

Research Article

Graph-Based Salient Region Detection through Linear Neighborhoods

Lijuan Xu,¹ Fan Wang,¹ Yan Yang,² Xiaopeng Hu,¹ and Yuanyuan Sun¹

¹School of Computer Science and Technology, Dalian University of Technology, No. 2 Linggong Road, Dalian 116024, China

²School of Computer and Information Technology, Liaoning Normal University, No. 850 Huanghe Road, Dalian 116024, China

Correspondence should be addressed to Xiaopeng Hu; xphu@dlut.edu.cn

Received 17 March 2016; Accepted 9 May 2016

Academic Editor: Chanho Jung

Copyright © 2016 Lijuan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pairwise neighboring relationships estimated by Gaussian weight function have been extensively adopted in the graph-based salient region detection methods recently. However, the learning of the parameters remains a problem as nonoptimal models will affect the detection results significantly. To tackle this challenge, we first apply the adjacent information provided by all neighbors of each node to construct the undirected weight graph, based on the assumption that every node can be optimally reconstructed by a linear combination of its neighbors. Then, the saliency detection is modeled as the process of graph labelling by learning from partially selected seeds (labeled data) in the graph. The promising experimental results presented on some datasets demonstrate the effectiveness and reliability of our proposed graph-based saliency detection method through linear neighborhoods.

1. Introduction

The goal of saliency detection is to identify and locate the most interesting and important region that pops out from the rest in an image, which has been widely used for applications in computer vision, including object detection and recognition [1, 2], image compression [3], image segmentation [4], content based image retrieval [5], image cropping [6], and photo collage [7].

Numerous researches have been conducted to design various algorithms for salient region detection. Among these works, graph-based saliency detection models have aroused considerable interest in recent years. Previous works on detecting salient regions from images represented as graphs include [8–15]. These models describe the input image as an undirected weight graph, in which vertices represent the image elements (pixels/regions) and edges represent the pairwise dissimilarity between vertices, and the salient object detection problem is formulated as random walks [8–10], binary segmentation [11, 12], labelling (ranking) task [13, 14], or distance metric [15] on the graph, which aims at finding the pop-out vertices at some local or global locations.

In methods of the random walks on graphs [8–10], the identification of salient regions is determined by the

frequency of visits to each node at equilibrium. In [8], while some results are presented on only two synthetic images, there is no evaluation of how the method will work on real images. In [9], Harel et al. constructed the full-connected directed graph to represent the image in which the weight of the edge between two vertices is proportional to their dissimilarity, as well as their closeness in the spatial domain. Nonsalient regions are defined as the most frequently visited vertices in a local context. Wang et al. [10] analyzed multiple cues in a unified energy minimization framework and used the model in [9] to detect salient objects. A major problem is that cluttered backgrounds usually yield higher saliencies for possessing high local contrasts. Lu et al. [11] and Liu et al. [12] regarded the saliency detection problem as binary segmentation on a graph. In [11], Lu et al. developed a hierarchical graph model and utilize concavity context to compute weights between nodes, from which the graph is bipartitioned for salient object detection. Gopalakrishnan et al. [13] and Yang et al. [14] defined the saliency as the labelling or ranking task on a graph and applied the semisupervised learning technique to infer the binary labels of the unlabeled vertices with the salient seeds. However, it is difficult to determine the number and location of salient seeds that the semisupervised method requires, which is a known problem

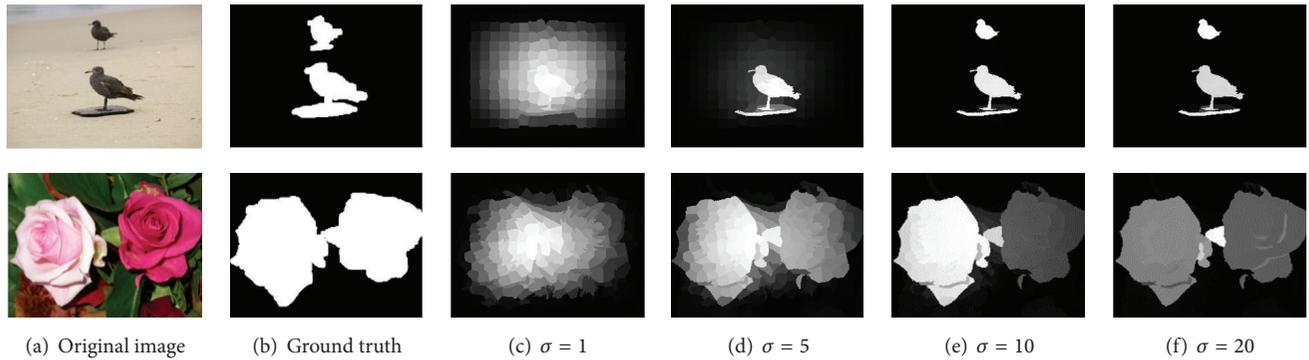


FIGURE 1: The results of different variances in graph-based saliency models. This figure shows the saliency maps obtained by the manifold ranking method proposed by Yang et al. [14] under different variances σ of the edge weight defined with Gaussian function in the graph. It is obvious that perturbation of the variance could make the detection results dramatically different.

with graph labelling. In addition, the geodesic distance metric was applied to measure the feature contrast along paths on the graph in [15].

The reason why the graph model can be associated with the saliency detection is that the prior consistency or cluster assumption [16, 17] observed in semisupervised learning or manifold learning problem, which have been demonstrated effectively to preserve the intrinsic data structure hidden in the dataset, is also appropriate for uncovering the relationships between pixels in the image. The prior consistency mainly consists of two aspects: (1) nearby pixels are likely to have the same saliency; (2) pixels on the same structure (such as an object or a homogeneous region) are likely to have the same saliency. Note that the first assumption is local, while the second one is global. The cluster assumption advises us to consider both local and global information contained in the image during learning. It is straightforward to apply cluster assumption to the graph-based saliency detection models developed in recent years, since the central idea of these methods is to find the pop-out or salient nodes while preserving the global structure hidden in the image.

Although there has been some success with the graph-based saliency detection approaches, identifying salient objects in natural scenes remains a challenge because factors such as the local or global structure information are not fully described. The graph-based semisupervised learning or manifold learning methods model the whole dataset as a graph. Similarly, the graph-based saliency detection method models the input image in the same way. In most graph-based models, the superpixels are extracted and denoted as the basic graph nodes in consideration of the computation efficiency and perception meaning. In addition, the complete graph [9, 13], k nearest neighboring graph or k -regular graph [13, 15], or the close-loop graph [14] is applied to simulate the local graph structure in different saliency models. However, how to estimate the weight of each edge has not been fully studied. More concretely, most of methods adopted a Gaussian function to calculate the edge weights of the graph [9, 13–15]. But the variance σ of the Gaussian function will affect the detection results significantly. This problem has been demonstrated in the semisupervised learning methods [18],

which occurs in the graph-based saliency models (illustrated in Figure 1) as well. However, there is no reliable approach for model selection if only very few labeled seeds are available; that is, it is hard to determine optimal σ , as pointed out by Zhou et al. [16].

To address the above issues, we propose a more reliable and stable graph-based saliency detection model in this paper. Firstly, the nodes of the graph are made up of a series of neighboring image superpixels, and the edges represent the neighborhood relationships between different image superpixels. Instead of considering pairwise neighborhood relationships adopted in current graph-based saliency detection methods, we apply the adjacent information provided by all neighbors of each image node to estimate the edge weights in the graph based on the locally linear assumption that nearby superpixels are likely to have the same saliency. Then the edge weights of all nodes are assigned to the edges for constructing the undirected weight graph. Finally, we model the saliency detection as the process of labelling by learning from partially selected seeds (labeled data) in the graph. The experiments on some datasets demonstrate the effectiveness and higher parameter stability of our proposed graph-based saliency method.

The remainder of the paper is organized as follows. Section 2 will describe our proposed model in detail. In Section 3, the experiments on some popular datasets are presented, followed by the conclusions and future works in Section 4.

2. Proposed Method

In the proposed method, a spatially neighboring graph, where superpixels are extracted and considered as the basic graph nodes and the linear relationships for all neighbors of each node are applied to estimate the weights of edges, is constructed to represent the local structure in the image. The problem of saliency detection can be tackled by modeling the task of labelling by using the selected seeds in the whole graph. The emphasis of this section and the major contribution of this paper are the construction of the graph by all neighbors of each node with the assumption of the prior

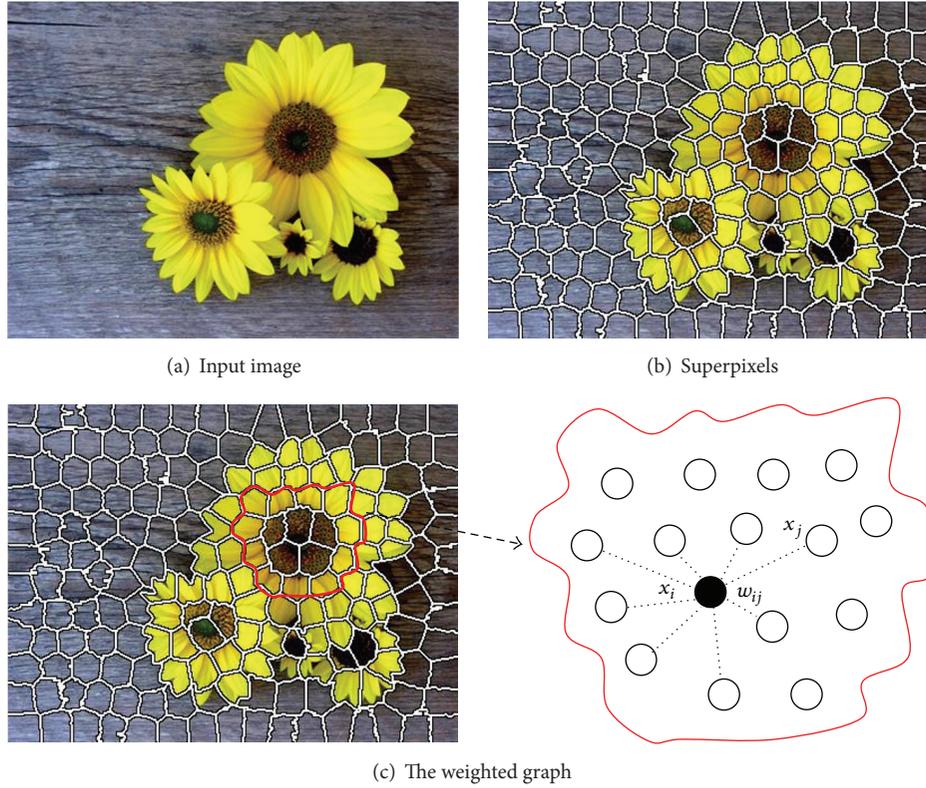


FIGURE 2: The illustration of the undirected weight graph. (a) An input image. (b) The image superpixels generated by SLIC segmentation. The construction of the weighted graph is shown in (c). If x_j belongs to the set of neighbors for node x_i , w_{ij} describes the contribution of node x_j to reconstruct node x_i .

consistency and the following graph labelling for saliency detection.

2.1. Graph Construction. As shown in Figure 2, an undirected weighted graph $G = (V, E)$ is constructed to represent the input image, where V is a set of nodes and E is a set of undirected edges with weight w_{ij} . In this paper, the nodes are visually homogeneous superpixels, which are computationally efficient and perceptually meaningful compared to regular image patches, and generated by the Simple Linear Iterative Clustering (SLIC) algorithm proposed by Achanta et al. [19]. The reason why we choose the SLIC is that the resulting superpixels are almost regular and compact image patches with better boundary adherence, which facilitate the preservation of the object edges in the saliency map [14, 15, 20]. In addition, the spatially neighboring superpixels are connected to simulate the local neighborhood relationships for all the nodes in the graph. Each edge is assigned a weight to represent the relationship between the two nodes.

Conventionally [9, 13–15], the weight w_{ij} of edge that links nodes x_i and x_j in E is computed as

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right). \quad (1)$$

This Gaussian weight function mainly describes the pairwise dissimilarity between two neighboring superpixels. However, it is parameter dependent and sensitive as most natural images are more cluttered and complicated. Moreover, the locally linear information is ignored. Thus a more reliable and stable way to construct the graph based on all neighbors of each node is derived, which is capable of recovering the global nonlinear structure from the locally linear fits in the graph.

Based on the cluster assumption that each superpixel and its neighbors lie on or close to a locally linear region of the same manifold in the image, we characterize this local structure of these superpixels by linear coefficients that reconstruct each node from its neighbors in the corresponding graph. We measure the reconstruction errors by the following cost function:

$$\varepsilon = \sum_i \left\| x_i - \sum_{x_j \in N(x_i)} w_{ij} x_j \right\|^2, \quad (2)$$

where $N(x_i)$ denotes all the neighbors of x_i and w_{ij} summarizes the contribution of the node x_j to the node x_i reconstruction. To estimate the weight w_{ij} , our objective is to minimize the cost function subject to three constraints: (1) each node x_i is reconstructed only from its neighbors $N(x_i)$; if x_j does not belong to the set of neighbors for node

x_i , w_{ij} is set to 0; (2) the sum of the contribution from all neighbors to node x_i equals 1; that is, $\sum_{x_j \in N(x_i)} w_{ij} = 1$; (3) we only consider the nonnegative edge weight, $w_{ij} \geq 0$, in our saliency detection model. Clearly, when x_j is more similar to the node x_i , w_{ij} will be larger, which means that x_j plays a more important role in reconstructing the node x_i . Thus the reconstruction weights of each node can be solved by the least-square algorithm with the three constraints:

$$\begin{aligned} \min_{w_{ij}} \quad & \left(\sum_i \left\| x_i - \sum_{x_j \in N(x_i)} w_{ij} x_j \right\|^2 \right) \\ \text{s.t.} \quad & \sum_j w_{ij} = 1, \\ & w_{ij} \geq 0. \end{aligned} \quad (3)$$

One question that should be noticed is that usually $w_{ij} \neq w_{ji}$; here we introduce the algorithm proposed in [18] to get a symmetric matrix. Once the reconstruction weights for all nodes in the graph are computed, we will obtain a sparse weight matrix $\mathbf{W} = w_{ij}$ for graph G . This weight graph describes the locally neighboring relationships through synthesizing the linear neighborhood around each node, which facilitates the discovery of the globally hidden structure and the following pop-out salient nodes in the image. Furthermore, our graph structure is capable of solving the problem of determining an optimal parameter that occurred in conventional Gaussian function method.

2.2. Saliency Model

2.2.1. Graph Labelling. In this paper, the problem of saliency detection is modeled as the task of graph labelling, which aims to propagate the labels of the selected seeds to the unlabeled nodes using the graph constructed in Section 2.1. Given a dataset $\{x_1, \dots, x_l, x_{l+1}, \dots, x_n\}$, the first l data belong to the set of labeled X_l and the remainders belong to the set of unlabeled X_u which need to be labeled according to the relevance to the labeled ones. Let $\mathbf{f} = (f_1, \dots, f_n)^T$ denote some classifying functions defined on X , which can assign a real value f_i to each data point x_i . Let $\mathbf{y} = (y_1, \dots, y_n)^T$ denote the label indicator for each data point. If $i \leq l$, $y_i = 1$; otherwise, $y_i = 0$. Thus, the problem of labelling the unlabeled data can be solved by an iterative procedure.

In each iteration, each data point can “absorb” a fraction of label information from its neighbors and retain some label value of its present state. In other words, all data spread their label scores to their neighbors via the weighted graph. Therefore, the label of data point x_i at time $t + 1$ becomes

$$f_i^{t+1} = \alpha \sum_{x_j \in N(x_i)} w_{ij} f_j^t + (1 - \alpha) y_i, \quad (4)$$

where $\mathbf{f}^t = (f_1^t, \dots, f_n^t)^T$ is the classifying function learned at iteration t and $\mathbf{f}^0 = \mathbf{y}$. The parameter α specifies the relative

contributions to the labelling scores from neighbors and the initial labels, which is in $[0, 1)$. Consider

$$\mathbf{f}^{t+1} = \alpha \mathbf{W} \mathbf{f}^t + (1 - \alpha) \mathbf{y}. \quad (5)$$

Iterating (5) to update the label scores of all the data until convergence, let \mathbf{f}^* be the limit of the sequence \mathbf{f}^t

$$\mathbf{f}^* = (1 - \alpha) (\mathbf{I} - \alpha \mathbf{W})^{-1} \mathbf{y}, \quad (6)$$

where \mathbf{I} is the n -dimensional identity matrix. In (6), the constant term $1 - \alpha \geq 0$, will not affect the labelling results of \mathbf{f}^* . Here, \mathbf{f}^* is equivalent to

$$\mathbf{f}^* = (\mathbf{I} - \alpha \mathbf{W})^{-1} \mathbf{y}. \quad (7)$$

The resulting label function \mathbf{f}^* provides a general framework for semisupervised learning. Here, multiple labels are predefined for applications including clustering, segmentation, and classification. In [14], Yang et al. applied \mathbf{f}^* to learning of the ranking scores for saliency detection according to the relations with the single label, which they call the manifold ranking [21] based algorithm.

2.2.2. Saliency Measure. For saliency detection, how to determine the number and location of salient labels that (7) requires remains a big problem. Reference [13] applied the most salient node and some background nodes together for label extraction of the salient region; however, the results are sensitive to the choosing of the seeds. Observing that background often presents local or global appearance consistent with the image boundary [15, 20, 22–24], Yang et al. [14] used the nodes on the image boundary to label most of the background nodes, which aims to produce some salient labels. Similar to the scheme proposed in [14], the salient regions are estimated by the process of two-phase graph labelling on the constructed weight graph (shown in Figure 3). In other words, the image boundary and the salient nodes are separated to generate the final saliency maps.

(1) *Labelling with the Background Nodes.* Based on the assumption that the image boundary is more likely to be background, we apply the superpixels on the image boundary as the seeds to remove some background clutters and in turn lead to better salient seeds for labelling. Considering the basic rule of the photographic composition that the four sides on the image boundary often show different appearances of the background, specially, they are marked with different labels for strong learning. If all of these boundary superpixels are characterized with the same label in the procedure of learning, the learned classifying function is usually less optimal as these nodes are dissimilar. Therefore, the first-phase saliency maps are generated using the four sides on the image boundary superpixels as the labeled nodes, respectively, and the rest as the unlabeled ones.

In this paper, we first obtain the four label indicator vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$ of the top, bottom, left, and right image boundary. Then all the nodes on the graph are labelled based on \mathbf{f}^* in (7) with the four label indicator vectors separately, and the results are four n -dimensional vectors, in which each



FIGURE 3: The framework of the two-phase saliency detection model. (a) Input image and (b) the ground truth labelled by humans. ((c) and (d)) The saliency maps generated by the first phase and second phase of graph labelling based saliency model, respectively.

element denotes the relevance to the labelled image boundary superpixels. We then normalize them to the range between 0 and 1. If the image elements are similar to the boundary, the labelling values obtained by \mathbf{f}^* are closer to 1. However, the saliency values are characterized with labelling with salient seeds in saliency detection models. So the final label score for each superpixel is modified as

$$F(i) = 1 - \mathbf{f}^*(i), \quad (8)$$

where i is a node on the constructed graph. In order to obtain some candidate salient superpixels in the image, (8) is applied to integrate the four label scores generated by the four sides of image boundary, respectively. Thus, the first-phase saliency is defined as

$$S_{\text{first}}(i) = F_1(i) * F_2(i) * F_3(i) * F_4(i). \quad (9)$$

Saliency maps generated by (9) can suppress most of the background superpixels, thus highlighting some candidate salient ones. To tackle this problem, we use the label function in (7) with the candidate salient nodes to further improve the performance of detection salient regions.

(2) *Labelling with the Candidate Salient Nodes.* The second phase aims to eliminate some background superpixels and highlight salient regions using the labelling function in (7) with the possible salient nodes produced in the first-phase labelling, since the possible salient regions inferred from the

image boundary are weak and prior dependent, in order to get stronger label seeds which are salient or belong to the actual salient regions, ‘‘Otsu’s method’’ [25] is applied to partition the saliency maps generated in the first phase adaptively, and the remaining foreground superpixels are treated as strong salient seed for labelling. Then the label indicator vector \mathbf{y} is established to compute the relevance function \mathbf{f}^* in (7). Here, the saliency is defined as the normalized labelling scores between 0 and 1. Consider

$$S_{\text{second}}(i) = \mathbf{f}^*(i). \quad (10)$$

Despite some imprecise foreground labels, the salient objects can be well detected in the final saliency maps. The reason is that the salient regions are usually compact and not dispersed in terms of spatial distribution compared to background regions and homogeneous and consistent when considering the appearance in the aspect of feature distribution, such as color and texture [14]. All of these priors affect the detection results greatly.

In conventional semisupervised learning methods for graph labelling, the diagonal elements of matrix \mathbf{A} ($\mathbf{A} = (\mathbf{I} - \alpha\mathbf{W})^{-1}$) play an important role in computing the relevance to the labelled data. Nevertheless, this diagonal matrix means that the relevance to each superpixel itself is considered in saliency detection if it does not equal 0, which can weaken the contributions of other labelled nodes as the self-relevance is usually to be large. In order to get better detection results, we set the diagonal elements of \mathbf{A} to 0.

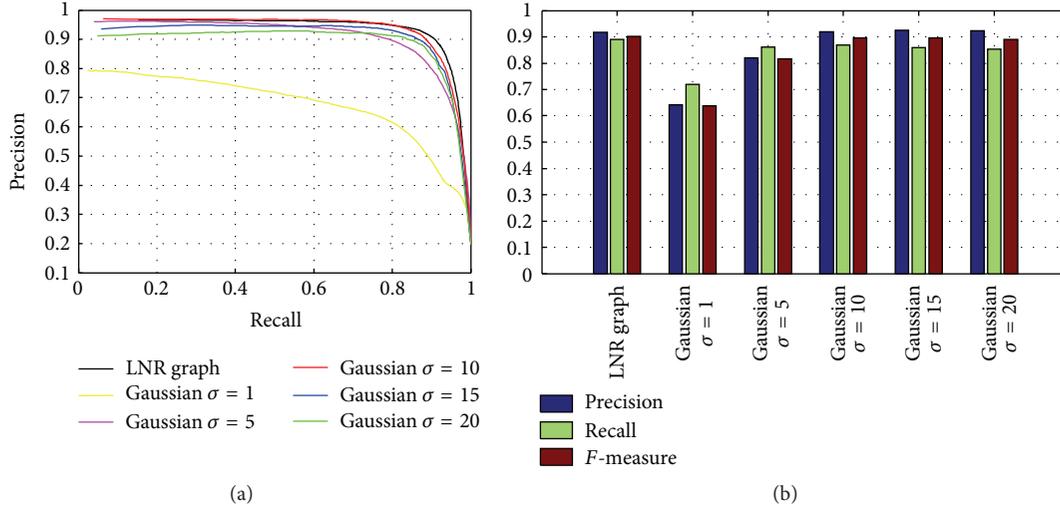


FIGURE 4: The comparison of different graphs. ((a) and (b)) The PR curves, as well as the precision, recall, and F -measure values for the comparison of LNR graph and graph weighted by the Gaussian function with different variances $\sigma = 1, 5, 10, 15, 20$, respectively. We note that the Gaussian weight function with the variance $\sigma = 10$ achieves the best performance. In fact, choosing an appropriate parameter is difficult.

3. Experiments

Dataset. Two standard benchmark datasets ASD and SOD are adopted to evaluate the proposed graph-based saliency model in this paper.

(1) *ASD.* It contains 1000 images with accurate human labeled segmentation masks for salient objects provided in [26], which has been used for testing almost all saliency models.

(2) *SOD.* The dataset comes from the well-known Berkeley segmentation dataset of 300 images [27], in which the images are cluttered and complex in terms of the scales, appearances, and positions of the foreground objects, as well as the appearance of the background regions. The pixel-labeled ground truth is obtained from [15].

Evaluation Metrics. For average performance evaluation, we use the standard PR (precision-recall) curve and F -measure. For each detected saliency map and the corresponding ground truth, precision rate corresponds to the ratio of salient pixels which are correctly detected in the saliency map, while recall rate is the percentage of all detected salient pixels belonging to salient objects in ground truth. To generate a PR curve, the saliency map is normalized into $[0, 255]$ first. A series of binary masks are then produced by segmenting the saliency map with a threshold varying from 0 to 255. We compare these binary masks with the ground truth to obtain the PR curve for each saliency map. The curves obtained from all images on each dataset are averaged to generate an overall PR curve.

Although commonly used, PR curve is limited in that it only considers whether the saliency of the object saliency is higher than that of the background. Since high precision can be achieved at the cost of decreasing the recall and vice versa,

the F -measure is used to trade off the overall performance of the precision rate and recall rate:

$$F\text{-measure}_\beta = \frac{(1 + \beta^2) \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}}. \quad (11)$$

And β^2 is set to 0.3 as in [26]. As different images have optimal binary threshold dissimilarly, a constant threshold adopted in the PR curve ignores it. Then an adaptive threshold value is proposed and determined as the mean saliency of all pixels in each saliency map as in [26] to measure the average precision, recall, and F -measure values on each dataset:

$$T = \frac{2}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} S(x, y), \quad (12)$$

where W and H are the width and height of the saliency map in pixels, respectively, and $S(x, y)$ is the saliency value of the pixel (x, y) .

3.1. Validation of the Graph Constructed through Linear Neighborhood. In this section, the proposed LNR (linear neighborhood relationship) graph is compared with the conventional pairwise relationship based graph weighted by the Gaussian function with different variances to generate saliency maps for all images on the ASD dataset in our framework of saliency detection. Figure 4(a) illustrates the results of average PR curves, and Figure 4(b) shows the precision, recall, and F -measure values generated by the adaptive threshold, respectively. The experimental results indicate the reliability and stability of our LNR graph. The reason is that our LNR graph is more capable of representing the local information through the linear neighborhood around each superpixel,

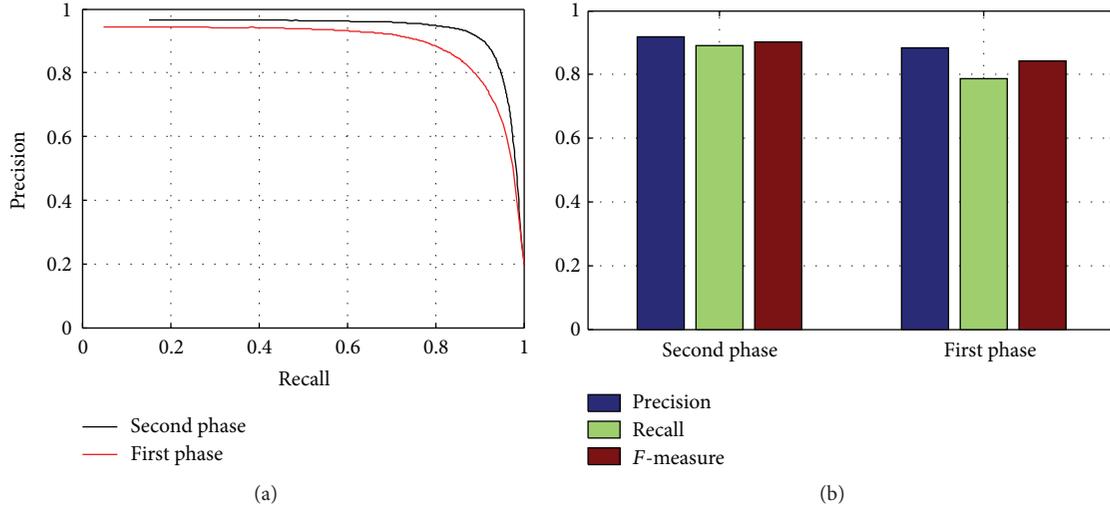


FIGURE 5: The comparison of two-phase saliency maps. ((a) and (b)) The PR curves and F -measure values for the comparison of the image boundary and some candidate salient nodes based graph labelling for saliency detection, respectively. We note that the second phase achieves better performance for more strong labels generated with the first phase and Otsu's method.

and discovering the global manifold structure through the semisupervised learning and labelling.

3.2. Validation of the Two-Phase Saliency. This paper applies the image boundary and some candidate salient nodes, respectively, to generate the label indicator vectors for graph labelling based saliency detection. We then compare the performance of the proposed approach for each phase. Figure 5 demonstrates that the second phase using the strong candidate foreground superpixels generated by Otsu's method further enhances the performance of the first phase just with the image boundary ones.

3.3. Comparison with the State of the Art on Some Datasets. To verify the effectiveness of proposed saliency detection method through linear neighbors on some datasets, we compare with the most recently state-of-the-art approaches (PCA [28], SF [29], FT [26], RC [30], and HS [31]), some graph-based methods (GS.SP [15], MR [14], and GB [9]), and two traditional ones (IT [32] and MZ [33]). We select these methods for their varieties in design of the saliency models or the description of saliency. In the experiment, we use the implementation from Achanta et al. [26] for FT, IT, GB, and MZ. For RC, PCA, HS, and MR, we run the authors' public codes. For SF and GS.SP, we directly use the provided saliency maps by authors. Figure 6 shows the comparison results of various saliency methods on the ASD datasets and demonstrates the performance of our method.

For the SOD dataset, we compare with the PCA, HS, GS.SP, RC, and MR methods. Example results of PR curves and the precision, recall, and F -measure values achieved from the adaptive threshold are illustrated in Figure 7, respectively. We note that when measured with the fixed threshold method, our PR curve cannot perform well compared to

some models at high recall rate (lower threshold). But as the threshold rises, the proposed method works best when compared with the methods mentioned in Figure 7(a), which means our method can highlight the whole object region uniformly. The reason is that the calculated pixel values of the saliency maps in our method are distributed within a fixed interval, which occupies a small range in the gray space. When the threshold is smaller than certain value, the segmented saliency maps cannot represent all the salient objects in the images, which results in the lower precision rate.

Results of visual comparison on the two datasets are shown in Figure 8.

3.4. Experimental Settings and Run Time. The proposed method is applied on the ASD and SOD datasets based on a machine with Intel Core i5-4590 3.30 GHz CPU and 8 GB RAM, and we implement the saliency model by using the Matlab language. Specifically, our method spends 0.201 s, 0.515 s, and 0.342 s on superpixel generation, graph construction, and saliency map computation, respectively, for each image on the ASD dataset, and the run time of superpixel generation is estimated by segmenting each image into 200 superpixels.

4. Conclusion

Recently, graph-based salient regions detection methods have applied the Gaussian weight function to measure the pairwise neighboring relationships between different image elements, which is parameter dependent and sensitive. To solve the problem, we propose the graph-based saliency detection model through linear neighbors, which means that each node is measured by a linear combination of

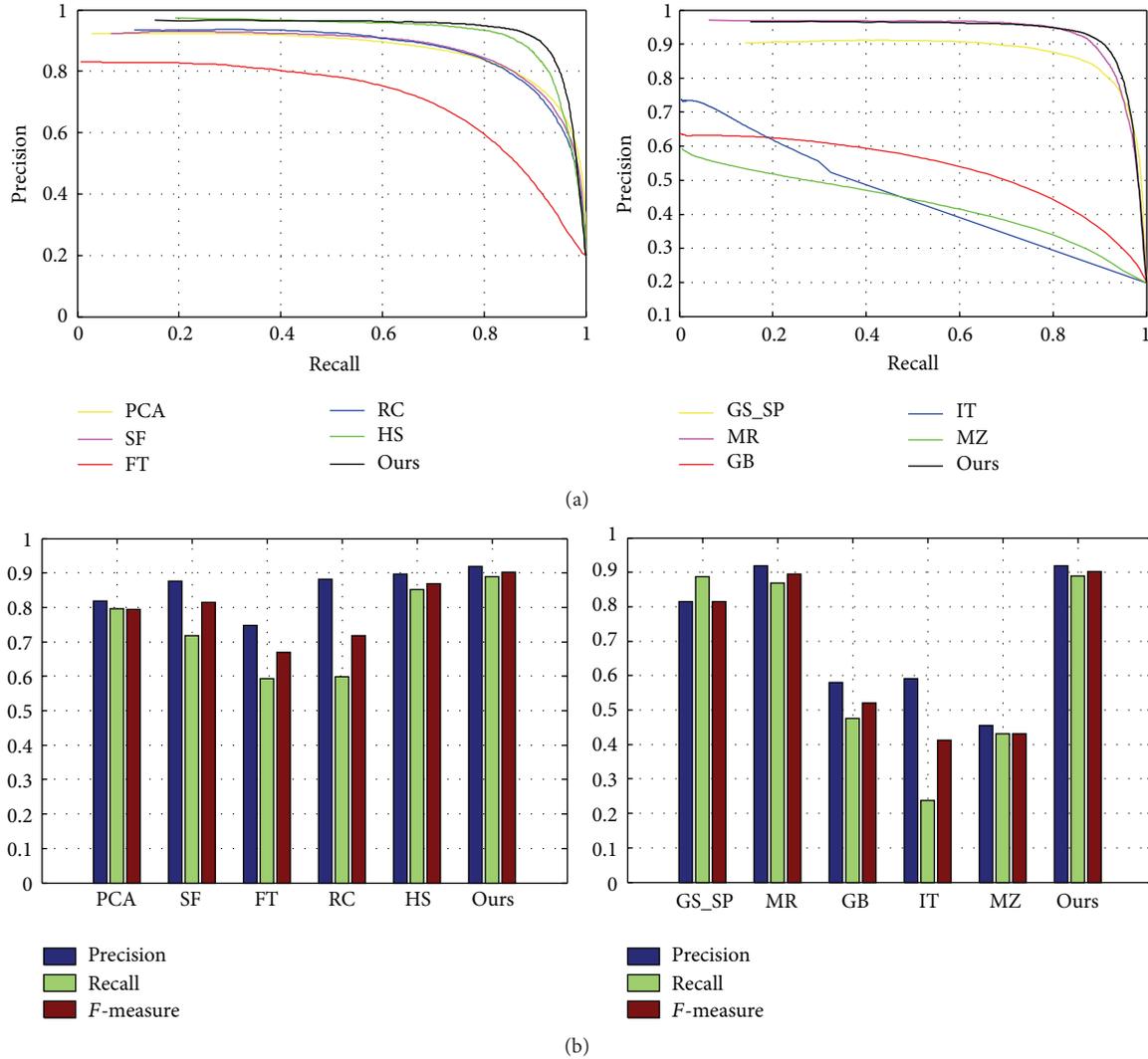


FIGURE 6: The comparison results on the ASD dataset. ((a) and (b)) The PR curves and F -measure values for the comprehensive evaluation with the state-of-the-art saliency models on the ASD dataset. We note that the proposed model shows the advantage of detecting salient regions when compared with other methods.

all its neighborhoods around to further represent the local information in the image. The saliency is defined as the process of semisupervised learning to label other nodes from partially selected seeds (labeled nodes) in the whole graph, which considers the global data structure hidden in the image for labelling. As a result, both the local grouping information and the global intrinsic structure of the cluster assumption are fully captured with the graph construction and labelling to learn the pop-out nodes in our linear neighborhood relationship based saliency detection method. In addition, our LNR graph has presented the reliability and stability of detecting salient regions, which does not need the predefined parameter for optimal model selection. We then evaluate the proposed model on some benchmark datasets in this paper. The experimental results indicate the effectiveness of our graph-based saliency detection method through linear

neighborhoods when compared with some other state-of-the-art approaches.

It should be noted that the proposed method can be further improved if more strong labels are provided. Our future work will focus on the choosing of the seeds.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by “National Natural Science Foundation of China” (no. 61272523, no. 61572103), “the National Key Project of Science and Technology of China”

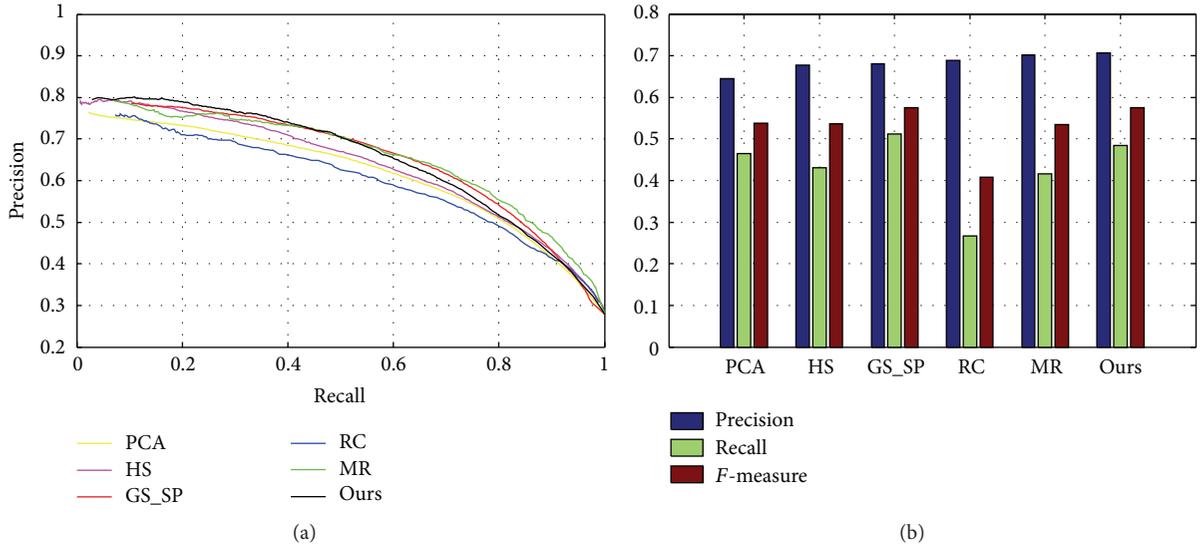


FIGURE 7: The comparison results on the SOD dataset. ((a) and (b)) The PR curves and F -measure values, respectively, for the comprehensive evaluation with the state-of-the-art saliency models on the SOD dataset. We note that our model performs better than other methods on this challenging dataset.

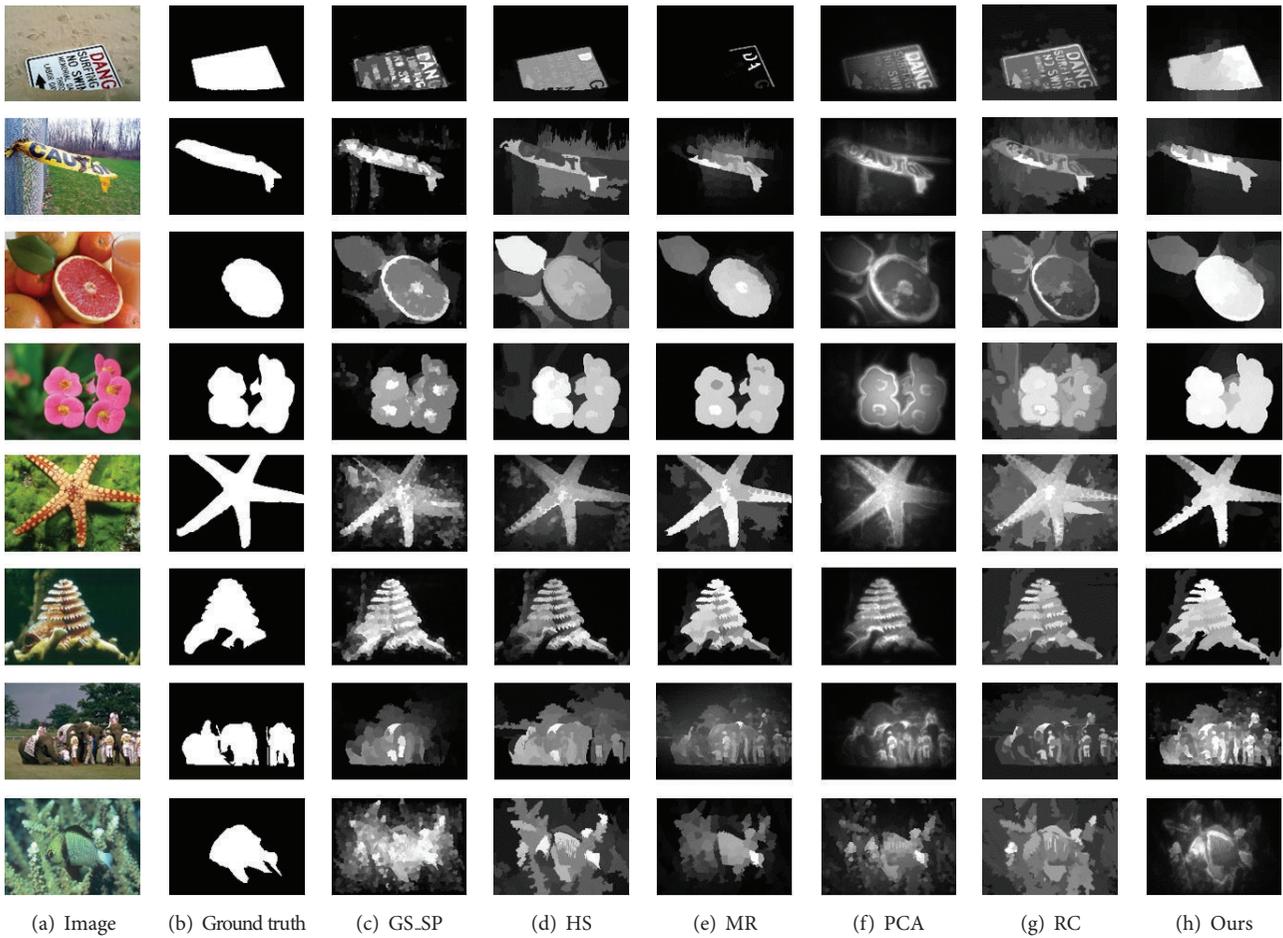


FIGURE 8: Visual results of the saliency maps generated by different methods on the two datasets.

(no. 2011ZX05039-003-4), and “the Fundamental Research Funds for the Central Universities” (no. DUT15QY33).

References

- [1] C. Kanan and G. Cottrell, “Robust classification of objects, faces, and flowers using natural image statistics,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2472–2479, IEEE, San Francisco, Calif, USA, June 2010.
- [2] U. Rutishauser, D. Walther, C. Koch, and P. Perona, “Is bottom-up attention useful for object recognition?” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II37–II44, July 2004.
- [3] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304–1318, 2004.
- [4] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.
- [5] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, “Sketch2Photo: internet image montage,” *ACM Transactions on Graphics*, vol. 28, no. 5, p. 124, 2009.
- [6] L. Marchesotti, C. Cifarelli, and G. Csurka, “A framework for visual saliency detection with applications to image thumbnailing,” in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2232–2239, IEEE, Kyoto, Japan, October 2009.
- [7] J. Wang, J. Sun, L. Quan, X. Tang, and H.-Y. Shum, “Picture collage,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 347–354, IEEE, June 2006.
- [8] L. da Fontoura Costa, “Visual saliency and attention as random walks on complex networks,” <http://arxiv.org/abs/physics/0603025>.
- [9] J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency,” in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 545–552, December 2006.
- [10] L. Wang, J. Xue, N. Zheng, and G. Hua, “Automatic salient object extraction with contextual cue,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 105–112, Barcelona, Spain, November 2011.
- [11] Y. Lu, W. Zhang, H. Lu, and X. Xue, “Salient object detection using concavity context,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 233–240, IEEE, Barcelona, Spain, November 2011.
- [12] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.
- [13] V. Gopalakrishnan, Y. Hu, and D. Rajan, “Random walks on graphs for salient object detection in images,” *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3232–3242, 2010.
- [14] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, “Saliency detection via graph-based manifold ranking,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3166–3173, IEEE, Portland, Ore, USA, June 2013.
- [15] Y. Wei, F. Wen, W. Zhu, and J. Sun, “Geodesic saliency using background priors,” in *Computer Vision—ECCV 2012*, pp. 29–42, Springer, Berlin, Germany, 2012.
- [16] D. Zhou, O. Bousquet, T. N. Lal et al., “Learning with local and global consistency,” *Advances in Neural Information Processing Systems*, vol. 16, no. 16, pp. 321–328, 2004.
- [17] O. Chapelle, J. Weston, and B. Schölkopf, “Cluster kernels for semi-supervised learning,” in *Proceedings of the 16th Annual Neural Information Processing Systems Conference (NIPS '02)*, pp. 585–592, Vancouver, Canada, December 2002.
- [18] F. Wang and C. Zhang, “Label propagation through linear neighborhoods,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.
- [19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.
- [20] W. Zhu, S. Liang, Y. Wei, and J. Sun, “Saliency optimization from robust background detection,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2814–2821, June 2014.
- [21] D. Zhou, J. Weston, A. Gretton et al., “Ranking on data manifolds,” in *Advances in Neural Information Processing Systems 16*, pp. 169–176, MIT Press, 2004.
- [22] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, “Salient object detection: a discriminative regional feature integration approach,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2083–2090, IEEE, Portland, Ore, USA, June 2013.
- [23] M. Xu and H. Zhang, “Saliency detection with color contrast based on boundary information and neighbors,” *The Visual Computer*, vol. 31, no. 3, pp. 355–364, 2015.
- [24] J. Zhang and S. Sclaroff, “Saliency detection: a boolean map approach,” in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 153–160, IEEE, Sydney, Australia, December 2013.
- [25] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.
- [26] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, “Frequency-tuned salient region detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1597–1604, Miami, Fla, USA, June 2009.
- [27] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proceedings of the 8th International Conference on Computer Vision*, pp. 416–423, July 2001.
- [28] R. Margolin, A. Tal, and L. Zelnik-Manor, “What makes a patch distinct?” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 1139–1146, Portland, Ore, USA, June 2013.
- [29] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, “Saliency filters: contrast based filtering for salient region detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 733–740, IEEE, Providence, RI, USA, June 2012.
- [30] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, “Global contrast based salient region detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, Providence, RI, USA, June 2011.
- [31] Q. Yan, L. Xu, J. Shi, and J. Jia, “Hierarchical saliency detection,” in *Proceedings of the 26th IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR '13), pp. 1155–1162, IEEE, Portland, Ore, USA, June 2013.

- [32] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [33] Y.-F. Ma and H.-J. Zhang, “Contrast-based image attention analysis by using fuzzy growing,” in *Proceedings of the 11th ACM International Conference on Multimedia (MM '03)*, pp. 374–381, November 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

