

Research Article

Airport Detection-Based Cosaliency on Remote Sensing Images

Zhen Hua ^{1,2}, Zhenzhu Bian ^{1,2} and Jinjiang Li ^{1,2}

¹School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China

²Institute of ZhongKe Network Technology, Yantai 264005, China

Correspondence should be addressed to Zhen Hua; huazhen66@foxmail.com

Received 7 August 2020; Revised 19 March 2021; Accepted 26 April 2021; Published 8 May 2021

Academic Editor: Silvestar Šesnić

Copyright © 2021 Zhen Hua 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.

This paper proposes a contour extraction model based on cosaliency detection for remote sensing image airport detection and improves the traditional line segmentation detection (LSD) algorithm to make it more suitable for the goal of this paper. Our model consists of two parts, a cosaliency detection module and a contour extraction module. In the first part, the cosaliency detection module mainly uses the network framework of Visual Geometry Group-19 (VGG-19) to obtain the result maps of the interimage comparison and the in-train image consistency, and then the two result maps are multiplied pixel by pixel to obtain the cosaliency mask. In the second part, the contour extraction module uses superpixel segmentation and parallel line segment detection (PLSD) to refine the airport contour and runway information to obtain the preprocessed result map, and then we merge the result of cosaliency detection with the preprocessed result to obtain the final airport contour. We compared the model proposed in this article with four commonly used methods. The experimental results show that the accuracy of the model is 15% higher than that of the target detection result based on the saliency model, and the accuracy of the active contour model based on the saliency analysis is improved by 1%. This shows that the model proposed in this paper can extract a contour that closely matches the actual target.

1. Introduction

Remote sensing image airport target recognition is an important topic in the field of remote sensing image processing. However, the pros and cons of detection algorithms seriously affect the accuracy of image recognition. Whether in the military or civilian fields, remote sensing image target recognition plays a vital role. In the military field, remote sensing image detection technology has become an important means of military reconnaissance and early warning [1]. In the civilian field, remote sensing image detection is often used in resource surveying, urban planning, map surveying, and mapping [2].

In response to the problem of remote sensing image airport detection, domestic and foreign researchers have achieved certain results. The previous remote sensing image airport detection methods are mainly divided into three types: edge-based, region-based segmentation, and saliency-based. With the development of deep learning, deep neural network algorithms have achieved significant development in the field of image detection, image saliency analysis, and

methods such as convolutional neural networks which have gradually emerged.

The edge-based method focuses on the artificial design representation of airport geometric features, runway. In 2011, Wang et al. [3] proposed a method based on perceptual organization, designed belief functions for the runway features. In 2012, Kou et al. [4] used line segment detection to identify airports. This type of method can achieve rapid detection of airport targets with better recognition results, but the presence of complex background noise and unrelated linear objects makes the airport targets difficult to predict.

The method based on region segmentation focuses on the salient structural features of the airport. And it is mainly based on the difference between the airport and the surrounding texture. Image segmentation achieves the purpose of extracting candidate regions, and then the features in the candidate regions are identified. Liu et al. [5] proposed the classifier based on the KMP algorithm, segmented aerial images and classified them into different textures, and extracted regions of interest based on the results of texture segmentation. This method needs to be analyzed pixel by

pixel during image segmentation, which is slow and complex. The recognition effect depends on the selection of the segmentation threshold.

The method based on the saliency region of the image mainly uses various saliency cues to replace the traditional airport feature description. In 2011, Tao et al. [6] proposed to use matching scale-invariant feature transform (SIFT) feature key points to cluster information and at the same time to extract regional information through image segmentation. In 2012, Wang et al. [7] proposed a method based on the saliency area of the image, introducing the human eye's attention selection calculation model to the remote sensing image airport detection, in which the Hough transform is used to initial screening whether there is an airport target in the remote sensing image, and then we use the proposed model to extract salient regions; the detection speed is faster. In 2015, Zhu et al. [8] proposed a top-down and bottom-up saliency method to form a saliency map, used SIFT and support vector machines (SVMs) to determine whether the candidate area for separating the airport target is correct.

Based on deep learning methods, the convolutional neural network is one of the representative algorithms of deep learning. The convolutional neural network can classify input information with translation invariance, which can perform supervised and unsupervised learning. Li et al. [9] summarized the existing representative deep learning-based target detection algorithms and proposed a large-scale, publicly available data set, but the purpose of the target detection in this article is slightly different, so we did not adopt it. Zhang et al. [10] and Xiao et al. [11] used convolutional neural networks to extract high-level features and hierarchical representations of targets. In 2017, Cai et al. [12] and Zhang et al. [13] applied convolutional neural networks to remote sensing images. In the airport detection and recognition task, they greatly improve the speed and accuracy of detection. Cai et al. achieved airport detection in a complex background by improving the area suggestion network and loss function method. However, due to insufficient samples, the detection frame may contain incomplete or multiple bounding frames at an airport. In the original convolutional neural network, Zhang et al. replaced the original RPN network with a weak semantic rule for line segmentation. Although efficiency has been improved, false detection targets have also increased, with an overall detection accuracy of 88.8%. Cheng et al. [14] aimed at problems such as target rotation and intraclass diversity that would interfere with the detection performance of the target, improved the detection performance of the target by optimizing the target function, built up the existing state-of-the-art object detection systems, and proposed a simple but effective method to train rotation-invariant and Fisher discriminative CNN models to further boost object detection performance.

Methods based on convolutional neural networks (CNNs) can uniformly present highly reliable detection results. Generally, supervised detection algorithms have better recognition rates than unsupervised methods. At the same time, there are the following shortcomings: (1) a large number of image samples accurately marked by the

researcher are required, which makes the work dependent on labor; (2) the reusability of the model is relatively reduced; (3) the process of pattern matching and sample training may be very time-consuming and largely determined the quality of the entire detection framework.

The concept of cosaliency first appeared in 2010. For a pair of similar images, Jacobs et al. [15] used machine learning to fuse multiple features and finally found the common salient objects in the two images, called cosaliency. To a certain extent, it provides better information for image classification. By 2013, Fu et al. [16] proposed cluster-based cosaliency detection, using bottom-up clues to express saliency, at the same time, through the comparison between images and images to determine whether the target is significant, to a certain extent of data processing speed. In 2014, multiple saliency maps generated using different single-image saliency models are combined based on the rank-one constraint to obtain cosaliency maps in [17]. In 2015, Ye et al. [18] combined the color and SIFT features of a group of images to form overall similarity, found the common salient area of each image as a sample, and obtained the sample saliency map using the saliency model of a single image. The local and global restoration of the salient area is combined, and the boundary connectivity and the focus of the area of interest are used to generate a common salient image set. In 2019, Zhang et al. [19] proposed a hierarchical image cosaliency detection framework as a coarse to fine strategy to capture this pattern and used a mask-guided fully convolutional network structure to generate the initial cosaliency detection result. Tsai et al. [20] proposed stacked autoencoder-enabled fusion (SAEF) to carry out the saliency proposal fusion via jointly exploring the image-level confidence based on the reconstruction error of stacked autoencoder (SAE) and the region-level confidence from cosalient object likelihood, and self-trained convolutional neural networks (STCNNs) can gradually learn cosalient objects in a self-taught fashion.

In summary, there are two main methods of airport target detection for remote sensing images. One is unsupervised based on airport feature modeling, and the other is a method of introducing a supervised learning mechanism into target detection. By detecting the runway area, the airport target is obtained. The first method focuses on the geometric characteristics of the airport, and the main basis method is line detection. The second method mainly uses the saliency clues of the image to replace the airport features. Generally, the second method has a better recognition rate, but the image needs to be manually labeled, resulting in lower repetitiveness of the model.

Our main contributions are summarized as follows:

- (1) This paper improves a straight-line detection algorithm LSD. According to the characteristics of the airport runway, we have improved the existing straight-line segmentation algorithm. The improvement is mainly based on the length, width, and parallelism of the runway. The line segment is screened in three directions, and the shortness and the difference from the airport runway are screened out. Too many line segments reserve the suspected

airport area. The improved line detection algorithm is more conducive to the location of airport targets than the original LSD algorithm.

- (2) This paper builds a cosaliency detection model. We use VGG-19 to extract the deep features of the image, merge the saliency within the image and the similarity between the images, and call it cosaliency. The model proposed in this paper is easy to reproduce, and the image does not need to be manually marked, which saves time and cost to a certain extent.

2. Related Work

2.1. Visual Saliency Detection. Visual saliency detection refers to the extraction of salient areas in an image by simulating human visual characteristics through intelligent algorithms, that is, the area that humans are most interested in when seeing an image, and the specific process is shown in Figure 1.

The development of image saliency detection is mainly divided into three stages. The first stage is as follows: Itti et al. proposed a saliency-based visual attention model in 1998 and further improved the model theory. Itti's saliency model has become the standard of the bottom-up visual attention model. The second stage is as follows: Liu et al. and Achanta et al. defined saliency detection as a binary segmentation problem. Since then, a large number of saliency detection models have appeared. The third stage is as follows: the convolutional neural network (CNN) [21, 22]. The model based on the convolutional neural network usually contains more than thousands of adjustable parameters and neurons with variable receptive field size [23, 24]. Neurons have a larger acceptance range to provide global information, which can help better identify the most prominent areas in the image. CNN can achieve unprecedented performance, which has gradually become the mainstream direction of salient object detection. Cornia et al. [25] combined convolutional neural networks and stacked autoencoders to predict eye attention points. In addition, there are methods based on frequency domain analysis [26, 27] and graph theory [28, 29].

2.2. Cosaliency Detection. Cosaliency detection is a weakly supervised extension of saliency detection. To a pair of images, Chen [30] proposed a progressive algorithm based on the distribution of sparse representations to enhance the response before the attention shift. Li and Ngan [31] combined conventional saliency detection methods to calculate the single-image saliency and then applied a comultilayer graph model to obtain the multi-image saliency.

Li et al. [32] linearly fused the saliency maps between images and the saliency maps within images and extracted the common significant objects from a set of images. In order to further improve the performance, the fusion-based method combined several saliency models to exclude a single prediction bias while retaining shared information. For this reason, such methods use fixed weight fusion, adaptive weight fusion [17], or regional adaptive fusion [33–36] to integrate the saliency suggestions generated by different models. The method based on deep learning [37–39] is very

effective in extracting semantic object information in complex scenes and greatly enhances cosaliency detection. However, these methods work in a supervised manner and require pretrained deep models or labeled training data. In addition, the supervised settings reduce their versatility for handling objects of invisible categories.

2.3. Active Contour Model. Contour refers to a curve that connects points with the same gray value or color. Extracting a contour is to extract these curves with the same color or gray value. Generally, a contour search is performed in a binary graph; that is, a white object is found on a black background. In 1985, Suzuki and Be [40] mainly introduced the surrounding relationship algorithm to determine the boundary of the binary image.

Generally speaking, active contour models include edge-based and region-based, and they are mainly used in medical images, such as cancer and tumors. Edge-based methods such as Sobel operator, Canny operator, and Laplacian operator. Jing et al. [41] proposed an edge detection technology based on local minimization, and Li et al. [42] proposed an edge detection technology based on deep learning, which is mainly for cancer images.

The region-based active contour model such as C-V is a typical regional active contour model, and many subsequent active contour models are improved based on it. It uses the pixel gray information of the image as energy, cleverly constructs the energy function, and then obtains the minimum value of the energy function to finally segment the target. After that, many people improved it and proposed an active contour model based on local area [43, 44].

2.4. Airport Detection. The development of deep learning has led the trend of industry and academia and also provided a new idea for airport target detection in remote sensing images. According to the characteristics of airport runways, Zhang et al. [45] and others used the transfer learning ability of convolution neural networks to identify airport runways and then used the transfer learning ability of convolution neural networks to directly identify airports. Xiao et al. [11] used convolution neural networks to extract multiscale deep fusion features of images and then used support vector machines for classification. Although airports across the country are different, most of them are similar. The above methods make full use of the powerful classification and recognition capabilities of convolution neural networks. Each image is analyzed separately, which increases the amount of network calculations to a certain extent. Therefore, this paper proposed cosaliency detection method and applied to remote sensing image airport target detection, which reduces the network running time to a certain extent and improves the accuracy of airport detection.

3. Model Building

In the field of computer vision, VGG-19 is mainly used for image recognition and classification, image feature extraction, etc. It can also be used as a pretrained model to apply to

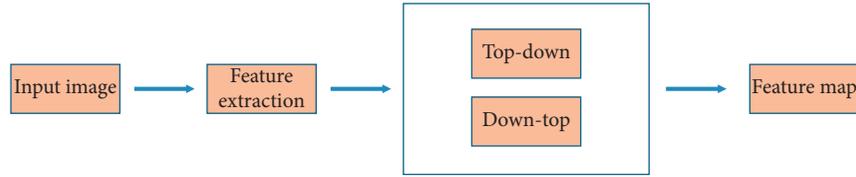


FIGURE 1: Visual saliency detection model flow. Feature extraction is performed on the input image pixel. Add or multiply the top-down and bottom-up features to obtain the feature map of the input image.

its tasks. This paper mainly uses the 16 convolution layers and 5 pooling layers of the VGG-19 network to preprocess remote sensing images and uses them to extract the depth features of the image. We call the feature map extracted by VGG-19 as a mask and let it as the input of the fourth and sixth layers of the cosaliency contour extraction module. The feature map fusion is used to generate a cosaliency prediction map. Finally, the saliency map of simple linear iterative clustering (SLIC), PLSD, and cosaliency prediction is combined with contour extraction, the cosaliency prediction map is refined, and finally, the remote sensing image airport target extraction based on cosaliency detection is completed. The flowchart is shown in Figure 2.

Algorithm steps:

- (i) Use SLIC for superpixel segmentation.
- (ii) Use PLSD algorithm to segment the image.
- (iii) Take 9000 training set images and their corresponding truth maps as the input of the VGG-19 network. After 16 layers of convolution and 5 layers of pooling, the final output image size is $7 \times 7 \times 512$ feature map mask.
- (iv) Design a cosaliency detection module to compare and fuse the clues in a single image and between multiple images to obtain a predicted cosaliency map.
- (v) Design a cosaliency contour extraction module to fuse the results of superpixel segmentation, line segmentation, cosaliency detection prediction map, and contour extraction method to obtain the airport target map of the remote sensing image.
- (vi) Optimize the entire network.

3.1. Superpixel Segmentation Model. In computer vision, image segmentation is a very important and basic research direction. Common terms in image segmentation include superpixel, semantic segmentation, instance segmentation, and panoramic segmentation. The concept of superpixel was first proposed by Ren [46] and others in 2003. One of the more commonly used methods in superpixel is SLIC (simple linear iterative clustering), which is a method proposed by Achanta et al. [47] in 2010. The algorithm is simple in thinking and easy to implement. The color image is converted into a 5-dimensional feature vector in CIELAB color space and XY coordinates, and then a distance metric is constructed for the 5-dimensional feature vector. The process of local clustering of image pixels is determined by

k-means algorithm evolved and has been widely used. By comparing with a variety of segmentation algorithms, the SLIC algorithm has a high comprehensive evaluation in terms of computing speed, object contour preservation, etc. SLIC can generate compact and approximately uniform superpixels, which is more in line with the expected segmentation effect, so this article uses the SLIC algorithm to preprocess the image.

As shown in Figure 3, the colored dots in Figure 3(a) represent different clustering centers. Each clustering center is called a superpixel, which is represented by K . Assuming an image has X pixels, the size of each pixel can be expressed as X/K . The distance between superpixels is as follows:

SLIC algorithm steps are as follows:

- (i) Initialize the seed point. Distribute the centers of K superpixels to the pixels of the image.
- (ii) Fine-tune the position of the seed. Within the range of s centered on K , move the center of the superpixel to the point with the smallest gradient among these 9 points, so as to avoid the superpixel point falling on the contour boundary with a larger gradient, so as not to affect the subsequent clustering effect.
- (iii) Initialize the data. Take an array label to store which superpixel each pixel belongs to. The disarray holds the distance from the pixel to the center of the superpixel to which it belongs.
- (iv) For each superpixel center x , it is a point within $2S$: if the distance (5 dimensions) from the point to the superpixel center x is less than the distance from this point to the superpixel center to which it originally belongs, then this point is a superpixel. Update label for the pixel x .
- (v) For each superpixel center, recalculate its position.
- (vi) Iterative optimization, repeat steps (iv) and (v) until the error converges. Generally, the number of iterations is 10.
- (vii) Enhance connectivity. Create a new label table, the elements in the table are -1 , follow the Z-shaped trend from left to right, from top to bottom, redistribute discontinuous or too small superpixels to adjacent super pixels, and traverse the pixels after points are assigned to the corresponding labels until all the pixels are traversed.

It can be seen from Figure 4 that the target regions after superpixel segmentation are roughly in the same cluster, which makes a good preparation for the subsequent contour extraction part.

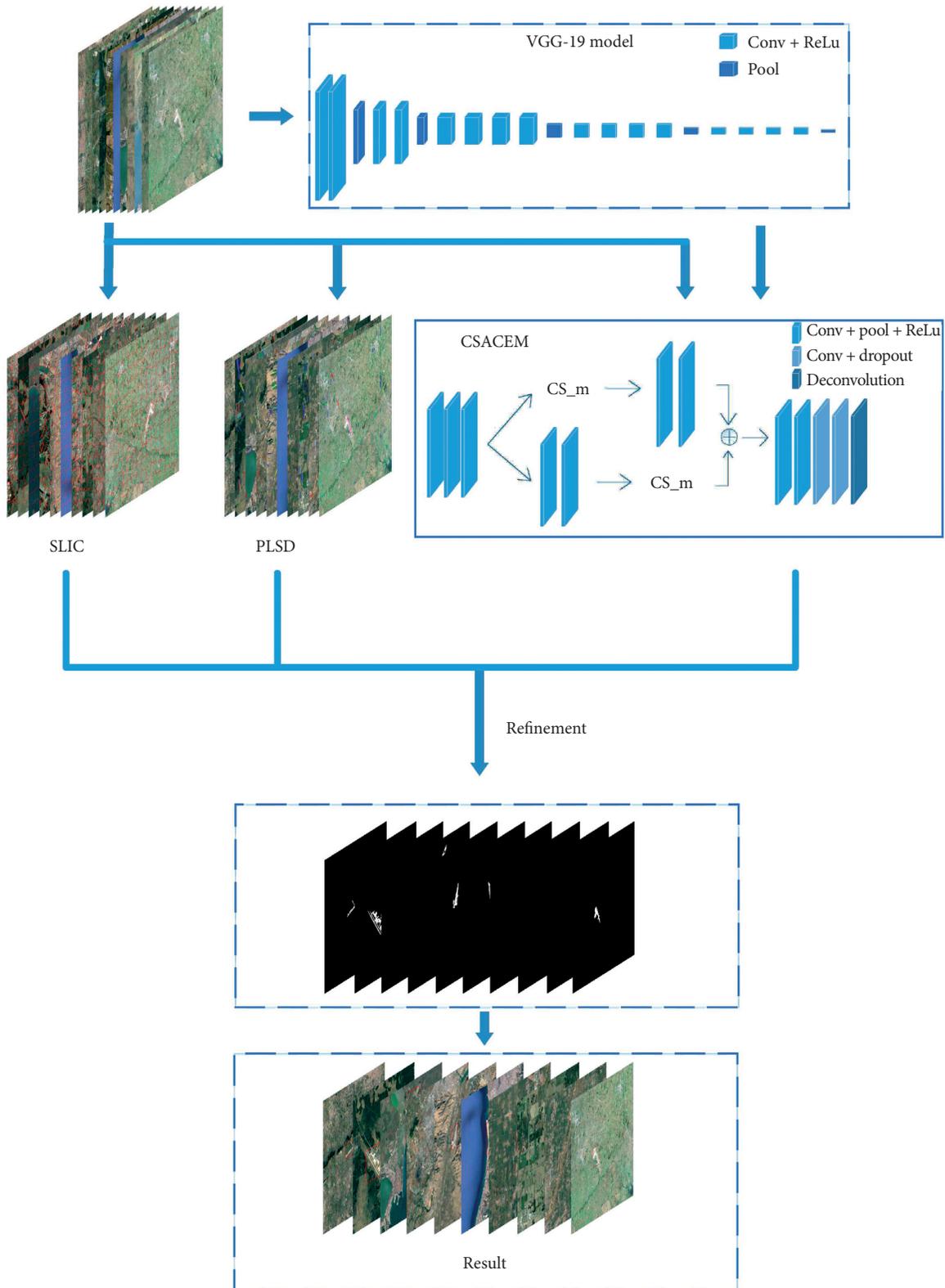


FIGURE 2: The airport detection model of remote sensing images.

3.2. *Straight Line Segmentation Module.* We all know that an airport is composed of a flight area, a ground transportation area, and a terminal area. From Figure 5, we can clearly see that the size of the airport in each place is

different, and the architectural style is also very different. However, the similarity of the airport is also obvious; that is, the airport runway is composed of parallel long straight lines.

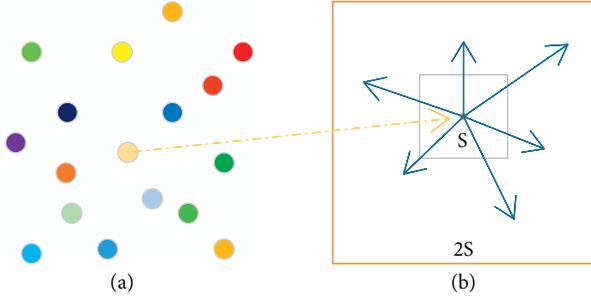


FIGURE 3: Selection of clustering centers in the SLIC algorithm: (a) the different cluster centers; (b) the area to look out for the cluster center.

There are three main algorithms for detecting line segments: line segment detector (LSD), edge drawing line detector (EDLD), and Hough line detector (HLD). After comparison, the line segment detection algorithm of LSD has the best effect. Therefore, we improve the LSD algorithm proposed in [48], mainly by adding the distance limit of parallel straight lines, which makes it easier to detect the airport runway and assist in locating the location of the airport in the remote sensing image.

The detection goal of LSD is to detect local straight contours in the image. The improved PLSD in this paper mainly detects the contours of parallel lines. The design feature of the airport runway is that the length and width are basically the same, and the distance between the two line segments is mostly fixed and consistent and can be well distinguished from street roads. The gradient is calculated as follows:

$$\begin{aligned} g_x(x, y) &= \log\left(\frac{I(x+1, y) + I(x+1, y+1)}{I(x, y) + I(x, y+1)}\right), \\ g_y(x, y) &= \log\left(\frac{I(x, y+1) + I(x+1, y+1)}{I(x, y) + I(x+1, y)}\right). \end{aligned} \quad (1)$$

The horizontal angle is calculated as follows:

$$T(x, y) = \arctan\left(\frac{g_x(x, y)}{-g_y(x, y)}\right). \quad (2)$$

If the horizontal angles in the above formula are the same, it proves that the two line segments are parallel. At this time, use the following formula to calculate the length of the two line segments:

$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)}. \quad (3)$$

LSD is a very effective line segment detection model. The runway feature extraction layer uses the improved LSD algorithm to extract straight lines and generate line density maps. For the superpixel area r_k , define the weight of line segment:

$$LD(r_k) = \frac{O_L(r_k)}{O(r_k)} \times Q_L(r_k). \quad (4)$$

In formula 5, $O_L(r_k)$ is the number of pixels in the parallel line segment detected by PLSD, and $O(r_k)$ is the

number of all pixels in the superpixel area. The $Q_L(r_k)$ items are used to measure the quality of the line segment and consist of two parts. The first part is the length of the line segment, and the second part is the distance between parallel line segments. The specific definition is as follows:

$$Q_L(r_k) = \sum_{i=1}^n \text{Length}(l_i) + \sum_{j=1}^n \text{Parallel}(p_j), \quad (5)$$

where $L(r_k) = \{l_1, l_2, \dots, l_i\}$, $i = 1, 2, \dots, N$, $P(r_k) = \{p_1, p_2, \dots, p_j\}$, $j = 1, 2, \dots, N$.

Based on the line density map generated by PLSD, the paper uses Gaussian form to formulate LKS, so high-density areas can be highlighted:

$$LKS(r_k) = \exp\left\{\frac{LD(r_k)}{\sigma^2}\right\}, \quad (6)$$

where $\sigma^2 = 0.4$, according to the survey, the actual runway width is generally 60 meters, and according to the satellite image and the actual ratio of 1 : 1000, therefore, if the length of the runway is equal and the distance is greater than 0.05 meters, it is considered an airport runway.

Compared with the LSD algorithm proposed in [49], the improved method PLSD in this paper reduces the false alarm rate and improves the accuracy of target detection. An example is shown in Figure 6.

It can be seen from Figure 6 that compared with the traditional LSD algorithm, the improved article contains fewer small line segments, and the quality of the detected straight line is relatively high. The length and parallel limit of the line segments effectively reduces the other target's interference with the airport runway.

3.3. Cosaliency Detection Module. Given N remote sensing images as the input of the network, where $A = \{I_n\}_{n=1}^N$, each single input remote sensing image I has the salient object of the airport. Our goal is to detect all salient objects in the data set A .

This article first designs two networks g_s and g_c , and both of which use the structure of the full convolutional neural network (FCN) model [50]. The structure of the convolutional layer and the pooling layer is the same as that of VGG-19. The g_s network is used to extract the saliency clues of each image, and the g_c network is used to extract the saliency features between images in a set of images. Secondly, the two detection results are multiplied and fused by pixels and finally obtained by deconvolution and input the predicted cosaliency map of the same size is marked as mask. The schematic diagram is shown in Figure 7, and the network layer design is shown in Table 1.

The feature map after the fusion of the two networks is called a cosaliency map, denoted as $B = \{F_n\}_{n=1}^N$, $F_n \in R^{w \times h \times c}$, where w represents width, h represents height, and c represents channel.

The objective function proposed in this paper is

$$Y(w) = \sum_{n=1}^N \psi_s(I_n; w_s) + \sum_{n=1}^N \psi_c(\{I_n\}_{n=1}^N; w_c), \quad (7)$$

where $\psi_s(I_n; w_s)$ is the saliency detection of I_n and $\psi_c(\{I_n\}_{n=1}^N; w_c)$ is the common saliency detection of a group

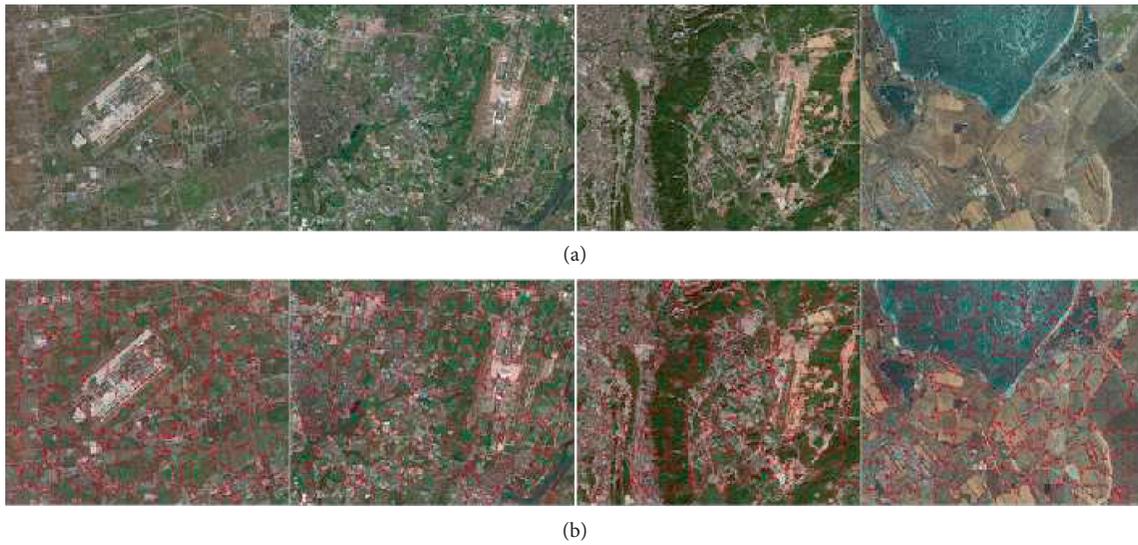


FIGURE 4: Superpixel segmentation map: (a) the original image; (b) superpixel segmentation map.

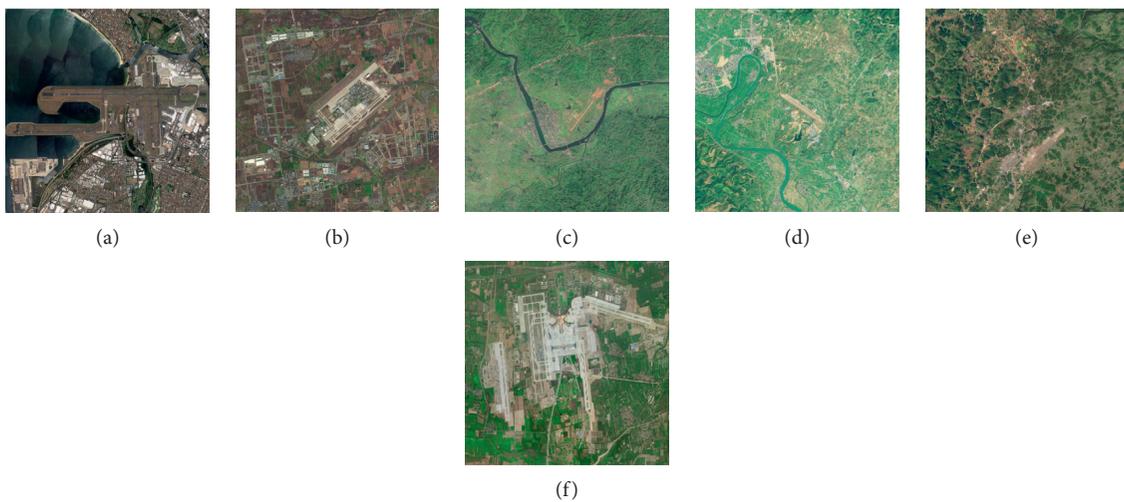


FIGURE 5: Remote sensing images of different airports. (a) Sydney Airport, (b) Xi'an Xianyang International Airport, (c) Zhijiang Airport, (d) Bama Airport, (e) Tongren Airport, and (f) Beijing Daxing International Airport.

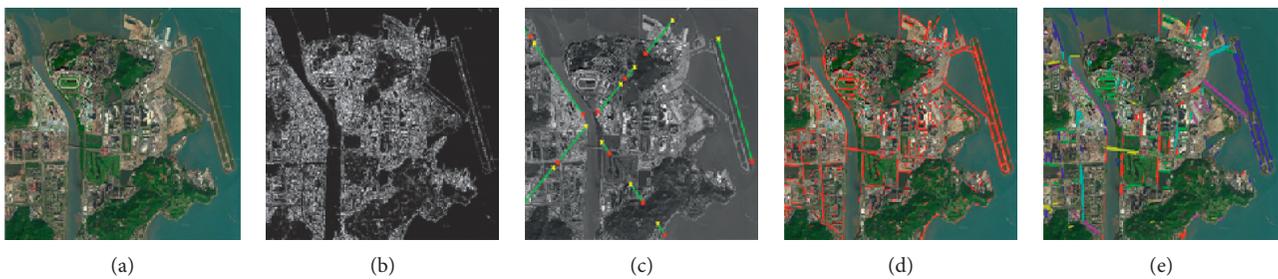


FIGURE 6: Comparison of results of different line segment detection algorithms: (a) the original image; (b) detected by the line detection operator, (c) the result of the line detected by the Hough transform, (d) the line segment image detected by the LSD algorithm, and (e) the improved LSD algorithm PLSD.

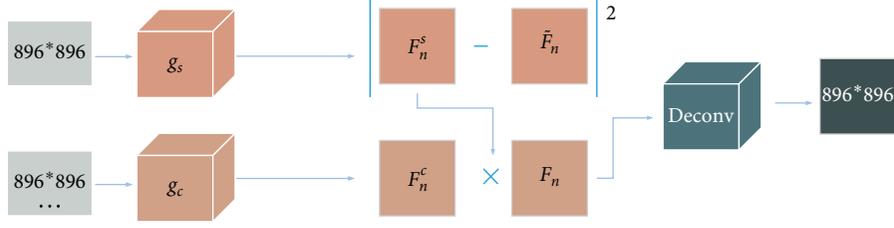


FIGURE 7: Schematic diagram of cosaliency detection.

TABLE 1: Network design of the deconvolution layer.

	Layer	Filter	Stride	Output size
Input				896 * 896 * 3
VGG-19				7 * 7 * 512
Feature concatenation	Concat			7 * 7 * 1024
	conv1	1 * 1	1	7 * 7 * 256
Cosaliency prediction	deconv2	4 * 4	2	14 * 14 * 512
	Addition			14 * 14 * 512
	deconv3	4 * 4	2	28 * 28 * 512
	Addition			28 * 28 * 512
	deconv4	4 * 4	2	56 * 56 * 256
	Addition			56 * 56 * 256
	deconv5	4 * 4	2	112 * 112 * 256
	Addition			112 * 112 * 256
	deconv6	4 * 4	2	224 * 224 * 256
	Addition			224 * 224 * 256
Output	deconv7	8 * 8	4	896 * 896 * 1
				896 * 896 * 1

of images, w represents the parameters of the model, $w = w_s \cup w_c$, w_s , and w_c are all learnable parameters of the network.

We learn two FCN models, denoted as g_s and g_c , respectively. g_s is used to generate the saliency map F_n^s of a single image, and g_c is used to extract the common saliency between the entire group of images and generate the saliency map F_n^c . The final saliency map is calculated by the following formula:

$$F_n = F_n^s \times F_n^c. \quad (8)$$

Term ψ_s : through training the FCN network, the saliency map F_n^s of image I_n is generated. At the same time, this paper uses the saliency detection algorithm proposed in [51] to detect the saliency of I_n and obtain the feature map \tilde{F}_n , which is used as the output reference target of the g_s network:

$$\psi_s(I_n; w_s) = \sum_{i \in I_n} r_n(i) |F_n^s(i) - \tilde{F}_n(i)|^2, \quad (9)$$

where i means pixel, $F_n^s(i)$ means saliency score, that is F_n^s , corresponding to pixel i , $\tilde{F}_n(i)$ means \tilde{F}_n value corresponding to pixel i , $r_n(i)$ means importance score of pixel i , and if pixel i belongs to the salient area, then let $r_n(i) \in (1, \delta)$, where δ is the salient pixel. According to the ratio of pixels to the entire image, the image F is divided into two parts. The image \tilde{F}_n is divided into two parts: the foreground and the background, namely, the salient area and the insignificant area. The distinguishing criterion is the average value of \tilde{F}_n . If the pixel i is greater than the average value of \tilde{F}_n , it is significant; otherwise, it is not significant.

Term ψ_c : the purpose of ψ_c item is to find the common part between a group of images, which is also the goal of learning network g_c . This also requires that the images input by the network have common objects, so that we can extract the similar parts between the images. For remote sensing images, the background is complex and there are many targets, and it is quite difficult to extract the objects of interest from them. We will have the airport target, but the remote sensing images in different backgrounds are used as a set of input, for example, mountain, city, seaside, and other backgrounds. In this way, it is relatively easier for us to extract the target. The airport area should look similar in a set of images, but it is not the same as the surrounding areas detected. Therefore, the similarity of the salient areas in each image is considered in the design of the network, which is the main factor considered by the g_c network.

Image I_n is generated through network g_c to generate a total saliency map $F_n^c = g_c(I_n)$. The last activation layer of the network uses the sigmoid function, and the value $F_n^c(i)$ of the total saliency map is between 0 and 1, which is $F_n^c(i) \in [0, 1]$. In this part, image I_n will be divided into two parts, the foreground and the background, which can also be called the cosaliency area and the remaining area of the image, denoted as I_n^o and I_n^b :

$$\begin{aligned} I_n^o &= F_n^c \times I_n, \\ I_n^b &= (1 - F_n^c) \times I_n. \end{aligned} \quad (10)$$

As a measure of the similarity between images, this paper uses feature extractor Φ to calculate the features of an image and at the same time applies feature extractor Φ to all mask images to obtain feature $\{\Phi(I_n^o), \Phi(I_n^b)\}_{n=1}^N$. Based on the obtained features, ψ_c can be defined as

$$\psi_c(\{I_n\}_{n=1}^N; w_c) = -\log(M_{\{I_n\}_{n=1}^N}), \quad (11)$$

where $M_{\{I_n\}_{n=1}^N}$ is the estimated score of the common significant area between the detected images, which is composed of the distance between the foreground and the background between any two images and the distance between the foreground and the background in a single image. After the learning of the two networks g_s and g_c , the cosaliency map of image F_n is produced by $F_n = g_s(I_n) \times g_c(I_n)$.

In remote sensing images, the target we pay attention to often occupies a small part of the entire image; that is, the proportions of salient and nonsalient areas are very unbalanced. Therefore, we need asymmetric weighted loss, which can improve the network performance [52].

According to the cosaliency maps generated by the two networks g_s and g_c , the loss function is defined as follows:

$$L(C, G) = -\frac{1}{MX} \sum_{i=1}^{MX} ((1-\gamma)g_i \log p_i + \gamma(1-g_i) \log(1-p_i)), \quad (12)$$

where $\gamma = 0.3$, it is the proportion of salient pixels in the truth map, and M and X indicate the number of training sets and the number of pixels in each training image, respectively. p_i represents the cosaliency map generated by the network in this paper, and g_i represents the truth map.

3.4. Cosaliency Contour Extraction Module. Contour detection uses certain techniques and methods to extract the contour of the target while ignoring the influence of the background and the internal texture of the target and noise interference. The contour extraction model is mainly for grayscale images, and the application direction is medical images. The general idea is to start from a certain point, search the surrounding area pixel by pixel, and determine the pixels inside and outside the area until the initial starting point is found. Influenced by the method of chan and Vese [53], this paper proposes an adaptive contour extraction model based on cosaliency detection, referred to as CSA-CEM, as shown in Figure 8.

In remote sensing images, the direct use of the model will greatly reduce the accuracy of contour recognition. Therefore, this article first uses VGG-19 to extract features from the input image and combines the results of cosaliency detection, PLSD, and SLIC segmentation to select the target frame, make a mask, which can ignore other image features, reduce the complexity of the image to a certain extent, increase the running speed, and fix the initial contour in the target area at the same time, avoiding the possibility of slow curve movement and false alarms.

3.4.1. Target Positioning Mask under Cosaliency.

$$r_n^k \in \begin{cases} S_n, & \text{if } \nu_n^k > \nu_n + \xi_n, \\ B_n, & \text{if } \nu_n^k < \nu_n - 0.25 \times \xi_n, \\ T_n, & \text{otherwise.} \end{cases} \quad (13)$$

In order to refine the edge part of the cosaliency map and improve the problem of the image being too smooth, this

paper takes into account the feature information obtained by superpixel segmentation and straight-line segmentation. Divide an image into K superpixel areas, r_k represents the K -th superpixel area, and the line segment in the K -th superpixel area is marked as l_k . The pixels in the superpixel are often the most similar parts in the adjacent area. Therefore, they generally belong to the salient object or background at the same time. We divide the superpixels into three groups according to the given cosaliency map.

In equation (11), $k = 1, 2, \dots, K$, S_n contains superpixels that belong to the salient target, B_n contains superpixels that belong to the background, and T_n contains the remaining superpixels; that is, it is not clear whether it is the foreground or the background, so it is not considered for the time being. ν_n^k represents the average significant value of the superpixel, ν_n and ξ_n represent the average and standard deviation of the significant value. The cosaliency map is optimized for training by the following formula:

$$\psi_m(I_n; w_m) = w_s \sum_{r \in S_n} \sum_{i \in r} |F_n^m(i) - 1|^2 + w_b \sum_{r \in B_n} \sum_{i \in r} |F_n^m(i) - 0|^2. \quad (14)$$

In equation (12), F_n^m is generated by the trained network g_m , w_s and w_b are the weights that balance the foreground and the background, recorded as constants, w_s is equal to the ratio of the background area to the foreground plus the background, and w_b is the opposite.

Therefore, our objective function is

$$Y(w) = \sum_{n=1}^N \psi_s(I_n; w_s) + \sum_{n=1}^N \psi_c(\{I_n\}_{n=1}^N; w_c) + \alpha_m \psi_m(I_n; w_m). \quad (15)$$

w_s , w_c , and w_m are all learnable parameters of the network. After optimizing the objective function, the cosaliency map can be expressed as follows: $F_n = F_n^s \times F_n^c \times F_n^m$, and the corresponding parts of the three networks are multiplied pixel by pixel.

3.4.2. Contour Extraction. For input image f , corresponding to two-dimensional gray image f_a , based on the LRCV model [43], level set function $\varphi(x)$ and CSACEM can be expressed as

$$E_e^{\text{CSACEM}}(c_{\text{in}}(x), c_{\text{out}}(x), C) = \lambda_1 \int_{\Omega} (f_a(x) - c_{\text{in}}(x))^2 dx + \lambda_1 \int_{\Omega} (f_a(x) - c_{\text{out}}(x))^2 dx + \alpha L_c(C). \quad (16)$$

According to the LRCV model, c_{in} and c_{out} are variables, not constants. They take into account the local characteristics of the image and are more sensitive to the initial contour. In the method of this article, the mask gives a good airport object, which makes the initial contour have an

accurate starting point. Use $\varphi(x)$ to represent contour C . If the pixel is inside the contour C , $\varphi(x) > 0$; if the pixel is outside the contour C , $\varphi(x) < 0$; if the pixel is on the contour C , $\varphi(x) = 0$. $L_c(C)$ represents the length of the contour. The calculation method of and in equation (14) is defined as

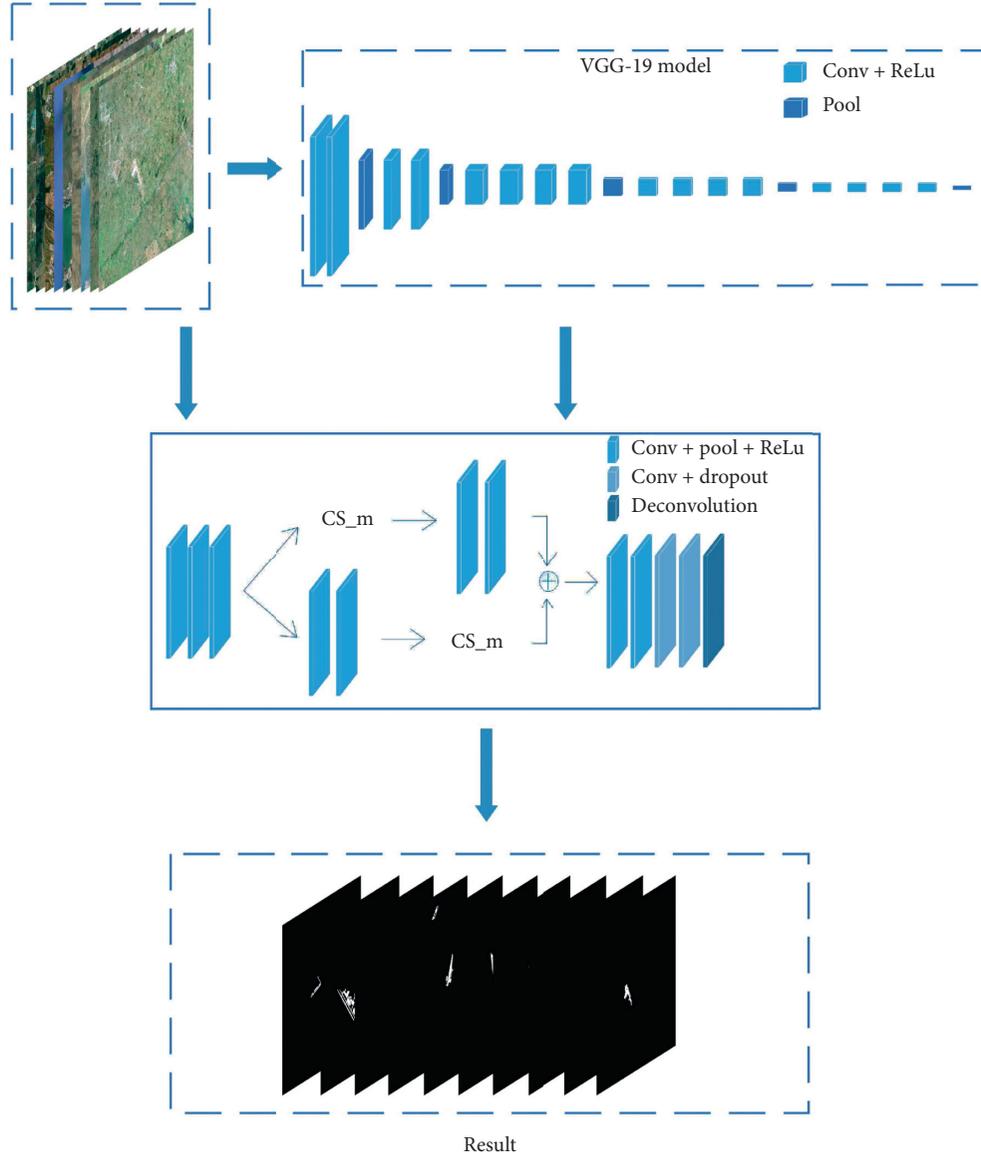


FIGURE 8: Cosaliency contour extraction model.

$$c_{in}(x) = \frac{\int_{\Omega} g_k(x-y)(f_a(y)H_{\varepsilon}(\varphi(x)))dy}{\int_{\Omega} g_k(x-y)H_{\varepsilon}(\varphi(y))dy}, \quad (17)$$

$$c_{out}(x) = \frac{\int_{\Omega} g_k(x-y)(f_a(y)(1-H_{\varepsilon}(\varphi(x))))dy}{\int_{\Omega} g_k(x-y)(1-H_{\varepsilon}(\varphi(y)))dy}. \quad (18)$$

The g_k in equation (16) represents the Gaussian kernel function, and Heaviside function $H(z)$ and Dirac function $\delta(z)$ are also used. Solve the minimization problem of functions E_{ε}^{CSACEM} by Euler-Lagrange equation, and get the level set function evolution equation:

$$\frac{\partial \varphi(x, t)}{\partial t} = \delta(\varphi) \left[\lambda_1 f_a((x) - c_{in}(x))^2 + \lambda_2 (f_a(x) - c_{out}(x))^2 + \alpha \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) \right], \quad (19)$$

with the cosaliency mask as the reference, our contour is closer to the airport target, and the fragmented area in the image set is successfully removed.

Optimization: in order to reduce memory loss and improve the training speed of the model, this article

proposes a method to optimize as follows: put all images into the network cycle 30 times, learn, and train network g_s , g_c , and g_m through the functions of equations (7), (9), and (12). Then, the result graph of cosaliency detection will be very stable.

4. Experiment

The network in this article is deployed under the NVIDIA GeForce GTX Titan 1080 GPU configuration. The hyperparameter learning rate is set to start as 10^{-8} , the momentum is 0.9, and the weight attenuation is set to 0.005. In order to verify the detection effect of the model in this article, we use the 323 remote sensing images collected from Google Earth, the resolution of the image is 896×896 , and each image has an airport target.

The data set used in this paper comes from Google Earth (Data SIO, NOAA, U.S. Navy, NGA, GEBCO, Landsat/Copernicus, IBCAO, and U.S. Geological Survey). We collected 323 remote sensing images including airport. Due to the small data set of remote sensing images, this paper performs data enhancement processing. The enhancement methods used are as follows: random rotation, blurring, illumination, and noise. In this way, we can get 10,000 images with a size of 896×896 , of which 9,000 are used as the training set, and the remaining 1,000 are used as the test set.

4.1. Experimental Results. Shadow removal has important research significance. Researchers at home and abroad have been extensively studying this issue, especially in the past decade, but a database specifically constructed for this problem is very rare. Currently, the most widely used databases are presented by [17, 20], but they only contain 76 and 37 pairs of shadow/shadow-free image pairs. In order to better evaluate the shadow removal algorithm proposed in this paper, we constructed a new RSDB database, which contained 2685 shadow and shadow-free image pairs. The comparison with other shadow removal databases in recent years is shown in Table 2.

Figure 9 shows the contour extraction based on the cosaliency map and detection results obtained by the method proposed in this paper. The experimental results show that the algorithm in this paper can accurately locate the location of airport target and has strong applicability. It has achieved detection results for different complex scenarios and different types of airports. For interference items that affect airport target detection such as urban areas, roads, and coastlines, our method uses superpixel segmentation and cosaliency map information fusion to effectively eliminate their interference. For small airports, the structure is simple and the farmland in the background. If there are a large number of parallel lines with the road, they bring great interference to the detection.

This paper eliminates airport target through local contrast. Compared with small airports, large international airports have more runways and larger areas, the linear characteristics and significance of the runways are more prominent, so the detection accuracy is higher than that of small airports. In addition, there are certain false alarms and missed detection in the results. The main reason is that some airports have simple structures and weak straight-line characteristics of runways. Effective positioning and detection cannot be performed based on runways. In addition, some urban areas have high similarities with airports, and their interference cannot be completely eliminated, so there are certain errors.

TABLE 2: Comparison of different airport extraction models.

Method	MAE	Precision	Recall	mAP	Average time (s)
Literature [48]	0.0725	0.8211	0.9334	0.83	0.12
Literature [49]	0.0512	0.8414	0.9387	0.81	0.09
Literature [50]	0.0074	0.9458	0.9632	0.87	1.25
Literature [51]	0.0148	0.93	0.9767	0.92	0.49
Ours	0.0063	0.97	0.98	0.94	0.34

4.2. Subjective Comparison. Compared with the airport detection algorithm proposed by Liu et al. [54], the cosaliency detection method proposed in this paper is more effective. The visualization results are shown in Figure 10. Compared with the baseline model without edge constraints, in this paper, the feature information fused plays a great role in target detection. Line segment detection and superpixel segmentation can improve the details of the target edge to a certain extent. In contrast, adding a contour extraction module incorporating superpixel segmentation after the original cosaliency detection method can more accurately locate and extract the complete target, which proves that the edge described in our method is more accurate.

Although liu's method can frame the area where the airport is located, the range that can be framed is too large to accurately locate the exact location of the airport, but the method in this article can accurately frame the area where the airport target is located. Frame selection is better than the method of Liu et al. in both aspects.

4.3. Objective Comparison. We use five indicators to evaluate the detection accuracy. They are as follows: precision, recall, mAP, IoU, and MAE.

The IoU of the predicted significant target and the real target is greater than 0.5 and the confidence is greater than 0.8 for evaluation. The IoU threshold used in this paper is 0.5, and the confidence threshold is 0.8.

4.3.1. Calculation Formula of Precision.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{N}, \quad (20)$$

where TP indicates a real case, FP indicates a false positive case, and N indicates a detected target. The accuracy reflects the proportion of the detection result that is correct.

4.3.2. Recall Rate Formula.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{P}. \quad (21)$$

In equation (19), FN represents a false negative example, P represents the target actually contained in the image, and the recall rate reflects the proportion of the target that the detection algorithm can detect.



FIGURE 9: This is the result obtained by the method in this paper: (a) the original image; (b) ground truth; (c) the contour extraction result combined with cosaliency detection; and (d) the target positioning selected by the box.

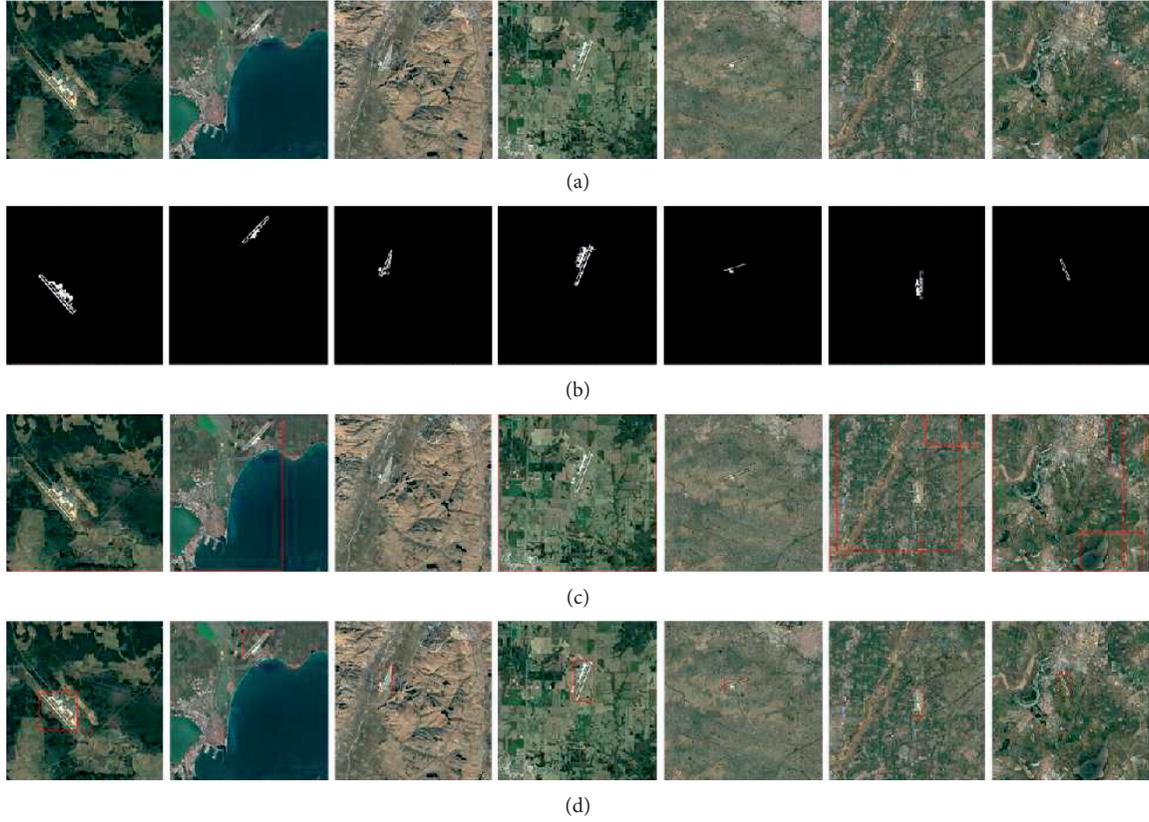


FIGURE 10: Method comparison: (a) the original image; (b) ground truth; (c) the suspected airport area selected by the method proposed in [54]; (d) the suspected airport area selected by our proposed method.

4.3.3. Average Detection Accuracy. Count the detection results of the test set, where the predicted target and the real target's IoU is greater than 0.5 and the confidence is greater than 0.8 for counting, thereby obtaining mAP:

$$\text{IOU} = \frac{\text{Area}(\text{Box}_f \cap \text{Box}_{\text{gt}})}{\text{Area}(\text{Box}_f \cup \text{Box}_{\text{gt}})} \quad (22)$$

In equation (20), $\text{Area}(\text{Box}_f \cup \text{Box}_{\text{gt}})$ and $\text{Area}(\text{Box}_f \cap \text{Box}_{\text{gt}})$, respectively, represent the union area and the intersection area of the detected target and the plane space of the true value. The IOU indicator reflects the accuracy of the positioning of the detection frame.

4.3.4. Mean Absolute Error. If the extracted area contains more than 50% of the absolute true value, we consider it a successful detection method. MAE is defined as the average pixel difference between the measurement result and the real situation:

$$\text{MAE} = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |R(x, y) - GT(x, y)|, \quad (23)$$

where W represents the width of the image, H represents the height of the image, $R(x, y)$ represents the pixel value of the prediction target, and $GT(x, y)$ is the true value corresponding to the image.

We also compare the model with four target detection related methods: Zhao et al. used a regression-based saliency model to detect airports [55], Redmon et al. [56] applied the YOLOv3 algorithm and a saliency-oriented active contour method [57], and the idea of multidimensional feature fusion is applied for target detection [58]. Table 2 shows the test data. As shown in Table 2, the algorithm proposed in this paper is slightly slower than the regression-based method and the YOLOv3 algorithm. However, the results of MAE show that our results are closest to the true value map compared to the four models. The method in this paper is better than the significance model detection the correct ratio is increased by 15%, which is 13% higher than that of the YOLOv3 algorithm and 1% higher than the method of contour extraction to detect airports. And the proportion of airport targets detected is also the largest. Overall, the accuracy of detection is higher.

4.4. Ablation Experiment. In order to prove the effectiveness of the various components in the method proposed in this paper, ablation experiments are carried out. The model in this paper is mainly composed of two parts. The first part is to use the PLSD algorithm to detect line segments in remote sensing images. This algorithm can remove particularly short line segments in remote sensing images, and it can also perform pixel completion on the suspected airport runway. To a large extent, make airport detection

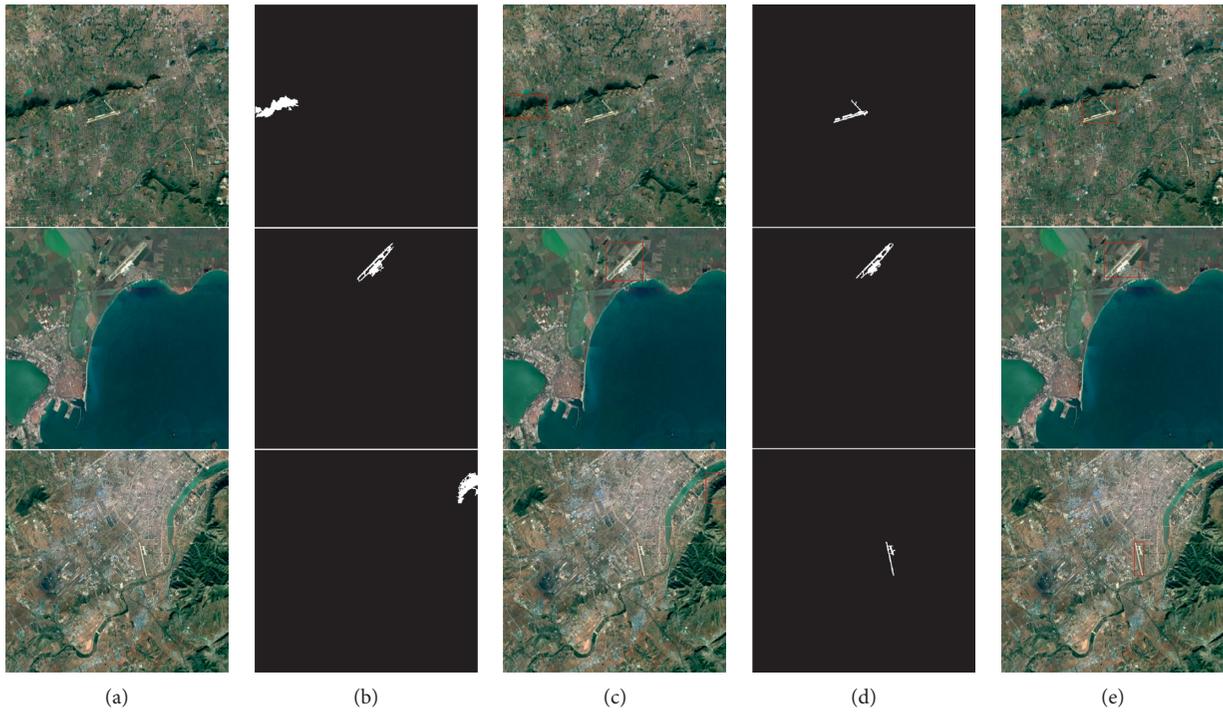


FIGURE 11: No straight-line segmentation algorithm result graph: (a) the original image; (b) the cosaliency map without line detection; (c) the frame selection map corresponding to (b); (d) the cosaliency map after line detection; and (e) the frame selection map corresponding to (d).

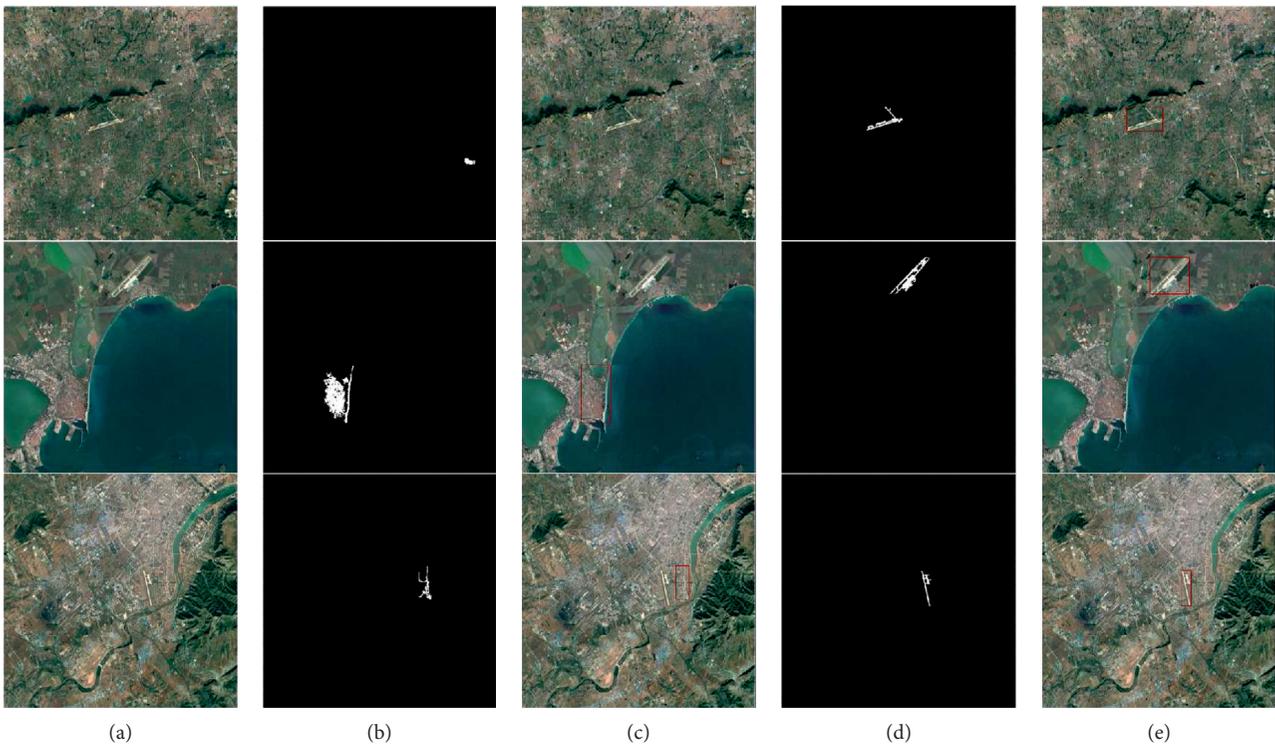


FIGURE 12: Results without cosaliency detection: (a) the original image, (b) the saliency map without cosaliency detection; (c) the frame selection map corresponding to (b); (d) the saliency map after cosaliency detection; and (e) the frame selection map corresponding to (d).

more accurate. The second part is the constructed cosaliency detection model, which is to fuse the features of the saliency in the image and the consistency between the images to extract the common saliency target in the remote sensing image.

In Figure 11, it can be clearly seen from the figure that when there is a large difference between the target and the background, the effect is still considerable when there is only a cosaliency detection model. However, for an image with a complicated background in the third row, without the PLSD algorithm, the target location cannot be accurately located. Therefore, the existence of the PLSD algorithm is of great significance.

In Figure 12, it can be clearly seen from the figure that when only the PLSD algorithm is used, the complete airport target cannot be detected. It must be used together with the cosaliency detection to achieve the purpose of accurately positioning the airport target.

Therefore, every component of the model proposed in this article is necessary.

5. Conclusions

This article focuses on airport target detection in remote sensing images. A contour extraction model based on cosaliency detection is proposed to estimate the place of airport. Our model multiplies the saliency maps, in-train image and inter-image pixel by pixel to obtain a cosaliency mask. The mask, superpixel segmentation, and PLSD detection play an auxiliary role to refine the airport contour, so that we can get a more accurate airport contour. At the same time, we compared the model with four commonly used methods. The experimental results show that the model can not only extract contours which highly consistent with the actual airport target, but also achieve high detection accuracy and efficiency.

Abbreviations

CNN:	Convolutional neural network
CSACEM:	Cosaliency active contour extraction model
EDLD:	Edge drawing line detector
FCN:	Fully convolutional network
HLD:	Hough line detector
LRCV:	Local region-based Chan–Vese model
LSD:	Linear segmentation detector
PLSD:	Parallel linear segmentation detector
SAE:	Stacked autoencoder
SAEF:	Stacked autoencoder-enabled fusion
SIFT:	Scale invariant feature transform
SLIC:	Simple linear iterative clustering
SVM:	Support vector machine
VGG-19:	Visual Geometry Group-19.

Data Availability

The data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after

publication of this article, will be considered by the corresponding author.

Disclosure

The funding sponsors had no role in the design of the study; in the collection of image; in the construction, training, and optimization of the model; in the writing of the manuscript, and in the decision to publish the results.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Z.H., Z.Z.B., and J.J.L. conceived and designed the review article and analyzed the literature and image data. Z.Z.B. drafted the manuscript, and Z.H. and J.J.L. performed critical review of the manuscript. J.J.L. contributed to the idea of building model. Z.Z.B. performed experimental verification. All the authors have agreed to have the paper submitted for publication and have contributed substantially to the work reported.

Acknowledgments

The authors acknowledge the support of the National Natural Science Foundation of China (grant nos. 61772319, 62002200, 61976125, 61976124, and 61773244) and Shandong Natural Science Foundation of China (grant no. ZR2017MF049).

References

- [1] M. Shimoni, R. Haelterman, and C. Perneel, "Hyperspectral imaging for military and security applications: combining myriad processing and sensing techniques," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 101–117, 2019.
- [2] K. E. Joyce, S. E. Belliss, S. V. Samsonov, S. J. McNeill, and P. J. Glassey, "A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters," *Progress in Physical Geography: Earth and Environment*, vol. 33, no. 2, pp. 183–207, 2009.
- [3] W. Wang, L. Liu, C. Hu, Y. Jing, and G. Kuang, "Airport detection in SAR image based on perceptual organization," in *Proceedings of the International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping*, pp. 1–5, IEEE, Xiamen, China, January 2011.
- [4] Z. Kou, Z. Shi, and L. Liu, "Airport detection based on line segment detector," in *Proceedings of the International Conference on Computer Vision in Remote Sensing*, pp. 72–77, Xiamen, China, December 2012.
- [5] D. Liu, L. He, and L. Carin, "Airport detection in large aerial optical imagery," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, p. 761, Montreal, Canada, May 2004.
- [6] C. Tao, Y. Tan, H. Cai, and J. Tian, "Airport detection from large IKONOS images using clustered SIFT keypoints and region information," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, pp. 128–132, 2010.

- [7] X. Wang, B. Wang, and L. Zhang, "Remote sensing image airport detection based on image saliency area," *Journal of Computer Aided Design and Graphics*, vol. 24, pp. 336–344, 2012.
- [8] D. Zhu, B. Wang, and L. Zhang, "Airport target detection in remote sensing images: a new method based on two-way saliency," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 1096–1100, 2015.
- [9] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: a survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.
- [10] F. Zhang, B. Du, L. Zhang, and M. Xu, "Weakly supervised learning based on coupled convolutional neural networks for aircraft detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5553–5563, 2016.
- [11] Z. Xiao, Y. Gong, Y. Long, D. Li, X. Wang, and H. Liu, "Airport detection based on a multiscale fusion feature for optical remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 9, pp. 1469–1473, 2017.
- [12] B. Cai, Z. Jiang, H. Zhang, D. Zhao, and Y. Yao, "Airport detection using end-to-end convolutional neural network with hard example mining," *Remote Sensing*, vol. 9, no. 11, p. 1198, 2017.
- [13] P. Zhang, X. Niu, Y. Dou, and F. Xia, "Airport detection on optical satellite images using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 8, pp. 1183–1187, 2017.
- [14] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2018.
- [15] E. Jacobs David, D. B. Goldman, and E. C. Shechtman, "Where people look when comparing images," in *Proceedings of the Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pp. 219–228, New York, NY USA, October 2010.
- [16] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3766–3778, 2013.
- [17] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng, "Self-adaptively weighted co-saliency detection via rank constraint," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 23, pp. 4175–4186, 2014.
- [18] L. Ye, Z. Liu, J. Li, W.-L. Zhao, and L. Shen, "Co-saliency detection via co-salient object discovery and recovery," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2073–2077, 2015.
- [19] K. Zhang, T. Li, B. Liu, and Q. Liu, "Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3095–3104, Long Beach, CA, USA, June 2019.
- [20] C. C. Tsai, K. J. Hsu, Y. Y. Lin, X. Qian, and Y. Y. Chuang, "Deep co-saliency detection via stacked autoencoder-enabled fusion and self-trained CNNs," *IEEE Transactions on Multimedia*, vol. 22, pp. 1016–1031, 2019.
- [21] J. J. Li, X. M. Feng, and Z. Hua, "Low-light image enhancement via progressive-recursive network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, In press.
- [22] H. Fan, M. Han, and J. J. Li, "Image shadow removal using end-to-end deep convolutional neural networks," *Applied Sciences*, vol. 9, no. 5, Article ID 1009, 2019.
- [23] J. Li, G. Li, and H. Fan, "Image dehazing using residual-based deep CNN," *IEEE Access*, vol. 6, pp. 26831–26842, 2018.
- [24] T. L. Zhang, J. J. Li, and Z. Hua, "Iterative multi-scale residual network for deblurring," *IET Image Processing*, 2021.
- [25] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," in *Proceedings of the 2016 23rd International Conference on Pattern Recognition*, pp. 3488–3493, Cancun, Mexico, December 2016.
- [26] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.
- [27] X. Hou, J. Harel, and C. Koch, "Image signature: highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 194–201, 2012.
- [28] T. Avraham and M. Lindenbaum, "Esaliency (extended saliency): meaningful attention using stochastic image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 693–708, 2009.
- [29] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 545–552, Minnesota, MN, USA, June 2007.
- [30] H. T. Chen, "Preattentive co-saliency detection," in *Proceedings of the 2010 IEEE International Conference on Image Processing*, Hong Kong, China, September 2010.
- [31] H. Li and K. N. Ngan, "A Co-saliency model of image pairs," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3365–3375, 2011.
- [32] H. Li, F. Meng, and K. N. Ngan, "Co-salient object detection from multiple images," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1896–1909, 2013.
- [33] C. C. Tsai, W. Li, and K. J. Hsu, "Image co-saliency detection and co-segmentation via progressive joint optimization," *IEEE Transactions on Image Processing*, vol. 28, pp. 56–71, 2018.
- [34] C. C. Tsai, X. Qian, and Y. Y. Lin, "Segmentation guided local proposal fusion for co-saliency detection," in *Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 523–528, Hong Kong, China, July 2017.
- [35] K. R. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Transactions on Multimedia*, vol. 18, pp. 1896–1909, 2018.
- [36] C. C. Tsai, X. Qian, and Y. Y. Lin, "Image co-saliency detection via locally adaptive saliency map fusion," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1897–1901, New Orleans, LA, USA, March 2017.
- [37] D. Zhang, J. Han, and J. Han, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, pp. 1163–1176, 2015.
- [38] L. Wei, S. Zhao, O. E. F. Bourahla, X. Li, and F. Wu, "Group-wise deep co-saliency detection," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Main Track*, pp. 3041–3047, Melbourne, Australia, August 2017.
- [39] J. Han, G. Cheng, and Z. Li, "A unified metric learning-based framework for co-saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 2473–2483, 2017.
- [40] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

- [41] Y. Jing, J. An, and Z. Liu, "A novel edge detection algorithm based on global minimization active contour model for oil slick infrared aerial image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2005–2013, 2011.
- [42] X. Li, H. Jiao, and Y. Wang, "Edge detection algorithm of cancer image based on deep learning," *Bioengineered*, vol. 11, no. 1, pp. 693–707, 2020.
- [43] S. Liu and Y. Peng, "A local region-based Chan-Vese model for image segmentation," *Pattern Recognition*, vol. 45, no. 7, pp. 2769–2779, 2012.
- [44] M. F. Talu, "ORACM: online region-based active contour model," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6233–6240, 2013.
- [45] P. Zhang, X. Niu, and Y. Dou, "Airport detection from remote sensing images using transferable convolutional neural networks," in *Proceedings of the 2016 International Joint Conference on Neural Networks*, pp. 2590–2595, Vancouver, Canada, July 2016.
- [46] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE, Nice, France, October 2003.
- [47] R. Achanta, A. Shaji, K. Smith et al., *Slic Superpixels*, 2010.
- [48] R. G. Von Gioi, J. Jakubowicz, J. M. Morel, and G. L. S. D. Randall, "A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 722–732, 2008.
- [49] R. G. Von Gioi, J. Jakubowicz, J. M. Morel, and G. L. S. D. Randall, "A line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.
- [50] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional models for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3–4, Boston, MA, USA, June 2015.
- [51] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: towards unsupervised learning of deep salient object detector," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4048–4056, Venice, Italy, October 2017.
- [52] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3376–3385, Boston, MA, USA, June 2015.
- [53] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [54] N. Liu, Z. Cui, Z. Cao, Y. Pi, and S. Dang, "Airport detection in large-scale SAR images via line segment grouping and saliency analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 434–438, 2018.
- [55] D. Zhao, Y. Ma, Z. Jiang, and Z. Shi, "Multiresolution airport detection via hierarchical reinforcement learning saliency model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2855–2866, 2017.
- [56] J. Redmon and A. Farhadi, *Yolov3: An Incremental Improvement*, 2018, <https://arxiv.org/abs/1804.02767>.
- [57] Q. Zhang, L. Zhang, W. Shi, and Y. Liu, "Airport extraction via complementary saliency analysis and saliency-oriented active contour model," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 7, pp. 1085–1089, 2018.
- [58] X. Wang, Y. Ban, and H. Guo, "Deep learning model for target detection in remote sensing images fusing multilevel features," in *Proceedings of the IGARSS IEEE International Geoscience and Remote Sensing Symposium*, pp. 250–253, Yokohama, Japan, August 2019.