

Received 12 December 2023, accepted 24 December 2023, date of publication 28 December 2023,
date of current version 17 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347812

RESEARCH ARTICLE

Saliency Detection via Manifold Ranking on Multi-Layer Graph

SUWEI WANG¹, YANG NING², XUEMEI LI¹, AND CAIMING ZHANG¹

¹School of Software, Shandong University, Jinan, Shandong 250101, China

²School of Computer Science and Technology, Shandong Jianzhu University, Jinan, Shandong 250101, China

Corresponding author: Caiming Zhang (czhang@sdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) Joint Fund With Zhejiang Integration of Informatization and Industrialization under Key Project U22A2033, and in part by NSFC under Grant 62072281.

ABSTRACT Saliency detection is increasingly a crucial task in the computer vision area. In previous graph-based saliency detection, superpixels are usually regarded as the primary processing units to enhance computational efficiency. Nevertheless, most methods do not take into account the potential impact of errors in superpixel segmentation, which may result in incorrect saliency values. To address this issue, we propose a novel approach that leverages the diversity of superpixel algorithms and constructs a multi-layer graph. Specifically, we segment the input image into multiple sets by different superpixel algorithms. Through connections within and connections between these superpixel sets, we can mitigate the errors caused by individual algorithms through collaborative solutions. In addition to spatial proximity, we also consider feature similarity in the process of graph construction. Connecting superpixels that are similar in feature space can force them to obtain consistent saliency values, thus addressing challenges brought by the scattered spatial distribution and the uneven internal appearance of salient objects. Additionally, we use the two-stage manifold ranking to compute the saliency value of each superpixel, which includes a background-based ranking and a foreground-based ranking. Finally, we employ a mean-field-based propagation method to refine the saliency map iteratively and achieve smoother results. To evaluate the performance of our approach, we compare our work with multiple advanced methods in four datasets quantitatively and qualitatively.

INDEX TERMS Manifold ranking, multi-layer graph, superpixel algorithm, saliency detection.

I. INTRODUCTION

Saliency detection can help people to find objects or regions that effectively represent the scene, therefore it has become a useful preprocessing step in complex vision problems, such as image segmentation [1], image edition [2], image classification [3], video compression [4], moving object detection [5], image correction [6], unmanned aerial vehicle detection [7], internet of surveillance thing [8], [9].

Over the past two decades, many saliency detection algorithms have been proposed. These methods can be divided into bottom-up methods and top-down methods. Bottom-up methods are data-driven and top-down methods are task-driven. The former attract much attention from vision

researchers and are widely used in previous saliency detection works. Bottom-up saliency detection work can be traced back to Itti's biologically inspired model [10]. He presented a central-surround difference operator to simulate human vision and compute the saliency of each pixel. Due to a lack of guidance from high-level knowledge, bottom-up methods require taking advantage of some perceptual priors, such as contrast prior [10], sparsity prior [11], background prior [12], center prior [13], compactness prior [11], smoothness prior and so on.

Recently, some successful graph-based saliency detection methods have been introduced [14], [15], [16]. These methods represent images as graphs and leverage the relationships and topological information within the graphs to enhance model performance. Nodes in the graph represent pixels or regions of the image, and edges are usually obtained

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato¹.

by connecting neighboring nodes in the space. Manifold Ranking is a graph-based semi-supervised learning method that maps the graph into a manifold space, aiming at discovering the streaming structure in the data. It exploits the connections and similarities between nodes in the graph by iteratively propagating labeling information between labeled and unlabeled nodes and finally assigning labels to previously unlabeled regions. Saliency detection based on manifold ranking assumes that neighboring nodes with similar appearances, i.e., belonging to the same manifold structure, exhibit similar saliency. Initial labeled seed nodes are selected based on heuristic assumptions. In the process of labeling propagation, the graph structure plays a pivotal role. It is crucial to construct a graph structure that can truly reflect the characteristics of the manifold space. This challenge necessitates a comprehensive consideration of data distribution under various circumstances and the construction of graphs to ensure the effective propagation of saliency values throughout the graph. Given scenarios where objects of the same class are scattered in different regions of the image, or where the appearance of an object is not smooth, resulting in the most similar superpixels not being spatially adjacent, just connecting neighboring nodes is insufficient to realistically represent the manifold structure in the image. Consequently, it fails to propagate saliency values effectively.

As the basic processing unit of a graph node, the shape of superpixels has a significant impact on graph-based saliency detection methods. Superpixels aggregate pixels with the same semantics into a single unit, significantly reducing the number of units to be processed, and can improve computational efficiency when performing saliency detection, e.g., [12], [15], [17]. Also, by combining spatially linked pixels into a single unit, superpixels reduce the effect of noise in the image, resulting in a smoother saliency map. Superpixel algorithms typically balance feature similarity with structural compactness. On the one hand, algorithms try to ensure that pixels within a superpixel are similar in feature space. On the other hand, algorithms also ensure that the superpixels are more structurally compact to represent the object accurately. The compactness of superpixels can lead to more regular shapes and similar sizes, but this usually diminishes the impact of feature similarity and results in missegmentation where regions unrelated to the semantic content of the superpixel are included, which can impair the final saliency detection performance. Segmentations produced by different superpixel algorithms can significantly differ in shape and distribution. Consequently, graphs constructed using superpixels as nodes exhibit distinct topological structures, significantly impacting the propagation of label information and ranking results. We have observed that saliency detection methods perform better when superpixels are characterized by regular shapes and similar sizes. However, when the hyperpixels are regular in shape and similar in size, they

are prone to missegmentation. We would like to capitalize on the differences in shape and distribution of superpixel segmentations. The design of such a collaborative approach needs to consider how to allow the nodes obtained from different superpixel algorithms to interact with each other. The design of such a collaborative approach needs to consider how to allow the nodes obtained from different superpixel algorithms to interact with each other.

As previously mentioned, the graph structure has a profound impact on the label propagation process. In this paper, we introduce a manifold ranking-based saliency detection method built upon a multi-layered graph. Constructing a multi-layered graph involves two key steps. Firstly, we construct each layer of the graph independently. To ensure that the graph reflects the true structure of the manifold space, we connect not only neighboring nodes in the spatial domain but also establish connections between neighboring nodes in the feature space. This additional connectivity allows for a more effective propagation of saliency values within image regions with the same semantics, thereby enhancing the completeness of detected salient objects. Subsequently, we establish connections between single-layer graphs to facilitate information exchange among different sets of superpixels. Due to significant differences in shape and distribution resulting from various superpixel segmentation algorithms, some algorithms generate superpixels that are regular in shape, and similar in size, but prone to over-segmentation. Others produce more accurate segmentations with fewer over-segmentation errors but generate superpixels with irregular shapes and significant size variations. We aim to fully leverage the complementary advantages of these two scenarios, enabling them to collaborate to enhance saliency detection. To achieve this goal, we employ edges to connect intersecting superpixels from different superpixel sets and calculate edge weights adaptively. These interlayer edges link superpixels located at the same positions in different graphs, ensuring consistency in saliency values. By constructing the multi-layered graph, saliency values can be propagated across different superpixel segments. Our method operates at a single scale to simultaneously reduce the computational cost of using multiple superpixel algorithms.

When calculating edge weights, we choose to measure the similarity between superpixels using both color and texture features. After constructing the graph, we utilize the manifold ranking algorithm for saliency detection, which consists of two stages. In the first stage, based on the background prior, we select superpixels along the image borders as background seeds and compute the correlation between each superpixel and the background seeds as the background probabilities. In the second stage, we threshold the results from the first stage to select foreground seeds and calculate the correlation between each superpixel and the foreground seeds to determine foreground probabilities, which indicate the saliency of each superpixel. Finally, we iteratively

refine the obtained saliency map to achieve the ultimate result.

The contributions of this paper are summarized as follows:

- 1) To mitigate the significance brought by superpixel segmentation errors, We take advantage of the complementary superpixel algorithms and construct a multi-layer graph to propagate saliency among superpixels segmented by different superpixel algorithms.
- 2) In a single-layer graph, we connect both neighbors in spatial space and feature space. Thus the propagation path between superpixels is expanded and the saliency can propagate globally and locally.
- 3) We propose a framework based on a multi-layer graph and improve the results significantly. Especially, the constructed multi-layer graph can be easily integrated into existing graph-based salient object detection works.

II. RELATED WORK

Many graph-based saliency detection methods have been proposed and have achieved great success in the past few years. In this section, we review these works briefly. The framework of the graph-based method generally includes graph construction, computation of similarity matrix, seed selection, and saliency propagation.

Some graph models have been successfully applied for saliency propagation. [14] constructs a pixel-level fully connected graph and uses the equilibrium state of Markov chains to calculate saliency maps. [15] proposes a two-stage graph model based on manifold ranking [18], [19]. [20] brings a novel view of the working mechanism of the diffusion process and promotes each individual diffusion before integration. [21] calculates a foreground probability and a background probability of each node on absorbing Markov chain and fuse them by cosine similarity measurement method. Since graph-based methods usually get smooth results, they can be treated as the post-processing of other methods, such as [22], [23], [24].

Since saliency propagates along edges in the graph, a well-constructed graph allows effective propagation. [25] connects every superpixel to boundary nodes and computes a dense similarity matrix based on geodesic distance. In [26], a four-layer graph is constructed using multi-scale segmentation, and a third rank is executed with obtained foreground probability as features. [27] substitutes the conventional k -regular graph with an adaptive irregular graph and proposes a new seeding strategy. [28] considers regionally spatial consistency, and connects all potential foreground nodes and background nodes respectively. [29] propose a multi-graph-based manifold ranking propagation framework to obtain a coarse map, where the edges of each graph and weights of edges are computed by color space and location space respectively.

Features are used to measure similarities of nodes, and various features have been integrated into the saliency

detection model. [30] uses filtered edge information to locate salient objects and converts the location information into foreground probability features. [31] learns a transition probability matrix with deep features and uses absorbed time in the Markov chain to represent saliency value. [32] integrates low-level and high-level deep features and learns a set of hyperparameters to distinguish the importance between different features. To learn a more adaptive similarity matrix, some linear models are used to simulate the local manifold structure. [33] adopts the locally linear embedding (LLE) scheme to guide the propagation process. [34] uses a local linear regression model to simulate the local manifold structure and fuses two different level deep feature metrics through cross-diffusion. [35] learn a joint affinity matrix based on low-rank representation using multiple appearance features and then compute diffusion-based compactness as saliency values.

Seeds selection provides the initial saliency value of nodes. Image boundary regions are usually regarded as initial background seeds. Considering the boundary regions may include salient regions, [36] removes the boundary which has the biggest difference with other boundaries from seeds. [37] constructs a superpixel-level graph with the addition of a virtual background node representing the global information. [38] compute the likelihood of each boundary superpixel belonging to backgrounds and propose a two-stage detection algorithm by combining complementary similarity metrics.

Previous methods usually use a single-layer graph for saliency detection. Inspired by the complement of different superpixel algorithms, we construct a multi-layer graph and each layer corresponds to one superpixel algorithm. In every layer, we fuse two k -regular graphs which are constructed in spatial space and feature space.

III. METHODOLOGY

In this section, we detail the proposed algorithm. Our method can be divided into four parts, including graph construction, foreground query, background query, and iterative propagation optimization. The framework of our method is shown in Fig.1. First, Our method segments the input image I into superpixel sets with different superpixel algorithms and constructs a multi-layer graph according to the rule sets. Then features are extracted and similarities between superpixels are measured. Once the graph is constructed, a two-stage manifold ranking resolves saliency. In the first stage, the background seeds are selected according to background priors, and the background probability of each superpixel is calculated by ranking superpixels. Since the summation of background probability and foreground probability equals one, low background probability means high foreground probability. In the second stage, our method segments the background probability map got in stage one to obtain foreground seeds, and ranks superpixels to calculate the saliency of each superpixel. Finally, our method optimizes the obtained saliency map by propagation to get the final result.

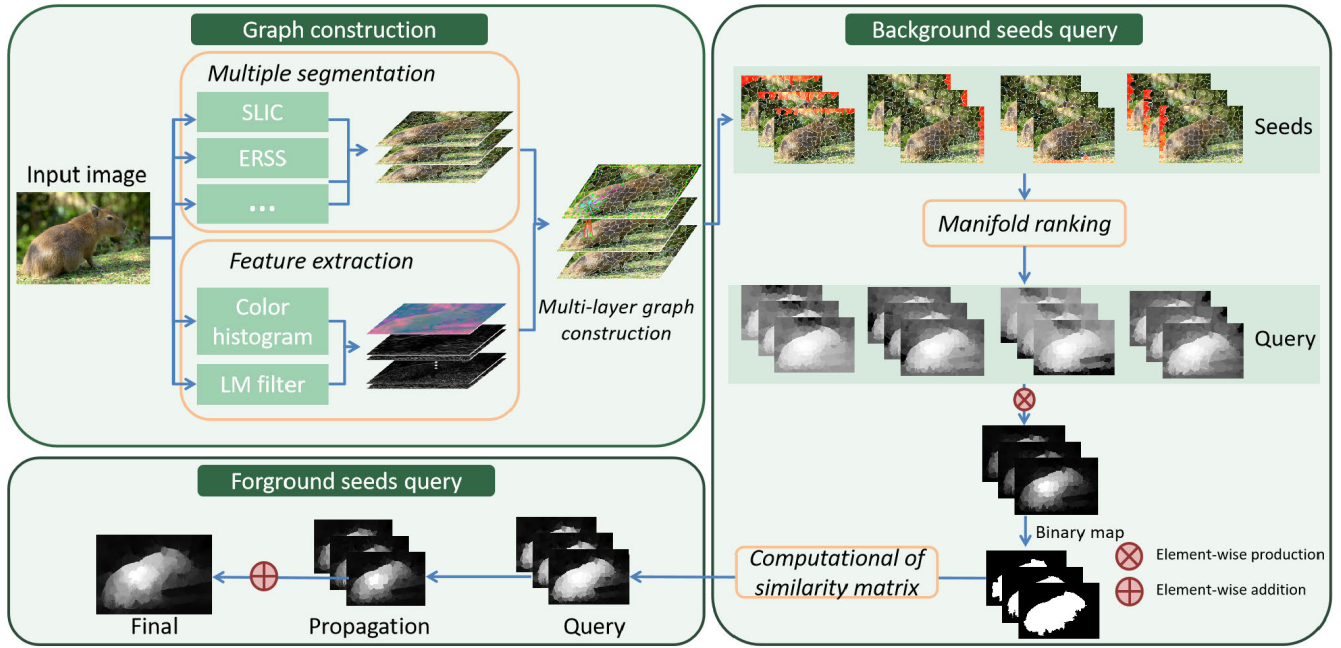


FIGURE 1. The framework of our method consists of three parts, including the construction of the multi-layer graph, the query of the foreground seeds and the query of the background seeds.

A. CONSTRUCTION OF MULTI-LAYER GRAPH

1) GRAPH MODEL

Given an image I , we first segment it into L superpixel sets with different algorithms. let S^l denotes the superpixel set got by the l th superpixel algorithm. Then our method treats each superpixel s_i^l as a node, and constructs an undirected graph $G = (V, E)$ according to given connection rules, where V represents the set of nodes and E represents the set of edges. Connection rules will unfold in detail later. For each set S^l , an adjacency matrix A^l can be constructed to describe the connection relationships between nodes. If superpixel s_i^l and superpixel s_j^l are directly adjacent, $A_{ij}^l = 1$, otherwise $A_{ij}^l = 0$.

The edges used in previous methods are usually composed of three parts, which are 1) edges connecting each superpixel and its neighbors, 2) edges connecting each superpixel and its 2-hop neighbors, and 3) edges connecting superpixels at the boundary. These rules have obvious limitations and can fail in some circumstances. For example, when the distribution of objects that belong to one class in an image is scattered, saliency values can not diffuse between separate objects, such as the flowers in the first line of Fig.2. The other case is that the distance of adjacent belonging to one object may not be the closest when the appearance features of the object are unevenly distributed, such as the leaves in the second line of Fig.2, which results in unsmooth saliency values in one object. To deal with the above two cases, our method connects each superpixel with its k nearest neighbors in feature space. Finally, our method connects different sets of overlapping superpixels, utilizing the complementarity between different superpixel algorithms to mitigate the imprecision of results

from a single segmentation scheme. Overall, our method uses five rules to construct edges. In the upper left corner of Fig.1, we show these five connection rules with two examples of superpixel segmentation. The above five rules are denoted by R_1, R_2, R_3, R_4 and R_5 :

$$\begin{aligned}
 R_1 : \varepsilon_1 &= \left\{ (s_i^l, s_j^l) \mid A_{ij}^l = 1 \right\} \\
 R_2 : \varepsilon_2 &= \left\{ (s_i^l, s_j^l) \mid A_{ir}^l = 1, A_{rj}^l = 1 \right\} \\
 R_3 : \varepsilon_3 &= \left\{ (s_i^l, s_j^l) \mid s_i^l, s_j^l \in B^l \right\} \\
 R_4 : \varepsilon_4 &= \left\{ (s_i^l, s_j^l) \mid s_j^l \in K_i^l \right\} \\
 R_5 : \varepsilon_5 &= \left\{ (s_i^m, s_j^n) \mid s_i^m \cap s_j^n \neq \emptyset, m \neq n \right\} \quad (1)
 \end{aligned}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$ are edge sets corresponding to five rules described above. B^l is the boundary superpixel sets which are selected from S^l . K_i^l is the k nearest neighbors of s_i^l in feature space. R_1 and R_2 connect each superpixel with its spatial neighbors, which ensures the connectivity of the graph. Since adjacent superpixels have large probabilities to belong to one object, they are more likely to have the same saliency values. R_1 and R_2 can ensure that superpixels belonging to one object get smooth saliency values. It is observed that the background regions may be scattered on different image boundaries, and R_1 and R_2 are not enough to propagate saliency effectively values between background regions on different boundaries. To solve this problem, our method connects superpixels on all boundaries according to R_3 . After connecting superpixels on the boundaries, superpixels in the image center have the longest shortest

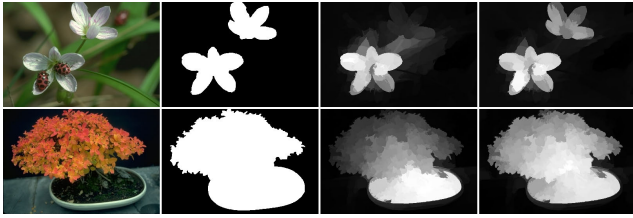


FIGURE 2. Comparison between results without R_4 and with R_4 . From left to right: input image, ground truth, result without R_4 , result with R_4 .

path to every boundary among all superpixels, which means R_3 satisfies the central prior in fact. R_4 connects each superpixel and its k neighbors in feature space so that connections are not limited to local connections. It can be seen from Fig.2 that when salient objects and backgrounds are scattered, the addition of R_4 improves the visual effect. R_5 connects overlapping superpixels in different sets. This connection guarantees that pixels that have the same location in different sets could obtain close saliency values. Besides, two regions that are not neighbors in one superpixel set may be neighbors in another superpixel set. Thus the introduction and connection between different superpixel methods can diversify the propagation path between two regions, which can improve the diffusion efficiency.

B. COMPUTATION OF SIMILARITY MATRIX

1) FEATURE EXTRACTION

Saliency detection can be seen as a classification problem, and the ideal features should minimize intra-class variation while maximizing inter-class variation. As the most commonly used feature, color means can bring a good classifying effect, although it is simple to extract. However, due to the loss of detail information, it is difficult for color means to deal with complex situations such as textures and gradations. Color histogram contains the distribution information of colors, thus retaining more detail. Therefore we use color histogram instead of color means. To further reduce the loss of detail, our method uses the Leung-Malik(LM) filter library to extract texture features. The LM filter library consists of 48 filters with multi-scale and multi-direction and has good application effects in image processing. Let h_i^{lc} and h_i^{lt} denote color histogram and texture histogram respectively of s_i^l . Distance d_{ij}^l between superpixels s_i^l and s_j^l is a weighted summation which can be seen in (2):

$$d_{ij}^l = \lambda_1 X^2(h_i^{lc}, h_j^{lc}) + \lambda_2 X^2(h_i^{lt}, h_j^{lt}) \quad (2)$$

where λ_1, λ_2 are control parameters, and $X^2(\cdot)$ represents the chi-square distance. The distance d_{ij}^l is normalized to $[0, 1]$. Fig.3 compares the results computed by two features separately and their weighted summation. It shows that when the salient object and the background have close colors, the combination of these two features can accurately find the salient object.



FIGURE 3. Comparison between results using different features. From left to right: input image, ground truth, results with color means, color histogram, and LM filter histogram respectively, and results with a weighted combination of the color histogram and LM filter histogram.

We construct a combined similarity matrix $W \in R^2$ to adapt to multiple superpixel sets. W is composed of two kinds similarity matrix, $W^l \in R^2$ and $W_{cl}^{lm,ln} \in R^2$. W^l characterizes the similarities between superpixels in set S^l . $W_{cl}^{lm,ln}$ describes the similarities between superpixels from different set S^{lm} and S^{ln} . Firstly, how to construct the internal similarity matrices of one superpixel set is described as follows. If there is an edge between s_i^l and s_j^l , the similarity is an exponential function of their distance and if there is no edge, the similarity is zero. As mentioned above, $\varepsilon_1^l, \varepsilon_2^l, \varepsilon_3^l$, and ε_4^l connect superpixels inner S^l . We split these four rules into two group E^{lb} and E^{lk} . E^{lb} is consisted of $\varepsilon_1^l, \varepsilon_2^l$, and ε_3^l . Let W^{lb} and W^{lk} denote similarity matrices corresponding to E^{lb} and E^{lk} separately. W_{ij}^{lb} is calculated as follows:

$$W_{ij}^{lb} = \begin{cases} \exp\left(-\frac{d_{ij}^l}{\sigma_1^l}\right) & \text{if } [s_i^l, s_j^l] \in E^{lb}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Since $W_{ij}^{lb} = W_{ji}^{lb}$, W^{lb} is a symmetric matrix. Similarly to (3), we compute W^{lk} . Then W^l can be get by adding W^{lb} and W^{lk} :

$$W^l = W^{lb} + W^{lk} \quad (4)$$

According to (4), the similarity matrices W^1, \dots, W^L are constructed.

Next, how to construct the similarity matrices between superpixel sets is described. According to rule R_5 , if s_i and s_j overlap, then an edge is connected between them, and the weight $W_{cl}^{lm,ln}(s_i^{lm}, s_j^{ln})$ is set to the mean of ratios of the overlapping areas:

$$W_{cl}^{lm,ln}(s_i^{lm}, s_j^{ln}) = \left(\frac{\text{area}(s_i^{lm} \cap s_j^{ln})}{\text{area}(s_i^{lm})} + \frac{\text{area}(s_i^{lm} \cap s_j^{ln})}{\text{area}(s_j^{ln})} \right) / 2 \quad (5)$$

According to (5), When calculating the similarity matrices between superpixel sets, the symmetry of the similarity matrices is guaranteed by taking the mean. Then, the combined similarity matrix W is obtained by splicing two

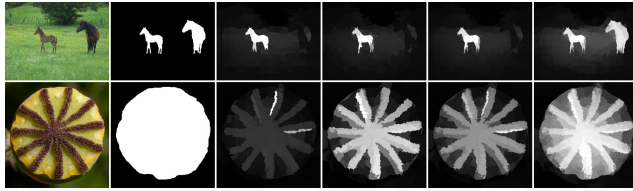


FIGURE 4. Comparison between results using different superpixels. From left to right: input image, ground truth, results of SLIC and ERSS respectively, averaging results of SLIC, ERSS, and results with inter-layer connections added.

kinds of similarity matrices that are constructed above as follows:

$$W = \begin{bmatrix} W^1 & \dots & W_{cl}^{1,L} \\ \vdots & \ddots & \vdots \\ W_{cl}^{L,1} & \dots & W^L \end{bmatrix} \quad (6)$$

Note to let $W_{ii} = 0$ to avoid self-inforcement. If we set W_{cl}^{1,l_2} to zero, the connections between different superpixel sets will be disconnected and our method will degrade to computing results with superpixel algorithms separately and then averaging these results directly. Fig.4 compares the results of different superpixel algorithms, directly averaged version, and our combination strategy. It is observed that saliency values are more smooth and salient regions are more complete with our combination strategy.

C. COMPUTATION AND PROPAGATION OF SALIENCY MAP

Saliency detection divides an image into foreground and background. If some superpixels are known to be background, then the background probabilities of other superpixels can be evaluated by their correlations to known background superpixels. Similarly, if some superpixels are known to be foreground, then the foreground probability of other superpixels can be obtained, which can represent saliency values. The process to obtain a salient map is divided into two stages. In stage one, the background probability is calculated by using background prior to selecting background seeds. In the second stage, the background probability map obtained in the first stage is segmented to select the foreground seeds, and the saliency values of each superpixel are obtained. We use manifold ranking to calculate correlations between labeled data and unlabeled data. Then, the resulting saliency values are further propagated to smoother saliency maps. Lastly, we integrate these maps and get the final saliency map.

1) QUERY THROUGH BACKGROUND PRIOR

According to the background prior, our method selects boundary superpixels as the background seeds. Then manifold ranking is used to rank correlations between other superpixels and seeds. Let F denote the needed background

probabilities, the cost function in this stage is:

$$F = \arg \min_F \left(\sum_{ij=1}^z W_{ij} \|f_i - f_j\|^2 + \mu \sum_{i=1}^z d_i \left\| f_i - \frac{Y_i^{bg}}{d_i} \right\|^2 \right) \quad (7)$$

where $F = [f_1, f_2, \dots, f_z]^T$ is the desired background probability, z is the number of superpixels, $d_i = \sum_j W_{ij}$, which is the sum of the similarities between s_i and its neighbors. Let Y_{bg} denotes background seeds indication vector, where $y_i^{bg} = 1$ if s_i belongs to image boundary, otherwise $y_i^{bg} = 0$. The cost function consists of two items. The first term is a binary smoothing term, which punishes the inconsistent value of connected superpixels. The weight is the similarity of the two connected superpixels, which ensures that connected similar superpixels obtain consistent saliency values. The second item is a unary retention term, which punishes the case that the ranking result is inconsistent with the indication vector Y_{bg} , which ensures values of superpixels that have more support from their neighbors will not change a lot. (7) is a convex optimization problem and the optimal solution is:

$$F = (D - \alpha W)^{-1} Y_{bg} \quad (8)$$

where D is a diagonal matrix, and the diagonal elements are d_i . $\alpha = \frac{1}{1+\mu}$. Let Y_{bg}^l denote seed indication vector corresponding to l th superpixel sets, concatenated seed indication vector $Y_{bg} = [Y_{bg}^1, \dots, Y_{bg}^L]$. After obtaining the background probability of each superpixel, we normalize it to $[0, 1]$, and saliency values F_{bg} can be obtained by subtracting background probabilities from 1:

$$F_{bg} = 1 - F \quad (9)$$

Since superpixels located on different boundaries usually belong to different objects, selecting all boundary superpixels as background seeds simultaneously will lead to high variance inside seeds, which results in inaccurate background probability results. It is observed that the variance between the superpixels in one boundary is much smaller, so our method computes saliency values using each boundary respectively. According to (7) and (9), $F_{bg}^t, F_{bg}^b, F_{bg}^l, F_{bg}^r$ corresponding upper, lower, left and right boundaries are calculated. F_{bg}^l is obtained by multiply these four results: $F_{bg} = F_{bg}^t F_{bg}^b F_{bg}^l F_{bg}^r$.

2) QUERY THROUGH FOREGROUND SEEDS

In this stage, our method segments result F_{bg} obtained in the first stage to select foreground seeds, and then the saliency values are obtained by computing the correlations between all superpixels and foreground seeds. The segmentation threshold is set to $\overline{F_{bg}}$ which is the mean of F_{bg} , and superpixels that have higher or equal values than the threshold are selected as foreground seeds. The indication vector Y_{fg} is

constructed as follows:

$$y_i^{fg} = \begin{cases} 1 & \text{if } f_i^{bg} \geq \overline{F_{bg}} \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Similarly to querying through background prior, the foreground probability is obtained by minimizing the cost function of manifold ranking. The optimal solution is:

$$F_{fg} = (D - \alpha W)^{-1} Y_{fg} \quad (11)$$

F_{fg} could be taken apart into $F_{fg}^1, \dots, F_{fg}^L$, which corresponds to different superpixel algorithms, according to the splicing order of W .

3) PROPAGATION OF SALIENCY

In order to get a smoother salient map, our method performs iterative propagation to adjust saliency values. Propagation matrix P is constructed first. The distance d_{ij}^l between adjacent superpixels s_i and s_j is calculated according to (12):

$$d_{ij}^l = \begin{cases} 1 - \exp\left(-\frac{d_{ij}}{\sigma_2^2}\right) & \text{if } A_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Then the shortest path d_{ij}^{\min} between any two superpixels are calculated. The similarity P_{ij} between s_i and s_j is calculated as follows:

$$P_{ij} = \exp\left(-\frac{d_{ij}^{\min}}{\sigma_3^2}\right). \quad (13)$$

Note to set elements on the diagonal elements of P to 0. Next iteratively propagation of F_{fg}^l is carried out. In order to simplify the expression, we have omitted the superscript l . Let $F_0 = F_{fg}$. Here, our method makes a slight modification on the commonly used mean field-based propagation (14) and obtains (15).

$$f_i^n = \delta f_i^{n-1} + (1 - \delta) \frac{\sum P_{ij} f_i^{n-1}}{\sum P_{ij}}. \quad (14)$$

Then after n iterations, the final result f_i^n is:

$$f_i^n = \delta f_i^{n-1} + (1 - \delta) \frac{\sum_{ij} P_{ij} \psi(f_i^{n-1} - thr) f_i^{n-1}}{\sum_{ij} P_{ij}}. \quad (15)$$

$$\psi(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

where δ is a weight parameter, and thr is a threshold parameter. $\psi(\cdot)$ is a positive operation that takes 0 for all values less than 0. (15) consists of two parts, the first part is the retention item, which ensures saliency values are consistent with the original value; the second part is the propagation term, which receives influences from similar superpixels. The second part can be regarded as adjusted saliency values, where thr takes a small value. Superpixels

whose values are below thr are considered to belong to the background. After adjustment, the contrast between background and foreground is increased. After assigning each pixel the value of the superpixel they belong to, propagated F_{fg} can be visualized to L saliency maps. The final saliency map SM is got by averaging these maps. The main steps of our methods are summarized in Algorithm 1.

Algorithm 1 Saliency Detection via Manifold Ranking on Multi-Layer Graph

Input: Input image I and parameters: λ_1 and λ_2 in (2), $\sigma_1^2, \sigma_2^2, \sigma_3^2$ in (3)(12)(13) respectively, μ in (7), δ and thr in (15), the number of nearest neighbors k and propagation iterations number t .

Output: Saliency map SM , each value indicating the saliency value.

- 1: Segment I using into superpixel sets $S = \{S^1, \dots, S^L\}$.
 - 2: Construct graph G using rules R_1, R_2, R_3, R_4 and R_5 , and compute similarity matrix W using (6).
 - 3: In stage one, construct background seeds indication vector Y_{bg} , compute saliency values according to (7), (9) using each boundary as seeds respectively and get saliency vectors $F_{bg}^t, F_{bg}^b, F_{bg}^l, F_{bg}^r$. Multiply these four vectors element-wise to get saliency vector F_{bg} .
 - 4: In stage two, construct foreground seeds indication vector Y_{fg} according to (10), and compute saliency vector F_{fg} using (7), and then split F_{fg} into $F_{fg}^1, \dots, F_{fg}^L$.
 - 5: Propagate $F_{fg}^1, \dots, F_{fg}^L$ using equation (15) by t times.
 - 6: Visualize adjusted saliency vectors to get a saliency map and average them to obtain the final result SM .
-

IV. EXPERIMENT

A. DATASETS

We experiment on four commonly used datasets, including ECSSD [39], MSRA10K [17], PASCAL-S [40] and DUTS [41]. These databases all have pixel-level labels. ECSSD consists of 1,000 images selected from the BSD dataset and every image contains at least one salient object. The MSRA10K contains 10,000 images randomly selected from the MSRA dataset and is an extension of MSRA1K. The PACSAL-S has been carefully designed to avoid design biases consists of 850 complex scene images and contains a small number of images without any salient objects. It is one of the most challenging datasets available. DUTS [41] is currently the largest saliency detection dataset. It is composed of two subsets DUTES-TR and DUTES-TE and. DUTES-TR is usually used to train deep models and contains 10553 images. DUTES-TE is usually used for testing and contains 5019 images.

B. EVALUATION CRITERIA

Evaluation criteria commonly used for saliency detection include precision-recall (PR) curve, F-measure curve, MaxF,

WF, MAE, and S-measure. Images can be segmented into salient regions and background regions with a threshold. Precision is defined as the area of the correctly detected salient region over the area of the detected salient region. The recall rate is defined as the ratio of the correctly detected salient region over the area of the ground truth. By changing the threshold from 0 to 255, we can get multiple pairs of precision and recall. The PR curve can be drawn with recall as the X-axis and precision as the Y-axis. F-measure is the weighted harmonic of precision and recall: $\frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$. By changing the threshold, we can get F-measure curves. MaxF is the max value of the FM curve. WF is obtained by setting β^2 to 0.3, which improves the influence of accuracy [42]. The larger the value of F-Measure, the better. MAE is the sum of squared differences between the saliency map and ground truth. The S-measure evaluates the structure similarity of the predicted saliency map and the ground truth. It takes object-aware structural similarity measure and region-aware structural similarity measure into consideration: $S = (1 - \alpha)S_r + \alpha S_o$, we set $\alpha = 0.5$ according [43].

C. PARAMETER SETTING

λ_1 and λ_2 in (2) are set to 0.4, 0.6 respectively. σ_1^2 , σ_2^2 , σ_3^2 in (3), (12), (13) are set to 0.1, which is consistent with [40]. μ in equation (7) is set to 0.1, which controls seed retention items and the boundary retention items respectively. μ in equation (7) is set to 0.1. δ and thr in (15) are set to 0.76 and 0.1, and the number of propagation iterations is 5. k is set to 3 by experience when selecting the nearest neighbors for R_4 . The superpixel algorithms chosen are SLIC, and ERSS [44].

D. COMPARISON WITH OTHER METHODS

We provide a quantitative and qualitative comparison between our method and high-level salient object detection methods including MR [15], DRFI [45], PISA [46], DSR [11], GraB [25], MILPs [47], EASD [30]2017, AMC_AE [31], CFSOD [48], GLJAF [35], LEGS [49], U2Net [50] to support the Contribution 3) mentioned in section I. Here MR, Grab, and EASD use a similar framework with us. The results are obtained by either running codes from paper homepages or downloading from paper homepages.

Fig.5 shows the PR curves and the F-measure curves of our method and other methods, where ‘ours’ is the result of our method. It can be seen that the PR curve of our method is superior to all other methods except for AMC_AE in different datasets. Our method even surpasses the supervised method DRFI, indicating that our method is very competitive. The F-measure curve of our method also surpasses methods that are based on graphs. Considering the complex scene of PASCAL-S, it is not surprising that AMC_AE surpasses our method by a large margin in PASCAL-S as it uses deep features. Table 1 compares the F-measure and MAE results. It can be seen that the F-measure of our method is improved,

which surpasses other methods other than AMC_AE. The MAE results are not very satisfactory due to the large area of low salient values in the background region. If the superpixels lie across the boundary between the object and the background, saliency will propagate from salient objects to backgrounds. How to deal with this phenomenon will be considered in our future work.

Fig.6 compares salient maps qualitatively. Examples are selected based on several tricky situations. The first and second rows contain two different salient objects. In the third and fourth rows, salient objects are partially obscured by the background. In the fifth row, the salient objects and the background have similar appearance properties. In the sixth and seventh rows, the salient objects are located on image boundaries. It can be seen that due to the simple background seed selection mechanism, although most of the salient regions are detected, the salient regions close to the boundary are not highlighted. In the sixth row, the background has complex textures. In the eighth and ninth lines, the background is composed of heterogeneous parts. In terms of visual contrast, the edge of the results of our method is kept well, and more complete salient objects are detected, and better results are obtained.

E. ABLATION ANALYSIS

In this section, we verify the effectiveness of each part by ablation analysis. Ablation analysis experiments on ECSSD. Experiments are divided into decremental experiments and incremental experiments. The effects of each part can be clearly viewed by contrasting these experimental results. We use MR [15] as the benchmark. Fig.7(a) shows the PR curves of the decremental experiments. ‘wo.MS’ means only SLIC is used. ‘wo.CL’ means that there is no edge connection between the superpixel sets. ‘wo.MF’ means that only the color mean is used as the feature. ‘wo.KNN’ means edges between k-nearest neighbor in feature space is not added to the graph structure. ‘wo.P’ means without propagation optimization. Fig.7(b) shows the PR curves of the incremental experiments. ‘w.MF’ means multiple features are used. ‘w.KNN’ means edges between k-nearest neighbor in feature space are added to the graph structure. ‘w.MS’ means multiple superpixel algorithms are used. ‘w.MS_CL’ means multiple superpixel algorithms are used and inter-layer connection are added. ‘w.P’ means (15) is used for adjustment. Here ‘wo.MS’, ‘wo.CL’, ‘wo.P’, ‘w.MS’, ‘w.CL’, and ‘w.P’ are used to support the Contribution 1) mentioned in section I, and ‘wo.MF’, ‘wo.KNN’, ‘w.MF’, and ‘w.KNN’ are used to support the Contribution 2) mentioned in section I.

As can be seen from Fig.7, the averaging results of multiple superpixel algorithms can bring an improvement as expected. Connections between superpixel sets further improve the performance by a large margin. We would design experiments to show performance variation of different superpixel combination strategies.

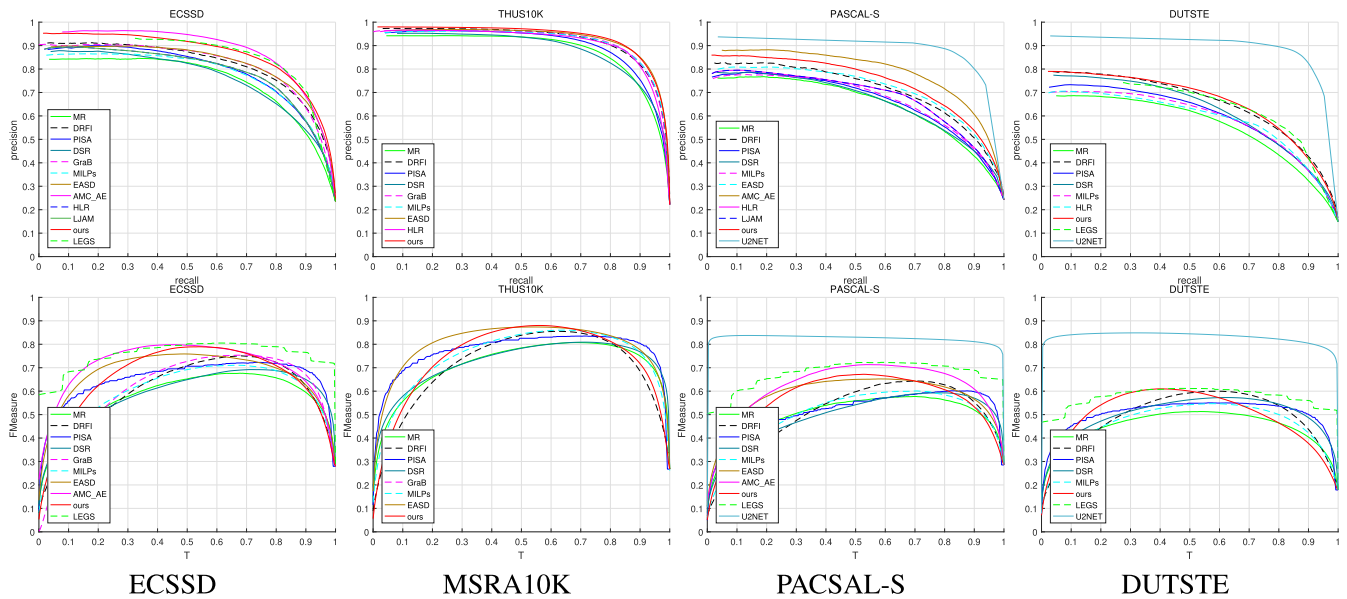


FIGURE 5. Quantitative comparison. The first and second rows are PR curves and F-measure curves. From left to right are results on ECSSD, MSRA10K, PASCAL-S, DUTSTE.

TABLE 1. Comparison of maxFM, WF, mae, and SM.

Dataset	Metric	MR	DRFI	PISA	GraB	MILPs	AA	HLR	LEGS	U2NET	Ours
ECSSD	MaxFm↑	0.677	0.751	0.723	0.754	0.711	0.798	0.725	0.805	-	0.789
	WF↑	0.691	0.732	0.725	0.748	0.713	0.784	0.719	0.783	-	0.773
	mae↓	0.186	0.17	0.157	0.168	0.177	0.168	0.172	0.118	-	0.168
	SM↑	0.692	0.732	0.719	0.734	0.718	0.775	0.742	0.786	-	0.763
THUS10K	MaxFm↑	0.807	0.855	0.835	0.859	0.855	-	0.857	-	-	0.88
	WF↑	0.833	0.846	0.845	0.864	0.864	-	0.858	-	-	0.869
	mae↓	0.126	0.126	0.108	0.124	0.11	-	0.104	-	-	0.122
	SM↑	0.785	0.822	0.81	0.813	0.827	-	0.847	-	-	0.837
PASCAL-S	MaxFm↑	0.577	0.643	0.602	-	0.601	0.713	0.625	-	0.838	0.672
	WF↑	0.581	0.61	0.593	-	0.599	0.691	0.612	-	0.763	0.645
	mae↓	0.229	0.207	0.198	-	0.217	0.195	0.208	-	0.074	0.211
	SM↑	0.625	0.658	0.62	-	0.644	0.713	0.669	-	0.844	0.687
DUTSTE	MaxFm↑	0.513	0.6	0.55	-	0.547	-	0.55	0.612	0.849	0.61
	WF↑	0.505	0.547	0.541	-	0.522	-	0.52	0.579	0.786	0.553
	mae↓	0.194	0.155	0.153	-	0.174	-	0.183	0.137	0.045	0.194
	SM↑	0.622	0.676	0.654	-	0.651	-	0.656	0.696	0.874	0.662

Fig.8(a) shows results of different superpixel algorithms, where ‘SLIC’, ‘ERSS’, ‘LSC’ [51], ‘EdgeBox’ [52], and ‘SEEDS’ [53] are the name of used superpixels algorithms. Among these algorithms, SLIC, EdgeBox, and SEEDS prefer regular shape, ERSS prefers edge retention, and LSC produces regular superpixels in smooth regions and irregular superpixels in texture regions. As shown by Fig.8(a), SLIC works best, which proves the effectiveness of SLIC.

To find out the optimal combination, we experiment with different combinations of superpixel algorithms. In Fig.8(b), we show results of different number and combination strategies, where legends without postfix ‘_CL’ means averaging results of different superpixel algorithms and

legends with postfix ‘_CL’ means inter-layer connections are added. Among combinations of two, three, and four algorithms, it is hard to say which combination is the best, so we select some representative combinations, which are ‘SLIC_ERSS’, ‘ERSS_SEEDS_EdgeBox’, ‘ERSS_LSC_SEEDS_EdgeBox’ respectively. As shown in Fig.8(b), ‘ERSS_SEEDS_EdgeBox’ works best in all combinations. Unexpectedly, it is not true that more superpixel algorithms bring better results. In summary, the use of three superpixel algorithms works best, which brings minor improvement over two superpixel algorithms. Four superpixel algorithms don’t bring any enhancements. Disappointingly, the use of five superpixel algorithms slightly

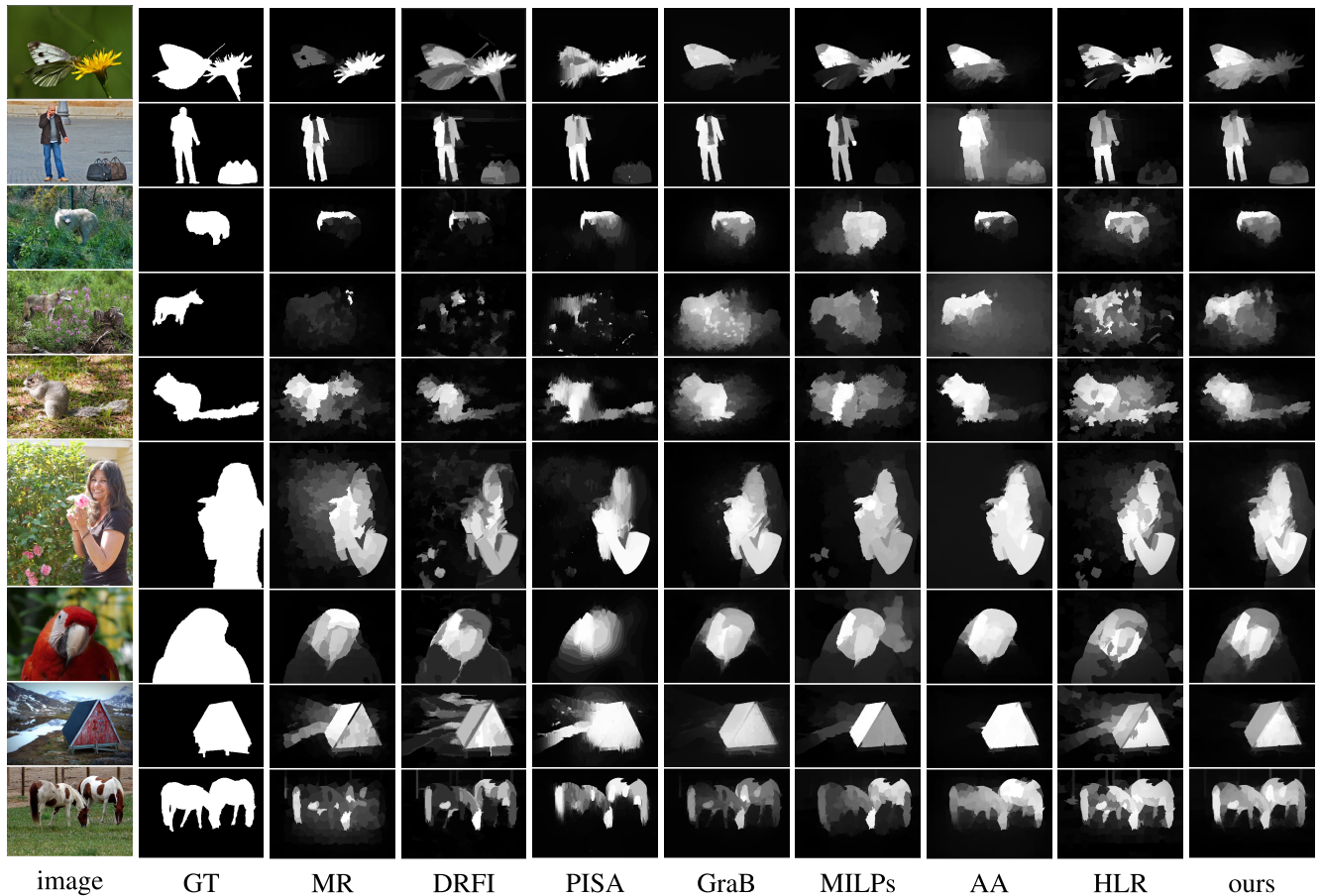


FIGURE 6. Qualitative comparison. Each row from top to bottom corresponds to one example image selected from ECSSD. The first and second columns are input image and ground truth, and the third to ten columns correspond to the results of MR, DRFI, PISA, GraB, MILPs, AA, and HLR respectively, and the last column is the result of our method.

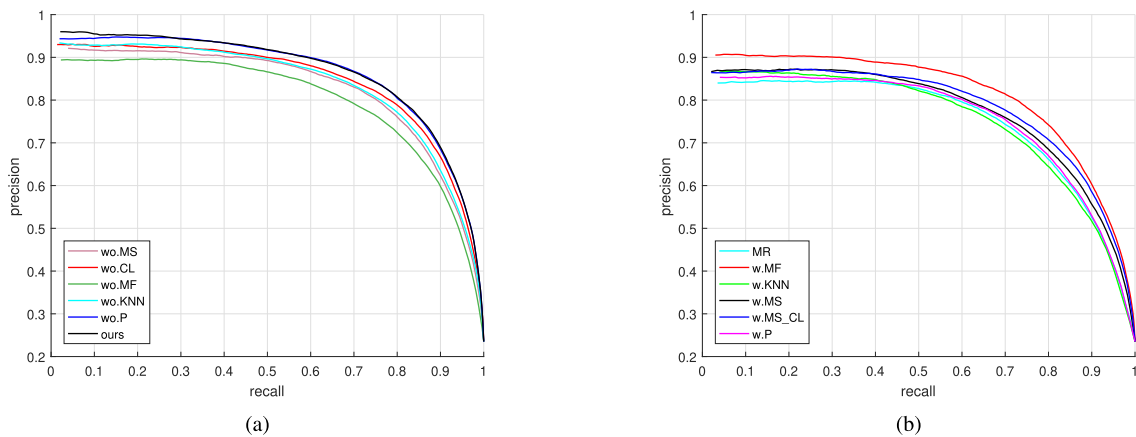


FIGURE 7. Ablation experiments on ECSSD. (a) Decremental experiments. (b) Incremental experiments.

degrades the effect. It is speculated that the used superpixel algorithms are not sufficiently diversified and there are much more algorithms preferring regular shapes. These algorithms with the same preference produce similar segmentation errors. Because wrongly segmented algorithms are more than rightly segmented algorithms, misclassified superpixels

receive fewer constraints from correctly segmented superpixels and more support from wrongly segmented superpixels. This kind of error can't be attenuated effectively. In fact, since superpixel algorithms weigh between consistency and compactness, two algorithms that are prone to compactness and consistency respectively would be enough. As mentioned

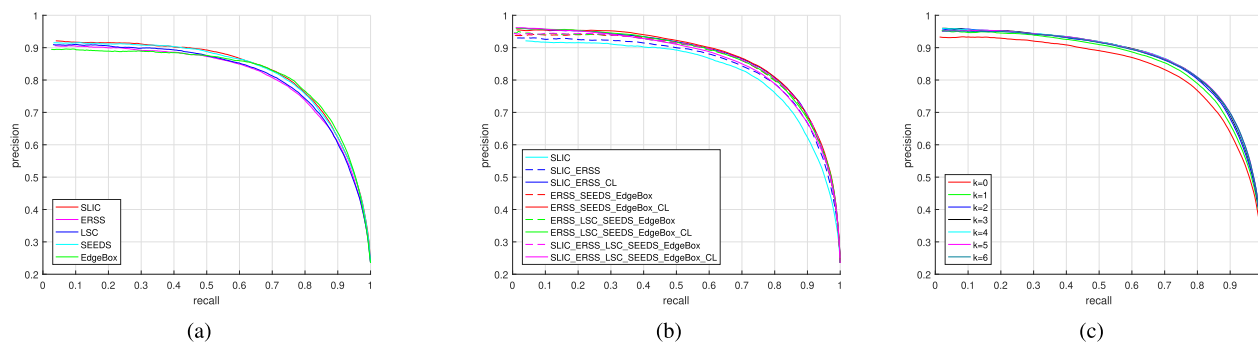


FIGURE 8. PR curves on ECSSD. (a) Different superpixel algorithms. (b) Different combinations of superpixel algorithms, where the solid lines are the results that use inter-layer connections, and the dashed lines are averaging results of different superpixel algorithms. (c) Different values for K in the K nearest neighbor connection.

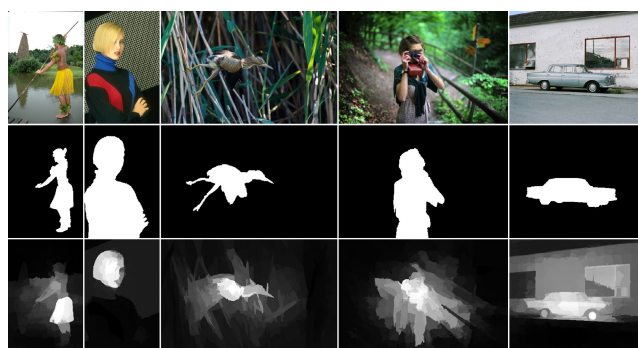


FIGURE 9. Failure cases. Rows from top to down are input images, ground truth, and the results of our method.

before, SLIC produces regular superpixels while ERSS produces irregular superpixels but preserves edges very well, and ‘SLIC_ERSS’ works best in the combination of two algorithms. In Fig.8(b), we also find that although averaging results of two algorithms are much worse than three, once inter-layer connections are added, combinations of two and three obtain similar results, which proves the effectiveness of the inter-layer connections.

In order to find out the optimal number of nearest neighbors, different values of k are used for experiments. Fig.8(c) shows the results when k varies from 0 to 6, 0 means that no similar superpixels are connected. As can be seen, when more neighbors are connected, the performance is first improved and then degraded. This is because when k takes a large value, superpixels that belong to different objects are likely to be connected mistakenly. In order to balance between good performance and lower computation cost, our method sets k to 3.

F. FAILURE CASES

Although our method gains great improvement, it still fails in some difficult images. Fig.9 shows some failure cases. In these images, some complex situations bring great challenges, including objects having similar appearances to the background, objects or backgrounds being composed of

heterogeneous parts, objects touching the image boundary, and backgrounds having complex textures and combinations. One way to cope with these tricky cases is optimizing background seeds and extracting more descriptive high-level features.

V. CONCLUSION

In this paper, a graph-based saliency object detection method is proposed, which improves the accuracy of detection by optimizing the graph structure. With the newly constructed multi-layer graph, saliency values diffuse across different superpixel sets, which integrates different superpixel algorithms naturally and attenuates the influence of segmentation errors that occur near the edge. Besides, by connecting k neighbors in feature space, saliency values can propagate globally and the obtained saliency map is smoother, and detected salient objects are more complete. We evaluated our method on several datasets and compared it with a variety of high-level methods qualitatively and quantitatively. Experimental results demonstrate the effectiveness of our method.

All previous methods use one superpixel algorithm to segment images at multiple scales. This means the multi-layer map constructed by our method is complementary with most existing graph-based methods, and the combination of the multi-layer map and existing methods is expected to improve results. In addition, there are two obvious flaws in our work. First, we use handcrafted features and it works not well enough in complex scenes. Manual features are usually low-level and lack of high-level semantic information and diversity, and deep features provide diverse features from low-level to high-level. Therefore, future work also includes the use of deep features and integrating multiple features adaptively. Secondly, we use all superpixels in the boundary as background seeds. If salient objects are located in the image boundary, it might be disastrous in such a simple way. So a background seeds filter will be of vital importance and it will be included in our future work. As mentioned before, how to deal with large areas of low salient background will be considered in our future work.

REFERENCES

- [1] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 3796–3805.
- [2] Y. Chen, Y. Pan, M. Song, and M. Wang, "Improved seam carving combining with 3D saliency for image retargeting," *Neurocomputing*, vol. 151, pp. 645–653, Mar. 2015.
- [3] J.-Y. Zhu, J. Wu, Y. Xu, E. Chang, and Z. Tu, "Unsupervised object class discovery via saliency-guided multiple class learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 862–875, Apr. 2015.
- [4] H. Hadizadeh and I. V. Bajic, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.
- [5] Y. Pang, L. Ye, X. Li, and J. Pan, "Incremental learning with saliency map for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 640–651, Mar. 2018.
- [6] J. Li, X. Feng, and H. Fan, "Saliency-based image correction for colorblind patients," *Comput. Vis. Media*, vol. 6, no. 2, pp. 169–189, Jun. 2020.
- [7] L. Chen, G. Liu, Y. Tan, Z. Sun, H. Ge, F. Duan, and C. Zhu, "A UA-net based salient object detection method for UAV," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2021, pp. 1662–1667.
- [8] Z. Gao, C. Xu, H. Zhang, S. Li, and V. H. C. de Albuquerque, "Trustful Internet of surveillance things based on deeply represented visual co-saliency detection," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4092–4100, May 2020.
- [9] J. Zhang, C. Xu, Z. Gao, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Industrial pervasive edge computing-based intelligence IoT for surveillance saliency detection," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5012–5020, Jul. 2021.
- [10] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
- [11] H. Lu, X. Li, L. Zhang, X. Ruan, and M.-H. Yang, "Dense and sparse reconstruction error based saliency descriptor," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1592–1603, Apr. 2016.
- [12] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.
- [13] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.
- [14] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 545–552.
- [15] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 3166–3173.
- [16] X. Ma, T. Zhao, Q. Guo, X. Li, and C. Zhang, "Fuzzy hypergraph network for recommending top-K profitable stocks," *Inf. Sci.*, vol. 613, pp. 239–255, Oct. 2022.
- [17] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Aug. 2015, vol. 37, no. 3, pp. 569–582.
- [18] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Scholkopf, "Ranking on data manifolds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 169–176.
- [19] X. Ma, X. Li, W. Feng, L. Fang, and C. Zhang, "Dynamic graph construction via motif detection for stock prediction," *Inf. Process. Manage.*, vol. 60, no. 6, Nov. 2023, Art. no. 103480.
- [20] P. Jiang, Z. Pan, C. Tu, N. Vasconcelos, B. Chen, and J. Peng, "Super diffusion for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 2903–2917, 2020.
- [21] W. Zhang, Q. Xiong, W. Shi, and S. Chen, "Region saliency detection via multi-feature on absorbing Markov chain," *Vis. Comput.*, vol. 32, no. 3, pp. 275–287, Mar. 2016.
- [22] Q. Zhang, Z. Huo, Y. Liu, Y. Pan, C. Shan, and J. Han, "Salient object detection employing a local tree-structured low-rank representation and foreground consistency," *Pattern Recognit.*, vol. 92, pp. 119–134, Aug. 2019.
- [23] Q. Zhang, J. Lin, W. Li, Y. Shi, and G. Cao, "Salient object detection via compactness and objectness cues," *Vis. Comput.*, vol. 34, no. 4, pp. 473–489, Apr. 2018.
- [24] X. Ma, X. Li, Y. Zhou, and C. Zhang, "Image smoothing based on global sparsity decomposition and a variable parameter," *Comput. Vis. Media*, vol. 7, no. 4, pp. 483–497, Dec. 2021.
- [25] Q. Wang, W. Zheng, and R. Piramuthu, "GraB: Visual saliency via novel graph model and background priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 535–543.
- [26] L. Zhang, C. Yang, H. Lu, X. Ruan, and M.-H. Yang, "Ranking saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1892–1904, Sep. 2017.
- [27] Y. Zhou, T. Zhang, S. Huo, C. Hou, and S.-Y. Kung, "Adaptive irregular graph construction-based salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1569–1582, Jun. 2020.
- [28] Y. Liu, J. Han, Q. Zhang, and L. Wang, "Salient object detection via two-stage graphs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1023–1037, Apr. 2019.
- [29] Y. Lu, K. Zhou, X. Wu, and P. Gong, "A novel multi-graph framework for salient object detection," *Vis. Comput.*, vol. 35, no. 11, pp. 1683–1699, Nov. 2019.
- [30] H. Yang and W. Li, "Edge-aware saliency detection via novel graph model," in *Advances in Multimedia Information Processing—PCM 2017* (Lecture Notes in Computer Science). Harbin, China: Springer-Verlag, 2017, pp. 45–55.
- [31] L. Zhang, J. Ai, B. Jiang, H. Lu, and X. Li, "Saliency detection via absorbing Markov chain with learnt transition probability," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 987–998, Feb. 2018.
- [32] C. Deng, X. Yang, F. Nie, and D. Tao, "Saliency detection via a multiple self-weighted graph-based manifold ranking," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 885–896, Apr. 2020.
- [33] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, "What is and what is not a salient object? Learning salient object detector by ensembling linear exemplar regressors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4399–4407.
- [34] Y.-Y. Zhang, S. Zhang, P. Zhang, H.-Z. Song, and X.-G. Zhang, "Local regression ranking for saliency detection," *IEEE Trans. Image Process.*, vol. 29, pp. 1536–1547, 2020.
- [35] F. Wang and G. Peng, "Graph-based saliency detection using a learning joint affinity matrix," *Neurocomputing*, vol. 458, pp. 33–46, Oct. 2021.
- [36] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. Dagan Feng, "Robust saliency detection via regularized random walks ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2710–2717.
- [37] Z. Liu, J. Li, L. Ye, G. Sun, and L. Shen, "Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2527–2542, Dec. 2017.
- [38] B. Wang, T. Zhang, and X. Wang, "Salient object detection based on Laplacian similarity metrics," *Vis. Comput.*, vol. 34, no. 5, pp. 645–658, May 2018.
- [39] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.
- [40] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.
- [41] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1265–1274.
- [42] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2009, pp. 1597–1604.
- [43] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4558–4567.
- [44] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2011, pp. 2097–2104.
- [45] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 2083–2090.
- [46] K. Wang, L. Lin, J. Lu, C. Li, and K. Shi, "PISA: Pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3019–3033, Oct. 2015.
- [47] F. Huang, J. Qi, H. Lu, L. Zhang, and X. Ruan, "Salient object detection via multiple instance learning," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1911–1922, Apr. 2017.
- [48] Q. Zheng, S. Yu, and X. You, "Coarse-to-fine salient object detection with low-rank matrix recovery," *Neurocomputing*, vol. 376, pp. 232–243, Feb. 2020.

- [49] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3183–3192.
- [50] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-Net: Going deeper with nested U-structure for salient object detection," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107404.
- [51] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1356–1363.
- [52] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
- [53] M. Van Den Bergh, X. Boix, G. Roig, B. De Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 13–26.



XUEMEI LI received the master's and Ph.D. degrees from Shandong University, Jinan, China, in 2004 and 2010, respectively. From 2013 to 2014, she was a Visiting Scholar with the University of Houston, USA. She is currently a Professor with the School of Software, Shandong University, and a member of the GD&IV Laboratory. She is engaged in research on geometric modeling, CAGD, medical image processing, and information visualization.



SUWEI WANG received the B.S. degree in software engineering from Shandong University, Jinan, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. Her research interests include saliency detection and image segmentation.



YANG NING received the master's degree in computer science from the Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics, Jinan, China, in 2016, and the Ph.D. degree in computer science from Shandong University, Jinan, in 2020. He is currently a Lecturer with the School of Computer Science and Technology, Shandong Jianzhu University. His research interests include computer vision, medical image analysis, deep learning, and machine learning.



CAIMING ZHANG received the B.S. and M.S. degrees in computer science from Shandong University, Jinan, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 1994. From 1998 to 1999, he was a Postdoctoral Fellow with the University of Kentucky, Lexington, USA. He is currently a Professor with the School of Software, Shandong University, and the School of Computer Science and Technology, Shandong University of Finance and Economics. His research interests include CAGD, CG, visualization in scientific computing, and image processing.

...