# Two-click based fast small object annotation in remote sensing images.

LEI, L., FANG, Z., REN, J., GAMBA, P., ZHENG, J. and ZHAO, H.

2024

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2024.3442732



# Two-click based Fast Small Object Annotation in Remote Sensing Images

Lu Lei, Zhenyu Fang, Jinchang Ren, Paolo Gamba, Jiangbin Zheng and Huimin Zhao

***Abstract*—In the remote sensing field, detecting small objects is a pivotal task, yet achieving high performance in deep learning-based detectors heavily relies on extensive data annotation. The challenge intensifies as small objects in remote sensing imagery are typically densely distributed and numerous, leading to a substantial increase in the cost of creating large-scale annotated datasets. This elevated cost poses significant limitations on the application and advancement of small object detection. To address this issue, a Point-Based Annotation method (PBA) is proposed, which generates bounding boxes through graph-based segmentation. In this framework, user annotations categorize nodes into three distinct classes - positive, negative, and to-cut - facilitating a more intuitive and efficient annotation process. Utilizing the max-flow algorithm, our method seamlessly generates Oriented Bounding Boxes (OBBOX) from these classified nodes. The efficacy of PBA is underscored by our empirical findings. Notably, annotation efficiency is enhanced by at least 40%, a significant leap forward. Moreover, the Intersection over Union (IoU) metric of our OBBOX outperforms existing methods like "Segment Anything Model" by 10%. Finally, when applied in training, models annotated with PBA exhibit a 3% increase in the mean Average Precision (mAP) compared to those using traditional annotation methods. These results not only affirm the technical superiority of PBA but also its practical impact in advancing small object detection in remote sensing.**

***Index Terms*—Remote Sensing, Small Object Detection, Data Annotation, Deep Learning, Cost-Efficiency in Data Processing.**

## I. Introduction

With the rapid advancement in remote sensing and sensor development, high-resolution optical imaging has become a cornerstone of object detection in Earth observation. Among various methodologies, deep learning techniques have revolutionized object detection by delivering unprecedented accuracy [1], [2]. These methods, and specifically relying on bounding box (BBOX) annotations, enable precise categorization and localization of objects in an end-to-end manner, provided that ample annotated training samples are available.

This work was supported in part by the National Natural Science Foundation of China under Grant 62202385, in part by the Fundamental Research Fund for the Central Universities under Grant G2021KY05103, and in part by the Basic Research Programs of Taicang under Grant TC2022JC21, and in part by the Guangdong Provincial Department of Education "Innovation and Strengthening University" Project (2022ZDJS015) (Corresponding author: Zhenyu Fang, email: zhenyu.fang@nwpu.edu.cn).

L. Lei, Z. Fang, and J. Zheng are with the School of Computer Software, Northwestern Polytechnical University (NPU), Xi'an, China.

Z. Fang is also with Yangtze River Delta Research Institute of NPU, Taicang, China

J. Ren and H. Zhao are with the School of Computer Sciences, Guangdong Polytechnic Normal University, Guangzhou, China.

J. Ren is also with Nationall Subsea Centre, School of Computing and Engineering, Robert Gordon University, Aberdeen, U.K.

P. Gamba is with Department of Electrical Computer and Biomedical Engineering, University of Pavia, Pavia, Italy

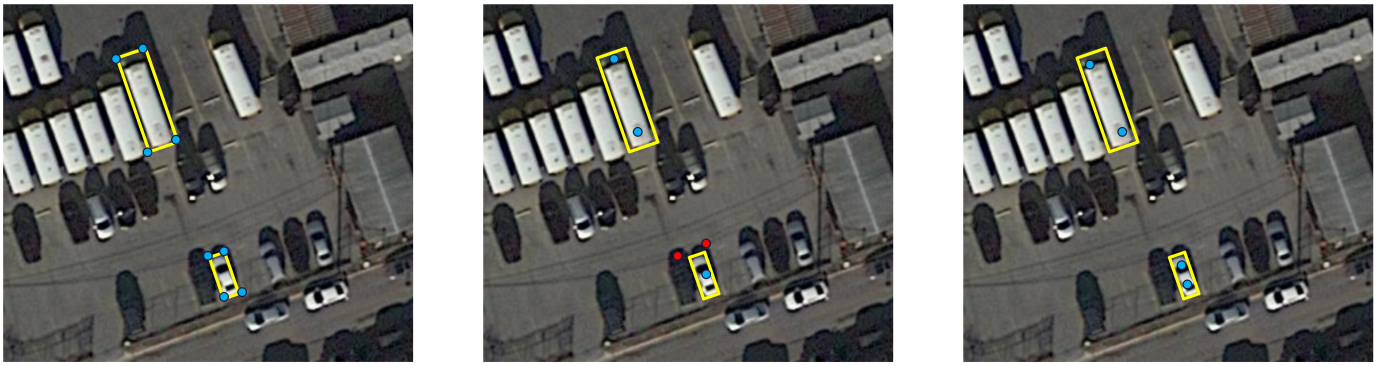
In the realm of remote sensing, the conventional horizontal BBOX (HBBOX), defined by center coordinates, width, and height $(x, y, w, h)$, often proves inadequate due to the unique bird's-eye perspective of imagery. This viewpoint necessitates consideration of object orientation, leading to the adoption of oriented BBOX (OBBOX), denoted as $(x, y, w, h, \theta)$. OBBOX is particularly crucial for delineating small objects, which are prevalent in high-resolution remote sensing images and often fall below the $32 \times 32$ pixel dimension threshold. The conventional HBBOX approach may result in significant overlaps among these small objects, leading to erroneous detections. However, the intricate nature of OBBOX annotations—requiring more precision in terms of angle and positioning—poses substantial challenges in terms of annotation effort, time, and cost [3], [4], [5]. As a result, the annotation fee of OBBOX may be far higher than that of HBBOX [6].

Recent advancements have seen the emergence of foundation models like the Segment Anything Model (SAM) [7], which introduce interactive point-based annotation methods. Annotators can now mark objects with a single point, and the OBBOX is generated from the segmentation results of SAM, using the input point as a "prompt" [8], [9], [10]. Despite these innovations, SAM's effectiveness in segmenting small objects in remote sensing is limited due to the considerable domain gap between its training dataset and the actual remote sensing scenes. Moreover, the substantial computational resources required by SAM's large model architecture exacerbate the processing burden for large-scale remote sensing images. Consequently, when applying SAM to remote sensing imagery, both annotation accuracy and efficiency are compromised.

To mitigate the challenges associated with small object annotation in remote sensing, a novel point-based bounding box annotation pipeline is proposed, termed PBA. Similar to SAM, PBA relies on minimal point annotations to indicate object locations. As seen in Fig. 1, there are no strict rules on the positions of points, which provides greater flexibility and can significantly reduce manual work. The bounding box of an object is then determined by the predicted segmentation mask. PBA's innovation lies in its unsupervised nature, eliminating the need for additional annotated data for model fine-tuning. This feature is particularly advantageous in remote sensing.

The major contributions of this paper can be summarized as follows:

i. An annotation pipeline is proposed that relies on the annotation of two points, termed the point-based annotation method (PBA), where the points do not need to be positioned at the object boundary. The bounding box of the annotated object is determined via an optimized



(a) (b) (c)

Fig. 1: Examples on the existing annotation methods. (a) The commonly used four-point annotation method; (b) Segment anything model and (c) the proposed PBA. The blue points are the positive annotation, the red points indicates the negative annotation, and the yellow box is the generated bounding box. As seen, the proposed PBA only requires two-point annotation, without other negative annotation on background region.

graph-based segmentation method. This approach is more flexible and can significantly reduce manual work.

ii. A local-graph-building method is proposed to improve segmentation efficiency. Unlike existing graph-based methods, only nodes near the annotated points are considered to build the graph. Nodes are clustered into three categories, i.e., "positive", "negative" and "to-segment", respectively. The goal of the graph cut is to classify the to-segment nodes as either positive or negative. This classification further improves the segmentation efficiency, especially considering the large-scale of remote sensing images;

iii. Experimental results on the Tiny-Dota [11] and DIOR-R [12] datasets demonstrate that PBA reduces annotation time by at least 40% and achieves a 10% higher IoU for OBBOX generation compared to SAM. Furthermore, models trained with PBA-annotated samples outperform those trained with SAM-annotated samples by 3% in mean average precision (mAP);

The remainder of this paper is structured as follows: Section II reviews related works, Section III details the proposed method, experimental results are presented in Section IV, and the conclusion is drawn in Section V.

## II. Related Work

This section will illustrate previous works in four aspects: manual annotation, automatic annotation, interactive image segmentation, and automatic point-based annotation, all of which are relevant to the proposed PBA approach.

### A. Manual Annotations

There are several methods to annotate objects in pattern recognition, i.e., including box annotation [13], point annotation [14], [15], pixel annotation [16], [17], [18], [19], line annotation [20], [21], etc. Box annotation is the main approach used in the remote sensing object detection task and can be further divided into horizontal box [22], [23], [24] and oriented box [25], [26], [27], [28], [29], [30]. Horizontal box is the most commonly used: the annotation of the horizontal box (HBBOX) typically stores the coordinates of the upper left and lower right corners in a rectangular bounding box, noted as $(x_{top-left}, y_{top-left}, x_{bottom-right}, y_{bottom-right})$. Another annotation format uses the center point coordinates along with the width and height, noted as $(x_c, y_c, w, h)$.

The shape and orientation of objects vary significantly in remote sensing images, such as densely distributed trucks in a car park, multiple ships in a harbor, and airplanes on an airfield. Therefore, it is difficult to box these objects using HBBOX, which can result in boxes containing a large amount of irrelevant background information. In some cases, objects with small intervals may even overlap, capturing parts of neighboring objects. Thus, the use of oriented box (OBBOX) annotations is more appropriate for remote sensing images.

Oriented box (OBBOX) annotation typically involves using a quadrilateral box with a specified angle of rotation. With the aid of annotation tools, annotators first draw a roughly aligned horizontal bounding box, then rotate it around the center point to align with the object, and finally make fine adjustments. Another annotation method involves clicking on one corner of the object, then sequentially clicking on the other three corners in a clockwise or counterclockwise direction, and finally clicking again on the first point. OBBOX directly saves the coordinates of the four vertices of the quadrilateral, noted as $\{(x_i, y_i), i = 1, 2, 3, 4\}$. OBBOX annotation solves the problem of inaccurate annotation due to the tilted attitude of the object in remote sensing images. However, it also takes more labor work to find the accurate rotation angle or the corner points of an object.

Among the existing annotation tools, the most widely utilized one is "LabelMe" [31], developed by MIT. This tool can create different types of annotations for images or videos, such as boxes, polygons, circles, lines, and points. Users can save the annotation results in COCO [32], VOC [33], or other

formats. Another open-source annotation tool is "CVAT" [1], developed by Intel Corporation, designed for professional data annotation teams. Manual annotation is highly accurate but comes at the expense of high labor costs and significant time. Moreover, annotation accuracy can vary significantly between individuals, leading to considerable variations in the annotation results.

### B. Automatic Annotations

When using automatic annotation, annotators simply input unannotated images, and the annotation tool adds the image annotations automatically. PaddleLabel[2] and LabelBee[3] annotation tools provide automated annotation plugins. They are both based on deep learning algorithms, with models typically trained on general scene datasets, such as COCO and ImageNet [34]. This means they cannot be directly tuned for use in other areas, such as remote sensing [35], medical [36] and industrial [37] scenes.

### C. Interactive Image Segmentation

Interactive image segmentation requires users set positive and negative samples through an interactive interface. Then, an algorithm automatically computes the segmentation result. The most classic method is Graph Cut [38], [39], [40], based on a graph-theoretic implementation. To be more specific, Graph Cut uses an interactive approach, allowing the user to set the seeds of positive and negative samples on an image. Then, it uses the maximum-flow algorithm [41] to obtain the segmentation result. As an example, Lazy-snapping [42] is an optimized Graph Cut based method. At first, it preprocesses the image using the Watershed algorithm [43], and then adds the preprocessed results to the Graph Cut algorithm. After the Graph Cut algorithm computation, an interactive interface for manual corrections is provided, allowing users to edit the segmented edges to make the segmented results more refined.

### D. Point-based Annotations

In recent years, more attention has been drawn on point annotation, in replace of direct box annotation. This approach can substantially reduce annotation costs. Chen et al. [44] proposed to combine weakly-supervised and semi-supervised approaches for object detection. First, a Point DETR [45] teacher network is trained to generate annotated boxes by point annotation. Then, a deep learning network is utilized to generate the corresponding pseudo-annotated boxes from the point annotated images. Finally, a student network for supervised object detection is trained with the dataset that includes all these annotated boxes. The method reduces the cost of annotation by bringing in point annotation.

Zhang et al. [46] designed a CNN-based point-to-box network Group R-CNN for weakly semi-supervised object detection task, based on point annotation. Group R-CNN surpasses Point DETR by nearly 4%, when using 5% of fully annotated and 95% of point annotated data. Group R-CNN introduces instance-level proposal grouping, instance-level proposals, instance-aware representation learning and other novel designs that allow the network to better perform than transformer-based methods, especially when there is a limited number of annotated box images.

Lee et al. [11] proposed a different approach. The proposed C3Det is an interactive annotation method for tiny object detection, where the user simply clicks on a target object on an image, and the system automatically fetch boxes of similar objects in the image. Even some objects of a different type than the user clicked object may be recognized. By substantially reducing the number of user's clicks, this method reduces the difficulty of annotating tiny objects.

In summary, manual annotation methods are more accurate, but come with high cost. Automatic annotation methods save labor works, but are based on trained deep learning models. Moreover, the domain adaption ability of object detection models is limited, making it difficult to reuse models from different domains. In this paper, an annotation method for small objects based on the Graph Cut algorithm is proposed to generate annotation boxes from low-cost point annotations using an unsupervised approach. This method not only reduces the high manual annotation costs, but also the corresponding computational costs required for deep learning training.

## III. The Proposed Method

In this section, the proposed point-based bounding box annotation method is illustrated. It consists of prompt collection, super-pixelization, graph construction, graph partitioning, and finally box generation.

### A. The Overall Pipeline of PBA

As seen in Fig.2, the PBA algorithm is designed to effectively generate bounding box annotations for small objects in remote sensing images. Compared with the overly costly manual annotation, PBA only needs low-cost point annotations. The segmentation procedure is unsupervised, meaning it can complete the annotation efficiently without any training step.

To improve the interference efficiency in densely distributed sets of small objects, the PBA algorithm first cuts a patch on the original image based on the point annotations provided by the user. Each annotation cuts a corresponding patch and the OBBOX annotation is generated. Finally, the boxes of all the sub-images will be merged back to the original image.

Note that, differently from the original four-point annotation method, here annotators only use two clicks along the object orientation, to label an object. This is a more user-friendly solution, which requires a lower annotation precision, as illustrated in Section III-B.

Foreground objects and background regions are segmented using a Graph Cut based method. However, since the scale of remote sensing images is usually larger than 1000 pixels, the computational costs quickly become unacceptable for building graph at the pixel level. Thus, the number of nodes is reduced via pixel clustering, a step that merges adjacent pixels with similar visual aspects. Graphs are then built using those nodes.

[1]https://github.com/opencv/cvat
[2]https://github.com/PaddleCV-SIG/PaddleLabel
[3]https://github.com/open-mmlab/labelbee

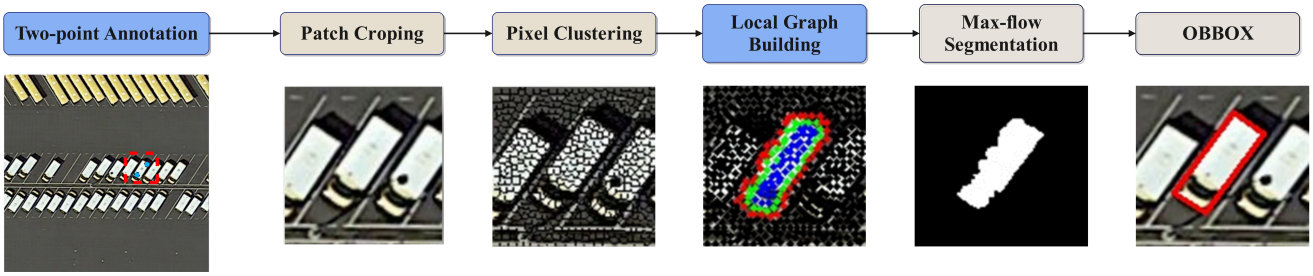



Fig. 2: The workflow of the proposed PBA (Point-Based Annotation) method, featuring two innovative modules (highlighted in blue) designed for optimization. Initially, the method streamlines the annotation process through two-point selection, which does not adhere to strict placement rules. Subsequently, to enhance computational efficiency, a novel local graph construction technique is employed to accurately represent the topology of individual objects. Bounding boxes are determined using a graph-segmentation approach.

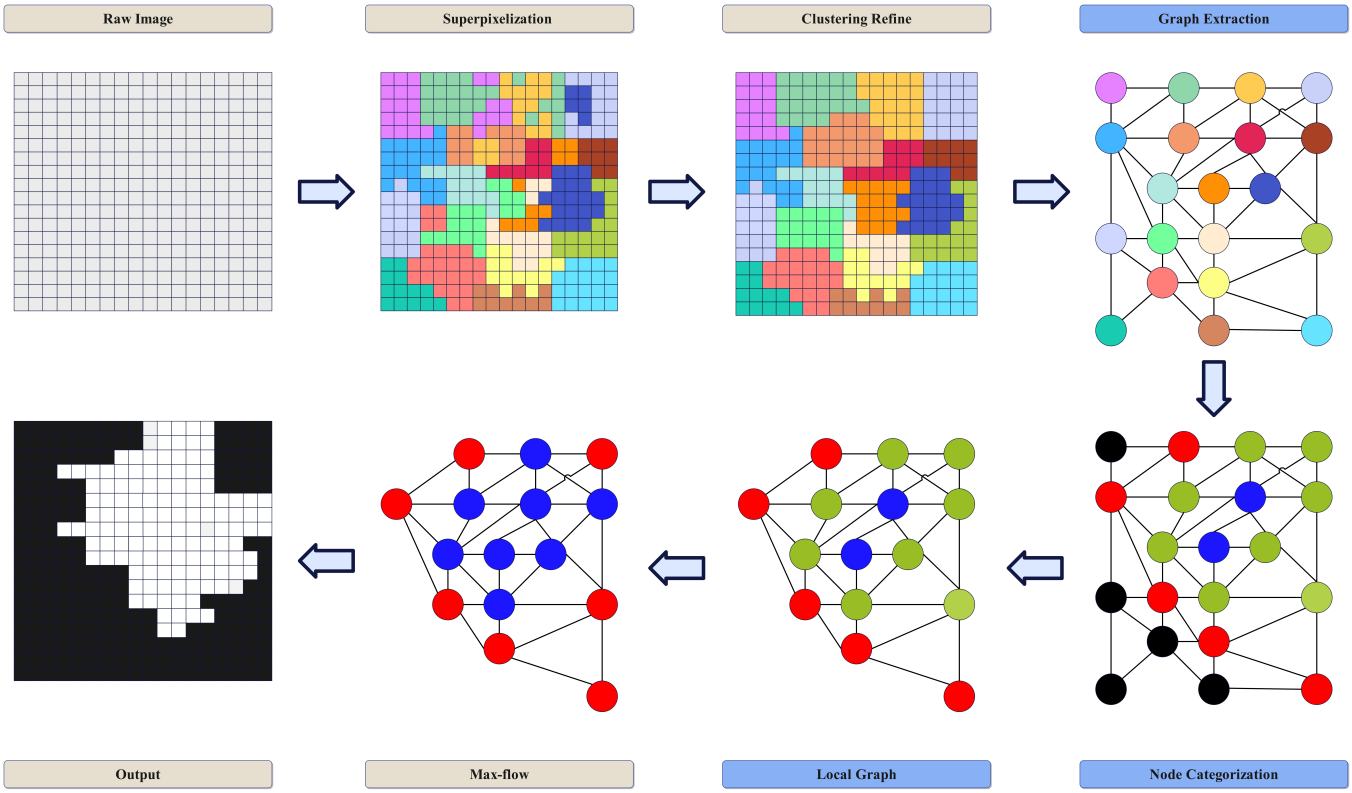



Fig. 3: An example of generating the segmentation mask using the proposed PBA method. Pixels are firstly clustered into superpixels using the SLIC method, and the extracted superpixels are then refined to remove the outlier or discrete patches. Taking those superpixels as nodes and their connections as edges, a local graph can be built, where positive nodes are firstly determined according to the user annotation (denoted by blue in the second row), followed by to-cut nodes (green) and negative nodes (red). The to-cut nodes and negative nodes are defined the same as in Sec III-D. After that, invalid nodes (black), where no positive or to-cut nodes connected, are excluded to alleviate the computational burden, with the Max-flow method being applied to predict the segmentation mask.

Compared with the original Graph Cut method, the number of nodes is reduced by a few percentage points, significantly improving annotation efficiency, as detailed in Section III-C.

Although the previous step significantly reduces computational costs, segmenting nodes in a large scene could still be a heavy burden. Thus, a local-graph-based segmentation model (LGSeg) is proposed to further reduce the computational costs in irrelevant regions of an image. Collectively, positive sample points, negative sample points, and points to be segmented are referred to as valid points. The edges connected to different types of valid points are assigned different weights, and the Graph Cut method segments the graph using the classical max-flow algorithm, dividing the graph into two parts with only positive and negative sample points, and mapping the

superpixel regions corresponding to the positive and negative sample points to the original graph, i.e., obtaining a bipartite graph, with the black region representing the background, and each white block representing a small object, as detailed in Section III-D.

Finally, each segmented block is framed with a rectangle to get the coordinates of the rectangular bounding box, as illustrated in section III-E.

### B. Two-point Annotation

The proposed PBA method first requires users to provide a two-point annotation for each object in an image. These two points are connected as a line, mainly used to select the seeds in the graph. The positions of these two points are not strictly ruled, a certain of offset in the position is acceptable.

The original four-point box annotation method, utilizes more constrains on the annotated points. Single point annotation methods cost less labor. However, they also offer less information, making is difficult to be directly utilized. The proposed method combines the advantages of box annotation and point annotation. It can not only obtain the location information of objects under the premise of low cost, but also have a preliminary knowledge of the size of objects, estimated through the size of the generated bounding box.

### C. Pixel Clustering

Building graph using the original pixels of the image as nodes could require a huge computational cost for segmentation, leading to extremely low computational efficiency. To reduce the number of nodes, an intuitive solution is to merge pixels into "superpixels", areas with similar visual aspect. To this aim, an off-the-shelf method is adopted: SLIC (Simple Linear Iterative Clustering) [47] for clustering pixels as preprocessing step. Compared with the traditional k-means algorithm, SLIC restricts the search space to a limited range, greatly reducing the computational complexity and improving the inference efficiency.

Due to the cluster process utilized in SLIC, some super-pixel regions are discontinuous, causing an erroneous determination of the center coordinates in the following steps. To tackle this, small regions that may cause bias are removed. Thus, each superpixel is re-clustered as below :

$$ID_i = \text{argmin}_{ID}\left(\sqrt{\left(I_i^{RGB} - I_j^{RGB}\right)^2}\right), \text{ if } N_i < R^2 \quad (1)$$

where $ID_i$ is the superpixel index, $N_i$ is the number of pixels contained in the superpixel, $R$ the region size. then, the pixel average intensity ($I_i^{RGB}$) through the RGB channels is applied to measure the similarity between superpixels. The nearest eight-directions (top, left, right, bottom, top-left, top-right, bottom-left, and bottom-right) are considered. Note that due to the different sizes of multiple super-pixels, so that the number of neighboring superpixels may be less than 8.

### D. Local-graph-based Segmentation Model

As visualized in Fig. 3, a graph-based method is utilized to distinguish objects from background. The graph structure, called $G$, includes nodes and edges, i.e. $G = \{V, E\}$. $G$ usually consists of a node set $V = \{V_{valid}, V_T\}$ and an edge set $E = \{E_{(V_{valid}, V_T)}, E_{(V_{(valid_i)}, V_{(valid_j)})}\}$. The graph structure used by the Graph Cut based method differs from the original graph structure: there are two additional terminal vertices, $S$ (source) and $T$ (sink), and every node in the graph structure must be connected to a terminal vertex. The edge set $E$ is divided into two main categories, t-links and n-links.

**t-links**: t-links are the connections between each node and the terminal vertex in the graph.

**n-links**: n-links are the connecting lines between adjacent nodes in the graph.

Before segment superpixels, the nodes and edges must be determined to build the graph structure.

*1) Get Center of Mass and Pixel Values:* In this paper, the geometric center of each superpixel region is adopted to represent the position of each superpixel, and the mean RGB value of the pixels in this region is utilized to calculate the edge value between two adjacent nodes.

For each superpixel region, the center coordinate is acquired using its image moments. Image moments are parameters estimated by a mathematical method used in image processing to describe the shapes and features of images, which can in turn describe the geometric properties of the shapes in an image at different scales and orientations. The moment is based on the concept of spatial distribution. Let $m_{00}$ denote the zero-order mixed moments of the origin of an image, and $m_{10}$ and $m_{01}$ denote the first-order mixed moments of the origin of an image about the x-axis and y-axis. The center of mass $(c_x, c_y, i)$ can be calculated according to the following equation, where $c_x$ and $c_y$ denote the x-axis and y-axis coordinates of the center of mass, respectively, and the marks to which superpixel this point belongs.

$$c_x = \frac{m_{10}}{m_{00}}, \quad c_y = \frac{m_{01}}{m_{00}} \quad (2)$$

*2) Building t-links:* The creation of t-links is relatively simple, directly connecting the points in the graph structure to the terminal vertices, while the weights of the edges need to be obtained based on the two-point annotation entered by users. Specifically, nodes are split into three types of seeds, each of which assigns a different weight to the edges, namely the positive sample seed, the seed to be cut, and the negative sample seed.

Positive sample seeds are superpixels covered by the two-point annotation and assigned to the positive sample set. The weights of the edges connected to the terminal vertex $V_s$ are set to 0 for positive sample seeds. Specifically, the weights of the edges connected to the terminal vertex $V_t$ are set to "MAXIMUM", a very large number in piratical, e.g., $10^5$. The positive sample seed is denoted as $V_{positive}$, as indicated by the blue dots in Fig. 4.

To-cut Seed. Superpixels as "To-Cut Seeds" have not been assigned a seed category yet. These nodes need to be computed by Graph Cut before they are automatically assigned to positive or negative samples as shown by the green point in Fig.4.

Negative sample seeds are instead of the adjacent superpixels of each to-cut seed which is visited. If a superpixel has

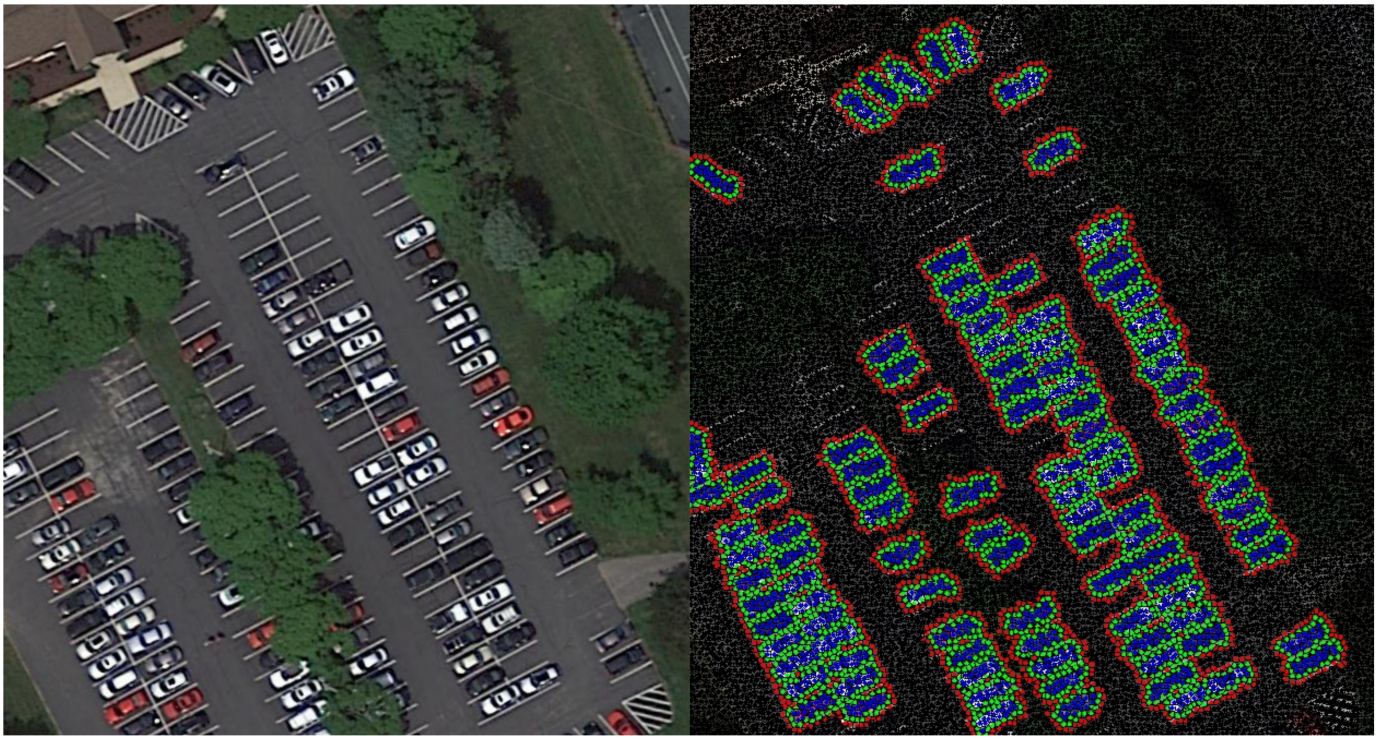


Fig. 4: Example of seed visualization. Positive seeds, to-cut seeds and negative seeds are colored in blue, green, and red, respectively. Specifically, invalid samples are illustrated in black.

not yet been assigned, it will be set as a negative sample seed, which can be noted as $V_{negative}$, as shown by the red point in Fig. 4.

In this paper, the positive sample seeds, the seeds to-be-segmented, and the negative sample seeds are uniformly referred to as valid nodes, denoted as $V_{valid} = \{V_{positive}, V_{negative}, V_{to-cut}\}$, respectively. Other unassigned superpixels are assigned as invalid nodes and are shown as black points in Fig. 4. The proposed local-graph-based segmentation will not involve those nodes in the graph, which can significantly reduce the computational cost. Finally, t-links are obtained based on the connection of valid points and terminal vertices by the following three kinds of edges, the connection of positive sample points and terminal vertices $E_{(V_{positive},V_T)}$, the connection of to-be-segmented points and terminal vertices $E_{(V_{to-cut},V_T)}$, and the connection of negative sample points and terminal vertices $E_{(V_{negative},V_T)}$.

*3) Building n-links:* Typically, in an image graph structure, each pixel acts as a node and is connected to its nodes in four neighbors: up, down, left, and right. This forms a grid-like graph structure. Moreover, in this paper the simplified graph structure consists of irregularly shaped superpixels, as shown in Fig. 5, resulting in the inability to directly calculate the center-of-mass coordinates of neighboring superpixels. Therefore, an edge-based neighbor searching method is proposed to find the neighboring superpixels of each superpixel.

Specifically, after pixel clustering, a matrix of superpixel boundaries is generated, recorded by different superpixel indexes. Since the pixels in the top, down, left or right directions of the boundary values must belong to different superpixels,

Fig. 5: Example of irregular superpixel distribution, where a superpixel (denoted by red) with its surroundings (denoted by green) are selected. This is caused by both size variation and angle rotation.

the connectivity between superpixels can be confirmed simply. A hash table is then built to record a pairs of connection, i.e., n-links, between these different superpixels. This table will be

used in building the local graph.

According to the graph structure, the nodes are divided into three types: positive samples, negative samples, and to be cut. Thus, two types of edges are formed between them: $E_{(V_{positive},V_{to-cut})}$ which connects positive sample nodes to to be cut nodes, and $E_{(V_{to-cut},V_{negative})}$ which connects to-be-cut nodes to negative sample nodes.

In addition, edges connecting valid nodes to unassigned points and edges between unassigned points to each other are not considered into the graph. The above two types of edges are collectively referred to as valid edges:

$$E_{\text{valid}} = \left\{ E_{(v_{\text{positive}}, V_{\text{to-cut}})}, E_{(v_{\text{to-cut}}, V_{\text{negative}})} \right\} \tag{3}$$

The weight values of two neighboring nodes $V_{valid_i}$ and $V_{valid_j}$ are calculated by their pixel values. The weight values, $w_{valid}$, of $E_{(V_{valid_i},V_{valid_j})}$ are given in the following equation:

$$w_{\text{valid}} = \frac{1}{1 + \sqrt[2]{\left(\bar{I}_i^{RGB} - \bar{I}_j^{RGB}\right)^2}} \tag{4}$$

where $\bar{I}^{RGB}$ is the mean superpixel intensity through RGB channels. By the above steps, the algorithm has obtained the set of nodes and the set of edges $E$ so that the graph $G$ can be successfully constructed.

### *E. Local-graph-based Segmentation and Annotation Box Generation*

Therefore, as the last step of the processing procedure, the proposed method performs a segmentation operation on the graph $G$ using the max-flow method. This divides the subgraph into two parts consisting of positive and negative sample nodes. Next, the segmented superpixels are mapped back into the subgraph to obtain a bipartite graph. As seen in Fig. 2, black represents the background and white represents a small object. The white region is framed with an adjoining rectangle, the coordinates of which are the generated box annotations. Each subgraph generates a unique box annotation based on the user-defined annotations and maps the annotation information back to the original graph; that is, all the box annotations of the whole graph are obtained. In addition, the category labels of annotations are provided by users when providing the two-point annotations.

## IV. Experimental Results

In this chapter, the experimental platform, dataset, and evaluation metrics are first be described in detail. Then, the proposed point-based bounding box annotation method (PBA) is analyzed in depth through extensive experiments to assess its validity and reliability.

### *A. Datasets*

**Tiny-Dota**: the Tiny-Dota dataset was proposed by Lee et al. [11] and is a subset on the Dota dataset, which is a large-scale target detection dataset for aerial images. The image sizes range from 800 × 800 to 20,000 × 20,000 pixels, and the size of each object varies greatly, from a few pixels to thousands of pixels. In the Dota 2.0 dataset, there are 11,268 images and 1,793,658 instances with a total of 18 object types. Although the Dota dataset contains many small objects, the categories such as baseball field, track and field, football field, basketball court, etc. still belong to the large-sized objects. The Tiny-Dota dataset is targeted instead to find out small objects with a total of 8 categories: PL (Plane), BR (Bridge), SV (Small Vehicle), LV (Large Vehicle), SH (Ship), ST (Storage Tank), SP (Swimming Pool), and HC (Helicopter). As introduced in Lee et al., the involved processing steps are as follows:

1). The original Dota dataset has no public test set annotation, so it is necessary to merge the training set and validation set, and re-divide it into the training set, the validation set and the test set with percentages of 70%, 10% and 20%, respectively;
2). As the image size in the Dota dataset is always very large, following the works in [48], [11], a method of cropping the images into 1024 × 1024 patches is adopted;
3). Finally, the instances are filtered and only the eight categories mentioned above are maintained.

**DIOR-R** [49]: DIOR is a large-scale, publicly available dataset of optical remote sensing images, which contains 23,463 images and 192,472 instances covering 20 object categories. Since the method in this paper focuses on small objects, as with the Tiny-Dota dataset screening eight categories are utilized : APL (airplane), SH (ship), STO (storage tank), BR (bridge), VE (vehicle), WM (windmill), BC (basketball court), and TC (tennis court).

### *B. Evaluation Metrics*

Three evaluation metrics are used to validate the performance of the proposed method: annotation time (time), IoU and mAP.

**Time**: The annotation time is an evaluation metric that measures the speed of algorithm execution. It is the total time required to complete the tasks of annotating an image. Time cost is crucial for annotation efficiency, because reducing annotation time means that more data can be annotated faster.

**IoU** (Intersection and Union Ratio): IoU is used to measure the degree of overlap between the predicted bounding box and the ground-truth bounding box, and is calculated by dividing the area of intersection between these two bounding boxes by the area of their union. IoU values range from 0 to 1, where higher values indicate larger overlaps between the predicted and actual bounding boxes. The annotation quality is evaluated by calculating the average IoU values between annotations produced by different annotation methods and the ground truth annotations.

**mAP** (Mean Accuracy): mAP is a comprehensive measure of the performances of the object detection algorithms, computed by means of precision and recall. Here, it is important to note that an IoU threshold (generally taken as 0.5) is considered, and the object detection is considered as correct only when the IoU value between the predicted bounding box and the actual bounding box is greater than this threshold [48], [50]. For each prediction the curve is plotted on the

detection rate-correctness plane. The area under the PR curve is calculated, and this can be done by discretizing the check-percentage (e.g. 0.0 to 1.0 with an interval of 0.1), then calculating the corresponding maximum check-percentage values, and eventually summing them up to get the AP. Averaging the AP values of all the categories gives the mAP value.

### C. Results on Time Consumption

The PBA method contains four main parts: pixel clustering, graph creation, graph segmentation, and bounding box generation. Since annotating all images requires quite a big labor work, a simplified method is adopted to simulate the manual annotation process. Specifically, two-point annotations are generated from the original annotation boxes with random oscillation added.

Both Tiny-Dota and DIOR-R dataset provide annotations in the form of $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$. In this paper, the center line coordinates are obtained by calculating the median of the first two points of the annotation frame $(c_1, c_{1_y})$ and the median of the last two points $(c_{2_x}, c_{2_y})$:

$$\begin{aligned} c_{1x} &= \frac{x_1 + x_2}{2}, \quad c_{1y} = \frac{y_1 + y_2}{2} \\ c_{2x} &= \frac{x_3 + x_4}{2}, \quad c_{2y} = \frac{y_3 + y_4}{2} \end{aligned} \tag{5}$$

After obtaining the coordinates of the center line, a translation is added to this as random noise, the translation range is sampled randomly from one of the box edge, with four directions of up, down, left and right. The simulated two-point annotation is obtained by the above operation. The coordinates are then rotated to convert $(x, y)$ to the rotated coordinates $(x_r, y_r)$:

$$\begin{aligned} x_r &= x \times \cos\theta - y \times \sin\theta \\ y_r &= x \times \sin\theta + y \times \cos\theta \end{aligned} \tag{6}$$

After generating pseudo two-point annotations for all the data in Tiny-Dota and DIOR-R respectively, they are fed into PBA to get the annotation frames. The statistics of the average time for each PBA step are shown in Table I.

TABLE I: Time consumption (s) for each PBA step (%).

| Modules in PBA | Tiny-Dota | DIOR-R |
|---|---|---|
| Super clustering | 1.5 (12.4%) | 1.9 (11.7%) |
| Building a Graph | 9.2 (76.1%) | 12.5 (77.2%) |
| Graph cut | 0.4 (3.3%) | 0.8 (4.9%) |
| Box generation | 0.1 (0.8%) | 0.1 (0.6%) |
| Others | 0.9 (7.4%) | 0.9 (5.6%) |
| **Total Time** | **12.1** | **16.2** |

As seen from the table above, the main time spent by the PBA is on the graph building part, which is also the most complex part of the algorithm. The time spent on the "other" items refers to the running time of the code for image loading, annotation loading and other coding execution intervals.

### D. Analysis of PBA annotation quality

The IoU between GT (Ground Truth) and PBA-generated annotations is compared to evaluate the quality of annotations generated by PBA. The time cost of annotating with the whole dataset is not affordable. Thus, 50 images in the dataset are selected. The same method as mentioned in the previous section is used to generate pseudo two-point annotations, as inputs to PBA for segmentation.

The proposed method is then compared with classical interactive segmentation methods, such as Graph Cut and the Lazy-snapping method improved based on Graph Cut. Additionally, the latest interactive segmentation method SAM (Segment Anything Model) is also included. the following modification is conducted to adopt those methods to segment small objects in the remote sensing field.

1). Line annotation is utilized to annotate objects, which is analogue to the two-point annotation as proposed in this paper. Apart from positive annotations, the Graph Cut and the Lazy-snapping also require negative annotations. Thus, negative regions are kept until all the positive instances are recognized.
2). In SAM, point prompts are utilized as inputs, because SAM does not support line prompts. As a result, each object is annotated as a single point, and the object region is automatically segmented via SAM. As introduced in the RSPrompter [35], additional bounding box annotation information on remote sensing is required. However, SAM is used here as an annotator rather than a segmentation model. As a result, no fine-tuning is applied.

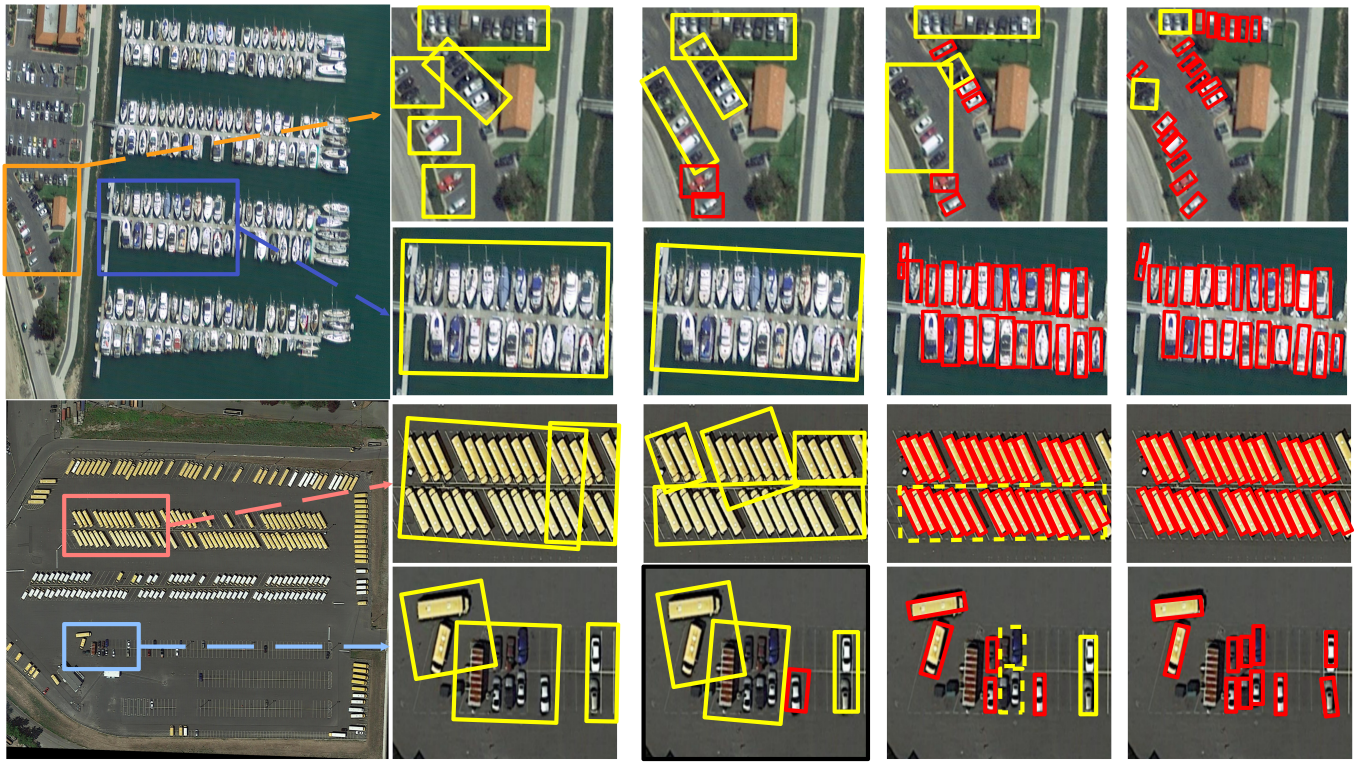
Table. II shows the comparison of IoU on Tiny-Dota and DIOR-R datasets, respectively.

TABLE II: Comparisons of IoU for different methods on two benchmark datasets, respectively.

| Method | Tiny-Dota | DIOR-R |
|---|---|---|
| Graph Cut | 42.1 | 43.6 |
| Lazy-snapping | 55.8 | 58.6 |
| SAM | 60.4 | 62.8 |
| PBA (ours) | **69.7** | **72.2** |

From the table, the IoU of the proposed method reaches 69.8% and 72.2% on the two datasets, respectively. In the object detection task, an IoU greater than 50% is recognized as a positive sample, which indicates that the annotations produced by the PBA method are applicable as labels to train a detector. The proposed PBA exceeds the classical Graph Cut method, Lazy-snapping, and SAM by about 28%, 14%, and 9%, respectively.

Since the Graph Cut method builds the graph structure for segmentation using the original pixels, due to the limited size, the graph construction may not be able to accurately capture the edge information of small objects. As a result, for densely placed small objects, Graph Cut may merge more objects into one bounding box, as shown in the second column of Fig. 6. Since the proposed PBA method is based on the graph



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

Fig. 6: Visualization on (a) the original images, (b) Graph Cut, (c) Lazy-snapping, (d) SAM and (e) the proposed PBA, respectively. The original positions of patches are annotated by orange, blue, pink and light blue, respectively. The false annotated boxes are highlighted in yellow, while the true positive annotations are marked in red. Occlusion and blurry are two main challenges on annotating small objects. Graph Cut and Lazy-snapping may fail on annotating dense small objects, which may merge multiple objects into one OBBOX. SAM annotates more accurately but is interfered by the object blurry and the background features. Meanwhile, SAM may fail occasionally caused by the user annotation, which is highlighted by dashed yellow boxes. As a comparison, the proposed is robust on occlusion and blurring, generating more accurate boxes.

structure constructed by superpixels, it highlights the internal structure and the edge information of small objects, achieving more accurate segmentation outputs. The overall structure of the Lazy-snapping method is analogue to that of PBA. However, the setting of positive and negative sample seeds is quite different. When applied to the actual annotation situation, Lazy-snapping needs to carry out interactive annotation several times. Compared to our method, it not only needs to set appropriate positive sample seeds, but also needs to set many negative sample seeds. As the backgrounds information of small object in the remote sensing scene is more complex, the quality of negative sample seeds has a great impact on the segmentation effect. Thus, this process requires massive delicate manual operations, which increases the burden of annotators, improving the probability of annotation mistakes, as well as the time cost of annotation.

The SAM method has achieved state-of-the-art results in the field of commonly used imagery. However, when applied to annotating remote sensing small objects, due to the limited size and vision details of small objects, it results into poor segmentation results, as shown in the fourth column of Fig. 6.

### *E. Effect on Training Object Detectors*

To further validate the practicality of the annotations generated by the proposed method, the annotations generated by Graph Cut, SAM, and PBA methods and the original manually annotated GT are utilized to train oriented object detection methods. During testing, the original GT is applied. In this paper, a Rotate RetinaNet [51] is used as the oriented object detection network. To ensure the fairness of the experiment, the network uses the same parameter settings. The rotation angle is in "OC" format and the backbone network is ResNet-50. The network is trained by 12 epochs with a batch size of 2. The initial learning rate is 0.0025. SGD is utilized as optimizer with a momentum of 0.9, and a weight decay of 0.0001. Table III and Table IV show the mAP results on Tiny-Dota and DIOR-R for the models trained by the four annotation methods.

The experimental results show that the network trained with the PBA-annotated samples is more reliable than other methods, with an average mAP around 2.8% higher on different datasets. For small objects annotating, SAM surpasses PBA in detecting storage tanks (Tiny-Dota) and oil tanks (DIOR-R) by about 2%, which may be caused by the seed setting. Indeed,

TABLE III: The mAP (%) comparison of the RetinaNet model trained Using different annotations on the Tiny-Dota Dataset. "GT" is the "ground truth annotation" supplied by the dataset publisher.

| Class | Graph Cut | Lazy-snapping | SAM | PBA (ours) | GT |
|---|---|---|---|---|---|
| PL | 35.4 | 53.4 | 55.2 | 59.3 | 81.1 |
| SH | 41.2 | 51.7 | 56.7 | 57.2 | 65.4 |
| ST | 45.7 | 48.9 | 57.3 | 55.8 | 63.5 |
| BR | 31.8 | 33.6 | 35.2 | 37.4 | 43.1 |
| LV | 52.3 | 54.1 | 58.3 | 64.3 | 71.2 |
| SV | 35.3 | 37.3 | 41.6 | 44.2 | 68.3 |
| HC | 15.2 | 18.8 | 21.7 | 25.1 | 31.4 |
| SP | 45.1 | 48.7 | 52.6 | 60.7 | 66.8 |
| mAP | 37.8 | 43.3 | 47.3 | 50.5 | 61.3 |

TABLE IV: The mAP (%) comparison of the RetinaNet model trained Using different annotations on the DIOR-R Dataset. "GT" is the "ground truth annotation" supplied by the dataset publisher.

| Class | Graph Cut | Lazy-snapping | SAM | PBA (ours) | GT |
|---|---|---|---|---|---|
| APL | 28.5 | 34.0 | 43.2 | 49.7 | 71.4 |
| SH | 54.1 | 64.8 | 72.3 | 76.4 | 81.1 |
| STO | 43.6 | 53.6 | 63.9 | 61.0 | 71.3 |
| BR | 14.2 | 18.7 | 21.9 | 23.9 | 33.1 |
| VE | 29.8 | 37.8 | 42.6 | 47.1 | 65.7 |
| WM | 37.2 | 48.7 | 55.1 | 60.2 | 71.2 |
| BC | 53.1 | 66.4 | 78.3 | 77.3 | 83.5 |
| TC | 54.7 | 69.2 | 78.1 | 78.2 | 83.1 |
| mAP | 39.4 | 49.2 | 56.9 | 59.2 | 70.1 |

in our seed setting paradigm, only superpixels around positive seeds are considered for the graph formation. However, for square-shape objects, some positive seeds may be missing as the annotation line cannot highlight positive seeds along other orientations. As a result, the annotated OBBOX may be smaller than the ground-truth. This issue requires some future work to optimize the segmentation pipeline for this type of objects.

The state-of-the-art results achieved by PBA are mainly caused by objects that have more distinctive features in the image, such as clear edges, high contrast, etc., which enable the proposed PBA to segment these objects more effectively. However, there is still a gap when compared with the manual annotations, with a legging around 11% on mAP. This is because some objects and backgrounds are not easily distinguishable, or the shapes of some small objects are unique. Another future work would be to annotate small objects with interference, such as illumination, blurry, occlusion, etc.

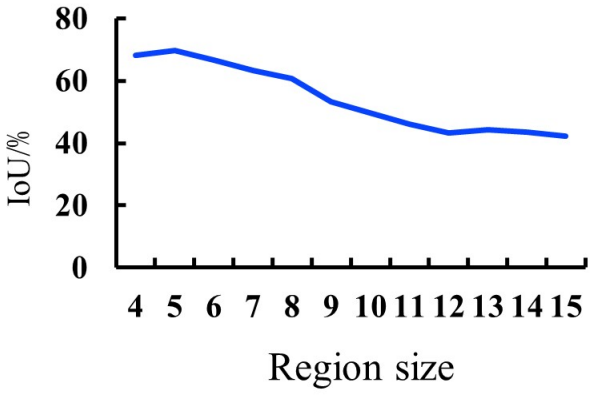



Fig. 7: The relationship between IoU and the bounding boxes generated with different region sizes.

### F. Hyper parametric analysis

In this subsection, some important hyperparameters that affect PBA results are analyzed, namely the region size and the line length ratio. As Tiny-Dota and DIOR-R have a similar imaging view and scenario, the experiments of hyper parametric analysis are performed on the Tiny-Dota dataset only.

**Region size**, i.e. the average size of the superpixels generated by the SLIC algorithm. When the region size gets smaller, the visual details are better preserved. However, extremely small superpixels are hard to be aggregated, which leads to under-segmentation. Finally, as the region size gets larger, superpixels may contain too much background information, causing over-segmentation, as shown in Fig. 7. In this paper, an optimal value of 5 pixels for region size is considered.

**Line length ratio**, i.e. the distance between two points and the object size. This parameter has effect on determining positive seeds. As seen in Fig. 8, a lower values may cause less superpixels as positive seeds. As a result, a smaller BBOX is generated. However, when the line length ratio exceeds 0.8, the quality of the annotation becomes slightly inferior. This is because more background may be involved from the boundary super-pixels. As a result, the recommended value of this parameter is 0.8.

### G. Comparison of Annotation Times in Practice

The proposed PBA aims at reducing the time cost of image annotation. To validate its effectiveness, time cost in practice is measured.

Similarly to previous works [42], [7], a subset of 20 images is extracted from the Tiny-Dota dataset as sample images, containing all the categories. The number of objects is approximately equal to the average number of objects in the whole Tiny-Dota dataset. To ensure the professionalism of the experiment, five volunteers with annotation-related experience were invited to participate in the experiment.

In the manual annotation part, each of the five participants used the LabelMe tool to annotate the 20 images with rotated boxes. During the annotation process, the annotation time of

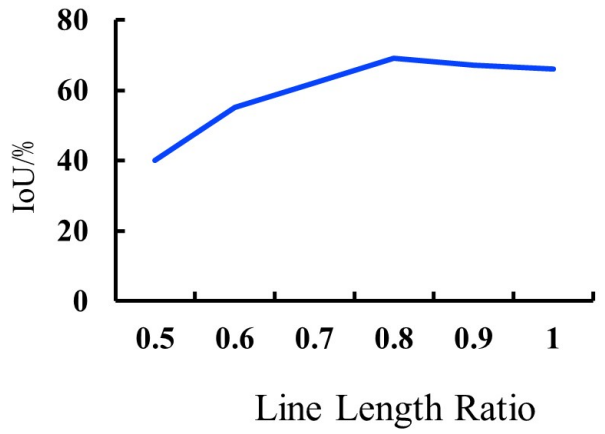



Fig. 8: The relationship between IoU and the line length ratio.

each person was recorded for comparative analysis. Each of the five participants firstly used the LabelMe tool to manually annotate via two points. Next, the annotated data was fed into the algorithm to generate the corresponding oriented annotations. During this process, the time required for the algorithm to run was also recorded. The total time cost of the PBA method was obtained by adding the manual annotation time and the time taken by generating the OBBOX annotations automatically. The experiment was repeated twice to reduce accidentally induced errors, and Table V shows the statistics of the two experiments.

TABLE V: Annotation time (s) of 5 annotators on the Tiny-Dota dataset (first and second time). The symbol "↑" indicates the ratio of time saving compared with the existing four-point annotation work.

| Category | First (mean ± std) | Second (mean ± std) |
|---|---|---|
| 4-point Anno. | 2761.40 ± 90.79 | 2713.80 ± 67.68 |
| SAM | 1835.00 ± 50.31 (↑ 33%) | 1652.40 ± 41.20 (↑ 39%) |
| 2-point Anno. | 1132.20 ± 40.37 | 1129.80 ± 113.24 |
| PBA processing | 318.80 ± 0.40 | 318.60 ± 0.49 |
| Overall | 1451.00 ± 40.52 (↑ 47%) | 1448.40 ± 113.37 (↑ 46%) |

For the DIOR-R dataset the same experimental design was adopted, except that the subset was made of 50 images, which contain all the categories, and the number of objects in each image is approximately equal to the average number of objects in each image of the whole DIOR-R dataset. Table VI shows the data statistics of the two experiments on the DIOR-R dataset.

The tables show that, the proposed PBA could significantly reduces the manual work by at least 50% because the data processing step does not require manual interaction. By comparing the two annotation experiments on each dataset, the second manual annotation phase generally takes less time than the first on, due to the annotators becoming more and more familiar with the annotation tools.

TABLE VI: Annotation time (s) of 5 annotators on the DIOR-R dataset (First and Second Time). The symbol "↑" indicates the ratio of time saving compared with the existing four-point annotation work.

| Category | First (mean ± std) | Second (mean ± std) |
|---|---|---|
| 4-point Anno. | 3082.00 ± 89.12 | 3049.20 ± 123.23 |
| SAM | 2751.10 ± 30.31 (↑ 11%) | 2325.60 ± 25.50 (↑ 24%) |
| 2-point Anno. | 1483.80 ± 113.77 | 1389.80 ± 97.33 |
| PBA processing | 601.00 ± 0.00 | 598.40 ± 0.55 |
| Overall | 2084.80 ± 113.77 (↑ 32%) | 2012.20 ± 111.41 (↑ 34%) |

Furthermore, the time consumption of the manual OBBOX annotation and the PBA annotation are estimated and the results are shown in Table VII. The two-point annotation time and the box annotation time are the average annotation time estimated by the results of the previous experiments. The average time for an annotator to complete the annotation of an object is about 2 seconds, while the average time for completing a rotated box annotation is about 6 seconds. As can be seen, the method proposed in this paper can significantly reduce the time cost of annotation.

TABLE VII: Annotating time cost (h) simulation on the Tiny-Dota and DIOR-R datasets.

| Dataset | 2-point Anno. | PBA | **Total** | **4-point Anno.** |
|---|---|---|---|---|
| Tiny-Dota | 357 | 139 | 506 | 1071 |
| DIOR-R | 107 | 78 | 194 | 321 |

Though the difference in time cost between the PBA method and the traditional manual annotation method is obvious, PBA exhibits a lower time cost. This will be of great practical significance for large-scale image annotation tasks, especially in the fields of computer vision and deep learning.

## V. Conclusion

Aiming at the problem of high cost on annotating small objects in the remote sensing field, this paper proposes a low-cost annotation method, named PBA. PBA is an efficient method for annotating small objects in remote sensing images, and only needs two-point annotations to generate oriented bounding boxes without pre-training or fine tuning. The algorithm firstly clusters pixels on the original image using the SLIC algorithm. Then, positive sample nodes are identified, as well as the to-be-segmented nodes and the negative sample nodes. Among that, the position and the superpixel intensities are collected to build a graph. Next, the graph is segmented by the max-flow method, to divide the graph into two parts, with positive and negative samples. These two parts are mapped

back to the original graph to form a binary mask. Finally, each object region is converted into an oriented rectangular box.

A series of experiments prove the effectiveness of PBA, since the annotation method and the average IoU in this paper exceeds the state-of-the-art SAM method by about 9%. Experimental results validate that the proposed PBA algorithm is an effective semi-automatic annotation method for small objects in remote sensing images.

However, there are still some limitations. At first, the effect on annotating objects with shapes similar to a square is poor, due to the above mentioned issue with positive seeds identification. It is important to subdivide the positive sample seeds, the seeds to be segmented, and the negative sample seeds, and a well-designed division policy can effectively improve the quality of the OBBOX annotation. Another future work will be the optimization of the annotation method with a seed template for each category, or to introduce deep learning feature extraction methods to assist seed division.

## REFERENCES


[1] H. F. Tolie, J. Ren, and E. Elyan, "Dicam: Deep inception and channel-wise attention modules for underwater image enhancement," *Neurocomputing*, vol. 584, p. 127585, 2024.

[2] Y. Li, J. Ren, Y. Yan *et al.*, "Cbanet: An end-to-end cross band 2-d attention network for hyperspectral change detection in remote sensing," *IEEE Trans. on Geoscience and Remote Sensing*, 2023.

[3] X. Li, J. Deng, and Y. Fang, "Few-shot object detection on remote sensing images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[4] G. Cheng, X. Xie, W. Chen *et al.*, "Self-guided proposal generation for weakly supervised object detection," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[5] Y. Yu, X. Yang, Q. Li *et al.*, "H2rbox-v2: Incorporating symmetry for boosting horizontal box supervised oriented object detection," in *37th Conf. on Neural Information Processing Systems*, 2023.

[6] J. F. Mullen Jr, F. R. Tanner, and P. A. Sallee, "Comparing the effects of annotation type on machine learning detection performance," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[7] A. Kirillov, E. Mintun, N. Ravi *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[8] A. Radford, J. W. Kim, C. Hallacy *et al.*, "Learning transferable visual models from natural language supervision," in *Int. Conf. on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[9] C. Jia, Y. Yang, Y. Xia *et al.*, "Scaling up visual and vision-language representation learning with noisy text supervision," in *Int. Conf. on Machine Learning*. PMLR, 2021, pp. 4904–4916.

[10] A. Ramesh, M. Pavlov, G. Goh *et al.*, "Zero-shot text-to-image generation," in *Int. Conf. on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[11] C. Lee, S. Park, H. Song *et al.*, "Interactive multi-class tiny-object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2022, pp. 14 136–14 145.

[12] 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 J. of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[14] M. Bernhard and M. Schubert, "Robust object detection in remote sensing imagery with noisy and sparse geo-annotations," in *Proc. the 30th Int. Conf. on Advances in Geographic Information Systems*, 2022, pp. 1–4.

[15] S. Moschos, P. Charitidis, S. Doropoulos, A. Avramis, and S. Vologiannidis, "Streetscouting dataset: A street-level image dataset for finetuning and applying custom object detectors for urban feature detection," *Data in Brief*, vol. 48, p. 109042, 2023.

[16] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi, "Medical image segmentation using deep learning: A survey," *IET Image Processing*, vol. 16, no. 5, pp. 1243–1267, 2022.

[17] Y. Yan, J. Ren, H. Sun *et al.*, "Nondestructive quantitative measurement for precision quality control in additive manufacturing using hyperspectral imagery and machine learning," *IEEE Trans. on Industrial Informatics*, 2024.

[18] P. Ma, J. Ren, G. Sun *et al.*, "Multiscale superpixelwise prophet model for noise-robust feature extraction in hyperspectral images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[19] Y. Yan, J. Ren, Q. Liu, H. Zhao, H. Sun, and J. Zabalza, "Pca-domain fused singular spectral analysis for fast and noise-robust spectral–spatial feature mining in hyperspectral classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2021.

[20] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognition*, vol. 111, p. 107623, 2021.

[21] E. Oğuz, A. Küçükmanisa, R. Duvar, and O. Urhan, "A deep learning based fast lane detection approach," *Chaos, Solitons & Fractals*, vol. 155, p. 111722, 2022.

[22] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. the IEEE/CVF Int. Conf. on Computer Vision*, 2019, pp. 9627–9636.

[23] P. Jiang, D. Ergu, F. Liu *et al.*, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[24] Z. Yang, S. Liu, H. Hu *et al.*, "Reppoints: Point set representation for object detection," in *Proc. the IEEE/CVF Int. Conf. on Computer Vision*, 2019, pp. 9657–9666.

[25] L. Hou, K. Lu, X. Yang, Y. Li, and J. Xue, "G-rep: Gaussian representation for arbitrary-oriented object detection," *Remote Sensing*, vol. 15, no. 3, p. 757, 2023.

[26] X. Yang, X. Yang, J. Yang *et al.*, "Learning high-precision bounding box for rotated object detection via kullback-leibler divergence," *Advances in Neural Infor. Proc. Systems*, vol. 34, pp. 18 381–18 394, 2021.

[27] X. Yang, Y. Zhou, G. Zhang *et al.*, "The kfiou loss for rotated object detection," in *The 11th Int. Conf. on Learning Representations*, 2022.

[28] Y. Pang, Y. Zhang, Q. Kong *et al.*, "Socdet: A lightweight and accurate oriented object detection network for satellite on-orbit computing," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[29] Z. Li, B. Hou, Z. Wu *et al.*, "Gaussian synthesis for high-precision location in oriented object detection," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[30] S. Zheng, Z. Wu, Y. Xu, and Z. Wei, "Instance-aware spatial-frequency feature fusion detector for oriented object detection in remote-sensing images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[31] B. C. Russell, A. Torralba, K. P. Murphy *et al.*, "Labelme: A database and web-based tool for image annotation," *Int. Journal of Computer Vision*, vol. 77, pp. 157–173, 2008.

[32] T.-Y. Lin, M. Maire, S. Belongie *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: Part V 13*. Springer, 2014, pp. 740–755.

[33] M. Everingham, S. A. Eslami, L. Van Gool *et al.*, "The pascal visual object classes challenge: A retrospective," *Int. J. Computer Vision*, vol. 111, pp. 98–136, 2015.

[34] J. Deng, W. Dong, R. Socher *et al.*, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[35] K. Chen, C. Liu, H. Chen, H. Zhang, W. Li, Z. Zou, and Z. Shi, "Rsprompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model," *arXiv preprint arXiv:2306.16269*, 2023.

[36] J. Wu, R. Fu, H. Fang *et al.*, "Medical sam adapter: Adapting segment anything model for medical image segmentation," *arXiv preprint arXiv:2304.12620*, 2023.

[37] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. on Cybernetics*, vol. 48, no. 3, pp. 929–940, 2017.

[38] Boykov and Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proc. 9th IEEE Int. Conf. on Computer Vision*, 2003, pp. 26–33.

[39] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *Int. Conf. on Systems and Informatics*. IEEE, 2012, pp. 1936–1941.

[40] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proc. 8th IEEE Int. Conf. on Computer Vision*, vol. 1. IEEE, 2001, pp. 105–112.

[41] L. Chen, R. Kyng, Y. P. Liu *et al.*, "Maximum flow and minimum-cost flow in almost-linear time," in *IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022, pp. 612–623.

[42] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. on Graphics (ToG)*, vol. 23, no. 3, pp. 303–308, 2004.
[43] A. Seal, A. Das, and P. Sen, "Watershed: An image segmentation approach," *Int. Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 2295–2297, 2015.
[44] L. Chen, T. Yang, X. Zhang, W. Zhang, and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 8823–8832.
[45] N. Carion, F. Massa, G. Synnaeve *et al.*, "End-to-end object detection with transformers," in *European Conf. on Computer Vision*. Springer, 2020, pp. 213–229.
[46] S. Zhang, Z. Yu, L. Liu *et al.*, "Group r-cnn for weakly semi-supervised object detection with points," in *Proc. the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2022, pp. 9417–9426.
[47] R. Achanta, A. Shaji, K. Smith *et al.*, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
[48] G.-S. Xia, X. Bai, J. Ding *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.
[49] G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han, "Anchor-free oriented proposal generator for object detection," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.
[50] G. Cheng, X. Yuan, X. Yao *et al.*, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2023.
[51] T.-Y. Lin, P. Goyal, R. Girshick *et al.*, "Focal loss for dense object detection," in *Proc. the IEEE Int. Conf. on Computer Vision*, 2017, pp. 2980–2988.

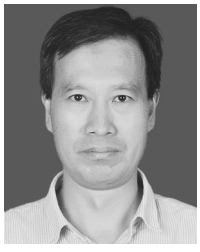
**Lu Lei** received received the B.E. degree in software engineering from Yunnan University, Kunming, China, in 2021, and the M.Eng. degree in software engineering from Northwestern Polytechnical University, Xi'an, China, in 2023, supervised by Dr. Zhenyu Fang. His research interests include computer vision and low cost annotation in remote sensing. He is currently working in the industry for gaining practical experience.

**Zhenyu Fang** received the Ph.D. degree in electronic and electrical engineering from the University of Strathclyde in July 2020. He is a currently an Associate Professor with the the School of Software, Northwestern Polytechnical University, Xi'an, China, and also an Associate Professor with Yangtze River Delta Research Institute of NPU, Taicang, China. His main interests are algorithm development for small object detection, self-supervised learning, semi-supervised learning and model compression.

**Jinchang Ren** (Senior Member, IEEE) received the B.E. degree in computer software, the M.Eng. degree in image processing, and the D.Eng. degree in computer vision from Northwestern Polytechnical University, Xi'an, China, in 1992, 1997, and 2000, respectively, and the Ph.D. degree in electronic imaging and media communication from the University of Bradford, Bradford, U.K., in 2009. He is a Professor with the National Subsea Centre, Robert Gordon University, Aberdeen, U.K., and also a Visiting Professor with Guangdong Polytechnic Normal University, Guangzhou. He has published over 350 articles. His research interests include computer vision and multimedia signal processing, especially on hyperspectral imaging, machine learning, and big data analytics. Dr. Ren acts as an Associate Editor for five international journals, including the IEEE Transactions on Geoscience and Remote Sensing, Journal of the Franklin Institute, Big Data Analytics, etc.

**Paolo Gamba** (Fellow, IEEE) received the Laurea (cum laude) and Ph.D. degrees in electronic engineering from the University of Pavia, Pavia, Italy, in 1989 and 1993, respectively. He is currently a Full Professor of telecommunications at the University of Pavia, Pavia, Italy. He has published more than 150 articles in international peer-reviewed journals and presented more than 300 research works in workshops and conferences. From 2009 to 2013, he served as the Editor-in-Chief for the IEEE Geoscience and Remote Sensing Letters. He has been a member of the GRSS AdCom of the IEEE Geoscience and Remote Sensing Society since 2009, the Chapter Committee Chair from 2014 to 2016, and a Latin American Activity Liaison from 2014 to 2016. He has served as the Vice President for Professional Activities in 2016, the Executive Vice President from 2017 to 2018, the Executive Vice President from 2019 to 2020, and the President and past President for GRSS. He is also the Editor-in-Chief of IEEE Geoscience and Remote Sensing Magazine (GRSM).

**Jiangbin Zheng** received the Ph.D. degree from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2002.,He is currently a Full Professor and the Dean of the School of Software, Northwestern Polytechnical University. His research interests include computer graphics, computer vision, and multimedia. He has authored over 100 articles in the above-related research area.



**Huimin Zhao** received the B.Sc. and M.Sc. degrees in signal processing from Northwestern Polytechnical University, Xi'an, China, in 1992 and 1997, respectively, and the Ph.D. degree in electrical engineering from Sun Yat-sen University, Guangzhou, China, in 2001.,He is currently a Professor and Dean with the School of Computer Sciences, Guangdong Polytechnic Normal University, Guangzhou. His research interests include image/video and information security technologies, and applications.