Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 7917021, 14 pages
https://doi.org/10.1155/2020/7917021

Research Article

A Deep Multiscale Fusion Method via Low-Rank Sparse Decomposition for Object Saliency Detection Based on Urban Data in Optical Remote Sensing Images

Cheng Zhang[1] **and Dan He**[2]

[1]*City Institute, Dalian University of Technology, China*
[2]*Dalian University of Finance and Economics, China*

Correspondence should be addressed to Cheng Zhang; zhangc@dlut.edu.cn

Received 6 February 2020; Accepted 16 April 2020; Published 8 May 2020

Academic Editor: Qingchen Zhang

The urban data provides a wealth of information that can support the life and work for people. In this work, we research the object saliency detection in optical remote sensing images, which is conducive to the interpretation of urban scenes. Saliency detection selects the regions with important information in the remote sensing images, which severely imitates the human visual system. It plays a powerful role in other image processing. It has successfully made great achievements in change detection, object tracking, temperature reversal, and other tasks. The traditional method has some disadvantages such as poor robustness and high computational complexity. Therefore, this paper proposes a deep multiscale fusion method via low-rank sparse decomposition for object saliency detection in optical remote sensing images. First, we execute multiscale segmentation for remote sensing images. Then, we calculate the saliency value, and the proposal region is generated. The superpixel blocks of the remaining proposal regions of the segmentation map are input into the convolutional neural network. By extracting the depth feature, the saliency value is calculated and the proposal regions are updated. The feature transformation matrix is obtained based on the gradient descent method, and the high-level semantic prior knowledge is obtained by using the convolutional neural network. The process is iterated continuously to obtain the saliency map at each scale. The low-rank sparse decomposition of the transformed matrix is carried out by robust principal component analysis. Finally, the weight cellular automata method is utilized to fuse the multiscale saliency graphs and the saliency map calculated according to the sparse noise obtained by decomposition. Meanwhile, the object priors knowledge can filter most of the background information, reduce unnecessary depth feature extraction, and meaningfully improve the saliency detection rate. The experiment results show that the proposed method can effectively improve the detection effect compared to other deep learning methods.

1. Introduction

With the rapid promotion of information technology, urban data has become one of the important information sources for human beings. And the amount of information received by people has increased exponentially [1, 2]. How to select the object regions of human interest from the mass of image information in urban becomes a significant research. Studies have found that under a complex scene, the human visual processing system will focus on several objects, named region of interest (ROI) [3]. ROI is relatively close to human visual perception. Saliency, as the image pretreatment process, can be widely applied in remote sensing areas such as visual tracking, image classification, image segmentation, and target relocation.

The saliency detection method mainly contains two aspects: top-down and bottom-up. The top-down-based saliency detection method [4–6] is a task-driven process. The ground-truth images are labeled manually for supervised training. It integrates more perceptions of humans to obtain the salient map. However, the bottom-up method is a data-driven process and pays more attention to the images'

Figure 1: Saliency detection instance.

features such as contrast, position, and texture to compute the saliency map (SM). Itti et al. [7] proposed a spatial visual model taking full advantage of local contrast and obtained the saliency map via the image differences from the center to the surrounding. Hou and Zhang [8] put forward a saliency detection algorithm based on Spectral Residual (SR). Achanta et al. [9] proposed a frequency-tuned (FT) method based on the image frequency domain to calculate saliency. A detection method combining histogram was presented to calculate global contrast [10]. Furthermore, other relevant methods were raised and showed better effect [11–15]. But they do not analyze the image from the dimensions.

Yan et al. [16] treated the saliency region of the image as sparse noise and the background as a low-rank matrix. It calculated the saliency of the image by using the sparse representation and robust principal component analysis algorithm. Firstly, the image was decomposed into 8×8 blocks. Every image block was sparsely encoded and merged into a coding matrix. Then, the coding matrix was decomposed by robust principal component analysis. Finally, the sparse matrix obtained by decomposition was devoted to establish the saliency factor of the corresponding image block. However, because the large-size saliency object contained many image blocks, the saliency object in each image block no longer satisfied the sparse feature; thus, it greatly affected the detection effect. Lang et al. [17] utilized a multitask low-rank recovery approach for saliency detection. The multitask low-rank representation algorithm was used to decompose the feature matrix and constrained the consistency of all feature sparse components in the same image blocks. The algorithm used the consistency information of multifeature description, and its effect was improved. However, since the large-size target contained a large number of feature descriptions, the feature was no longer sparse. The reconstruction error could not solve this problem, so this method could not completely detect the saliency object with a large size. To perfect the result of the above method, Shen and Wu [18] proposed a low-rank matrix recovery (LRMR) algorithm combining bottom-up and top-down algorithm (providing high-level and low-level information, respectively). First, it performed the superpixel segment in the image and several features were extracted. Then, the feature transformation matrix and a priori knowledge, including size, texture, and color, were obtained by network learning to transform the feature matrix. Finally, the low-rank and sparse decomposition of the transformed matrix were carried out by using the robust principal component analysis algorithm. This method improved the deficiency to some extent. However, due to the limitation of center prior and the failure of color prior to complex scenes, this algorithm was not ideal for detecting images with complex backgrounds.

The saliency detection method using different low-level features is usually only effective for a specific type of image, which is not suitable for multiobject images in complex scenes [19–21]. Figure 1 is the instance of saliency detection. The low-level features of visual stimuli lack an understanding of the nature of saliency objects and cannot represent the features at a deeper level. For noisy objects in the image, if they are similar to the low-level features but do not belong to the same category, they are often wrongly detected as saliency objects. Yang et al. [22] showed a bag of word model to detect saliency. Firstly, the prior probability saliency map could be obtained through the object feature, and a word bag model representing the middle semantic features was established to calculate the conditional probability saliency graph. Finally, two saliency images were synthesized by Bayesian inference. The middle semantic features could represent the image content more accurately than the bottom features. Therefore, the detection effect was more accurate. Jiang et al. [23] took saliency detection as a regression problem and integrated regional attributes, contrast, and feature vectors of regional background knowledge at multiscale segmentation conditions. The saliency map was obtained by supervised learning. Due to the introduction of background knowledge features, the algorithm had a better ability to identify background objects, and thus obtained more accurate foreground detection results.

Deep learning (DL) combines low-level features to form more abstract high-level features, a typical representative is a convolutional neural network (CNN). Many saliency detection methods have adopted CNN to optimize the result. Li et al. [24] proposed deep CNN to detect saliency. Firstly, region and edge information were obtained by using the hyper-pixel algorithm and bilateral filtering. DCNN was utilized to extract the regions and edge features in raw images. Finally, the region confidence graph and edge confidence graph generated by CNN were integrated into the conditional random field to judge the saliency. Wang et al. [25] proposed recurrent fully CNN (i.e., RFCNN) for saliency detection, which mainly included two steps: pretraining and fine-tuning. RFCN was used to train the original image to correct the saliency prior image. Then, the traditional algorithm was used to further optimize the modified saliency graph.

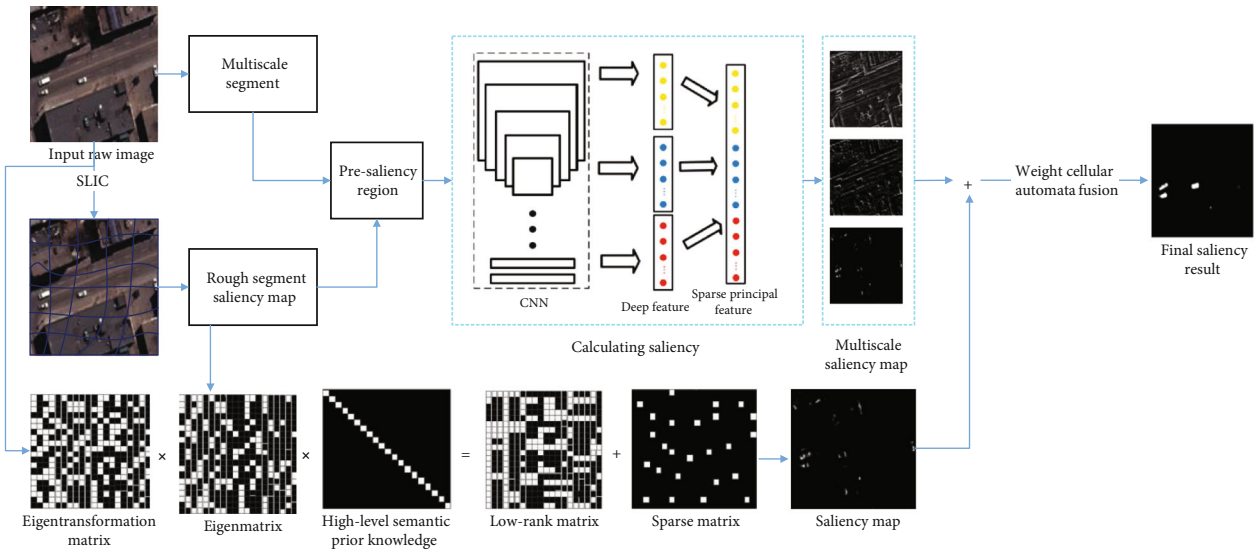


Figure 2: The framework of proposed saliency detection.

Lee et al. [26] proposed a deep saliency (DS) algorithm for saliency detection using low and high-level information in a unified CNN framework. VGG-Net was used to extract the advanced features. It mainly extracted the low-level features. Then, the CNN was used to encode the distance graph. Finally, the coded low-level distance graph was connected with higher features. A full-connected CNN classifier was adopted to evaluate the features' information and obtain the saliency graph [27]. The above DL methods show the excellent performance in terms of saliency detection rate. But there are still some disadvantages such as slow speed and highly complex calculations.

In this paper, we propose a deep multiscale fusion method via low-rank sparse decomposition for object saliency detection in optical remote sensing images. The main contributions are as follows.

(a) First, multiscale segmentation is executed for remote sensing images. For the first segmentation graph, the depth features of all the superpixel blocks are extracted by CNN

(b) Then, we calculate the saliency value, and the proposal region is generated. The superpixel blocks of the remaining proposal regions of the segmentation graph are input into the CNN network. By extracting the depth feature, the saliency value is calculated and the proposal regions are updated. Meanwhile, the color, texture, and edge feature mean values of all the pixels in each superpixel are calculated to construct the feature matrix. In order to make the image background facilitate low-rank sparse decomposition, the above feature matrices need to be transformed so that the background can be represented as a low-rank matrix in the new feature space

(c) To make use of the high-level information and improve the detection effect of the ROI, the fully convolutional neural network is used for learning features, and the high-level semantic prior knowledge matrix is obtained. The feature matrix is transformed by using the feature transformation matrix and the high-level semantic prior knowledge. The robust principal component analysis algorithm is used to decompose the transformed matrix into a low-rank sparse decomposition to obtain a saliency map. The process is iterated continuously to obtain the saliency map on each scale

(d) Finally, the weight cellular automata method fuses the multiscale saliency graphs. It is shown that the proposed method can effectively improve the detection effect compared to other DL methods

The remainder of the paper is organized as follows. The proposed deep multiscale fusion method for saliency detection is analyzed in section II. Section III introduces the saliency region extraction based on multiscale segmentation. Saliency is calculated based on the deep features in section IV. The performance and robustness are evaluated in section V. Conclusion is drawn in section VI.

2. Deep Multiscale Fusion for Saliency Detection

The proposed deep multiscale fusion method for saliency detection in optical remote sensing images is shown in Figure 2.

Firstly, the image I is segmented into a small number of superpixel blocks by using the superpixel segmentation algorithm. The deep feature is extracted from all the superpixel blocks. The color, texture, and edge feature mean value of all the pixels in each superpixel are calculated to construct the feature matrix. In order to make the image background facilitate low-rank sparse decomposition, the above feature matrix needs to be transformed so that the background can be represented as a low-rank matrix in the new feature space. And the multidimensional feature containing the key

information of the image is extracted by PCA (principal component analysis). The rough segmentation saliency graph is obtained based on the calculation of key features, where we can extract the initial saliency region to obtain the superpixel set *Suppix*. Then, we adopt *Suppix* to centralize the similarity degree between superpixel and the nonobject region. The input image is segmented at different scales. The region containing the superpixel block in the *Su ppix* set is selected for depth feature extraction. Saliency maps and *Suppix* sets at the next scale are obtained based on the same method. The robust PCA is used to decompose the transformed matrix into a low-rank sparse decomposition to obtain a saliency map. Weight cellular automata fusion is used to obtain the final SM M_{final}.

3. Saliency Region Extraction Based on Multiscale Segmentation

Superpixel segmentation is to gather adjacent similar pixel points into image regions with different sizes according to the low-level features such as brightness, thus reducing the complexity of significance calculation. The superpixel segmentation algorithm mainly includes watershed [28] and simple linear iterative clustering (SLIC) [25] method. We combine their respective characteristics, SLIC method is used to obtain the segmentation results with regular shape and uniform size during rough segmentation, and the watershed algorithm is used to obtain better object contour during fine segmentation in this study.

For N segmentation scales $(s_1,\cdots,s_n)$. $Sup_j = \{Sp_i^j\}_{i=1}^{N_j}$ denotes the obtained superpixel set at a certain segmentation scale. N_j denotes the superpixel number at scale s_j. $Sp_i^j(v) = \{R, G, B, L, a, b\}$ is pixel's color feature vector in the superpixel.

For the input image, we extract color, texture, and edge features to construct the feature matrix.

(i) Color feature. The gray value of R, G, B, hue, and saturation are extracted to describe the color feature of the image

(ii) Edge feature. Steerable pyramid filter is used to decompose the image in multiple scales and directions. Filters with 3 scales and 4 directions are selected to obtain 12 responses as the edge features of the image

(iii) Texture feature. Gabor filter is used to extract texture features at different scales and directions. Here, 3 scales and 12 directions are selected to obtain 36 responses as the texture features

It calculates the mean value of all pixel features in each superpixel to represent the eigenvalue f_i. All the eigenvalues constitute the eigenmatrix $F = [f_1, f_2,\cdots,f_N]$, $F \in R^{d\times N}$.

The saliency region of the image is regarded as sparse noise and the background as a low-rank matrix. In the complex background, the image background similarity degree after clustering is still not high. Therefore, the features in the original image are not conducive to low-rank sparse decomposition. In order to find a suitable feature space, most image backgrounds can be represented as low-rank matrices; in this paper, the eigentransformation matrix is obtained based on the gradient descent method. The process of obtaining the eigentransformation matrix is as follows:

(a) Construct marker matrix $Q = \text{diag}\{q_1, q_2,\cdots,q_N\}$. If the superpixel p_i is within the marked saliency region manually, $q_i = 0$. Otherwise, $q_i = 1$

(b) According to the following formula, the optimal model of transformation matrix T is utilized to learn the features of raw image

$$T_{optimal} = \arg\min_T O(T) = \frac{1}{K}\sum_{k=1}^{K} \|TF_kQ_k\|_\circ - \gamma\|T\|_\circ \quad (1)$$

Where $F_k \in R^{d\times N_k}$ is the feature matrix of kth image. N_k represents the superpixel number of kth image. $Q_k \in R^{N_k\times N_k}$ is the labeled matrix of the kth image. $\|\cdot\|_\circ$ represents the kernel norm of the matrix, that is, the sum of all singular values of the matrix. γ is the weight coefficient. $\|T\|_2$ denotes the ℓ_2 norm of the matrix T. c is a constant to prevent T from arbitrarily increasing or decreasing. If the eigentransformation matrix T is appropriate, then TFQ is low rank. $-\gamma\|T\|_\circ$ is to avoid obtaining the general solution when the rank of T is arbitrarily small.

(c) Find the $T_{optimal}$ gradient descent direction, that is

$$\frac{\partial O(T)}{\partial T} = \frac{1}{K}\sum_k \frac{\partial\|TF_kQ_k\|_\circ}{\partial T} - \gamma\frac{\partial\|T\|_\circ}{\partial T} \quad (2)$$

(d) Adopt the following formula to update the eigentransformation matrix T until the algorithm converges to the local optimal. α is the step size

$$T_{t+1} = T_t - \alpha\frac{\partial O(T)}{\partial T} \quad (3)$$

3.1. Extracting Proposal Region. The segmentation graph of a rough segmentation scale s_j is taken as input. The saliency map Map_j is obtained by depth feature extraction and saliency value calculation. The Map_j, as the object prior knowledge in the next segmentation, is used to guide the proposal region extraction. The saliency Map_j is binarized. The value of Map_j is divided into K channels by the adaptive threshold strategy. $p(i)$ is used to represent the number of pixels in the channel i. The channel k with

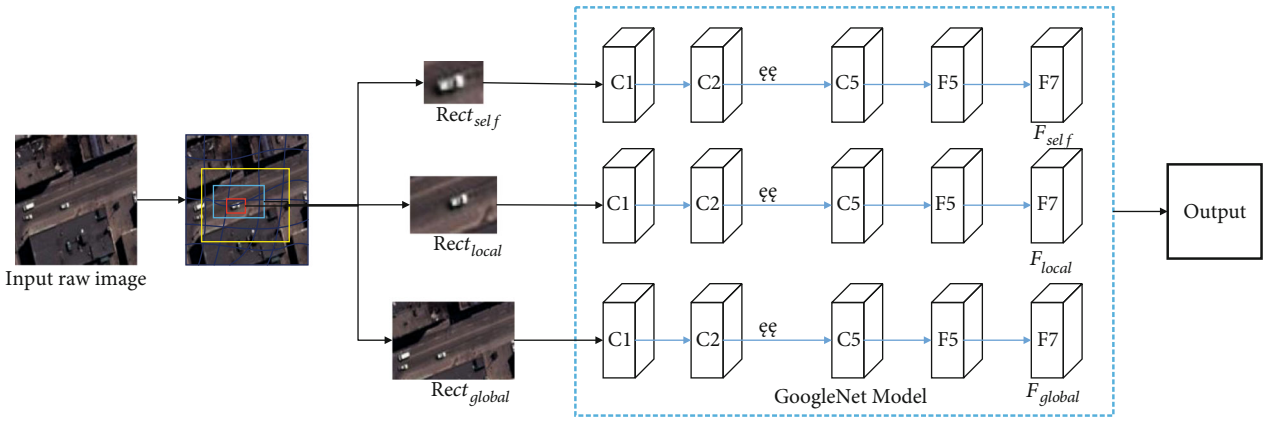


Figure 3: Deep features extraction based on CNN.

the largest number of pixels in all channels is determined. The threshold value T is calculated by the formula (4).

$$T = (k+1)/K \tag{4}$$

In order to prevent T from getting larger, the significant pixel will not be binarized to 0 when the saliency object occupies the most space in the image. The pixel number in each channel must satisfy $p(i)/area(I) < \varepsilon$. Where $area(I)$ is the pixel number of image I. $\varepsilon \in [0.6, 0.9]$ is an experience value. The binarization object a priori map is denoted as $MapB_j$. We adopt $MapB_j$ as the prior knowledge. The corresponding superpixel area of superpixel set $Sup_{j+1} = \{Sp_i^{j+1}\}_{i=1}^{N_{j+1}}$ in the next scale s_{j+1} constitutes the proposal saliency superpixel set $Suppix_{j+1} = \{Sp_i^{j+1}\}_{i=1}^{M_{j+1}}$. M_{j+1} is the number of proposal saliency superpixel at the scale s_{j+1}, $M_{j+1} < N_{j+1}$. Assume that Num_i is the total number of the superpixel Sp_i^{j+1}. num is the pixel number with a value of 1 at the corresponding position of the binary map $MapB_j$. If $num/Num_j > 0.5$, the superpixel at the corresponding position is considered to belong to $Suppix_{j+1}$.

3.2. Region Optimization. The proposal object superpixel set may contain some background areas or missing saliency areas. It needs to optimize the proposal object area. It removes the possible background area in $Suppix_{j+1}$ and adds the possible saliency area in the background area. According to the Euclidean distance between the two color spaces, the difference matrix is $Difmat$. It is a symmetric matrix with N_{j+1} order.

$$Difmat(i,j) = Difmat\left(Sp_i, Sp_j\right) = \sqrt{\sum_{k=1}^{6}\left(F_{i,k} - F_{j,k}\right)^2} \tag{5}$$

Where $F_{i,k}$ is the kth feature of superpixel region Sp_i. $k = [1,\cdots,6]$ corresponds to R, G, B, L, a, and b, respectively. For $Sp_k \in Suppix_{j+1}$, it calculates the local average dissimilarity degree through equation (6),

$$MavDif(Sp_k) = \frac{\sqrt{\sum_{l=1,l\neq k}^{M_{j+1}} Difmat(Sp_k, Sp_l)^2}}{M_{j+1}} \tag{6}$$

Where $Sp_k, Sp_l \in Suppix_{j+1}$, M_{j+1} is superpixel number in the proposal saliency region set $Suppix_{j+1}$. We calculate the average dissimilarity degree of each superpixel Sp_k in $Suppix_{j+1}$ and its adjacent background region:

$$MavDif(Sp_k)' = \frac{\sqrt{\sum_{l=1,l\neq k}^{M'_{j+1}} Difmat(Sp_k, Sp_l)^2}}{M'_{j+1}} \tag{7}$$

Where $Sp_k \in Suppix_{j+1}$, $Sp_l \notin Suppix_{j+1}$ and Sp_k is adjacent to Sp_l. M'_{j+1} represents the number of superpixels adjacent to Sp_k in the background area. If $MavDif(Sp_k)' > MavDif(Sp_k)$, it indicates that Sp_k is more similar to the adjacent background area, then Sp_k will be removed from $Suppix_{j+1}$.

Similarly, for any $Sp_h \notin Suppix_{j+1}$, the average dissimilarity $MavDif(Sp_h)'$ between Sp_h and adjacent background region, and average dissimilarity $MavDif(Sp_h)$ between Sp_h and adjacent proposal saliency region can be calculated. If the condition $MavDif(Sp_h)' > MavDif(Sp_h)$ is satisfied, then the similarity between Sp_h and the adjacent saliency region is higher than that of other background regions. Therefore, Sp_h is added to $Suppix_{j+1}$. $Suppix_{j+1}$ is constantly updated by comparing the superpixel in $Suppix_{j+1}$ with other saliency regions and background regions. Until the superpixel in $Suppix_{j+1}$ is no longer changed.

3.3. Deep Feature Extraction of Proposal Region. This deep feature extraction method based on CNN is as shown in Figure 3. In the first superpixel segmentation, the deep

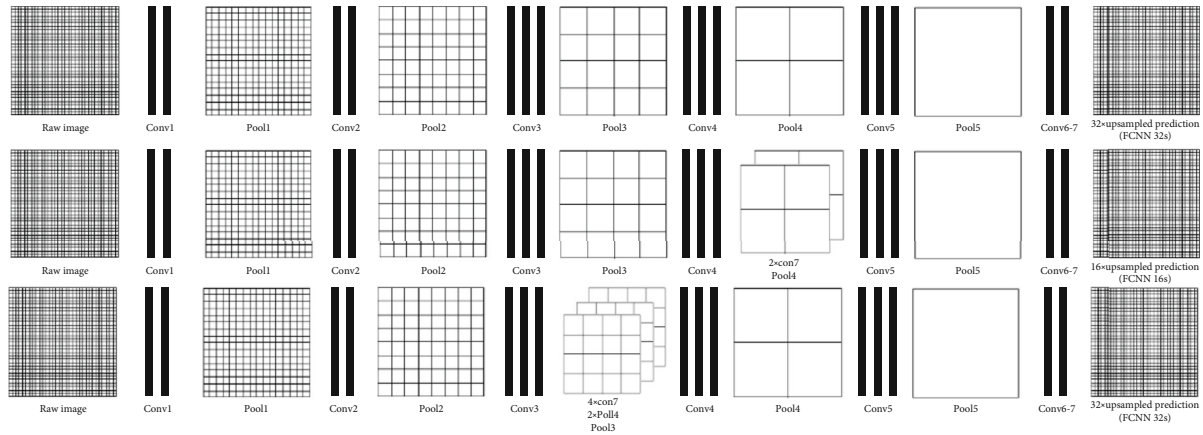


Figure 4: FCNN model.

features of all superpixels are extracted. In the subsequent deep feature extraction process, only the superpixel in *Su ppix* set is extracted. Under a certain segmentation strategy, the computation is greatly reduced and the computation speed is increased.

Assuming it is not the first time to segment the superpixel, the local and global features are extracted for superpixel Sp_i. The local features of the superpixel include two parts: (1) the deep feature F_{self} containing its own region; (2) deep feature F_{local} containing itself and the adjacent superpixel region.

First, according to *Suppix* set, it extracts the minimum rectangular region Re ct_{self} of each superpixel Sp_i. Since most superpixels are not regular rectangles, the extracted rectangles must contain other pixels. These pixels are represented by the average value of the superpixel. The depth feature F_{self} only containing its own region can be obtained through the deep CNN.

If we only adopt the saliency calculation of F_{self} to acquire saliency detection value is meaningless. It is impossible to determine whether it is saliency without comparing it with the saliency of other adjacent superpixels. Therefore, it still needs to extract Re ct_{local} to further obtain F_{local} of the deep local feature. The location of the region in the image is an important factor to judge whether it is saliency or not. It is generally believed that the area in the center of the image is more likely to be saliency than the region at the edge. Therefore, the whole image is taken as the input, and the deep feature F_{global} of the global region is extracted.

If it only uses the bottom feature to extract the saliency map, due to many interference objects, the final saliency map is not ideal. Therefore, the high-level information needs to be added to improve the detection effect. The adopted high-level semantic prior knowledge is mainly to predict the most likely ROI based on previous experience (i.e., training samples). The FCNN is used to train the high-level semantic prior knowledge, which is integrated into the feature transformation process to optimize the final saliency map. Higher-order features can be learned from the primitive data without preprocessing in the multi-stage global training process of CNN.

FCNN can accept input images with any size. The difference between FCNN and CNN is that the deconvolution layer replaces the full connection layer. Finally, pixel classification is carried out on the feature map of the upsampling. A binary prediction is produced for each pixel, and a classification result at the pixel level is output. Thus, the problem of image segmentation at the semantic level is solved. Semantic a priori is an important high-level information in the detection of the ROI, which can assist the detection of the ROI. Therefore, this paper adopts FCNN to obtain high-level semantic prior knowledge and applies it to the detection of the ROI.

The network structure of FCNN is shown in Figure 4. Based on the original classifier, this paper utilizes the back propagation algorithm to fine-tune the parameters in all FCNN layers. In the network structure, the first row gets the feature map after alternately seven convolutional layers and five pooling layers. The last step of the deconvolution layer is to conduct the upsampling of the feature map with a step size 32 pixels. The network structure in this paper is denoted as FCNN-32s. It is found that the precision decreases because of the maximum pool operation. It directly executes upsampling for the feature map of downsampling, which will result in very rough output and details loss. Therefore, in this paper, the features with step size 32 pixels obtained from the upsampling are extended by 2 times, which is summed with the feature with step size 16 pixels. Then, the obtained feature is recovered to the original image for training, and the FCNN-16s model is obtained. So more accurate detailed information is obtained than that of FCNN-32s. We adopt the same method to train the network to obtain the FCNN-8s model, the prediction of detailed information is more accurate. Experiments show that although lower-level feature fusion for training networks can make detailed information prediction more accurate, the effect of low-rank sparse decomposition on the result is not significantly improved. Since the training time will increase sharply, this paper adopts FCNN-8s model to acquire the high-level priori knowledge of images.

The deep CNN model comprises an input layer, multiple convolution layers, downsampling layer, full connection layer, and output layer. The downsampling layer and

convolution layer form the intermediate structure of the neural network. The former is used for feature extraction, and the latter is for feature calculation. The fully connected layer is connected with the downsampling layer, which can output the feature. The output of the convolution layer is:

$$d_n^l = f\left(\sum_{\forall m}\left(d_m^{i-1} \cdot k_{m,n}^l\right) + b_m^l\right) \tag{8}$$

Where d_n^l and d_m^{i-1} are the feature maps of the current layer and the previous layer. $k_{m,n}^l$ is the convolution kernel of the model. $f(x) = 1/[1 + e^{-x}]$ is the neuron activation function. b_n^l is neuron bias. The feature extraction result of the downsampling layer is:

$$d_n^l = f\left(k_n^l \times \frac{1}{s^2}\sum_{s\times s} d_n^{l-1} + b_n^l\right) \tag{9}$$

Where $s \times s$ is the downsampling template scale. k_n^l is the template weight. In this paper, the trained GoogleNet model is used to extract the depth features of the proposal object region. On the strength of this model, the labeled output layer is removed to obtain a depth feature. The convolution layer C1 uses 96 filters with $11 \times 11 \times 3$ size to filter the input image with size $224 \times 224 \times 3$. The convolution layers C2, C3, C4, and C5 take the output of the downsampling layer as their input, respectively. The convolution processing is carried out by using the self-filter, and several output feature graphs are obtained and transmitted to the next layer. The full connection layers F6 and F7 have 4096 features. The output of each full connection layer can be denoted as:

$$d_n^{\text{out}} = f\left(\sum\left(d_n^{\text{out}-1} \times k_{m,n}^{\text{out}}\right) + b_n^{\text{out}}\right) \tag{10}$$

3.4. Saliency Calculation Based on Deep Feature. PCA [28] is the common method for dimension reduction of high-dimensional data, which can replace p high-dimensional features with a smaller number of m features. For n superpixels, the output features can constitute a sample matrix W with $n \times p$ dimension. The correlation coefficient matrix $R = (r_{ij})_{p\times p}$ of the sample is calculated by the formula (11):

$$r_{ij} = \frac{\sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{kj} - \bar{x}_j)^2 \sum_{k=1}^n (x_{ki} - \bar{x}_i)^2}}, i, j = 1, 2, \cdots, p. \tag{11}$$

Where $\bar{x}_i = 1/n\sum_{i=1}^n x_{ij}$. By solving the equation $|\lambda I - R| = 0$, we find the eigenvalues and order them. Then, we calculate the contribution rate and cumulative contribution rate of each eigenvalue λ_i:

$$con_{rate} = \lambda_i / \sum_{k=1}^p \lambda_k, cum_{rate} = \sum_{k=1}^i \lambda_k / \sum_{k=1}^p \lambda_k, i, j = 1, 2, \cdots, p. \tag{12}$$

We calculate the corresponding orthogonal unit vector $z_i = [z_{i1}, z_{i2}, \cdots, z_{ip}]^T$ of each eigenvalue λ_i. The unit vector corresponding to the first m features with a cumulative contribution rate 95% is selected to form the transformation matrix $Z = [z_1, z_2, \cdots, z_m]_{p\times m}$. The high-dimensional matrix m is reduced by formula (13). $Sp_i(df) = (f_{i,1}, f_{i,2}, \cdots, f_{i,m})$ denotes the m-dimension principal component feature. The principal component features are extracted by the same transformation matrix in the segmentation maps with different scales.

$$Sp_i(df) = W_{n\times p} Z_{p\times m} \tag{13}$$

3.5. Contrast Feature. The contrast feature reflects the difference degree between the region and its adjacent region. The contrast feature $w^c(Sp_i)$ of the superpixel Sp_i is defined by its distance from other superpixels features, as given in equation (14):

$$w^c(Sp_i) = \frac{1}{n-1}\sum_{i=1, i\neq k}^n \left\| Sp(df)_i - Sp(df)_k \right\|_2 \tag{14}$$

Where n denotes the number of superpixel. $\|\cdot\|_2$ is 2-norm.

3.6. Spatial Feature. In the human visual system, we pay different attentions in different spatial positions. The distance between the pixel at different positions and the image center satisfies the Gaussian distribution. For any superpixel Sp_i, its spatial feature $w^s(Sp_i)$ is calculated as:

$$w^s(Sp_i) = e^{\frac{-d(Sp_{i,x}, c)}{\sigma^2}} \tag{15}$$

Where $Sp_{i,x}$ is the central coordinate of superpixel Sp_i. c is the central region. If the average distance from the image center is smaller, the spatial feature is larger. The saliency value of the superpixel Sp_i is denoted as:

$$Map(Sp_i) = w^c(Sp_i) \times w^s(Sp_i) \tag{16}$$

We obtain the SM of the first segmented image and use it as the object prior knowledge to guide the proposal region extraction and optimization.

3.7. Saliency Detection Based on Low-Rank Sparse Decomposition. The background in the image can be expressed as a low-rank matrix. The saliency region can be regarded as sparse noise. For an original image, the eigenmatrix $F = [f_1, f_2, \cdots, f_N] \in R^{d\times N}$ and the eigentransformation matrix T are obtained. Then, we use the FCN to obtain the high-level prior knowledge P. The low-rank sparse decomposition of the transformed matrix is carried out by robust PCA.

$$(L^*, S^*) = \arg\min_{L,S}(\|L\|_\circ + \lambda\|S\|_1) \quad s.t.\ TFP = L + S \tag{17}$$

Input: Raw image I, multiscale segment number N and segment parameter in each scale.
Output: Saliency map.
for $i = 1 : N$
{
if i=1 then
(1) According to the determined parameters, we use SLIC to segment image I;
(2) Determine the input region Re ct_{self}, Re ct_{local}, Re ct_{global} of each superpixel;
(3) The above is input GoogleNet to extract deep feature F_{self}, F_{local}, F_{global};
(4) The deep features of all superpixels constitute a matrix W, and the transformation matrix A of W is calculated by using PCA to obtain the principal component features;
(5) According to the principal component features, saliency values without object priors are calculated to obtain the first segmentation saliency map Map_1;
else
(6) According to the determined parameters, we use Watershed algorithm to segment image;
(7) The saliency map Map^{i-1} is taken as object priori map. Then it extracts and optimizes proposal object set *Suppix*;
(8) Determine the input region Re ct_{self}, Re ct_{local}, Re ct_{global} in *Suppix*;
(9) The above is input GoogleNet to extract deep feature F_{self}, F_{local}, F_{global};
(10) The deep features of all superpixels constitute a matrix W, and the transformation matrix A of W is calculated by using PCA to obtain the principal component features;
(11) According to the principal component features, saliency values with object priors are calculated. And we obtain the saliency map Map_i;
end if
}
(12) Calculate the saliency map weight w_i at each scale;
(13) Adopt weight cellular automata to fuse the obtained N saliency maps and get final SM.

Algorithm 1: Proposed saliency detection method.

Where F is the eigenmatrix. T is the learned eigentransformation matrix. P is a high-level prior knowledge matrix. L is a low-rank matrix. S represents the sparse matrix. $\|\cdot\|_\circ$ represents the kernel norm of the matrix, that is, the sum of all singular values of the matrix. $\|\cdot\|_1$ represents the ℓ_1-norm of the matrix, the sum of the absolute values of all the elements in the matrix. Supposing that $S*$ is the optimal solution for the sparse matrix. The saliency map can be calculated by the following equation.

$$Sal(p_i) = \|S^*(:,i)\|_1 \tag{18}$$

Where $Sal(p_i)$ represents the saliency value of superpixel p_i. $\|S^*(:,i)\|_1$ represents the ℓ_1-norm of the ith column vector of $S*$, that is, the sum of the absolute values of all the elements in the vector.

3.8. Saliency Map Fusion Based on Weight Cellular Automata. Wang and Wang [29] adopted the multilayer cellular automata (MCA) for object fusion. Each pixel represents a cell. In the m-layer cellular automata, the cellular of the saliency map has m-1 neighbors. They are at the same positions in other saliency maps.

If cellular i is labeled as foreground, the foreground probability of its neighbor j at the same position in other SMs is $\lambda = P(\eta_i = +1 \mid i \in F)$. Saliency maps obtained by different methods are considered to be independent. When updating synchronously, all saliency maps are considered to have the same weight. There are guiding and refining relationships between the saliency maps at different segmentation scales. The weights cannot be considered as equally during the fusion process. In different segmentation scales, it is assumed that the weight of the SM obtained by the first segmentation scale is λ_1, represented by $w_i = \lambda_1$. The SM weight with different scale is expressed as:

$$w_i = \lambda_{i-1} + (1 - o_i/O_i), i = 1, 2, \cdots, 6 \tag{19}$$

Where O_i denotes the total pixel number in the proposal object set. o_i is the superpixel number in the ith saliency map. Set $\lambda_1 = 1$. Synchronous updating mechanism $f : Map^{M-1} \longrightarrow Map$ is defined as:

$$l\left(Map_m^{t+1}\right) = w_m \sum_{k=1,k\neq m}^{M} \operatorname{sign}\left(Map_k^t - \gamma_k \cdot I\right) \cdot \ln\left(\frac{\lambda}{1-\lambda}\right) + l\left(Map_m^t\right) \tag{20}$$

Where $Map_m^t = [Map_{m,1}^t, \cdots, Map_{m,H}^t]^T$ represents the saliency value of all the cellular of the mth SM at time t. Matrix I is a matrix with H elements. If the neighbor of cellular is judged as foreground, then the saliency value should be increased. We obtain the final saliency map by formula (21). $T2$ is the next time.

$$Map_{final} = \frac{1}{N} \sum_{m=1}^{M} \left(Map_m^{T_2} + Sal(p_i)\right) \tag{21}$$

The proposed deep multiscale fusion method for object saliency detection is summarized as depicted in Algorithm 1.

4. Experiments and Analysis

In this section, we obtain the experiment data from Google Earth. The remote sensing image size is from 512×512 pixel to 2048×2048 pixel. The spatial resolution is 1 m. The experiment environment is Intel(R), Core(TM), i7-8750, CPU2.2 Hz, Geforce GTX1060 with MATLAB 2017a platform.

4.1. Evaluation Index and Parameter Setting. In the experiment, the PR curve, F-measure, and mean absolute error (MAE) of the saliency map are compared to evaluate the effect of saliency detection to select a better segmentation scale.

Precision and Recall are the two most commonly used evaluation criteria in image saliency detection. If the PR curve is higher, the effect of saliency detection is better. Otherwise, it is poor. For the given manual labeled Ground Truth G and the saliency map S, the definition of Precision and Recall is given in equation (22):

$$\Pr ecision = \frac{sum(S, G)}{sum(S)}, \operatorname{Re} call = \frac{sum(S, G)}{sum(G)} \tag{22}$$

Where $sum(S, G)$ represents the sum of the value after the pixels of visual feature graph S multiplying that of G. $sum(S)$ is the sum of all pixels in the visual feature graph S. $sum(G)$ represents the sum of all pixels in G.

When calculating F-measure, the adaptive threshold T of each image is used to segment the image.

$$T = \frac{2}{W \cdot H} \sum_{x=1}^{W} \sum_{y=1}^{H} S(x, y) \tag{23}$$

Where the W and H denote the width and height of the image, respectively. It calculates the average precision and recall of the SM. The average F-measure value is calculated according to equation (24). The effect of saliency is better if the F-measure value is excellent. F-measure value is used for the comprehensive evaluation of accuracy and recall. β^2 is often set to 1.

$$F = \frac{2Pecision \cdot \operatorname{Re} call}{Pecision + \operatorname{Re} call} \tag{24}$$

MAE is used to evaluate the saliency model by comparing the difference between the SM and the GT. We use formula (25) to compute the MAE value of each input image. The calculated MAE value can be used to draw a histogram. If the MAE value is lower, the proposed algorithm is better.

$$MAE = \frac{1}{W \cdot H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x, y) - G(x, y)| \tag{25}$$

4.2. Segment Scale Determination. The main parameter of this algorithm is the segmentation scale. Many segmentation scales can increase the computational complexity. Few scales will affect the accuracy of saliency detection. Therefore, 15 segmentation scales are set according to experience. We conduct experiments on randomly selected remote sensing image data. Then, we extract the depth features of all superpixels in the segmentation graphs and calculate the saliency map. The histogram of the PR curve with different segmentation scales is shown in Figure 5. Three segmentation scales with better effects are selected from them. Through comparative analysis, it is found that the three segmentation scales 10, 11, 12 have a relatively better saliency detection effect. The three segmentation scales are selected as the final segmentation scales of the proposed method.

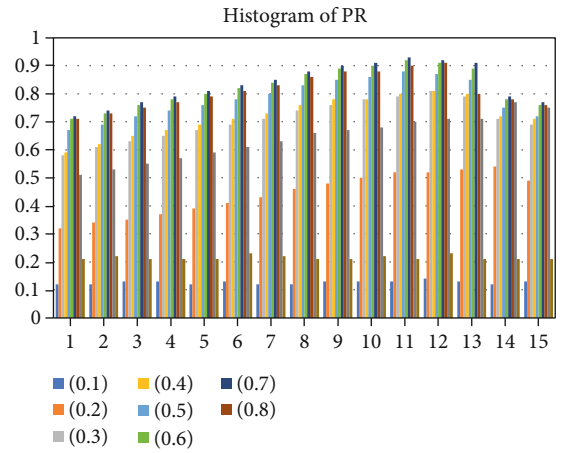


Figure 5: Histogram of PR curve with different segmentation scales.

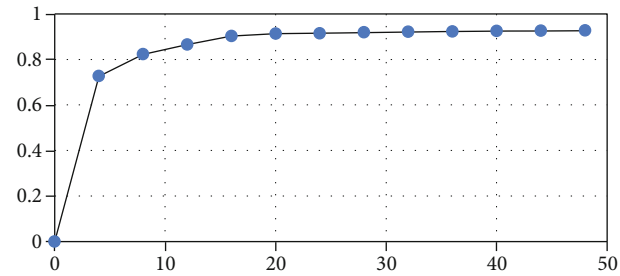
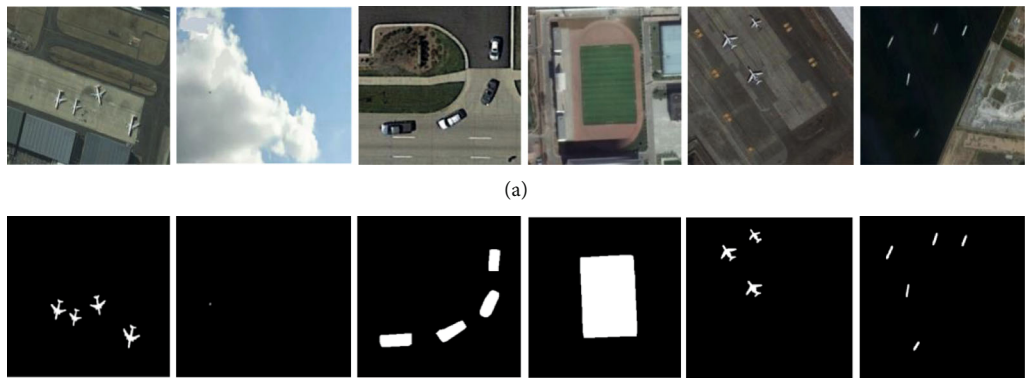


Figure 6: Relation between PEV and the principal components.

4.3. PCA Parameter Determination. To verify the effectiveness of PCA on selecting principal components from depth features, this section adopts the depth features extracted from each superpixel block as the data set. The percentage of explained variance (PEV) is used to measure the importance of the principal component in the overall data as formula (26). PEV is a main index to describe the distortion rate of data.

$$PEV = \sum_{i=1}^{m} R_{ii}^2 / tr\left(\sum\right) \tag{26}$$

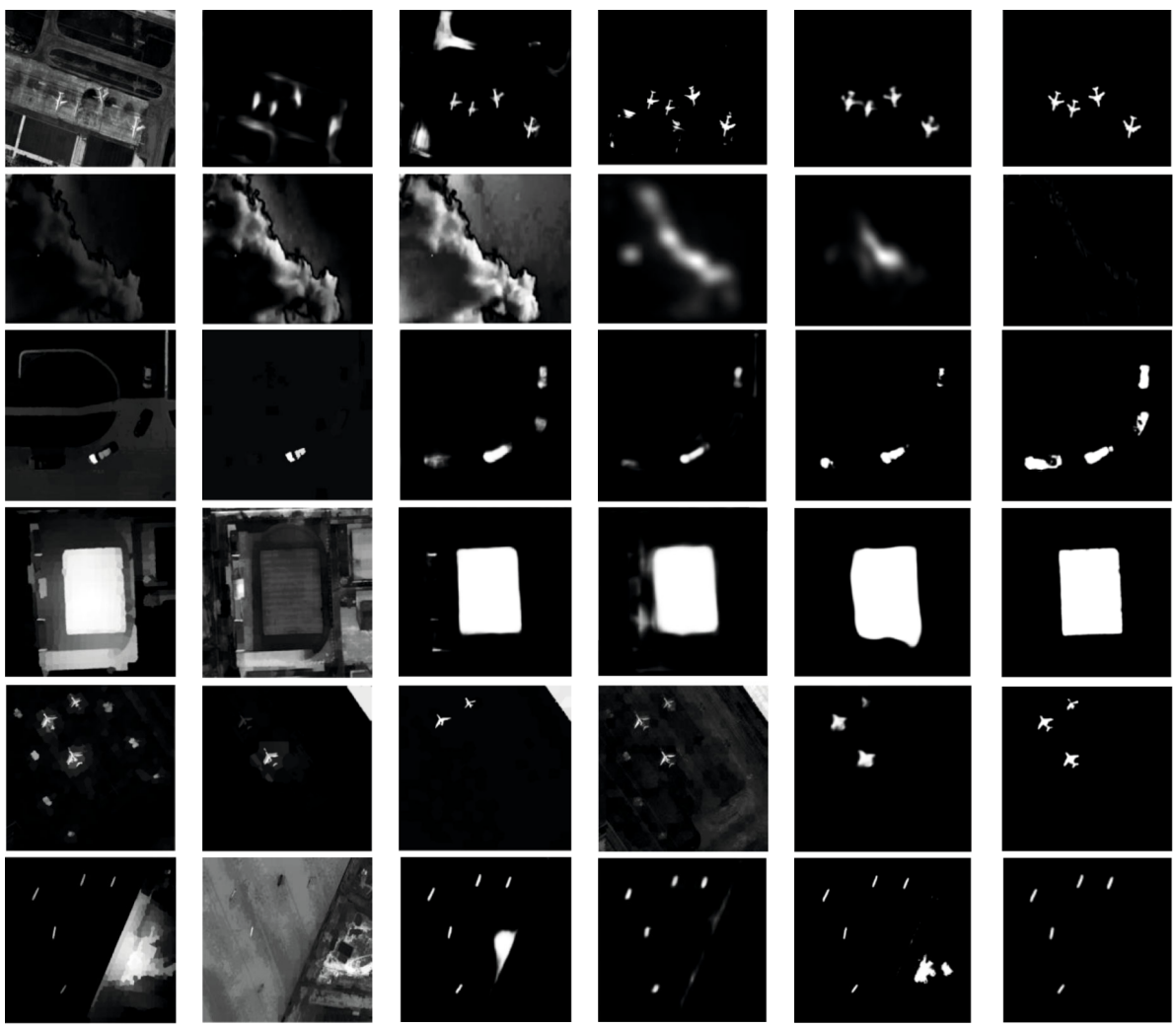
Where R_{ii}^2 is the right matrix of the main component matrix M' after singular value decomposition. $\sum$ denotes the covariance matrix. Figure 6 shows the relation between PEV and the top 50 principal components. It reveals that



(a)

(b)

Figure 7: Test images: airplane1, cloud, vehicle, playground, airplane2, boat. (a) Raw remote sensing images; (b) Ground Truth.



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

Figure 8: Comparison of saliency images. (a) RA. (b) RB. (c) SC. (d) RAD. (e) SCLR. (f) Proposed.

with the increase of principal component number, PEV shows an upward trend. But the trend grows slowly. When the number of principal components exceeds 20, the PEV reaches to 90%, which is considered to represent the overall information of the data. In this paper, the top 20 principal components are selected for saliency calculation.

Table 1: Different indexes with different methods on different objects.

Object	Method	Precision	Recall	F-measure	MAE
Airplane1	RA	79.9%	73.5%	72.8%	19.3%
	RB	81.1%	75.8%	76.4%	17.2%
	SC	81.8%	74.6%	77.2%	15.7%
	RAD	87.4%	77.4%	79.5%	14.6%
	SCLR	91.7%	75.4%	81.8%	12.5%
	Proposed	95.6%	65.3%	82.5%	9.8%
Cloud	RA	84.6%	68.9%	73.8%	21.2%
	RB	89.1%	71.8%	75.4%	17.8%
	SC	90.7%	74.1%	76.2%	14.1%
	RAD	91.6%	73.7%	78.4%	12.6%
	SCLR	93.6%	72.8%	80.9%	11.3%
	Proposed	97.4%	71.5%	83.6%	7.6%
Vehicle	RA	89.2%	77.4%	78.7%	19.5%
	RB	91.5%	79.9%	80.8%	17.6%
	SC	93.1%	79.3%	82.1%	13.8%
	RAD	93.6%	79.5%	82.5%	13.1%
	SCLR	94.3%	78.6%	83.7%	11.7%
	Proposed	98.2%	72.4%	89.1%	8.7%
Playground	RA	85.7%	77.8%	81.9%	16.5%
	RB	87.8%	74.2%	83.7%	14.8%
	SC	89.9%	74.6%	84.1%	13.1%
	RAD	91.8%	73.3%	84.5%	12.5%
	SCLR	92.4%	71.8%	86.7%	10.2%
	Proposed	97.2%	59.6%	89.7%	9.4%
Airplane2	RA	86.4%	78.9%	77.1%	15.8%
	RB	87.6%	78.3%	78.6%	14.6%
	SC	88.2%	77.1%	79.4%	13.5%
	RAD	88.3%	76.6%	80.8%	12.9%
	SCLR	91.3%	75.8%	81.6%	11.9%
	Proposed	96.3%	62.4%	83.9%	8.1%
Boat	RA	85.4%	84.1%	72.4	24.5%
	RB	88.6%	78.2%	74.1%	21.2%
	SC	89.7%	76.1%	76.4%	19.4%
	RAD	91.6%	74.8%	79.2%	15.7%
	SCLR	92.7%	73.4%	82.5%	11.4%
	Proposed	95.2%	68.3%	89.7%	7.4%

4.4. Saliency Detection Comparison with Other State-of-the-Art Methods. In this section, five state-of-the-art methods including RA [30], RB [31], SC [32], RAD [33], and SCLR [34] are conducted comparison with the proposed deep multiscale fusion method. And we conduct experiments on some optical remote sensing images based on urban data, namely airplane (512×512 pixel), playground (1024×1024 pixel), boat (1024×1024 pixel), vehicle (512×512 pixel), and cloud (2048×2048 pixel). Due to the limited space, we only display the results of several remote sensing objects. The test images

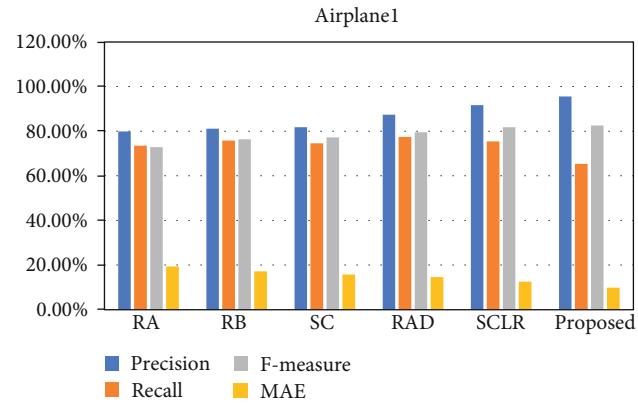


Figure 9: Airplane1 comparison with different methods.

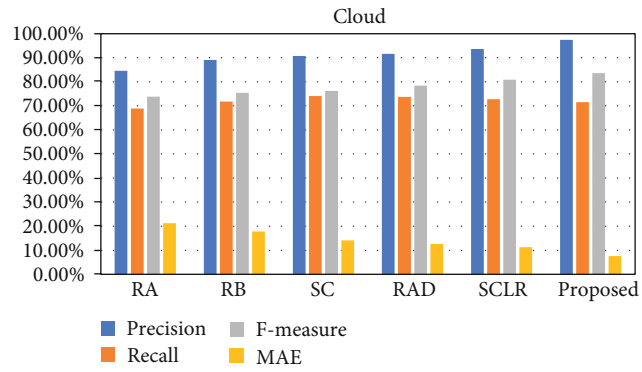


Figure 10: Cloud comparison with different methods.

along with their relevant GT maps are shown in Figure 7. Figure 8 displays the saliency results with different methods.

Figure 8 shows the comparison of saliency detection results with different methods. It can be seen that the detection effect of this algorithm is obviously better than other algorithms.

Table 1 is the F-measure result. With the change of Recall, the Precision of the method in this paper has better value and keeps a high level. However, in terms of F-measure value, our method is 7.18% higher than the second better method. Under the condition of complex background information, both the PR curve value and F-measure value of the proposed method are significantly higher than other algorithms. It fully demonstrates the advantages of the proposed algorithm in relatively complex image information. Similarly, the MAE of this proposed algorithm is lower than that of other algorithms. Figures 9–14 are the subjective evaluation results for the six objects.

We also adopt IoU (Intersection-Over-Union) to illustrate the effectiveness of the proposed method [35, 36]. The IoU is calculated as follows:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{27}$$

The greater IoU shows a better effect. The results are shown in Table 2.

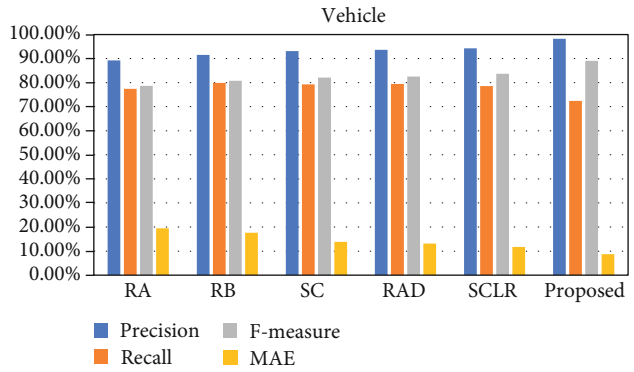


Figure 11: Vehicle comparison with different methods.

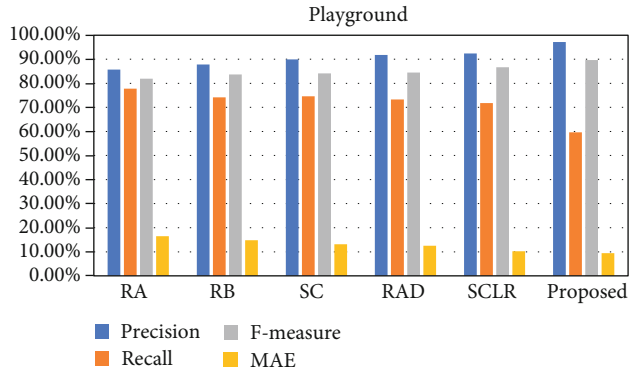


Figure 12: Playground comparison with different methods.

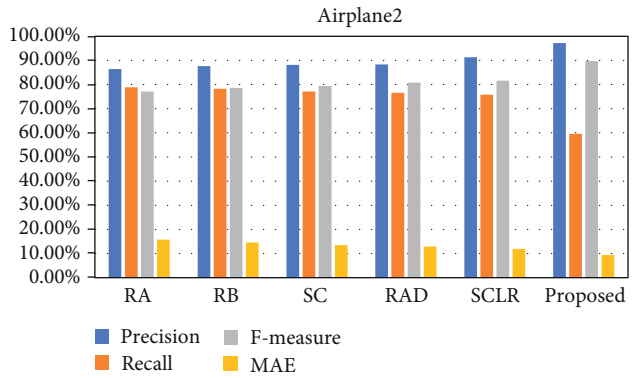


Figure 13: Airplane2 comparison with different methods.

From Table 2, we can see that our proposed method has a better saliency detection effect than other methods.

There are also apparent differences in the detection time among different algorithms. In terms of the speed of saliency detection, the proposed method is faster than other methods as given in Figure 15. Though deep learning-based algorithms need to train many samples, compared with other deep learning methods, the processing efficiency is improved by about 4%. Overall, the deep multiscale fusion method has a better effect on saliency detection for remote sensing images.

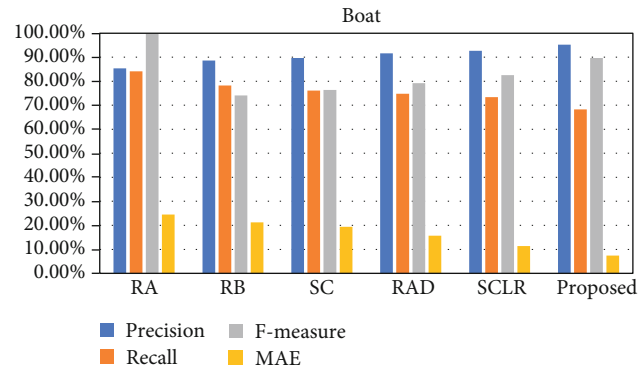


Figure 14: Boat comparison with different methods.

Table 2: IoU comparison.

Method	RA	RB	SC	RAD	SCLR	Proposed
Airplane1	69.3%	72.5%	76.7%	77.5%	79.8%	82.4%
Cloud	68.4%	72.7%	77.1%	79.2%	81.6%	83.5%
Vehicle	69.7%	73.8%	76.5%	78.5%	79.4%	81.6%
Playground	66.4%	72.9%	80.1%	81.4%	83.7%	86.4%
Airplane2	63.9%	71.3%	74.9%	76.4%	78.2%	81.9%
Boat	65.4%	70.8%	73.2%	75.9%	77.2%	79.5%

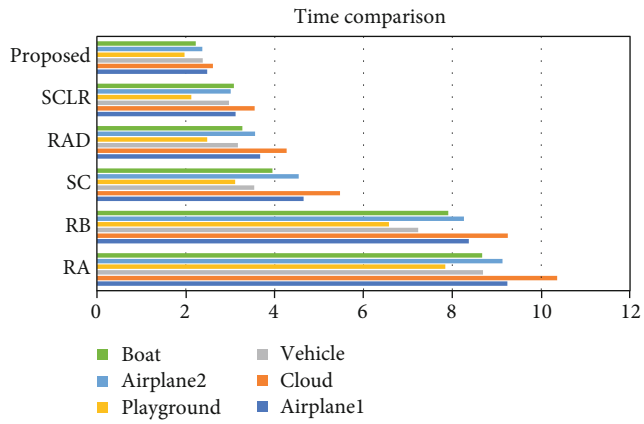


Figure 15: Time comparison with different methods.

5. Conclusions

The saliency detection algorithm based on DL can overcome the shortcomings of the traditional saliency detection algorithms. However, the detection efficiency is obviously insufficient. Therefore, we present a deep multiscale fusion method for object saliency detection in optical remote sensing images based on urban data. Through the deep feature extraction, we calculate the saliency value and use the weight cellular automata to integrate and optimize the scale saliency map. Results reveal that the proposed method can efficiently acquire the saliency detection results than other methods. In the future, some new models based on deep learning will be researched. And the new methods will be applied to practical engineerings.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Q. Zhang, L. Yang, Z. Chen, and P. Li, "Incremental Deep Computation Model for Wireless Big Data Feature Learning," in *IEEE Transactions on Big Data*, no. article 1, 2019.

[2] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep Convolutional Computation Model for Feature Learning on Big Data in Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.

[3] S. Yin, Y. Zhang, and K. Shahid, "Large Scale Remote Sensing Image Segmentation Based on Fuzzy Region Competition and Gaussian Mixture Model," *IEEE Access*, vol. 6, pp. 26069–26080, 2018.

[4] H. Quan, S. Feng, and B. Chen, "Two Birds With One Stone: A Unified Approach to Saliency and Co-Saliency Detection via Multi-Instance Learning," *IEEE Access*, vol. 5, pp. 23519–23531, 2017.

[5] C. Lang, J. Feng, S. Feng, J. Wang, and S. Yan, "Dual Low-Rank Pursuit: Learning Salient Features for Saliency Detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1190–1200, 2016.

[6] J. Zhu, Y. Qiu, R. Zhang, J. Huang, and W. Zhang, "Top-Down Saliency Detection via Contextual Pooling," *Journal of Signal Processing Systems*, vol. 74, no. 1, pp. 33–46, 2014.

[7] 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.

[8] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, Minneapolis, MN, USA, 2007.

[9] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, IEEE, Miami, FL, USA, 2009.

[10] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 409–416, IEEE, Providence, RI, 2011.

[11] Q. Zhang, C. Bai, T. Yang, Z. Chen, P. Li, and H. Yu, *A Unified Smart Chinese Medicine Framework for Healthcare and Medical Services*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2019.

[12] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An Incremental Deep Convolutional Computation Model for Feature Learning on Industrial Big Data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1341–1349, 2019.

[13] G. Lee, Y. W. Tai, J. Kim et al., "ELD-Net: An Efficient Deep Learning Architecture for Accurate Saliency Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1599–1610, 2018.

[14] Q. Zhang, C. Bai, Z. Chen et al., "Deep Learning Models for Diagnosing Spleen and Stomach Diseases in Smart Chinese Medicine with Cloud Computing," in *Concurrency and Computation: Practice and Experience*, p. e5252, 2019.

[15] W. Wang, J. Chen, J. Wang, J. Chen, and Z. Gong, "Geography-Aware Inductive Matrix Completion for Personalized Point of Interest Recommendation in Smart Cities," *IEEE Internet of Things Journal*, 2019.

[16] J. Yan, M. Zhu, H. Liu, and Y. Liu, "Visual saliency detection via sparsity pursuit," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 739–742, 2010.

[17] C. Lang, G. Liu, J. Yu, and S. Yan, "Saliency detection by multitask sparsity pursuit," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1327–1338, 2012.

[18] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 853–860, IEEE, Providence RI, USA, 2012.

[19] C. Dai, X. Liu, J. Lai, P. Li, and H. Chao, "Human Behavior Deep Recognition Architecture for Smart City Applications in the 5G Environment," *IEEE Network*, vol. 33, no. 5, pp. 206–211, 2019.

[20] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-Enhanced Collaborative Filtering for Personalized Point of Interests Recommendation," *IEEE Transactions on Industrial Informatics*, p. 1, 2019.

[21] C. Dai, X. Liu, and J. Lai, "Human action recognition using two-stream attention based LSTM networks," *Applied soft computing*, vol. 86, p. 105820, 2020.

[22] S. Yang, C. Zhao, and W. Xu, "A novel salient object detection method using bag-of-features," *Acta Automatica Sinica*, vol. 42, pp. 1259–1273, 2016.

[23] 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 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090, IEEE, Portland, OR, USA, 2013.

[24] Y. Li, Y. Xu, S. Ma, and H. Shi, "Saliency detection based on deep convolutional neural network," *Journal of Image and Graphics*, vol. 21, pp. 53–59, 2016.

[25] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proceedings of the Computer Vision-ECCV 2016. Lecture Notes in Computer Science, vol. 9908*, pp. 825–841, Springer, Amsterdam, Netherlands, 2016.

[26] G. Lee, Y. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 660–668, IEEE, LasVegas, NV, USA, 2016.

[27] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An Improved Stacked Auto-Encoder for Network Traffic Flow Classification," *IEEE Network*, vol. 32, no. 6, pp. 22–27, 2018.

[28] C. Yang, J. Pu, Y. Dong, G. S. Xie, Y. Si, and Z. Liu, "Scene classification-oriented saliency detection via the modularized prescription," *The Visual Computer*, vol. 35, no. 4, pp. 473–488, 2019.

[29] A. Wang and M. Wang, "RGB-D Salient Object Detection via Minimum Barrier Distance Transform and Saliency Fusion,"

IEEE Signal Processing Letters, vol. 24, no. 5, pp. 663–667, 2017.

[30] S. Chen, X. Tan, B. Wang et al., "Reverse attention for salient object detection," in *European Conference on Computer Vision*, Springer, Cham, 2018.

[31] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2814–2821, 2014.

[32] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5300–5309, 2017.

[33] X. Hu, L. Zhu, J. Qin, C.-W. Fu, and P.-A. Heng, "Recurrently aggregating deep features for salient object detection," *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 6943–6950, 2018.

[34] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3264–3274, 2018.

[35] J. Gao, P. Li, Z. Chen, and J. Zhang, "A Survey on Deep Learning for Multimodal Data Fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.

[36] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.