can be seen that deep learning based methods achieve much better performance than traditional methods. Compared with the existing state-of-the-art methods, our method achieves better performance where both precision and recall are high (top-right region of the P-R curve). Good segmentation results are usually generated using a threshold within this range. As shown in the second row of Fig. 6, the proposed method can achieve better adaptive-threshold segmentation performance than other methods. Comparisons of MAE and ω-F_β are shown in Table 1. Our model achieves the best performance on most datasets. Among these models, the recent state-of-the-art methods ELD and MTDS are both trained with the MSRA10K dataset and built on the VGG16-net, which is the same setting as ours. Given an input, ELD processes the superpixels one by one and takes about 0.799 seconds to produce the final saliency map (implemented in $C++/caffe$). MTDS solves an optimization problem for every input based on superpixels and coarse saliency map output by a fully convolutional neural network, and takes about 5.6 seconds per image (implemented in $python/caffe$). These are two typical methods to refine the saliency maps with superpixels. Different from them, our method efficiently and effectively incorporates superpixel information via the guided superpixel filtering layer. During testing, the proposed network processes an

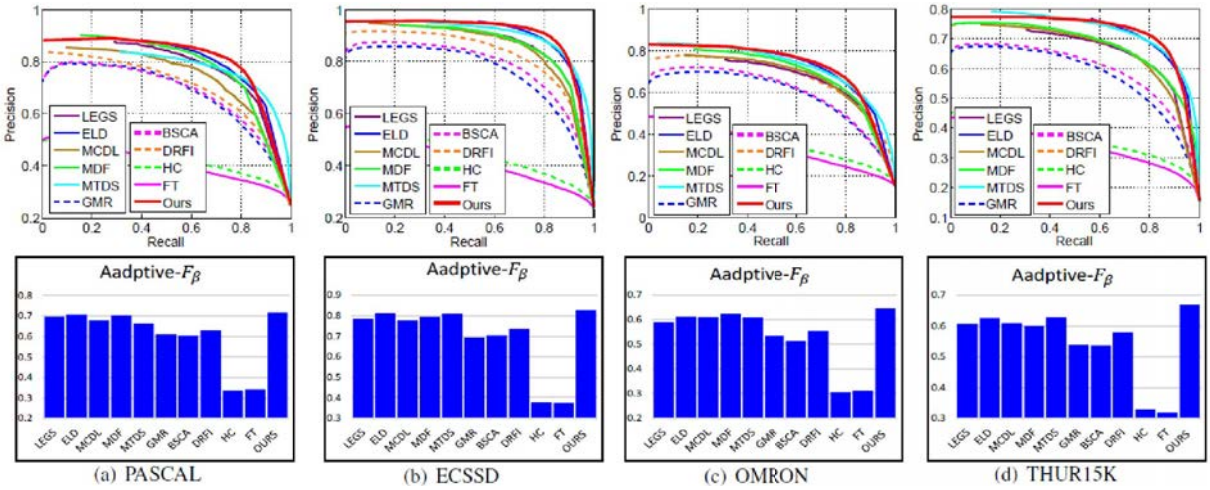


Figure 6. Precision-Recall curves and adaptive-F_β on the datasets.

Dataset	Metrics	FT	HC	DRFI	GMR	BSCA	MCDL	LEGS	MTDS	MDF	ELD	Ours
SED2	MAE	0.198	0.180	0.125	0.167	0.157	0.119	0.123	0.118	0.107	0.105	0.084
	ω-F_β	0.333	0.517	0.613	0.568	0.526	0.630	0.621	0.611	0.674	0.687	0.733
PASCAL	MAE	0.295	0.348	0.204	0.217	0.225	0.157	0.161	0.179	0.151	0.132	0.136
	ω-F_β	0.205	0.309	0.514	0.421	0.439	0.573	0.596	0.537	0.582	0.658	0.651
ECSSD	MAE	0.293	0.334	0.166	0.190	0.185	0.116	0.122	0.125	0.112	0.092	0.090
	ω-F_β	0.244	0.319	0.585	0.484	0.509	0.679	0.682	0.663	0.692	0.756	0.766
OMRON	MAE	0.258	0.320	0.156	0.187	0.190	0.098	0.134	0.120	0.094	0.096	0.093
	ω-F_β	0.194	0.242	0.467	0.379	0.370	0.548	0.520	0.486	0.557	0.581	0.591
HKU-IS	MAE	0.228	0.291	0.154	0.181	0.178	0.106	0.122	0.081	0.130	0.084	0.072
	ω-F_β	0.196	0.311	0.553	0.443	0.460	0.634	0.607	0.711	0.567	0.718	0.748
THUR	MAE	0.219	0.293	0.153	0.179	0.183	0.110	0.127	0.118	0.132	0.103	0.099
	ω-F_β	0.173	0.256	0.471	0.372	0.384	0.542	0.535	0.525	0.502	0.596	0.621

Table 1. Comparison between the proposed method and methods FT [1], HC [10], DRFI [21], GMR [56], BSCA [47],MCDL [62], LEGS [50], MTDS [31], MDF [29], ELD [26]. The best one is labeled in red, the second-best one is in green, and the third is in blue.

input by simply performing one single feed-forward pass, which takes only 0.078 second on average with the same experiment environment. Some qualitative comparisons are shown in Fig. 7. Our model is able to produce saliency maps that highlight salient regions accurately and uniformly.

4.4. Analysis of the Proposed Method

There are two important hyperparameters in the proposed method, the ε in Approximated Heaviside Function and the graph radius D in Guided Superpixel Filtering. Fig. 8 shows the performance of the proposed network with different parameter values on the datasets. The two charts on the top are for the graph radius D in the Guided Superpixel Filter. We tried six values and found that increasing D from 1 leads to slightly improved performance, when $D > 4$, the performance decreases rapidly. A large D will propagate saliency within a large region, and this may produce noise in complex scenes. The bottom charts show performance for different ε in the Approximated Heaviside Function. Both large and small values result in the decrease of the performance. A smallε results in a narrow support range and may get stuck in a local minimum, on the other hand, a large ε may fail to learn a good level set function and cannot generate compact saliency map. Therefore, in the final system, we set $D = 3$, and $\varepsilon = \frac{1}{32}$.

Analysis of the proposed energy function defined in Eq. 4 is shown in Fig. 9. We train the VGG16-based CNN with different parameter settings to evaluate the contribution of different terms. As shown in the figure, the Length terms (associated with γ) and the compactness term (associated with λ) help to improve the performance. We also evaluate the performance of different components in our network as shown in Fig. 3. The VGG16-based CNN is trained with BCE loss for 15 epochs and fine-tuned with the proposed level set method for 15 epochs (denoted by "CNN+LS"). By inserting a GSF layer, we get the complete network and denote it by "CNN+LS+GSF". For comparison, another VGG16-based CNN is trained for 30 epochs

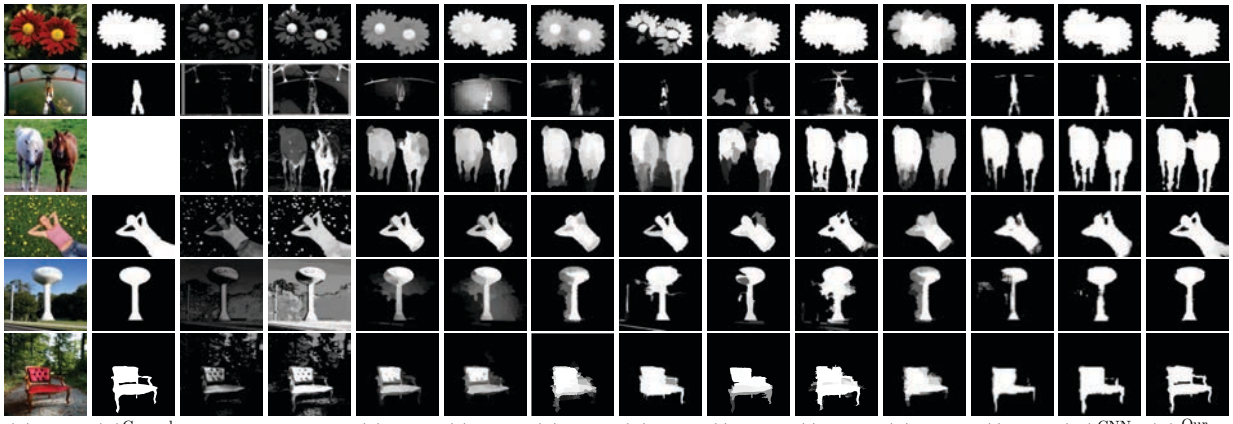


Figure 7. Qualitative comparison with FT [1], HC [10], GMR [56], BSCA [47], MTDS [31], MDF [29], MCDL [62], ELD [26], LEGS [50]. "CNN" is output by the CNN trained with BCE loss. "CNN+LS" is output by the CNN trained with the level set method.

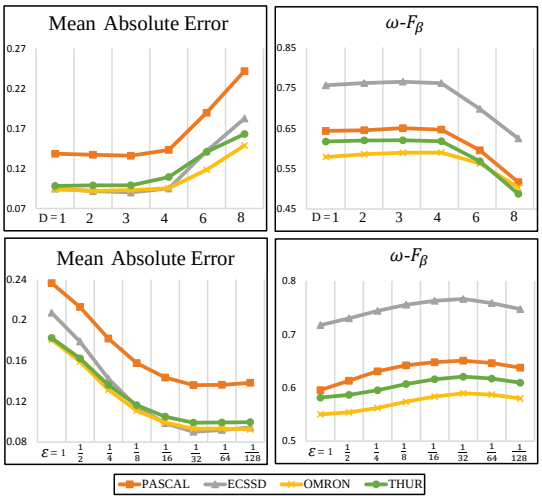


Figure 8. Parameter Analysis. Top: performance for different radius D in Guided Superpixel Filtering layer. Bottom: performance for different ε in Approximated Heaviside Function layer.

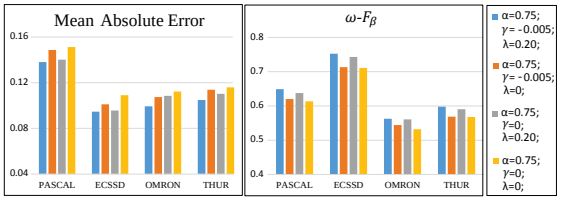


Figure 9. Performance of different parameter settings.

only using BCE loss (denoted by "CNN"). As shown in Table 2 and Fig. 7(l)-(n), using Level Sets helps the network to detect more compact salient regions and detect more details of shape. And by adding the GSF layer, the network can further improve saliency maps and accurately segment salient objects from the background. We also report time cost in the table. The guided superpixel filtering layer takes about 28ms to process an image with $D = 3$. Our final network can perform at a speed over 12FPS.

		PASCAL	ECSSD	OMRON	THUR	Time(ms)
CNN	MAE	0.143	0.104	0.102	0.108	50
	ω-F_β	0.619	0.715	0.534	0.569	
CNN+ LS	MAE	0.138	0.095	0.099	0.105	50
	ω-F_β	0.646	0.752	0.569	0.593	
CNN+LS +GSF	MAE	0.136	0.090	0.093	0.099	78
	ω-F_β	0.651	0.766	0.591	0.621	

Table 2. Method Analysis. "CNN" represents the CNN trained with BCE loss. "CNN+LS" is for the CNN trained with the level set method. "CNN+LS+GSF" represents the complete network. The last column reports the average time cost for one image.

5. Conclusion

In this paper, an end-to-end deep level set network have been proposed to detect salient objects. Trained to learn a level set function instead of binary groundtruth directly, the network can deal with object boundaries more accurately. Furthermore, the proposed method extends the guided image filter to deal with superpixels so that saliency can be further propagated between pixels and the saliency map can be recovered to full resolution. Experiments on benchmark datasets demonstrate that the proposed deep level set network can detect salient objects effectively and efficiently.

Acknowledgement

The research is in part supported by Singapore Ministry of Education (MOE) Tier 2 ARC28/14, and Singapore A*STAR Science and Engineering Research Council PSF1321202099. The authors gratefully acknowledge the support of NVAITC (NVIDIA AI Technology Centre) for their donation of Tesla K40 and K80 cards used for our research at the Rapid-Rich Object Search (ROSE) Lab.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009.

[2] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Trans. on PAMI*, 34(2):315–327, 2012.

[3] T. Andersson, G. Lathen, R. Lenz, and M. Borga. Modified gradient search for level set based image segmentation. *IEEE Trans. on image processing*, 22(2):621–630, 2013.

[4] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics*, volume 26, page 10, 2007.

[5] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. Pixelnet: Towards a general pixel-level architecture. *arXiv preprint arXiv:1609.06694*, 2016.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, 1997.

[7] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM journal on applied mathematics*, 66(5):1632–1648, 2006.

[8] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.

[9] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

[10] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Trans. on PAMI*, 37(3):569–582, 2015.

[11] L. Evans. Partial differential equations. *Province: American Mathmatical Society*, 1998.

[12] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Trans. on PAMI*, 34(10):1915–1926, 2012.

[13] H. Hadizadeh and I. V. Bajic. Saliency-aware video compression. *IEEE IEEE Trans. on Image Processing*, 23(1):19–33, 2014.

[14] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006.

[15] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[17] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[18] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[19] L. Itti, C. Koch, E. Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on PAMI*, 20(11):1254–1259, 1998.

[20] Y. Jia and M. Han. Category-independent object-level saliency detection. In *ICCV*, 2013.

[21] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013.

[22] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *ICCV*, 2013.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *CVPR*, 2016.

[25] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, 2016.

[26] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016.

[27] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. on Image Processing*, 17(10):1940–1949, 2008.

[28] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: a new variational formulation. In *CVPR*, 2005.

[29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, 2015.

[30] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016.

[31] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Trans. on Image Processing*, 25(8):3919 – 3930, 2016.

[32] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

[33] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016.

[34] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Trans. on PAMI*, 33(2):353–367, 2011.

[35] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, 17(2):158–175, 1995.

[36] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *CVPR*, 2014.

[37] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[38] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[39] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *CVPR*, 2016.

[40] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

[41] C. Y. Ren, V. A. Prisacariu, and I. D. Reid. gSLICr: SLIC superpixels at over 250Hz. *ArXiv preprint arXiv:1509.04232*, 2015.

[42] B. Schauerte and R. Stiefelhagen. Quaternion-based spectral saliency detection for eye fixation prediction. In *ECCV*. 2012.

[43] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, 2012.

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[45] M. Sun, A. Farhadi, B. Taskar, and S. Seitz. Salient montages from unconstrained videos. In *ECCV*, 2014.

[46] Y. Tang and X. Wu. Saliency detection via combining region-level and pixel-level predictions with cnns. In *ECCV*, 2016.

[47] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In *CVPR*, 2015.

[48] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien. Real-time salient object detection with a minimum spanning tree. In *CVPR*, 2016.

[49] B. Vemuri and Y. Chen. Joint image registration and segmentation. In *Geometric level set methods in imaging, vision, and graphics*, pages 251–269. 2003.

[50] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, 2015.

[51] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016.

[52] L. Wang, H. Wu, and C. Pan. Region-based image segmentation with local signed difference energy. *Pattern Recognition Letters*, 34(6):637–645, 2013.

[53] Q. Wang, W. Zheng, and R. Piramuthu. Grab: Visual saliency via novel graph model and background priors. In *CVPR*, 2016.

[54] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, 2012.

[55] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013.

[56] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.

[57] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[58] F. Zhang, B. Du, and L. Zhang. Saliency-guided unsupervised feature learning for scene classification. *IEEE Trans. on Geoscience and Remote Sensing*, 53(4):2175–2184, 2015.

[59] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In *ICCV*, 2013.

[60] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, 2015.

[61] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

[62] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015.

[63] J.-Y. Zhu, J. Wu, Y. Xu, E. Chang, and Z. Tu. Unsupervised object class discovery via saliency-guided multiple class learning. *IEEE Trans. on PAMI*, 37(4):862–875, 2015.

[64] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, 2014.