



# Aerial Image Object Detection Based on Superpixel-Related Patch

Jiehua Lin<sup>(✉)</sup>, Yan Zhao, Shigang Wang, Meimei Chen, Hongbo Lin,  
and Zhihong Qian

Jilin University, Changchun, China

linjh20@mails.jlu.edu.cn, {zhao\_y, wangsg, chenmm,  
hblin}@jlu.edu.cn

**Abstract.** Aerial image object detection and recognition has attracted increasing attention in recent years. Many excellent detectors have been proposed. However, due to the high-resolution of aerial images, these detectors are difficult to directly apply to aerial images. In order to solve the problem of hard processing caused by high resolution, it is generally to resize the high-resolution images into low-resolution images or cut the high-resolution images into small image patches. Cutting high-resolution aerial images into small image patches without overlap may cut an object into multiple parts which may lose the integrity of the object and causes one object to be detected as multiple objects. We design a new baseline to cut high-resolution aerial images into small image patches by using superpixel. Firstly, we use pixel-related GMM (Gaussian mixture model) to segment the high-resolution aerial images into superpixel images. Then we utilize superpixel label to cut high-resolution aerial images into low-resolution image patches with integrity of the object. Finally, we use YOLOv5 with CSL (Circular Smooth Label) to detect oriented objects. Our method effectively preserves the integrity of the object and improves the AP (Average Precision) of the object detection. This baseline can be applied not only to object detection, but also to aerial image segmentation, classification and so on. Experiments on the UCAS-AOD dataset show the effectiveness of the proposed method.

**Keywords:** Superpixel · Object detection · Aerial image · YOLOv5

## 1 Introduction

In recent years, with the development of deep learning, object detection has been improved rapidly. Many object detection methods with good performance based on deep learning have been proposed. Although these object detection methods have achieved remarkable results in natural scene images, they cannot be directly applied to aerial image object detection because of the differences between aerial images and natural scene images. Aerial image has the characteristics of dense object, arbitrary orientation and high resolution. In order to solve the problems of dense objects and arbitrary directions, researchers have proposed some oriented object detection methods such as

CSL (circular smooth label) [1], BBAVectors [2] and ROI transformer [3]. Because the resolution of remote sensing image is very high, if the detector is applied to the original image, it will consume a lot of hardware resources, so researchers generally have two ways to apply the detector. One is to resize high-resolution aerial image into low-resolution image for object detection, but it cannot extract enough features resulting in inaccurate detection. The other is to cut the aerial image into small image patches for object detection, and then merge image patches into a high-resolution aerial image. This may cut a whole object into multiple parts, which results in an object being detected as multiple objects. If there is overlap areas when cutting into small image patches, it will cause great information redundancy and resource consumption.

In this paper, we aim to find a new way to transform high-resolution image into low-resolution image for object detection, which can preserve the integrity of the object, retain more information, and avoid information redundancy. Superpixel will be a small area composed of adjacent pixels with similar characteristics such as color, brightness, texture, etc. And pixels belonging to the same object will be assigned the same superpixel label. Therefore, we propose a new baseline for high-resolution aerial image object detection. Specifically, we use the pixel-related GMM (Gaussian mixture model) superpixel segmentation method to pre-process the high-resolution aerial image, and then cut the high-resolution aerial image into low-resolution image patches according to the result of superpixel segmentation. When cutting, we keep all the pixels of one object in one superpixel at the edge area to ensure the integrity of the object, and there is no overlapping area between image patches. Finally, we take the image patches as input for object detection. YOLOv5 object detection algorithm has good performance in speed and detection performance. But because the aerial image object has arbitrary orientation, and YOLOv5 can only detect the object in the horizontal orientation, we cannot directly use YOLOv5 to detect objects. Therefore, we adopt the YOLOv5 detector combined with CSL as object detector, which introduces the angle variable to control the orientation in the representation of the bounding box of the object.

The rest of this paper is structured as follows: Sect. 2 introduces the related work about our method including the superpixel segmentation and oriented object detection. In Sect. 3, we briefly describe the proposed method. The results of the proposed method are provided in Sect. 4. At last, we conclude the whole work in Sect. 5.

## 2 Related Work

### 2.1 Superpixel Segmentation

The concept of superpixel is an image segmentation technology proposed and developed by Ren and Malik [9]. It refers to irregular pixel blocks with a certain visual significance composed of adjacent pixels with similar texture, color, brightness and other characteristics. Superpixel uses the similarity between pixels to group pixels, and uses a small number of superpixels instead of a large number of pixels to express image features, and it has been widely used in image segmentation, pose estimation, object tracking, object recognition and other computer vision applications. SLIC [10] converts the colorful image into a 5-dimensional feature vector of the color and XY coordinates in the

CIELAB color space, and then constructs a distance metric for the 5-dimensional feature vector, and performs local clustering of image pixels to generate superpixels. Zhihua Ban [11] proposed a pixel-related Gaussian mixture model (GMM) to segment images into superpixels. GMM is a weighted sum of Gaussian functions. Each function corresponds to a superpixel to set the label for pixels into superpixels. SpixelFCN [12] uses an encoding-decoding full convolutional network to implement an end-to-end superpixel prediction network.

## 2.2 Oriented Object Detection

The difference between the oriented object detector and the horizontal object detector is that the oriented object detector relies on oriented bounding boxes (OBB), and the horizontal object detector uses horizontal bounding boxes (HBB). The horizontal object detector is mainly classified into two-stage and single-stage object detectors. RCNN [4] is a typical two-stage object detection network. It first uses convolutional neural network to extract features, then uses region proposal network (RPN) to get the proposals and performs ROIpool on the region of interest (ROI), and finally classifies objects and regresses the bounding box of the proposal. Typical single-stage object detectors are YOLO [5], RetinaNet [6], CenterNet [7], etc. Compared with the two-stage object detector, the single-stage object detector directly predicts the bounding box of the object, and its speed is faster than the two-stage detector. Most of the current oriented object detector is extended from the horizontal object detector, and the angle variable is introduced to control the orientation in the representation of the object's bounding box. For example, R2CNN [8] uses a two-stage Faster RCNN architecture, first obtains the horizontal bounding box (HBB) proposals through the RPN network, then uses multi-scale pooling (ROIpooling) for each proposal, and finally predicts the orientation and obtains the oriented object bounding box (OBB). Based on RetinaNet, CSL [1] introduces a classification method to predict the orientation of the object to obtain an oriented bounding box (OBB) when regressing the bounding box. BBAVectors uses a U-shaped network based on CenterNet to generate heatmaps and obtain the center point position of the object, and then regress to a box boundary-aware vectors (BBAVectors) to obtain an oriented bounding box to achieve the result of oriented object detection.

## 3 Method

The framework of our proposed method is shown in the Fig. 1. The method we proposed is divided into the following steps. First, we use the GMM-based superpixel segmentation algorithm to segment the high-resolution aerial image, and then use the superpixel segmentation results to cut the high-resolution aerial image into small image patches. In this process, the pixels belonging to one superpixel at the edge area of the patch will be reserved. The start position of the next image patch is the end position of the superpixel at the edge of the previous image patch. In this way, cutting the high-resolution aerial image into small image patches not only avoid one object being cut into multiply parts, and there is no redundant information between patches. Then we use YOLOv5 [13] combined with CSL bounding representation to detect oriented objects.

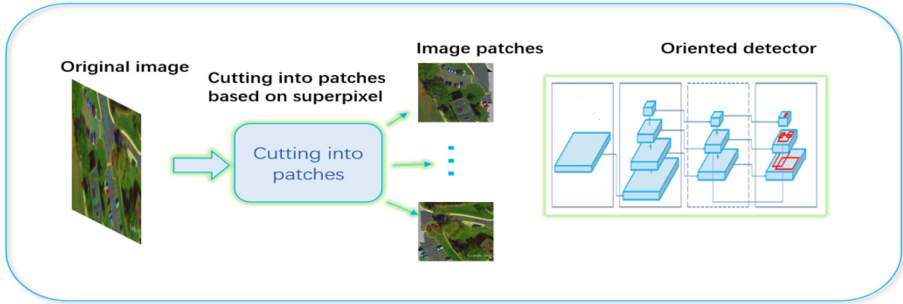


Fig. 1. Framework of our proposed method

### 3.1 Cutting Aerial Image into Image Patches Based on Superpixel

Superpixel segmentation aggregates some pixels with similar characteristics to form a larger “element” that is more representative. And this new element will serve as the basic unit of the latter image processing. The pixels in the same superpixel generally belong to the same object, which can effectively separate the object from the background, and has strong integrity.

SLIC converts the colorful image into a 5-dimensional feature vector which contains color and XY coordinates in the CIELAB color space, and then constructs a distance metric for the 5-dimensional feature vector, and performs local clustering of image pixels to generate superpixels. Assuming that the image has  $N$  pixels and is to be segmented into  $K$  superpixels, then the size of each superpixel is  $N/K$ . The distance between super pixels is  $S = \sqrt{N/K}$  (the side length of super pixels under regular conditions). The specific steps are as follows: First, it distributes the centers of  $K$  superpixels to the pixel points of the image and fine-tunes the position of the seed. The center of the superpixel is moved to the point with the smallest gradient among the 9 points in the  $3 \times 3$  range, to avoid superpixels falling on noise or boundaries. Then, two matrix **LABEL** and **DIS** are initialized, which are respectively used to store the superpixel label each pixel belonging to and the distance between the pixel to the center of the superpixel it belongs to. And the distance between each pixel within  $2S$  and the center of the superpixel is calculated. If the distance from the point to the center of the superpixel  $x$  is less than the distance from the point to the center of the superpixel it originally belongs to, then the point belongs to the superpixel  $x$ . Furthermore, the **DIS** matrix and **LABEL** matrix are updated. Finally, the above steps are iterated to obtain the minimum cost function, that is, the sum of the distances from the pixel to the center of its corresponding super pixel.

The main idea of superpixel segmentation based on GMM is to use Gaussian distribution to relate pixels. The main procedure of the algorithm is as follows: let  $I$  represent the input image,  $W$  and  $H$  represent the width and height of the image,  $i \in V \triangleq \{0, 1, \dots, N - 1\}$  ( $N$  is the number of pixels in the image),  $(x_i, y_i)$  represent the position of the  $i$ th pixel, and  $c_i$  represent gray value of the pixel (the color image is the RGB value) of the  $i$ th pixel,  $z_i = (x_i, y_i, c_i)$  represent pixel  $i$ . Let  $v_x$  and  $v_y$  denote the width and height of the superpixels,  $K$  is the number of super pixels. When  $K$  is known,  $v_x$  and  $v_y$  can be obtained as follow.

$$v_x = v_y = \left\lceil \sqrt{\frac{W \cdot H}{K}} \right\rceil. \tag{1}$$

When  $v_x$  and  $v_y$  are known,  $K$  can be obtained from the following formula:

$$n_x = \left\lfloor \frac{W}{v_x} \right\rfloor, n_y = \left\lfloor \frac{H}{v_y} \right\rfloor, K = n_x \cdot n_y. \tag{2}$$

Let  $\theta_k = \{\hat{\mathbf{u}}_k, \hat{\Sigma}_k\}$  denote the parameters of the Gaussian distribution model corresponding to the  $k$ -th superpixel, and  $I_k$  is used to denote the area where the  $k$ th superpixel is distributed (this area is initially limited to an area with width of  $3 v_x$  and height of  $3 v_y$ ), Where  $\hat{\mathbf{u}}_k$  represents the mean value vector and  $\hat{\Sigma}_k$  represents the covariance matrix. Then the Gaussian distribution probability density function corresponding to a superpixel can be expressed by

$$p(\mathbf{z}, \theta) = \frac{1}{(2\pi)^D \sqrt{|\det(\Sigma)|}} \exp\left\{-\frac{1}{2}(\mathbf{z} - \mathbf{u})^T \Sigma^{-1}(\mathbf{z} - \mathbf{u})\right\}. \tag{3}$$

where  $D$  represents the number of elements in the pixel vector  $\mathbf{z}$ . The  $K_i$  is used to represent the label of the pixel in the area where the  $k$ th superpixel is distributed. Let  $L_i$  denote the random variable of the superpixel label of pixel  $i$ , the pixel-related Gaussian mixture model can be expressed by

$$P_i(\mathbf{z}) = \sum_{k \in K_i} P_r(L_i = k) p(\mathbf{z}; \theta_k), \forall i \in V. \tag{4}$$

where  $P_r(L_i = k)$  represents the probability that the superpixel label of pixel  $i$  is  $k$ , which is represented by  $P_i$  and defined as a constant, so (4) can be simplified to:

$$P_i(\mathbf{z}) = P_i \sum_{k \in K_i} P(\mathbf{z}; \theta_k), \forall i \in V. \tag{5}$$

When parameter set  $\theta_k = \{\hat{\mathbf{u}}_k, \hat{\Sigma}_k\}$  is determined, the label  $L_i$  of pixel  $i$  is determined by

$$L_i = \operatorname{argk} \max_{k \in K_i} \frac{p(\mathbf{z}_i; \theta_k)}{\sum_{k \in K_i} p(\mathbf{z}_i; \theta_k)} \tag{6}$$

If labels  $L_i$  of each pixel  $i$  is determined. The parameter set  $\theta$  can be obtained by maximum likelihood estimation:

$$\begin{aligned} f(\theta) &= \sum_{i \in V} \ln p_i(\mathbf{z}_i) \\ &= \sum_{i \in V} \ln P_i + \sum_{i \in V} \ln \sum_{k \in K_i} p(\mathbf{z}_i; \theta_k). \end{aligned} \tag{7}$$

where  $\sum_{i \in V} \ln P_i$  is a constant, and maximizing  $f(\theta)$  is equivalent to maximizing

$$\begin{aligned}
 L(\theta) &= \sum_{i \in V} \ln \sum_{k \in K_i} p(z_i; \theta_k) = \sum_{i \in V} \ln \sum_{k \in K_i} R_{i,k} \frac{p(z_i; \theta_k)}{R_{i,k}} \\
 &\geq \sum_{i \in V} \sum_{k \in K_i} R_{i,k} \ln \frac{p(z_i; \theta_k)}{R_{i,k}} \quad (8)
 \end{aligned}$$

After initializing parameter set  $\theta$ ,  $\mathbf{R}$  and  $\theta$  are iterative updated using the EM algorithm to obtain its best estimate. Until EM algorithm converges, the result of super pixel segmentation is obtained.

We preprocess the high-resolution aerial image by using superpixel segmentation method, and retain the super pixel segmentation results. Let  $I$  denote the high-resolution aerial image,  $W$  and  $H$  represent the width and height of the origin high-resolution aerial image. Let  $(x_i, y_i)$  represent the position of the  $i$ th pixel, where  $x_i \in \{1, 2, \dots, W\}$ ,  $y_i \in \{1, 2, \dots, H\}$ . We use matrix  $\mathbf{SP}$  denote the superpixel label of each pixel obtained by the superpixel segmentation algorithm. Let  $L_i$  represent the superpixel label of pixel  $i$ , as shown in the following formula:

$$L_i = \mathbf{SP}(x_i, y_i) \quad (9)$$

Take the first image patch (starting from the upper left corner of the image) as an example to illustrate our method based on the superpixel to cut high-resolution aerial image into image patches. We initialize the width and height of the image patch as  $w$  and  $h$ , and the starting position of each row and column of the patch is 0. We use superpixel labels to cut the high-resolution aerial image into image patches. We use the vector  $\mathbf{flag1}$  to save the superpixel label of pixels (when  $x_i = w$  in each row of the image), as shown in the following formula:

$$\mathbf{flag1}(i) = \mathbf{SP}(w, i), i \in \{1, 2, \dots, h\} \quad (10)$$

Let  $i \in \{1, 2, \dots, h\}$ , find the pixel  $i$  with the largest  $x_i$  which superpixel label is  $\mathbf{flag1}(i)$  in the  $i$ th row, and record the value of  $x_i$  in the vector  $\mathbf{flag\_x}$ , and save it in  $\mathbf{flag\_w}$ , and record the maximum value of  $x_i$  in  $\mathbf{flag\_w}$  as  $x\_max$ .

We use vector  $\mathbf{flag2}$  to save the superpixel label of each column of the image when  $y_i = h$ , as shown in the following formula:

$$\mathbf{flag2}(j) = \mathbf{SP}(j, h), j \in \{1, 2, \dots, x\_max\} \quad (11)$$

Let  $j \in \{1, 2, \dots, x\_max\}$ , find the pixel  $j$  with the largest  $y_j$  whose superpixel label is  $\mathbf{flag2}(j)$  in the  $j$ th column, and record its  $y_j$  value in the vector  $\mathbf{flag\_y}$ , and save it in  $\mathbf{flag\_h}$ , and record the maximum value of  $y_j$  in  $\mathbf{flag\_y}$  as  $y\_max$ .

At this time, the width and height of image patch become  $x\_max$  and  $y\_max$ . And we record the position of pixels the image patch, which should take from each row and each column of the original high-resolution aerial image. Same as the above steps, the row starting position of the second patch is the value stored in the  $\mathbf{flag\_x}$  vector, and the column starting position is the value stored in  $\mathbf{flag\_y}$ . This is to avoid information redundancy. The width and height of the image patch are initialized as  $w$  and  $h$ , and

the second image patch can be obtained according to the above steps until all the image patches are obtained.

The Fig. 2 is an example of using our method to cut high-resolution aerial image into image patches, in which the resolution of the aerial image is  $1280 \times 659$  and image patches are taken with the width of 640 and height of 659. We only take the first patch as example. In Fig. 2 (a), we directly cut the high-resolution aerial image into image patches without superpixel segmentation. The two cars in the bottom right corner of the patch are cut into two parts. And Fig. 2 (b) is the result of cutting the high-resolution aerial image into image patches based the proposed method. We use the red dots to show the edge positions, which is obtained using superpixel labels to expand the edges of patch. The horizontal coordinate of each row of red dots is the value stored in the vector  $flag_w$ , and the largest value is  $x_{max}$ .



**Fig. 2.** An example of the low-resolution patch. The result of cutting high-resolution aerial image into image patches without superpixel is shown in (a). The result of cutting high-resolution aerial image into image patches with surperpixel is shown in (b). (Color figure online)

### 3.2 Oriented Object Detection Based on YOLOv5

The object detector based on neural network generally consists of the following parts: Input, Backbone, Neck, Prediction. Input is the input terminal, which is generally an image or image batches. Backbone performs feature extraction on the input data. Neck realizes the extraction of multi-scale features. Prediction uses the extracted features to predict the location of the objects and the object category.

The network framework of YOLOv5 is shown in Fig. 3.

YOLOv5 has the advantages of fast detection and high accuracy. However, it is based on horizontal bounding boxes (HBB). We use a combination of CSL and YOLOv5 [14] to realize oriented object detection.

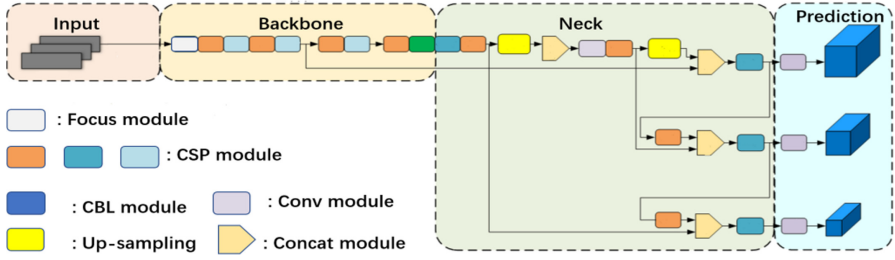


Fig. 3. The network framework of YOLOv5

## 4 Experiments

### 4.1 Datasets

We use the UCAS\_AOD [15] dataset for experiments. UCAS\_AOD is annotated by the Pattern Recognition and Intelligent System Development Laboratory of the University of Chinese Academy of Sciences, and it contains two types of objects and background negative samples. The resolution of aerial images ranges from  $1280 \times 659$  to  $1372 \times 972$ , and the number of samples is given in Table 1. For this dataset, we cut the high-resolution aerial image into 2 image patches in the horizontal direction. All experiments are implemented on a desktop machine equipped with an Intel(R) Core (TM) i5-8600k CPU @ 3.60 GHz and 16.0 GB RAM.

Table 1. UCAS\_AOD dataset

	Plane image	Plane sample	Car image	Car sample	negative
Version1	600	3591	310	4475	492
Version2	400	3891	200	2639	408
Total	1000	7482	510	7114	910

### 4.2 Superpixel Segmentation Comparison Experiments

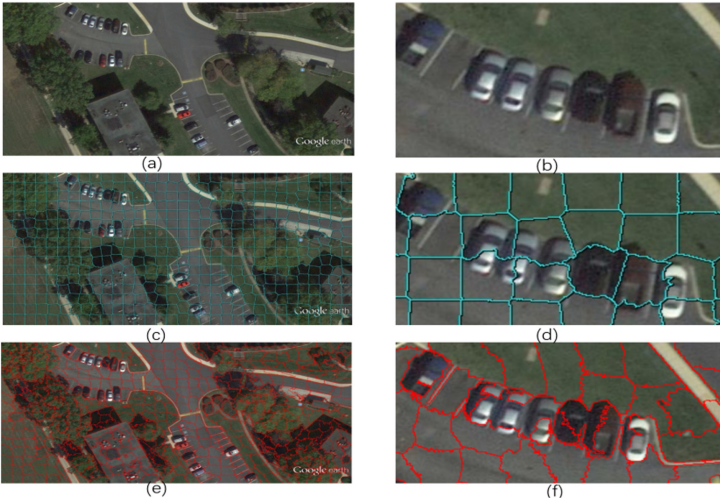
For SLIC and GMM-based superpixel segmentation algorithms, we have done comparative experiments, and the experimental results are shown in Fig. 4. Where, (a) is the original aerial high-resolution image for super pixel segmentation, (b) is a partial enlarged image of aerial image, (c) is the result image of SLIC superpixel segmentation, (d) is a partial enlargement of the result image of SLIC superpixel segmentation, (e) is the result of GMM-based super pixel segmentation, and (f) is a partial enlarged image of GMM-based superpixel segmentation. It can be intuitively observed from the segmentation result image and the partial enlarged image that the superpixel segmentation based on GMM is better than SLIC in preserving the integrity of the object. In addition, the SLIC algorithm took 15.2200 s on the image with the resolution of  $1280 \times 659$ ,



while the GMM-based superpixel segmentation only took 0.67967 s, therefore we chose the GMM-based superpixel segmentation algorithm to pre-process the high-resolution aerial image.

### 4.3 Experimental Results of Cutting High-Resolution Aerial Image

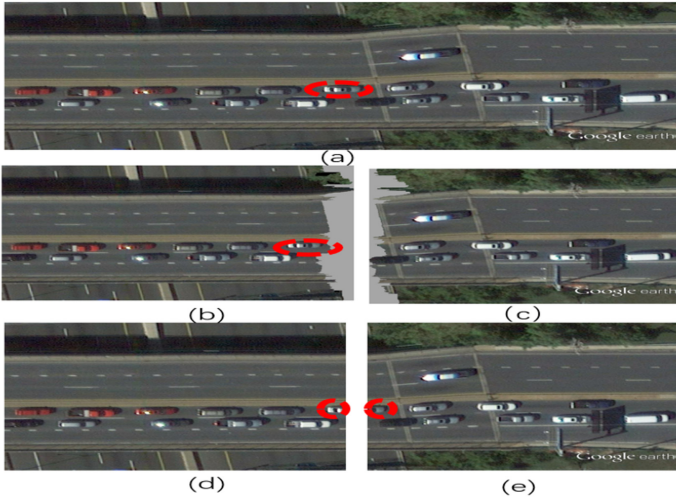
The results of the comparison between cutting the high-resolution aerial image into image patches with superpixel and without superpixel are shown in Fig. 5. Where, (a) is the original high resolution aerial image for cutting into image patches, (b) and (c) are image patches obtained by cutting the high-resolution aerial image into image patches based on superpixels, (d) and (e) are image patches obtained by cutting high-resolution aerial image into image patches without superpixels. It can be clearly seen that when the high-resolution aerial image is directly cut into patches, the car circled in red in the original high-resolution image is cut into two parts, and our method avoids the car from being cut into two parts and retains the integrity of the whole object.



**Fig. 4.** Superpixel segmentation comparison experiments. (a) is the original aerial high-resolution image for super pixel segmentation, (b) is a partial enlarged image of aerial image, (c) is the result image of SLIC super-pixel segmentation, (d) is a partial enlargement of the result image of SLIC super-pixel segmentation, (e) is the result of GMM-based super pixel segmentation, and (f) is a partial enlarged image of GMM-based superpixel segmentation.

### 4.4 Oriented Object Detection

We first cut the UCAS\_AOD dataset into low-resolution image patches, and then randomly divide it into a training set and a testing set at a ratio of 9:1. And then, we use YOLOv5 network combined with CSL to realize oriented object detection. The object detection AP (Average Precision) and mAP (mean Average Precision) of two methods

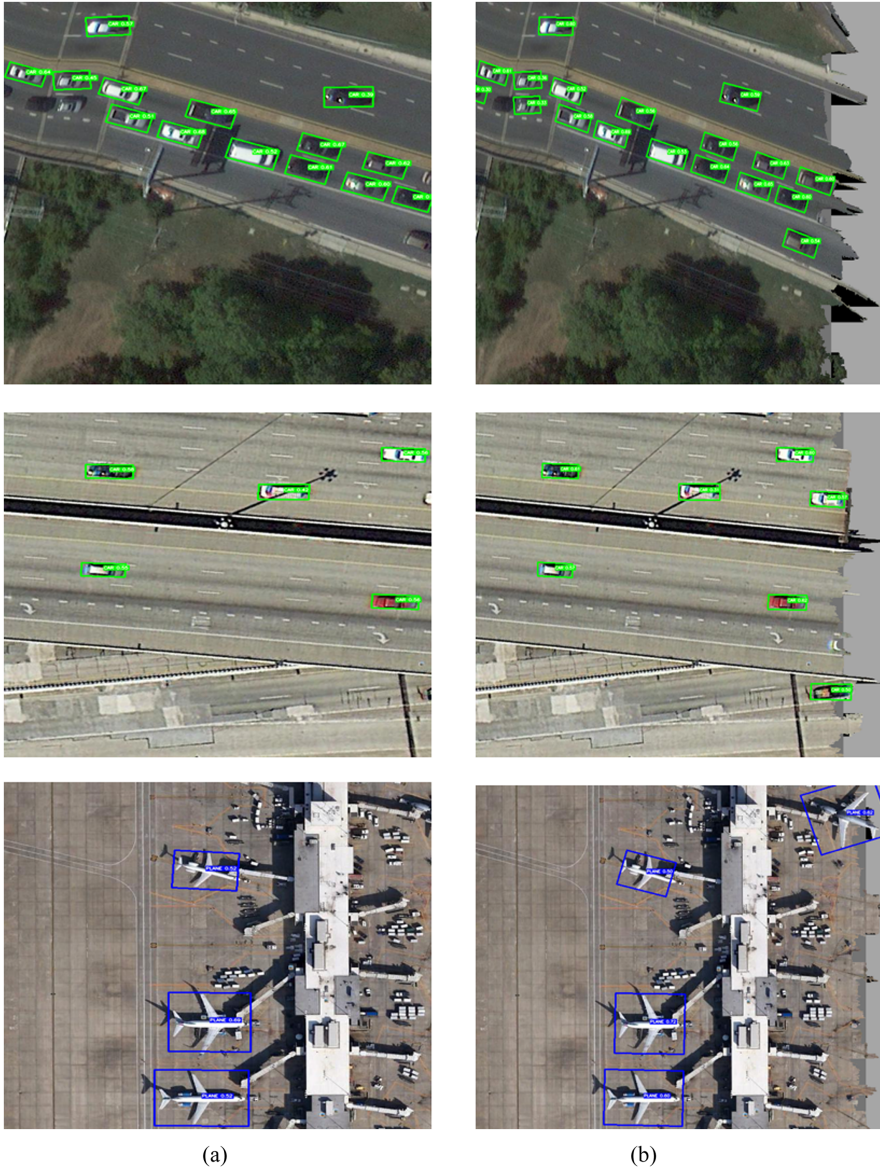


**Fig. 5.** Comparison between using superpixel-based cutting high resolution aerial image into image patches and direct cutting. (a) is the original high resolution aerial image for cutting into image patches, (b) and (c) are image patches obtained by cutting the high-resolution aerial image into image patches based on superpixels, (d) and (e) are image patches obtained by cutting high-resolution aerial image into image patches without superpixels. (Color figure online)

to cut high-resolution aerial image to small image patches are shown in Table 2. As shown in Table 2, our method improved by 0.223% in car category AP and improved by 0.071% in plane category AP compared with cutting high-resolution aerial image into image patches without superpixel. Our method has improved on the mAP by 0.147%. And the oriented object detection results of cutting high-resolution into image patches with superpixel or not are shown in Fig. 6. Figure 6 (a) is the result of detection in patches cutting from high-resolution aerial image without superpixel and Fig. 6 (b) is the result of detection in patches cutting from high-resolution aerial image based on superpixel. As shown in Fig. 6 (a) and Fig. 6 (b), the detection performance at the edge area of cutting has been improved by our method. If cutting the high-resolution aerial image into image patches without superpixel, the objects in the edge area will be cut into multiple parts which will not be detected in the following object detection. Cutting high-resolution image into image patches by our method can preserve the integrity of objects in the edge area that is helpful for the detector to detect the object correctly.

**Table 2.** The object detection mAP of two cutting methods

	AP of CAR	AP of PLANE	mAP
Our method	44.137%	45.047%	44.591%
Without superpixel	43.914%	44.976%	44.444%



**Fig. 6.** Comparison of object detection in patches between using superpixel-based cutting high resolution aerial image into image patches and direct cutting, (a) are the results of object detection in patches from cutting high-resolution aerial image into images patches directly, and (b) are the results of object detection in patches from cutting high-resolution aerial image into images patches based on superpixel.

## 5 Conclusion

In this paper, we propose a new baseline for object detection in high-resolution aerial image. In general, the resolution of remote sensing image is very high. Therefore, if the detector is applied to the original image, it will consume a lot of hardware resources. Cutting the high-resolution aerial image into image patches can be divided into two cases. In the case of no overlapping areas between patches, the object located at the edge area of the patches will be cut into multiple parts, causing the detector to fail to accurately detect these objects. In the case of overlapping areas between patches, this will lead to a lot of information redundancy and consume a lot of resources. Compared to the previous cutting method, our proposed cutting method based on superpixel will not cut a whole object into multiple parts and cause information redundancy, and improves the performance of detector.

**Acknowledgment.** This work is supported by the National Natural Science Foundation of China (No. 61631009, No. 61771220).

## References

1. Yang, X., Yan, J.: Arbitrary-oriented object detection with circular smooth label. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 677–694. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58598-3\\_40](https://doi.org/10.1007/978-3-030-58598-3_40)
2. Yi, J., Wu, P., Liu, B., et al.: Oriented object detection in aerial images with box boundary-aware vectors. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2150–2159 (2021)
3. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.K.: Learning ROI transformer for oriented object detection in aerial images. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 2844–2853 (2019)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580–587 (2014)
5. Redmon, J., Farhadi, A.: YOLOv3: an Incremental Improvement. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–6 (2018)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
7. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6569–6578 (2019)
8. Jiang, Y., et al.: R2CNN: rotational region CNN for orientation robust scene text detection. arXiv preprint. [arXiv:1706.09579](https://arxiv.org/abs/1706.09579) (2017)
9. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings ICCV, pp. 10–17 (2003)
10. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. In: IEEE Trans. Pattern Anal. Mach. Intell. 34, 2274–2282 (2012)

11. Ban, Z., Liu, J. Cao, L.: Superpixel segmentation using Gaussian mixture model. *IEEE Trans. Image Process.* 27(8), 4105–4117 (2018)
12. Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 13961–13970 (2020)
13. Ultralytics.: yolov5. <https://github.com/ultralytics/yolov5>
14. YOLOv5\_DOTA\_OBB. [https://github.com/hukaixuan19970627/YOLOv5\\_DOTA\\_OBB](https://github.com/hukaixuan19970627/YOLOv5_DOTA_OBB)
15. Zhu, H., Chen, X., Dai, W., Fu, K., Ye, Q., Jiao, J.: Orientation robust object detection in aerial images using deep convolutional neural network. In: 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, pp. 3735–3739 (2015)