Supplemental Materials

Salient Region Detection using Diffusion Process on a 2-Layer Sparse Graph

Li Zhou, Zhaohui Yang, Zongtan Zhou, and Dewen Hu

1. Influence of close-loop constraint

In constructing the graph model, we enforce the condition that the nodes on the four sides of the image are connected. This close-loop constraint significantly improves the performance of the proposed method because it tends to reduce the geodesic distance between similar superpixels on the four edges of the image. Many recent diffusion-based methods [1-4] use the close-loop constraint when constructing the graph model, and these works have analyzed the influence of the close-loop constraint. Thus, we have not conducted a detailed analysis in our paper.

To investigate the influence of the close-loop constraint in the proposed method, we conduct a series of experiments. The results are as follows.

The average precision, recall, F-Measure, MAE, and MOR using different approaches are shown in Table 1. We can see that the proposed method using the close-loop constraint achieved the best result for all five criteria.

The precision-recall curves of different graph construction methods with and without the close-loop constraint are shown in Fig. 1, in which the close-loop constraint improves performance significantly.

Several natural example images are shown in Fig. 2 for a visual comparison of the saliency map with and without the close-loop constraint. From the example images, we can observe that the proposed close-loop constraint enhances the quality of the saliency map efficiently.

Table 1

Evaluation of the influence of the close-loop constraint on saliency detection performance in the ASD dataset. For each criterion, the top three results are highlighted in **red**, **blue** and **green**, respectively.

Criterion	2-layer sparse graph + **close-loop constraints**	2-layer sparse graph	1-layer neighborhood graph + **close-loop constraints**	1-layer neighborhood graph	2-layer neighborhood graph + **close-loop constraints**	2-layer neighborhood graph
Precision	**0.9286**	0.9043	**0.9230**	0.8985	**0.9247**	0.8957
Recall	**0.8961**	0.8559	**0.8950**	0.8458	**0.8676**	0.8095
F-measure	**0.9132**	0.8731	**0.9098**	0.8654	**0.8960**	0.8459
MAE	**0.0697**	0.1058	**0.0780**	0.1177	**0.0802**	0.1168
MOR	**0.8432**	0.7952	**0.8388**	0.7842	**0.8142**	0.7543

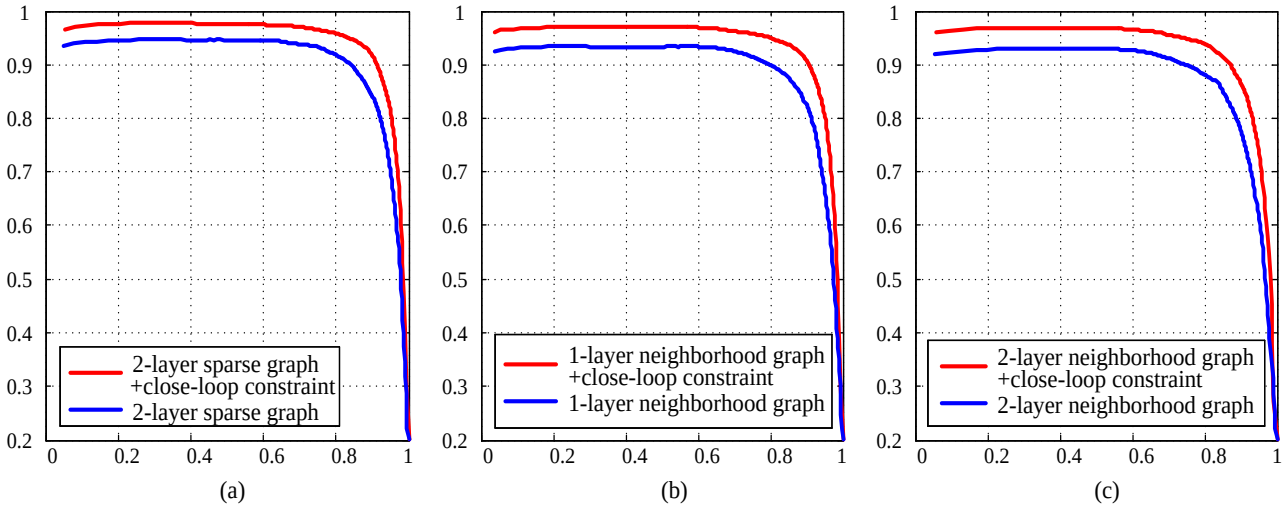


Fig. 1. Evaluation of the influence of the close-loop constraint on the ASD dataset with different graph construction methods. (a) Precision-recall curves of our 2-layer sparse graph. (b) Precision-recall curves of the 1-layer neighborhood graph. (c) Precision-recall curves of the 2-layer neighborhood graph.

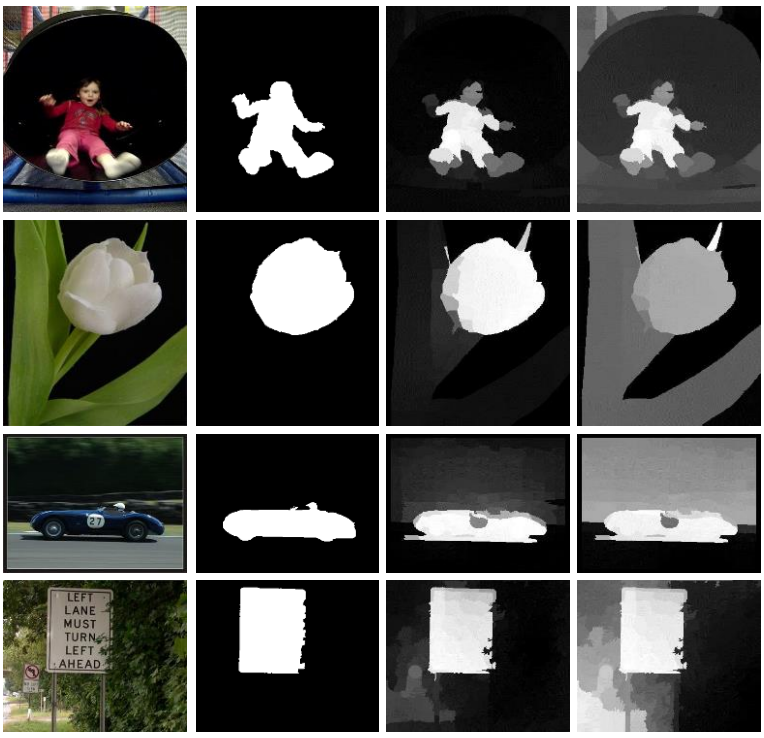


Fig. 2. Visual comparison of the saliency map with and without the close-loop constraint on the ASD dataset. (a) Input image. (b) Ground truth. (c) Our 2-layer sparse graph with the close-loop constraint. (d) 2-layer sparse graph without the close-loop constraint.

2. Influence of Parameters *N*

We used the SLIC model to abstract the input image into uniform and compact regions, where *N* controls the number of superpixel nodes. It is well known that *N* can have a significant effect on

performance. If N is too small, many different objects will be mapped to the same superpixel, which leads to a decrease in the detection of salient objects. If N is too large, salient objects will be mapped to many different superpixels, which may incorrectly suppress the salient regions.

To investigate the influence of N in the proposed method, we conduct a series of experiments. The results are as follows.

The precision-recall curves are shown in Fig. 3(a), and the average precisions, recalls, and F-Measures using an adaptive threshold are shown in Fig. 3(b). Figure 3 shows that, increasing N up to approximately 200 significantly improves performance, but makes little difference beyond this. These experimental results show that the proposed method is robust to the parameter N.

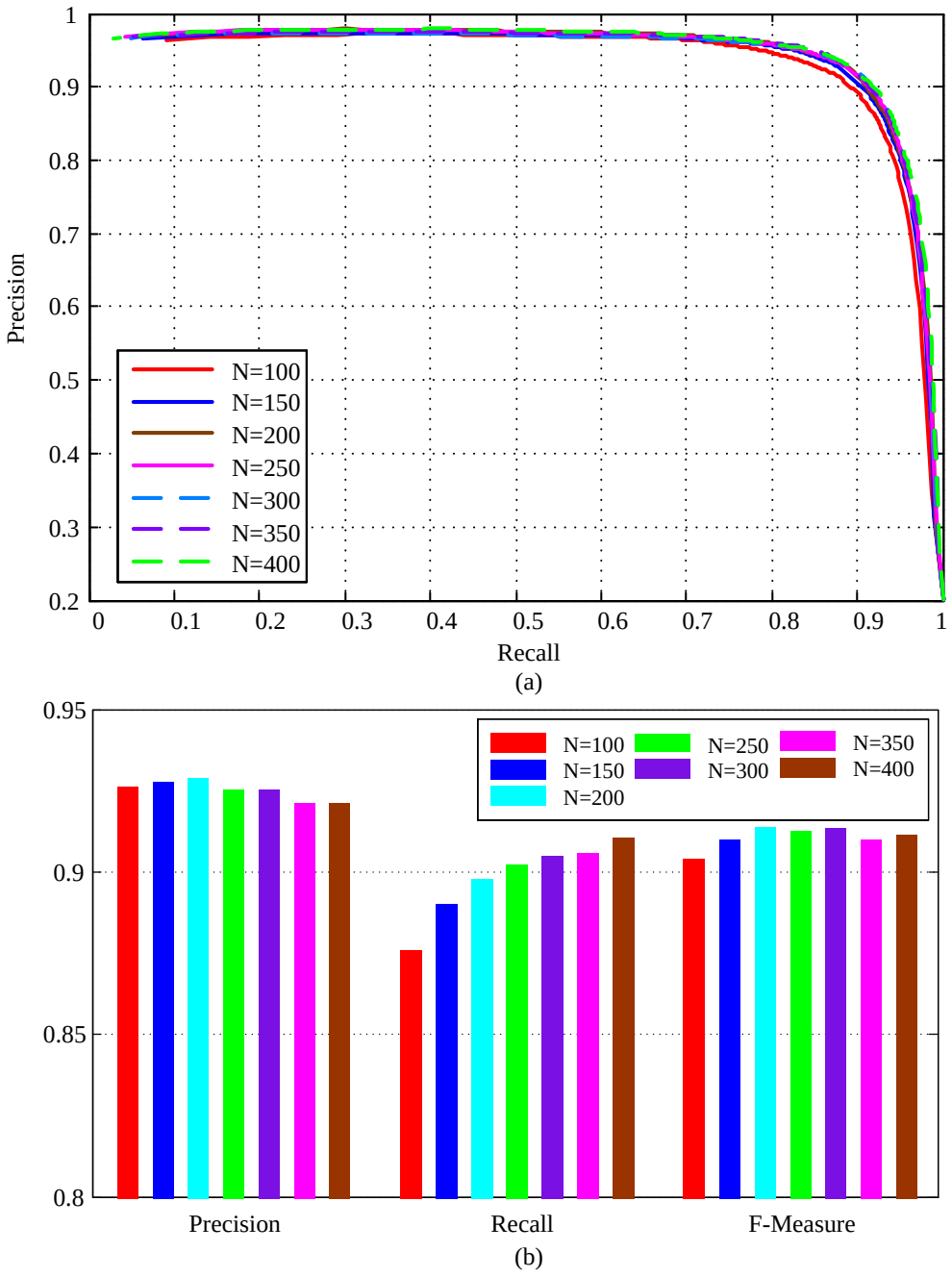


Fig. 3. Influence of the number of superpixel nodes N. (a) Precision-recall curves for different N. (b) Precision, recall and F-Measure for different N.

3. Illustration of the 2-layer sparse graph

Figure 4 shows an illustration of the graph edge for different graph construction methods.

As shown in Fig. 4 (b), a node (illustrated by a red dot) of the 1-layer neighborhood graph only connects to its adjacent nodes (green dot and connection). We can observe that, the neighborhood of the node is too small, with the result that character "T" cannot be connected with characters "S" and "O"; that is, the 1-layer neighborhood graph cannot effectively use the local spatial relationship. Thus, the nodes corresponding to "STOP" in the traffic sign are mistakenly marked as the background.

As shown in Fig. 4 (c), a node (illustrated by a red dot) of the 2-layer neighborhood graph connects to both its adjacent nodes (green dot and connection) and nodes that share common boundaries with neighboring nodes (yellow dot and connection). We can observe that, although the neighborhood of the node becomes larger, it also brings many dissimilar redundant nodes (the red shading of the traffic sign). Because the 2-layer neighborhood graph brings many dissimilar redundant nodes, it usually marks the background incorrectly as a salient region, as shown in Fig. 5 (f).

As shown in Fig. 4 (d), a node (illustrated by a red dot) of the proposed 2-layer sparse graph connects to both its adjacent nodes (green dot and connection) and the most similar node (yellow dot and connection) that shares common boundaries with its adjacent nodes. It can be observed that compared with the 1-layer neighborhood graph and 2-layer neighborhood graph, the proposed 2-layer sparse graph does not only effectively use the local spatial relationship but also removes the dissimilar redundant nodes.

Figure 5 shows a visual comparison of the aforementioned three graph construction methods, it is clear that the proposed 2-layer sparse graph performs better compared with the 1-layer neighborhood graph and 2-layer neighborhood graph.

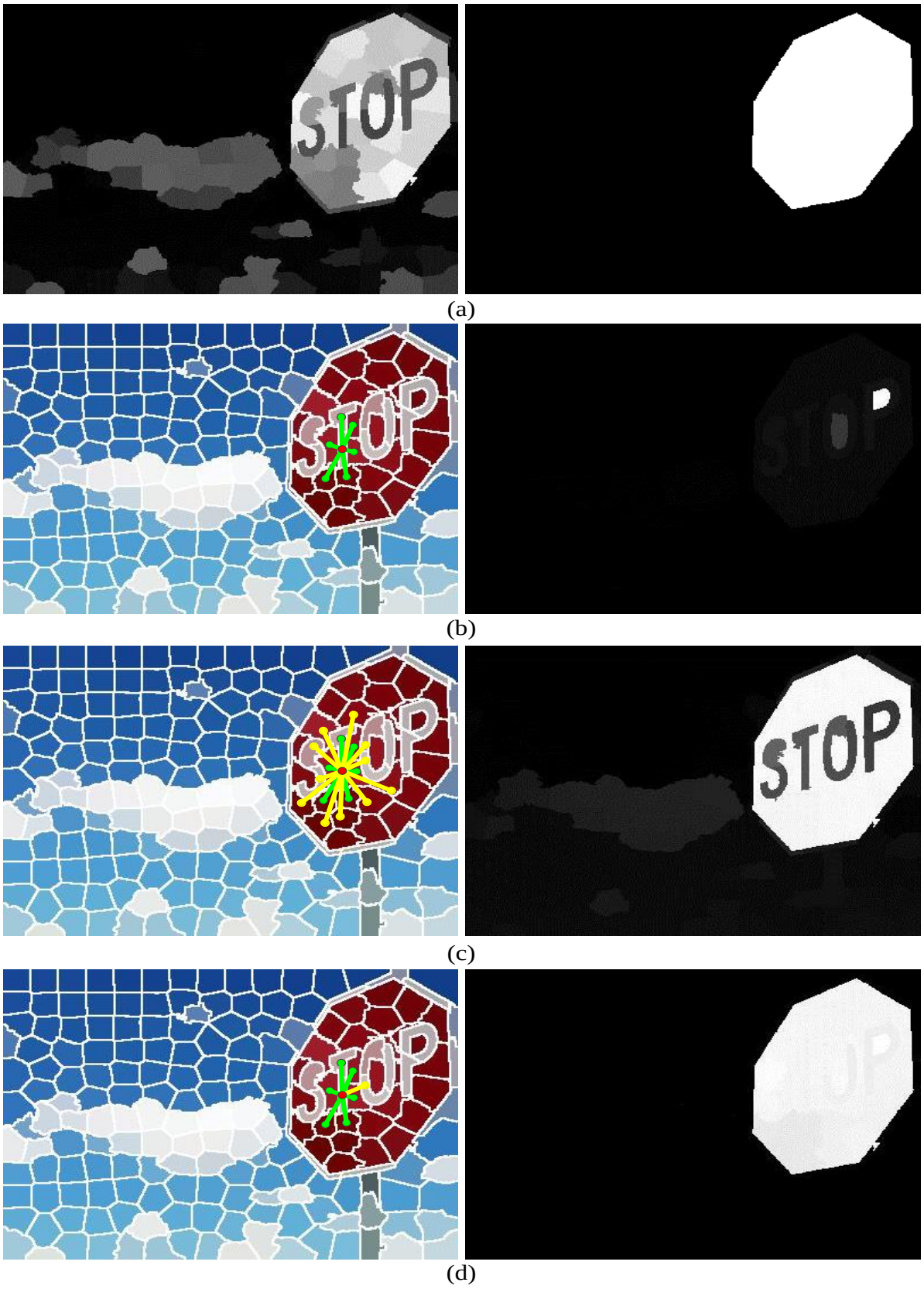
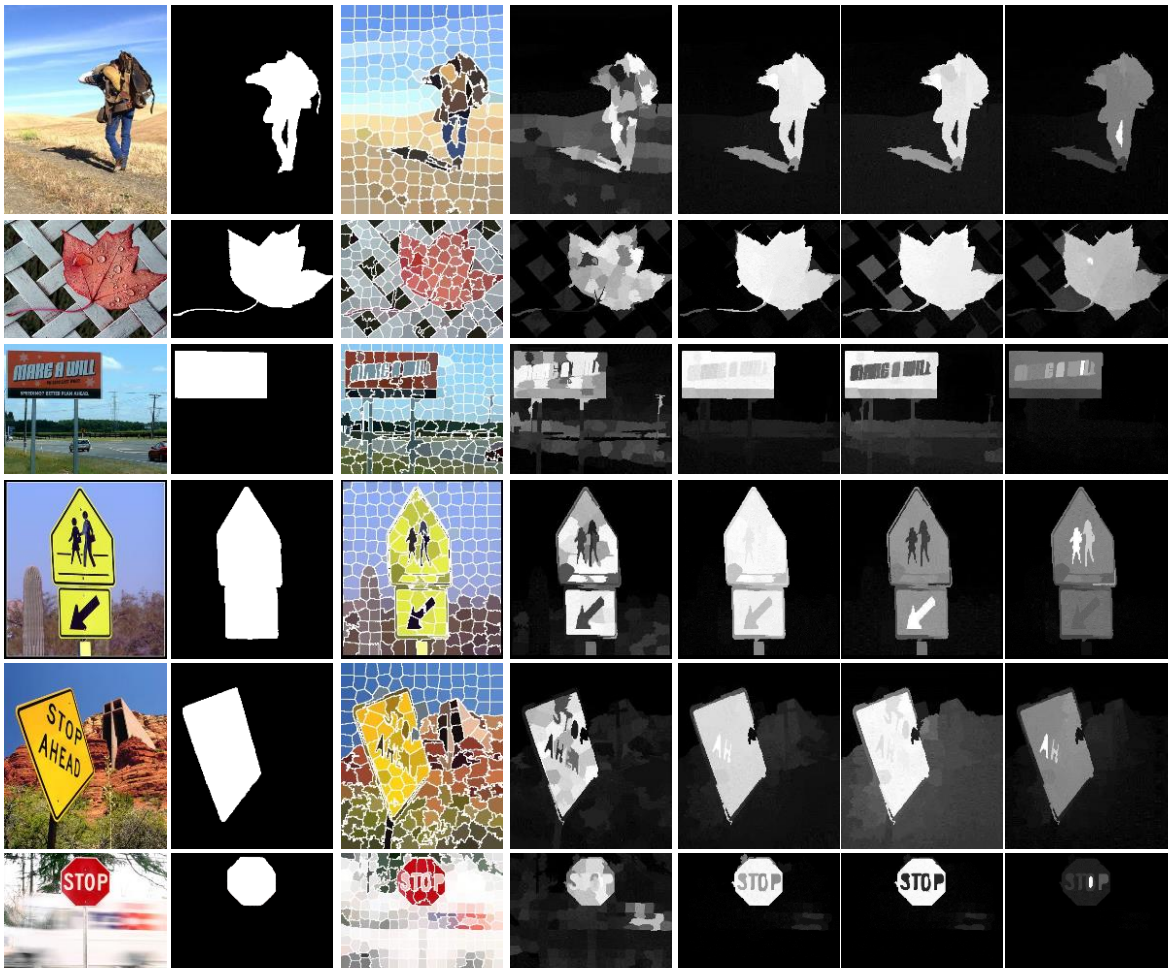


Fig. 4. Illustration of the graph edge for different graph construction methods, and the saliency map obtained after the diffusion process. (a) Image used for the diffusion process and the ground truth. (b) 1-layer neighborhood graph. (c) 2-layer neighborhood graph. (d) 2-layer sparse graph.



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

Fig. 5. Visual comparison of different graph construction methods. (a) Input image. (b) Ground truth. (c) Superpixel. (d) Images used for the diffusion process. (e) Saliency maps produced by the diffusion process on our 2-layer sparse graph. (f) Saliency maps produced by the diffusion process on a 2-layer neighborhood graph. (g) Saliency maps produced by the diffusion process on a 1-layer neighborhood graph.

4. Influence of different graph construction methods

As shown in Fig. 6, when salient objects contain regions with different appearances, the proposed 2-layer sparse graph can still detect the entire salient object. To investigate the effectiveness of the proposed 2-layer sparse graph, we conduct a series of experiments in the ASD, DUT-OMRON, ECSSD, MSRA5K, and MSRA10K datasets. From Figs. 7-11, it is clear that our proposed 2-layer sparse graph outperforms the other two graph construction methods for all five datasets, especially when there is no post-processing. These experimental results also show that post-processing can effectively improve the performance of the three graph construction methods. Additionally, because the 2-layer sparse graph obviously outperforms the other two graph construction methods, the degree of performance improvement of the proposed 2-layer sparse graph is not as large as the other two graph construction methods, using post-processing.

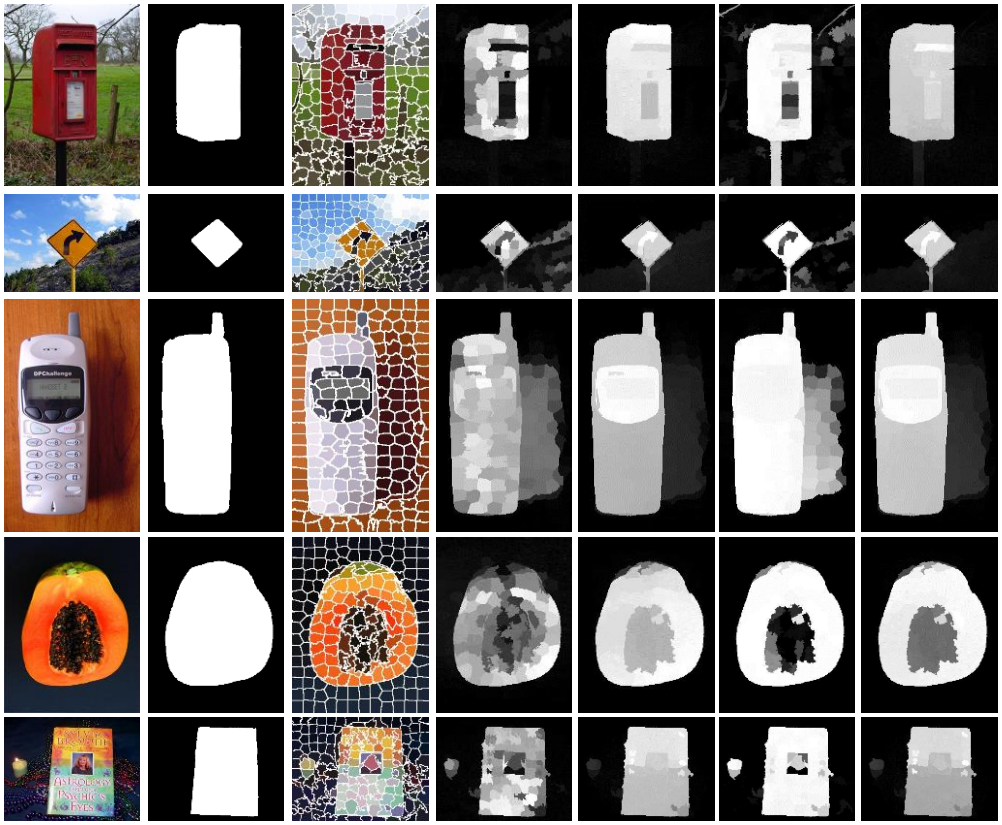


Fig. 6. Graph labeling results produced by our 2-layer sparse graph when salient objects contain regions with different appearances. (a) Input image. (b) Ground truth. (c) Superpixels. (d) $\boldsymbol{S}_{fg\_p}$ saliency maps. (e) $\boldsymbol{S}_{fg\_mr}$ saliency maps. (f) $\boldsymbol{S}_{bg\_p}$ saliency maps. (g) $\boldsymbol{S}_{bg}$ saliency maps.

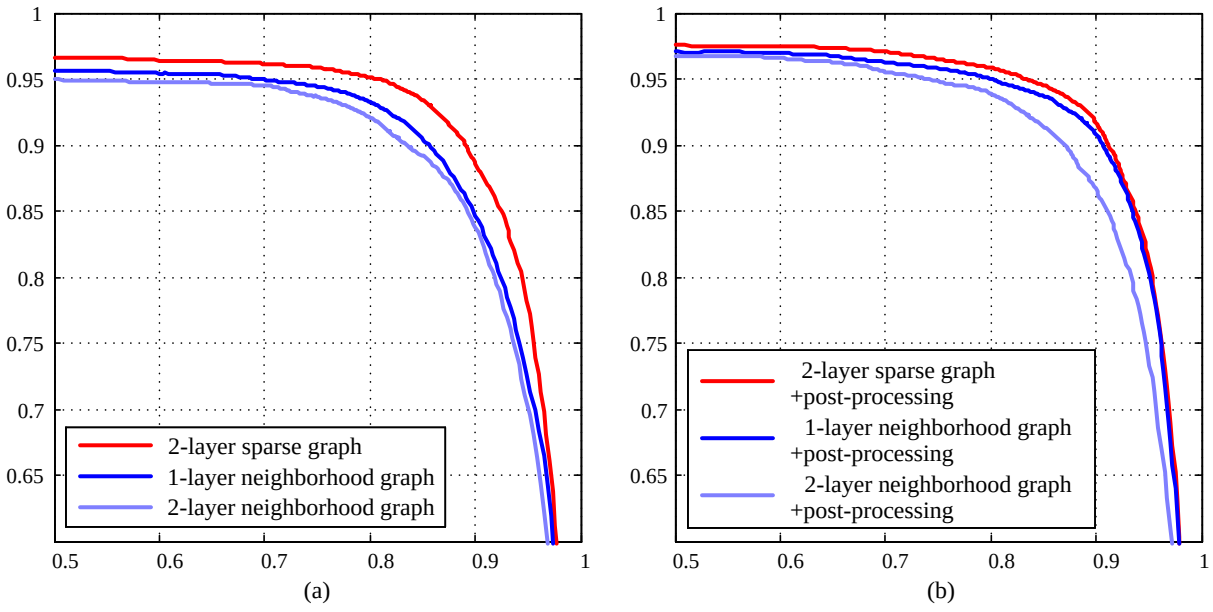


Fig. 7. Influence of different graph construction methods on the ASD dataset. (a) Precision-recall curves of different graph construction methods without post-processing. (b) Precision-recall curves of different graph construction methods with post-processing.

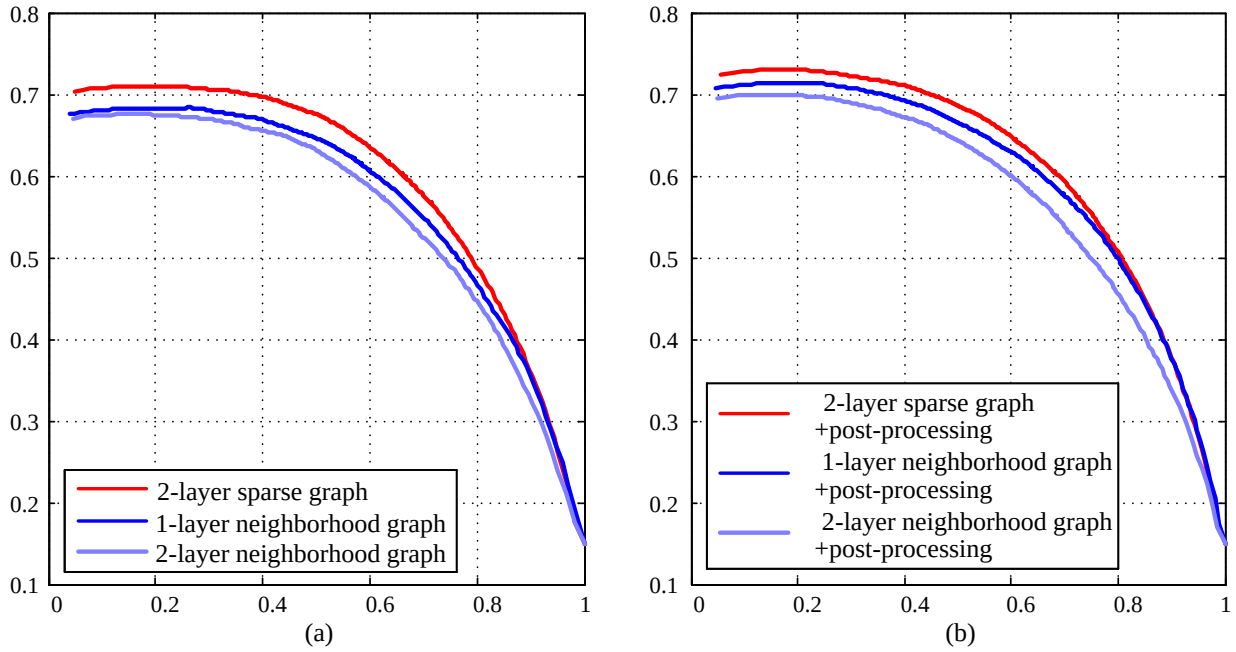


Fig. 8. Influence of different graph construction methods on the DUT-OMRON dataset. (a) Precision-recall curves of different graph construction methods without post-processing. (b) Precision-recall curves of different graph construction methods with post-processing.

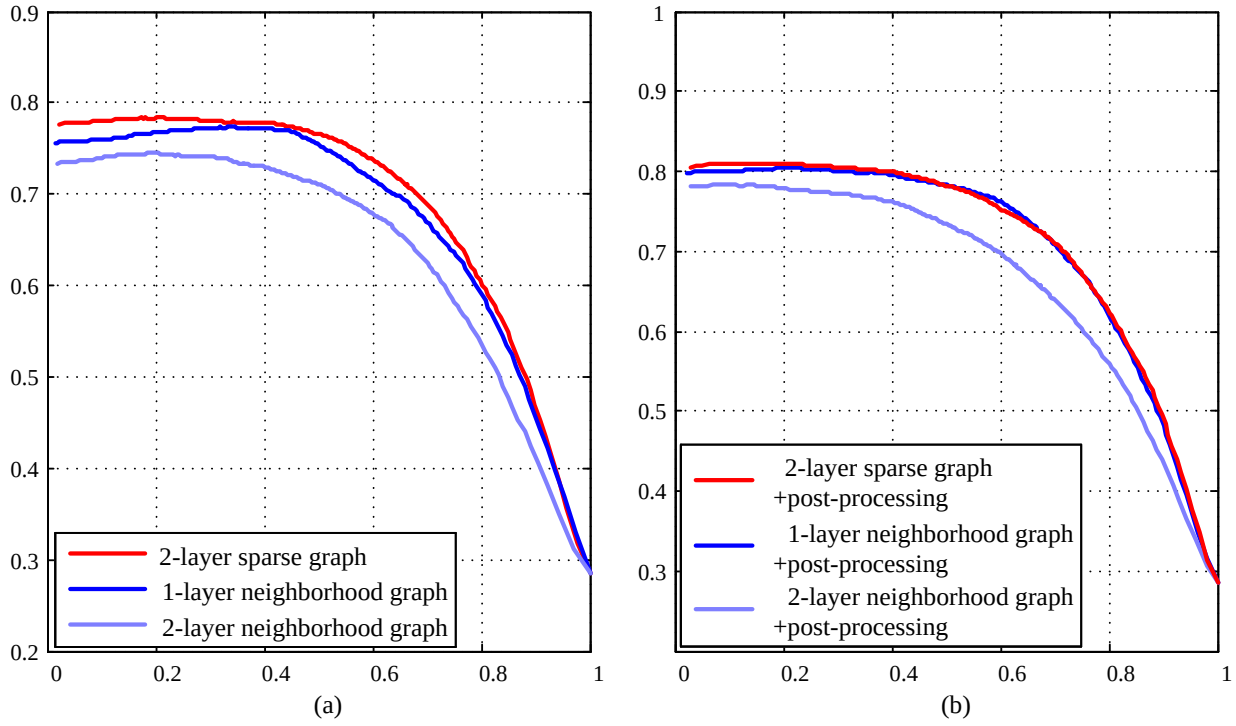


Fig. 9. Influence of different graph construction methods on the ECSSD dataset. (a) Precision-recall curves of different graph construction methods without post-processing. (b) Precision-recall curves of different graph construction methods with post-processing.

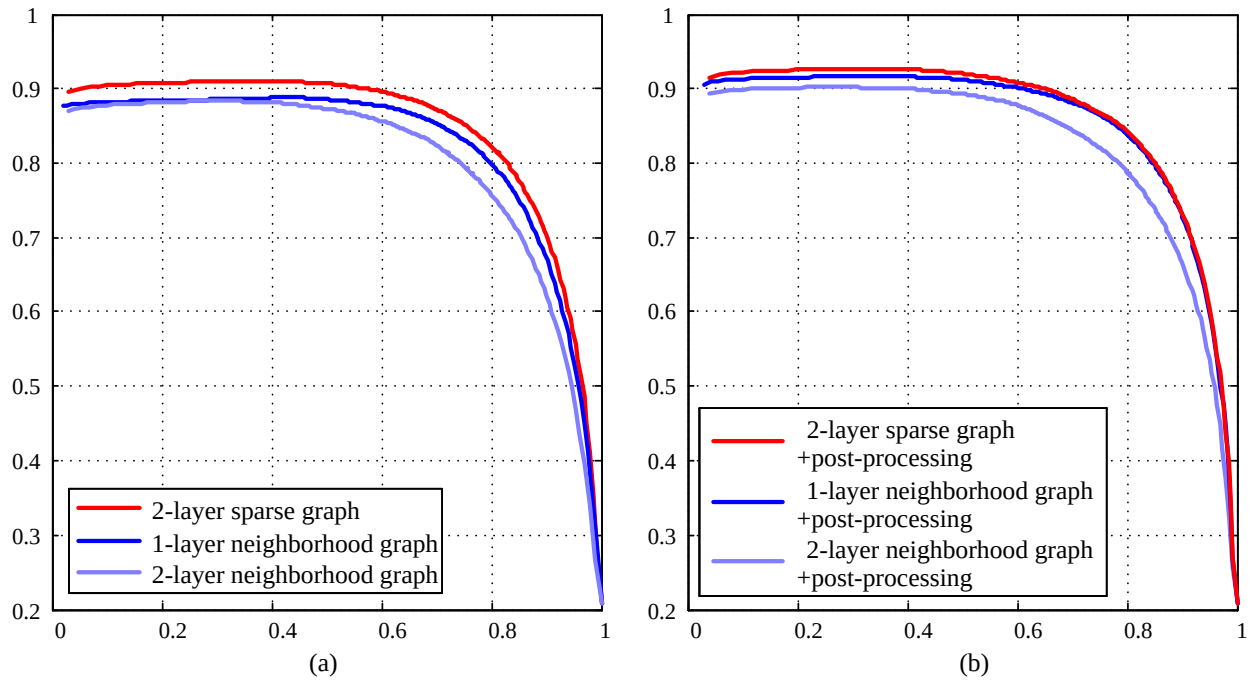


Fig. 10. Influence of different graph construction methods on the MSRA5K dataset. (a) Precision-recall curves of different graph construction methods without post-processing. (b) Precision-recall curves of different graph construction methods with post-processing.

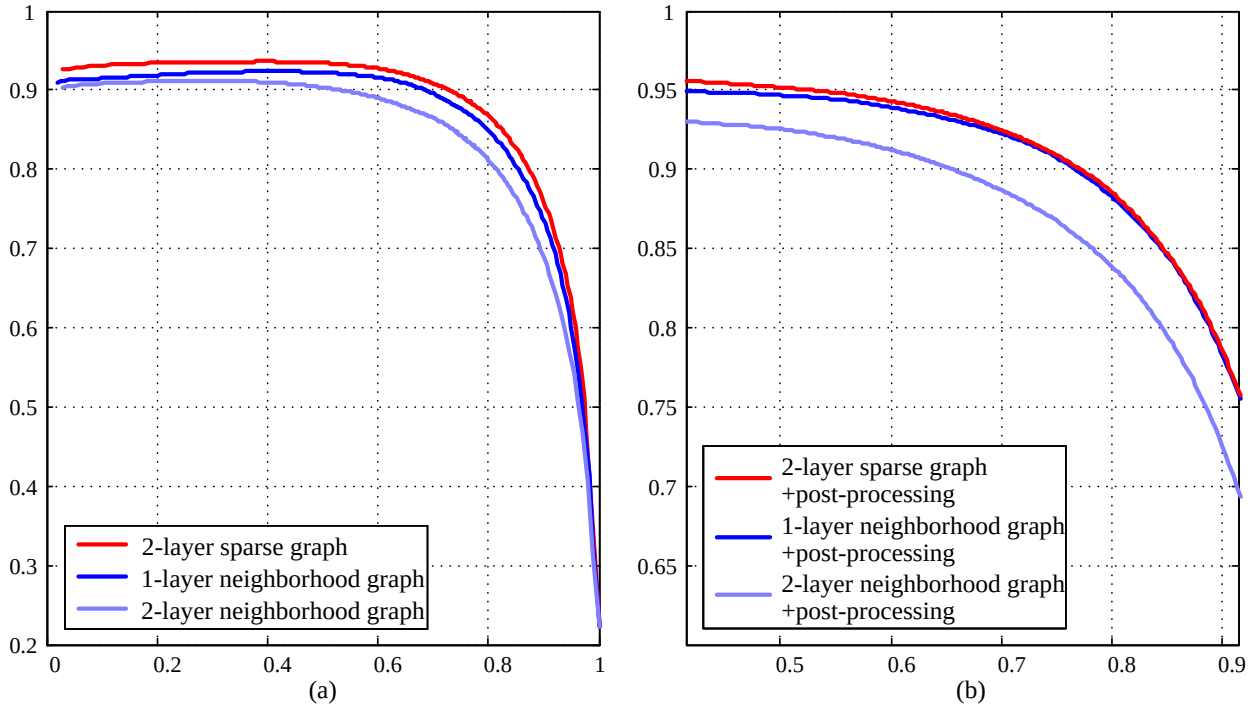


Fig. 11. Influence of different graph construction methods on the MSRA10K dataset. (a) Precision-recall curves of different graph construction methods without post-processing. (b) Precision-recall curves of different graph construction methods with post-processing.

5. Influence of graph-based post-processing

To investigate the effectiveness of the proposed graph-based post-processing, we conduct a series of experiments in the ASD, DUT-OMRON, ECSSD, MSRA5K, and MSRA10K datasets. From Figs. 12-16, it is clearly observed that the proposed graph-based post-processing can effectively improve the performance of the three graph construction methods for all five datasets.

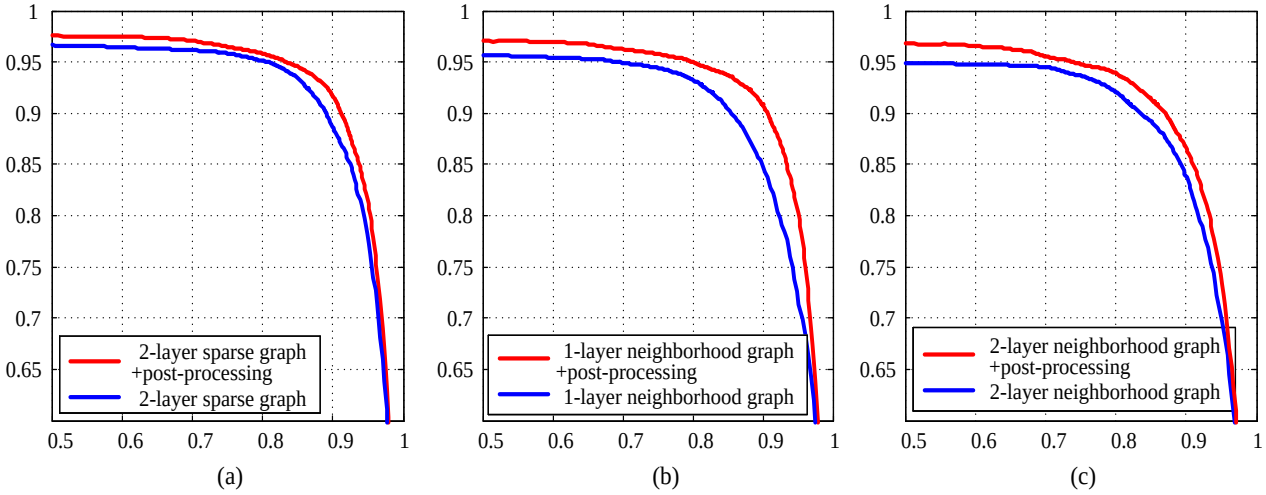


Fig. 12. Influence of the proposed graph-based post-processing on the ASD dataset. (a) Precision-recall curves of a 2-layer sparse graph with and without post-processing. (b) Precision-recall curves of a 1-layer neighborhood graph with and without post-processing. (c) Precision-recall curves of a 2-layer neighborhood graph with and without post-processing.

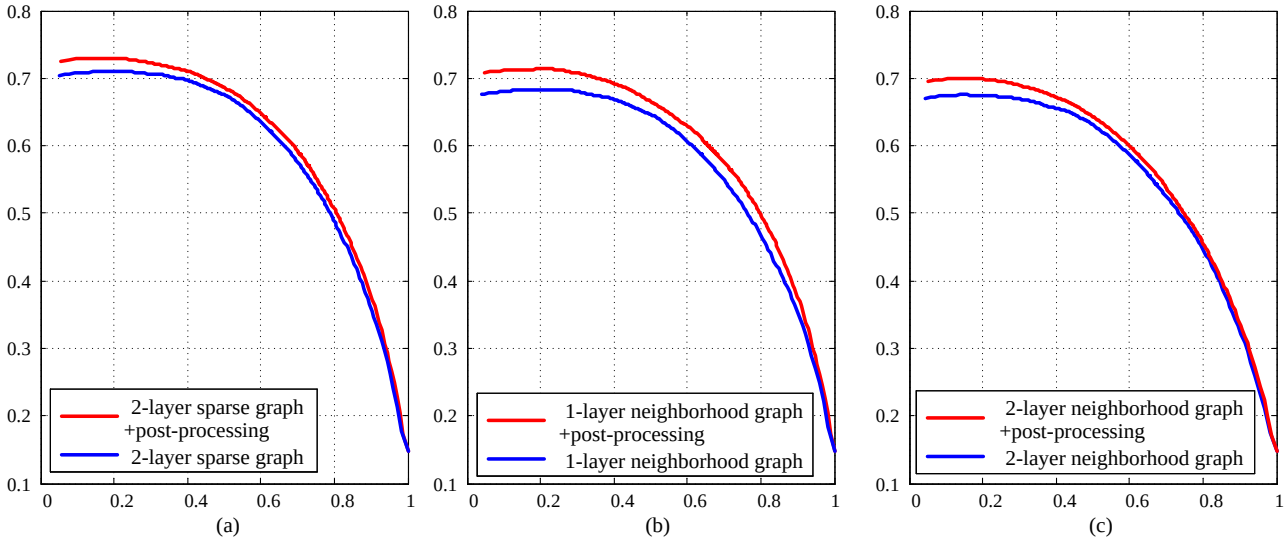


Fig. 13. Influence of the proposed graph-based post-processing on the DUT-OMRON dataset. (a) Precision-recall curves of a 2-layer sparse graph with and without post-processing. (b) Precision-recall curves of a 1-layer neighborhood graph with and without post-processing. (c) Precision-recall curves of a 2-layer neighborhood graph with and without post-processing.

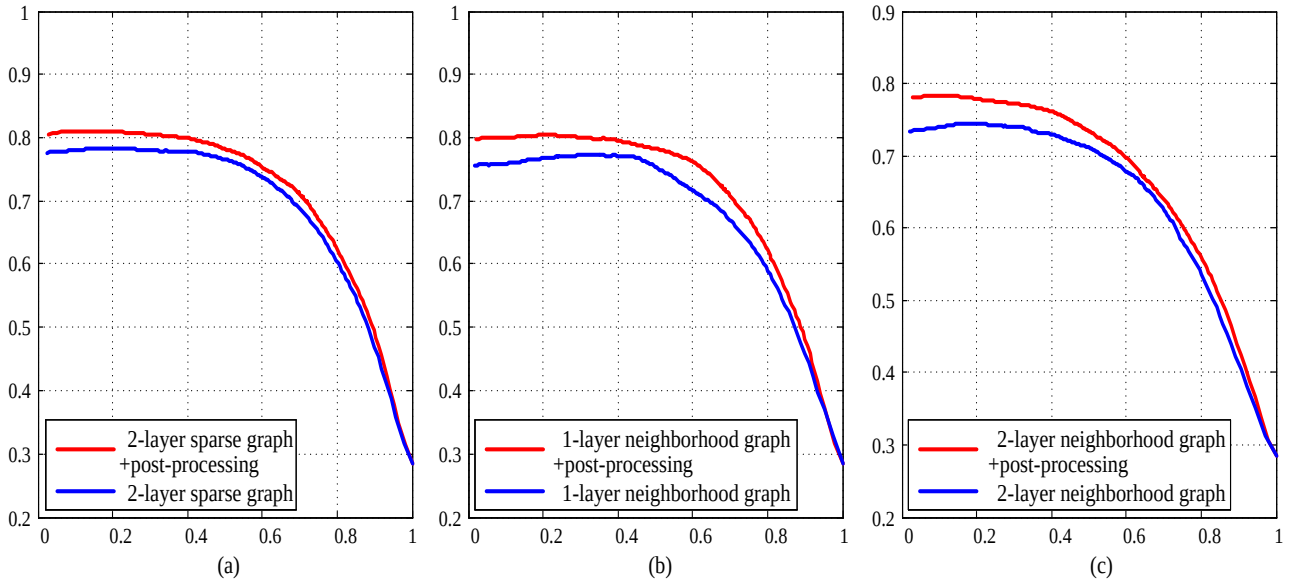


Fig. 14. Influence of the proposed graph-based post-processing on the ECSSD dataset. (a) Precision-recall curves of a 2-layer sparse graph with and without post-processing. (b) Precision-recall curves of a 1-layer neighborhood graph with and without post-processing. (c) Precision-recall curves of a 2-layer neighborhood graph with and without post-processing.

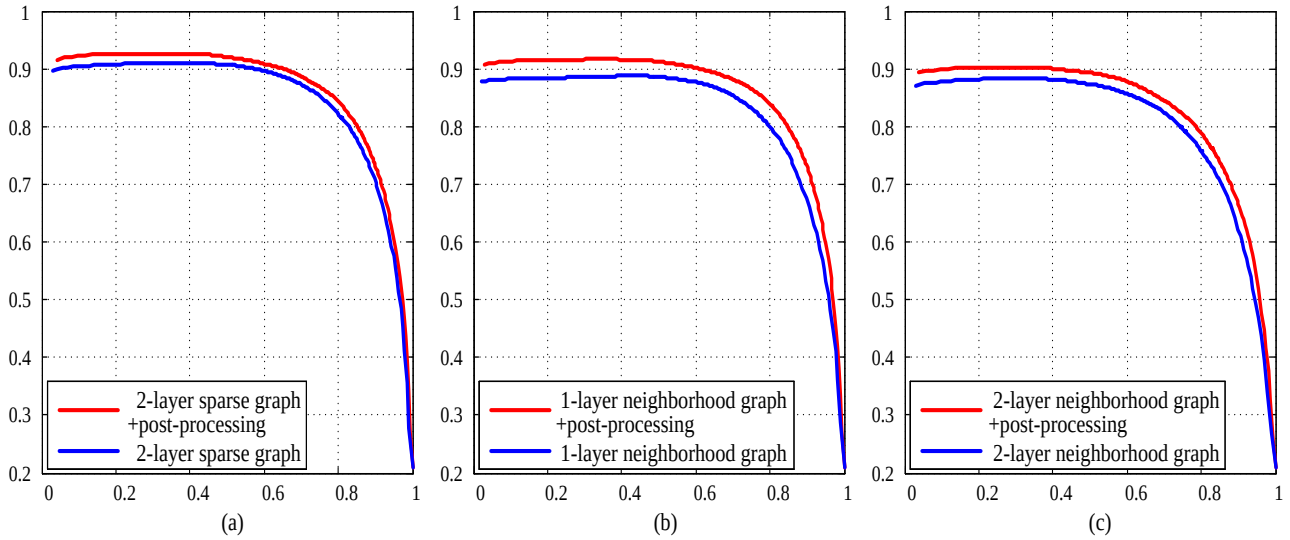


Fig. 15. Influence of the proposed graph-based post-processing on the MSRA5K dataset. (a) Precision-recall curves of a 2-layer sparse graph with and without post-processing. (b) Precision-recall curves of a 1-layer neighborhood graph with and without post-processing. (c) Precision-recall curves of a 2-layer neighborhood graph with and without post-processing.

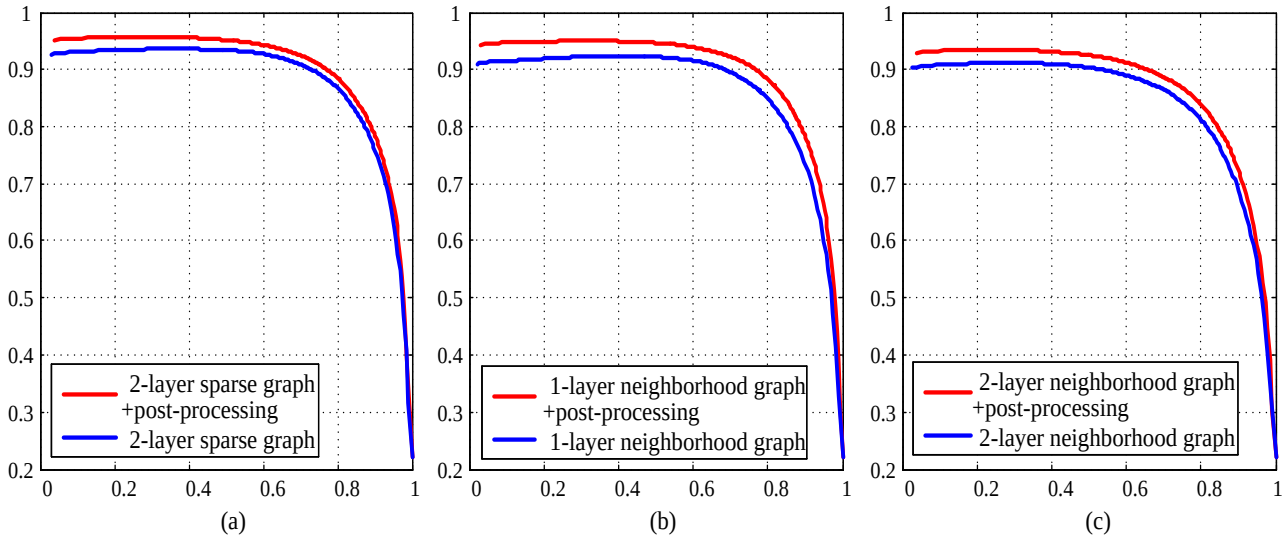


Fig. 16. Influence of the proposed graph-based post-processing on the MSRA10K dataset. (a) Precision-recall curves of a 2-layer sparse graph with and without post-processing. (b) Precision-recall curves of a 1-layer neighborhood graph with and without post-processing. (c) Precision-recall curves of a 2-layer neighborhood graph with and without post-processing.

6. The relationship between deep learning based algorithms and our work

As shown in Fig. 17, of the salient region detection algorithms include the following main steps: pre-processing, feature representation, saliency computation, and post-processing. The deep learning-based algorithms [5-7] focus on the feature representation step, whereas our proposed work focuses on the saliency computation and post-processing steps; that is, we are concerned with the different stages of salient region detection.

For the feature representation step, the deep learning-based algorithms usually use a deep convolutional neural network (CNN) to extract features. The feature extracted by CNN has a high dimension, which is usually several thousand or even tens of thousands. Our proposed work uses the Lab color space to present feature, and the extracted feature only with three dimensions.

For the saliency computation step, the deep learning-based algorithms usually train fully connected neural network layers, and the extracted CNN features are fed into these layers that are trained using a collection of labeled saliency maps. Our proposed work calculates the saliency value using a diffusion process on the proposed two-layer sparse graph.

Because the deep learning-based methods and our proposed method focus on different aspects of salient region detection, they complement each other well.

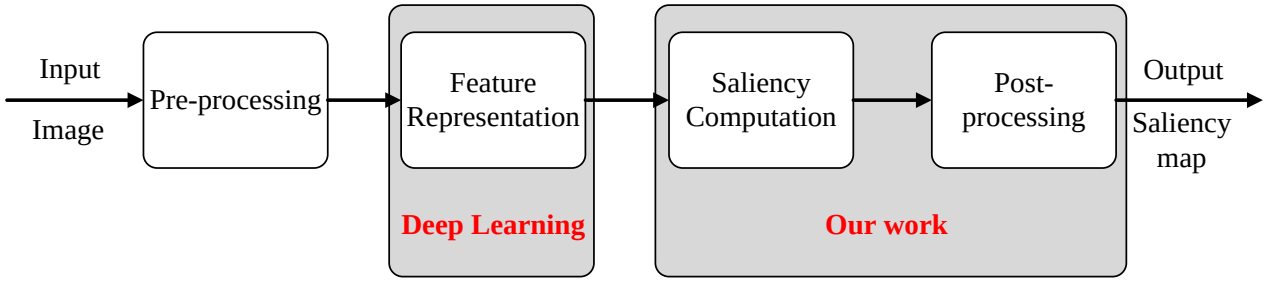


Fig. 17. Main steps of salient region detection.

7. Performance comparison with deep learning based method

Convolutional neural networks (CNN) have recently achieved many successes in visual recognition tasks, including image classification, object detection, and scene parsing. Razavian *et al*. [8] concluded that CNN-based deep learning can be a strong candidate for any visual recognition tasks.

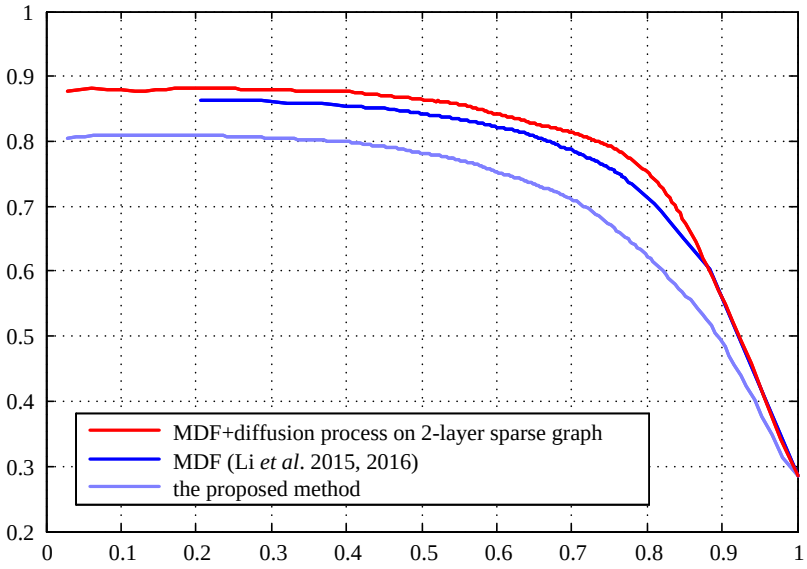
As shown in Fig 18, the deep learning based method [6, 7] performed better than our proposed method, but the proposed method has its own value.

First, deep learning based methods such as [6, 7] perform feature extraction using a CNN originally trained over the ImageNet dataset (this dataset contains 1.2 million images). The final feature extracted by CNN has a high dimension, which is usually several thousand or even tens of thousands. Thus, these deep learning based methods have a high demand for storage and computing performance. Our proposed work uses Lab color space to present feature, and the extracted feature only with three dimensions.

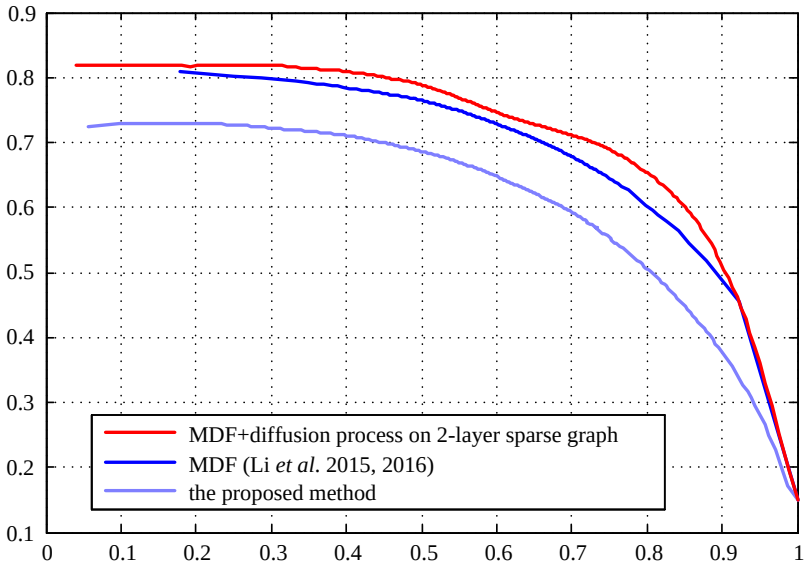
Second, for the saliency computation, the deep learning based algorithms usually train fully connected neural network layers, and the extracted CNN features are fed into these layers trained using a collection of labeled saliency maps. By contrast, our method is unsupervised and does not need manually annotated images. Without resorting to any supervised learning process, our method uses the properties of the diffusion process and some refinement techniques to capture saliency on a 2-layer sparse graph.

As a pre-processing procedure, saliency detection can be used for many vision tasks, such as visual tracking, object retargeting, image classification, object detection and recognition, image compression, and image segmentation. Because it is used for pre-processing, a saliency detection method is required that has faster computing speed and less computing resource consumption. Based on the aforementioned description, our proposed method is more suitable for this situation.

Third, because the deep learning based methods and our proposed method focus on different aspects of salient region detection, they complement each other well, which can be observed in Fig. 18. As shown in Fig. 18, the performance of the MDF method is greatly improved after we apply the proposed diffusion process to it.



(a) ECSSD dataset



(b) DUT-OMRON dataset

Fig. 18. Average precision-recall curves of the proposed method compared with the deep learning based method [6, 7]. (a) ECSSD dataset. (b) DUT-OMRON dataset.

Reference

[1] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*., 2013, pp. 3166-3173.

[2] B.-W. Jiang, L.-H. Zhang, H.-C. Lu, C. Yang, and M.-H Yang, "Saliency detection via absorb Markov chain," in *Proc. IEEE Int. Conf. Comput. Vis*., 2013, pp. 1665-1672.

[3] J. Sun, H. Lu, and X. Liu, "Saliency Region Detection based on Markov Absorption Probabilities," *IEEE Trans. Image Process*., vol. 24, no. 5, pp. 1639-1649, May 2015.

[4] L. Zhou, Z. Yang, Q. Yuan, Z. Zhou, and D. Hu, "Salient region detection via integrating diffusion-based compactness and local contrast," *IEEE Trans. Image Process*., vol. 24, no. 11, pp.3308-3320, Nov. 2015.

[5] L. Wang, H. Lu, R. Xiang, and M.H. Yang, "Deep networks for saliency detection via local estimation and global

search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2015, pp. 3183-3192.
[6] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2015, pp. 5455–5463.
[7] G. Li, and Y. Yu, "Visual Saliency Detection Based on Multiscale Deep CNN Features," *IEEE Trans. Image Process*., vol. 25, no. 11, pp. 5012-5024, 2016.
[8] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proc. IEEE Conf. CVPRW*, 2014, pp. 806–813.