Object Detection Using Clustering Algorithm Adaptive Searching Regions in Aerial Images

Yi Wang, Youlong Yang(✉), and Xi Zhao

School of Mathematics and Statistics, Xidian University, Xi'an, China
wangyi0102@stu.xidian.edu.cn, ylyang@mail.xidian.edu.cn

Abstract. Aerial images are increasingly used for critical tasks, such as traffic monitoring, pedestrian tracking, and infrastructure inspection. However, aerial images have the following main challenges: 1) small objects with non-uniform distribution; 2) the large difference in object size. In this paper, we propose a new network architecture, Cluster Region Estimation Network (CRENet), to solve these challenges. CRENet uses a clustering algorithm to search cluster regions containing dense objects, which makes the detector focus on these regions to reduce background interference and improve detection efficiency. However, not every cluster region can bring precision gain, so each cluster region difficulty score is calculated to mine the difficult region and eliminate the simple cluster region, which can speed up the detection. Then, a Gaussian scaling function(GSF) is used to scale the difficult cluster region to reduce the difference of object size. Our experiments show that CRENet achieves better performance than previous approaches on the VisDrone dataset. Our best model achieved 4.3% improvement on the VisDrone dataset.

1 Introduction

Equipped with cameras and embedded systems, Unmanned Aerial Vehicles (UAV) are endowed with computer vision ability and widely used for traffic monitoring, pedestrian tracking, and infrastructure inspection. With the rapid development of deep neural networks, the object detection framework based on deep neural networks has gradually become the mainstream technology of object detection. Although correlation detectors (such as R-CNN family [9–11,29], YOLO family [1,26–28], SSD family [7,19], etc.) have achieved good performance in natural images, they cannot achieve satisfactory results in aerial images.

Compared to natural images such as COCO [18], ImageNet [4] and Pascal VOC [6], aerial images have the following features:

(1) Small objects with the non-uniform distribution. Generally, a small object refer to object with the area of less than 32×32 in an image. The main problems of small objects are low resolution and small amount of information, which lead

A. Bartoli and A. Fusiello (Eds.): ECCV 2020 Workshops, LNCS 12538, pp. 651–664, 2020.
https://doi.org/10.1007/978-3-030-66823-5_39

to weak feature expression. The traditional method of processing small objects is to enlarge the image, which will increase the processing time and the memory needed to store large feature maps. Another common method is uniform cropping an image into several regions [8,20] and then detect in each region, which solves the problem of storing a large feature map. However, the uniform cropping ignores the sparsity of objects, and some regions may have few or no objects, which will waste a lot of computing resources. As can be seen from Fig. 1, the object distribution in the aerial image is uneven and the object is highly clustered in a certain region. Therefore, one method to improve detection efficiency is to focus the detector on these regions with a large number of objects.

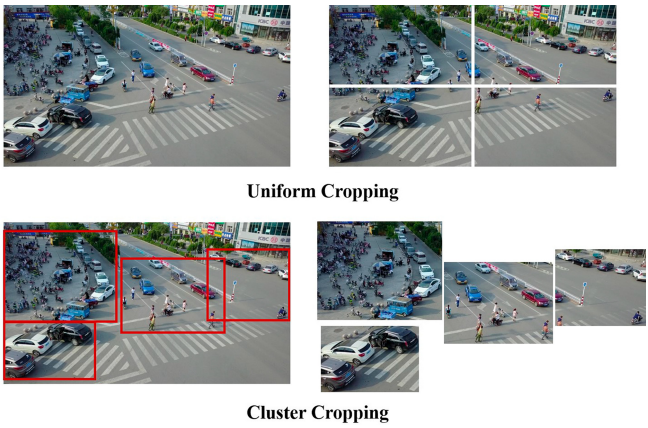

Fig. 1. Visualization of uniform cropping vs cluster cropping. The first row is an example of uniform cropping. The second row is an example of cluster cropping. Compared with uniform cropping, cluster cropping has the following advantages: (1) concentrate on computing resources in cluster regions with a large number of objects, (2) have no background interference.

(2) Diversity of object size. When collecting UAV images, the shooting height varies from tens of meters to hundreds of meters, which leads to huge difference in the size of the same category of objects. It is a big problem for the anchor-based detector to set the size of the anchor. For anchor-free detector, it is difficult to directly regress the width and height of the object. Therefore, it is necessary to reduce the difference in object size among images as much as possible.

To solve the above problems, this paper proposes a coarse-to-fine detection framework CRENet. As shown in Fig. 2, CRENet is composed of three parts: a coarse detection network (CNet), a cropping module, and a fine detection network (FNet). The aerial image is first sent to the coarse detection network CNet to get the initial detection results, which will get a rough distribution of the object. Then the initial detection results are sent to the crop module. The first we mentioned, the cluster region is obtained through a clustering algorithm.

The second step is to calculate the difficult score of each cluster region, it is believed that the cluster region with higher difficult score can bring greater accurate gain to the detector, and the cluster region with a small score can be deleted to improve the detection efficiency of the model. In the third step, we plug the remaining difficult cluster region into a Gaussian scaling function(GSF), which calculates the scaling factor for each of the difficult cluster regions. In particular, we refer to the difficult cluster region after scaling as ROI[1] (region of interest). Finally, ROI is sent to the fine detection network to obtain fine detection results, and the fine detection results are fused with the coarse detection results.

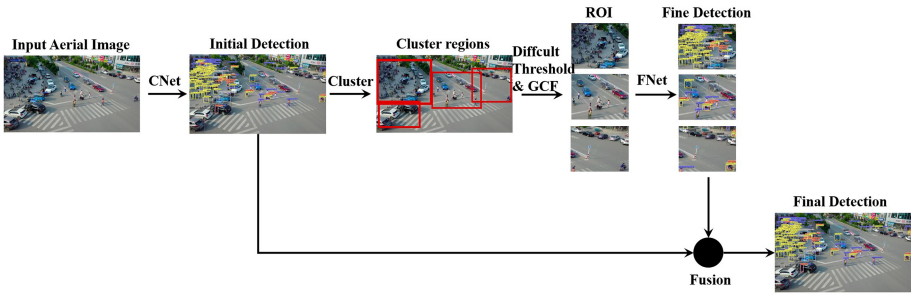

Fig. 2. Overview of CRENet framework. Firstly, CRENet sends the aerial image to the coarse detector CNet to get the initial prediction. Then a clustering algorithm is used to generate cluster regions for the initial prediction. And we mine difficult cluster regions, and then use the Gaussian scaling function(GSF) to scale difficult cluster regions. The difficult cluster regions are sent to the fine detector. Finally, the detection result of the global image is fused with the detection result of ROI to generate the final detection result. See Sect. 3 for more details.

Compared with previous detectors, the proposed CRENet has the following advantages: (1) The computational resources are concentrated on a dense region with a large number of objects, which reduces the computational cost and improves the detection efficiency; (2) Because the cluster region has different size, clustering algorithms are directly used to replace the network to predict the cluster region, which avoids the problem of anchor setting and the cluster region overlapping processing; (3) Calculating the difficult score of each cluster region and eliminating the cluster region that can hardly bring the accuracy gain can speed up the calculation; (4) Using Gaussian scaling function(GSF) can reduce the difference of object size among different images.

To sum up, the contributions of this paper are as follows:

1) A new CRENet detector is proposed that it can adaptively search and sacle regions with dense object for fine detection.

[1] The ROI here is different from Faster RCNN [29]. The ROI of this paper contains not just one object but multiple objects of interest, and it is used to represent the region with dense objects.

2) A Gaussian scaling function(GSF) is proposed to solve the problem of the large size difference of objects in aerial images and improve the detection accuracy.
3) We achieve more advanced performance on representative aerial image dataset VisDrone [45] with fewer images.

The rest of this paper is organized as follows. Section 2 briefly reviews relevant work. In Sect. 3, the proposed approach is described in detail. Section 4 for experimental results and Sect. 5 for the conclusion.

2 Related Work

In this section, we will review the benchmark of anchor-based detectors and anchor-free detectors for natural images and some recent efforts in aerial images. Finally, we focus on searching the region of interest for fine detection.

2.1 Generic Object Detection

At present, the mainstream object detection algorithms are mainly based on deep convolutional neural network, which can be divided into two types: anchor-based detectors and anchor-free detectors. The anchor-based detectors can be further divided into two categories: two-stage detector and single-stage detector. The two-stage detector consists of two steps: proposal region extraction and region classification. The first stage produces proposal regions, containing approximately location information of the object. In the second stage, the proposal regions are classified, and the positions are adjusted. Representatives of two-stage detectors include R-CNN family [9,10,29] and Mask RCNN [11]. The single-stage detector does not need the stage of producing proposal region , but directly generates the classification confidence and position of objects in only one stage. Representatives of single-stage detectors include SSD family [7,19], YOLO family [1,26–28], RetinaNet [17], etc. In general, the two-stage detector has higher accuracy, and the single-stage detector has higher detection speed. However, the anchor-based detector depends on the good prior anchor, and it is difficult to estimate a suitable prior anchor for the large variation in object size. In addition, in order to improve the recall rate, dense anchors are set, and most anchors are negative. This leads to an imbalance between negative anchors and positive anchors, which seriously affects the final detection performance.

The anchor-free detector is a method of object detection based on point estimate. CornerNet [14] uses a convolutional neural network to predict the upper left and lower right corner of an object and predicts embedding Vector of each diagonal corner to determine whether the upper left or lower right corner belong to the same object. CenterNet [45] directly predicts the center of the object and regresses its length and width. ExtremeNet [44] detects vertex, left point, bottom point, rightmost point, and the center point of the object.

It is difficult to find the suitable anchor size for all objects because of the large difference of object size caused by the change of UAV flying height. Therefore,

in this paper, the anchor-free detector CenterNet [45] is used as the detection framework to solve this problem. Experiments also show that the anchor-free detector has better performance on the aerial image datasets.

2.2 Aerial Image Detection

Compared with natural image object detection, aerial image object detection faces more challenges: small objects, objects with uneven distribution, objects with various perspectives and objects vary in size. According to the characteristics of aerial images, people have proposed various solutions. Because the focus of this works is deep learning, this paper only reviews the related work of aerial image detection using deep neural networks. In [24], the tile method was used in the training stage and testing stage to improve the detection ability of small objects. In [35], the free metadata recorded by drones are used to learn Nuisance Disentangled Feature Transform (NDFT) to eliminate the interference of the detector caused by flying altitude change, adverse weather conditions, and other nuisances. Objects in the aerial image can be in any direction and any position. [5,21,38] uses rotate anchors to detect objects in any direction. In [16], the shape mask is allowed to flexibly detect objects in any direction without any pre-defined rotate anchors. In [34], the researcher studied the scale variation of aerial image object detection and proposed a Receptive Field Expansion Block (RFEB) to increase the receptive field size for high-level semantic features and a Spatial-Refinement Module (SRM) to repair the spatial details. In [25], a multi-task object detection and segmentation model is proposed. The segmentation map is used as the weight of self-attention mechanism to weight the feature map of object detection, which reduces the signal of non-correlated regions.

2.3 Region Search in Detection

In object detection, searching the region of interest for fine detection is usually used to detect small objects. The work of [20] proposes an adaptive detection strategy, which can continuously subdivide the regions that may contain small objects and spend computing resources in the regions that contain sparse small objects. The method in [31,37], the clustering algorithm was used to get ROI's ground truth on the original datasets, then a special CNN was used to predict ROI, and finally, ROI was sent to the fine detector. [42], using sliding window method on the feature map, then calculating the difficulty score for each window, and send the difficulty region to the fine detector. [8,32,33] solved the problem of small object detection in large images, and used a reinforcement learning method to find ROI for fine detection. [15] proposed an aerial image object detection network based on a density map. According to the density map, we can get a rough distribution of objects to search for the ROI.

Among the methods reviewed above, some use the network to predict the ROI, some use fixed windows to slide on the feature map to search the ROI, and some directly uniform crop the original image to get the ROI. Due to the different shapes and sizes of ROI, it is difficult to set the size of an anchor or regress the

width and height of ROI by the network. The size of ROI obtained by using the fixed window sliding method and the uniform crop of the original image is fixed, which is difficult to adapt to the real ROI. Therefore, this paper sends the images to the coarse detection network to get the approximate distribution of objects, and then uses the clustering algorithm to adaptively get the ROI. Through the clustering algorithm, the ROI of various sizes can be obtained, which is more in line with the actual situation.

3 Methodology

As shown in Fig. 2, detection of an aerial image can be divided into three stages: the difficult cluster region extraction, fine detection of the ROI, and fusion of the detection results. In the first stage, aerial images are first sent into CNet to obtain an initial prediction. Then the cluster region is obtained by mean shift [39] for initial detection. Besides, the difficulty score of each cluster region is calculated, and the region with a higher score is regarded as a difficult cluster region. In the second stage, firstly, we use the Gaussian scaling function(GSF) to scale the difficult cluster region, so as to reduce the scale difference of objects. The ROI, scaled difficult cluster region, is then finely detected using the FNet. Finally, the third stage fuses the detection results of each ROI and global image with soft-NMS [2].

3.1 Difficult Cluster Region Extraction

Difficult cluster region extraction consists of three steps: Firstly, aerial images are fed into the trained CNet to obtain coarse detection results of objects. Then the results are used to obtain a cluster region. Finally, the difficulty score for each cluster region is calculated, and the non-difficult region is removed to speed up the detection.

In previous work, [37] proposed to use the clustering algorithm to generate the ground truth of the cluster region for each image and then trained a detector to predict the cluster region. However, there is a large overlap between the cluster regions predicted by the network. It is necessary to use the Iterative Cluster Merging (ICM) for the cluster region, and the number of the cluster region obtained is fixed. Especially, in aerial images, due to different camera angles, shooting time, and other reasons, the number of cluster regions may be different. Therefore, a fixed number of cluster regions is not suitable for all cases. Another problem is that the cluster regions vary in shape. It is difficult to manually set the anchor size in Faster-RCNN [29], and it is also difficult to regress anchor. In this paper, We use the clustering algorithm instead of the network to search the clustering region and avoid the problem of anchor setting. Specifically, aerial images are sent into the CNet, so as to obtain the initial prediction results of the object. The cluster region is obtained by using mean shift [39] from the initial prediction. Because an object can only belong

to one region, the overlap between cluster regions is very small. Unlike [37], our algorithm is an unsupervised algorithm, whereas [37] is a supervised algorithm.

The aerial image is acquired at high altitude, so the background is complex and the objects is small. As can be seen from Fig. 1, the objects are usually gathered together. We can use a clustering algorithm to get the cluster region, and then crop and enlarge it for fine detection, which can not only solve the problem of small object detection but also reduce the interference of background. The mean shift [39] is a dense-based clustering algorithm, which assumes that the data of different clusters belong to different probability density distributions. By inputting the initial detection into the mean shift [39] algorithm, the cluster region of the image can be obtained adaptively.

It is worth noting that not every cluster region can get accurate gain. In order to improve the detection efficiency of the detector, it is necessary to eliminate the regions which cannot bring accurate gain or small accurate gain. This paper assumes that the denser the objects are in the cluster region, the lower the average confidence score is. The cluster region with denser objects or low average confidence score can obtain greater the accuracy gain from the fine detection in this region. According to this assumption, similar to [42], the initial prediction results of aerial images are used to calculate a score for each cluster region, and the regions whose score is greater than the difficulty threshold are retained.

$$M = \frac{\sum_{i=1}^{N} score_i}{N} \tag{1}$$

$$S = \frac{N^2}{A \times M} \tag{2}$$

Using Eqs. (1) and (2) to calculate the difficulty score for region p, where N is the number of the predicted boxes in p, M is the average of the confidence scores of all the prediction boxes by the coarse detector for region p. It is believed that the smaller the value of M is, the greater the accuracy gain will be. Therefore, we place it in the denominator. Where A is the area of p, S is the final score of p. $\frac{N^2}{A}$ represents the density of region p. It is believed that the denser the region is, the more accuracy gain it can bring. Because in places where objects are dense, they are usually accompanied by occlusion between objects. When the occlusion is serious, the detector will miss detection. Therefore, for dense regions, enlarging it can effectively reduce the missed detection.

3.2 Fine Detection on Region of Interest

After obtaining the difficult cluster region, a special detector FNet is utilized to perform fine detection on these regions. But the difficult cluster region has different shapes and different sizes of objects, it will bring a problem that it is difficult to regress the width and height of objects. Different from the existing approaches, [8,13,20] that directly send these regions to fine detection, inspired by [41], this paper proposes a Gaussian scaling function(GSF) to reduce the

size difference of objects. [41] uses the transformation function Scale Match to scale the extra dataset, so that there is little difference in object size between the targeted dataset and the extra dataset. In [41], MS COCO [18] is used as an extra dataset to pre-train the detector and improve its performance on the targeted dataset. Unlike [41], we do not scale the extra dataset, but the targeted dataset. We use Gaussian scale functions(GSF) to scale difficult cluster regions. First, select a mean value that is suitable for the receptive field of the backbone. Then, we select the standard deviation based on the three Sigma rule of thumb. The Gaussian scale functions(GSF) is made up of the mean value and standard deviation. The Gaussian scale function (GSF) can be used to shrink large objects and enlarge small objects. The implementation process is shown in Algorithm 1.

Algorithm 1: Gaussian Scaling Function

Input: Initial difficult cluster regions $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^{N_\mathcal{R}}$
Output: ROI $\mathcal{R}^{'} = \{\mathcal{R}_i\}_{i=1}^{N_\mathcal{R}}$
Note:R_i is i-th region in $\mathcal{R}$, B_i represents all predicted boxes set in R_i. *Scaleregion* is a function to scale region and predicted boxes with a given scale factor. *Uniform Crop* is a function that uniform crop an region into two regions; *Padding* is a function that uses the scaling factor to fill in the region from the original image.

$\mathcal{R}^{'} \longleftarrow \emptyset$
for (R_i, B_i) *in* $\mathcal{R}$ **do**
  sample $k \sim gauss(\mu, \sigma)$;
  $s \longleftarrow Mean\,(B_i)$;
  $c \longleftarrow s/k$;
  $R_i, B_i \longleftarrow ScaleRegion\,(R_i, B_i, c)$;
  if *the size of* R_i *is to large* **then**
    $\mathcal{R}^{'} \longleftarrow \mathcal{R}^{'} \cup UniformCrop\,(R_i)$;
    else if *the size of* R_i *is to small* **then**
      $\mathcal{R}^{'} \longleftarrow \mathcal{R}^{'} \cup Padding\,(R_i, c)$;
      else
        $\mathcal{R}^{'} \longleftarrow \mathcal{R}^{'} \cup R_i$;

$$AS\,(G_{ij}) = \sqrt{w_{ij} \times h_{ij}} \tag{3}$$

In Algorithm 1, Eq. (3) is used to calculate the absolute size of each object for each difficult cluster region. The ratio of the value randomly sampled from a Gaussian function to the average absolute size of all objects in the region is used as the scale factor for scaling the difficult cluster region. If the size of the difficult cluster region after scaling is less than a certain range, the padding function is used to pad the region proportionally. Otherwise, the uniform crop function is used to divide it into two equal regions.

After scaling the difficult cluster region, we get the ROI. Then, the detection network (FNet) performs fine object detection. The architecture of the FNet can be any state-of-the-art detectors. The backbone of the detector can be any standard backbone networks, e.g., VGG [30], ResNet [12], Hourglass-104 [23].

3.3 Final Detection with Local-Global Fusion

NMS [22] is a post-processing step commonly used in object detection, which is used to remove the duplicate detection box to reduce false detection. When there are multiple prediction boxes on the same object, NMS will eliminate the remaining prediction boxes whose IOU is greater than the threshold value with the prediction box with the maximum confidence score. It can be seen that NMS is too strict and soft-NMS [2] replaces the original score with a slightly lower score instead of zero. The final detection of an aerial image is obtained by fusing the detection results of ROI and global detection results of the whole image with the soft-NMS [2] post-processing.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. The VisDrone [45] dataset was collected by using drones from 14 different cities in China under different weather and different lighting. The objects in this dataset are mostly small, and the objects are often clustered together. It contains a total of 10,209 images, including 6,471 training images, 548 validation images, 1,610 test-dev images, and 1,580 test-challenge images. Except for the test challenges, all other annotations are publicly available. The dataset contains a total of 10 categories, and its image resolution is about 2000*1500 pixels. In order to make a fair comparison with existing works [15,37], we evaluate the detection performance in the validation set.

Evaluation Metric. We use the same evaluation protocol as proposed in MS COCO [18] to evaluate our method. Six evaluation metrics of AP, AP_{50}, AP_{75}, AP_{small}, AP_{medium}, and AP_{large} are reported. The AP is the average precision of all categories on 10 IOU thresholds, ranging from 0.50 to 0.95 with a step size of 0.05. AP_{50} is the average precision of ten categories when the IOU threshold is set to 0.5, and the IOU threshold of AP_{75} is set to 0.75. The AP_{small} means that the AP for objects with area less than 32×32. The AP_{medium} means that the AP for objects with area less than 96×96. The AP_{large} means that the AP for objects with area greater than 96×96. The number of ROI will affect inference time. In the following experiments, we use $\#img$ to record the total number of images, send to the detector, including both original images and cropped ROI.

4.2 Implementation Details

We implemented the proposed CRENet on pytorch 1.4.0. Using an RTX 2080Ti GPU to train and test the model. In common with training many deep CNNs, we use data augmnetation. Specifically we use horizontal flipping. In this article, CNet and FNet use the same detector, which is CenterNet [43] with the backbone network Hourglass-104 [23], and different detectors may be selected. After obtaining the detection result of the aerial image through CNet, use mean shift [39] to get the preliminary cluster region. The region with an area of fewer than 10000 pixels, the aspect ratio of more than 4 or less than 0.25, and the number of objects less than 3 are excluded. The cluster regions with difficulty scores less than 0.01 will be eliminated. The image input resolution for CNet and FNet both are 1024*1024. We train the baseline detector for 140 epoch with Adam, and the initial learning rate was 2.5×10^{-4}.

4.3 Quantitative Result

CenterNet [43] with Hourglass-104 [23] is chosen as the baseline model. Table 1 shows that our approach with baselines, UC, RC, ClusDet [37], and DMNet [15]. We achieve the best performance using fewer images than other methods, and even the small backbone network DLA-34 [40] with deformable convolution layers [3], modified by CenterNet [43], achieves better performance than ClusDet [37] and DMNet [15] both use Faster R-CNN [29] with ResNeXt [36]. We find that the AP value of RC was lower than the baseline, possibly because RC truncates the object when cropping the image. Experiments show that AP_{small} and AP_{medium} have more improvements, indicating that the method, the clustering algorithm adaptive cropping regions we proposed, is of great help to the detection of small and medium objects.

Table 1. The quantitative results on the validation set of VisDrone. $\#img$ is the number of images that send to the detector. UC refers to the uniform cropping of the aerial image into four parts, while RC refers to the random cropping of four 1024*1024 regions from the image each time.

Method	Backbone	$\#img$	AP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AP_{large}	s/img
Baseline	Hourglass-104	548	30.6	53.1	29.9	21.6	44.0	57.6	0.194
ClusDet [37]	ResNeXt 101	2716	28.4	53.2	26.4	19.1	40.8	54.4	0.773
DMNet [15]	ResNeXt 101	2736	29.4	49.3	30.6	21.6	41.0	56.7	–
Baseline+RC(4)	Hourglass-104	2740	30.2	52.3	29.5	21.6	42.7	56.6	1.046
Baseline+UC (2×2)	Hourglass-104	2740	32.3	**55.6**	32.8	24.3	44.1	55.2	1.040
CRENet	DLA-34	2413	30.3	53.7	29.2	21.6	41.9	50.6	0.561
CRENet	Hourglass-104	2337	**33.7**	54.3	**33.5**	**25.6**	**45.3**	**58.7**	0.901

4.4 Ablation Study

In this experiment, we show how the three components of the framework, clustering algorithm, difficulty threshold, and Gaussian scaling function(GSF), affect the final performance. We consider five cases: (a) Baseline: we use CenterNet [43] with hourglass-104 [23] as the baseline model; (b) CRENet w/o difficult threshold and GSF: a clustering algorithm is added to the baseline model to search cluster regions, but the difficulty threshold and the Gaussian scaling function(GSF) are not used to process regions; (c) CRENet w/o difficult threshold: clustering algorithm produces regions that are not filtered using difficulty threshold; (d) CRENet w/o GSF: difficult regions are directly sent to the fine detector without Gaussian scaling function(GSF); (e) CRENet: The complete implementation of our method. As can be seen from Table 2, the performance improvement of searching regions using only the clustering algorithm is limited. Therefore, it is necessary to use a Gaussian scaling function(GSF) to scale regions. Using the difficulty threshold to filter the cluster regions, 767 regions were eliminated without lowering the AP. It shows that the difficulty threshold can effectively eliminate the region which can hardly bring the precision gain, thus speeding up the detection speed. The above experiments show that the two components of our proposed, clustering algorithm, and Gaussian scaling function(GSF), are very important for the full improvement of detection performance. And the component of difficulty threshold is crucial to achieve a high inference speed.

Table 2. Ablation study of detection result on validation set of VisDrone.

Method	$\#img$	AP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AP_{large}
Baseline	548	30.6	53.1	29.9	21.6	44.0	57.6
CRENet w/o difficult threshold and GSF	2807	32.0	54.8	31.8	23.8	43.8	57.8
CRENet w/o difficult threshold	2836	33.7	**57.6**	33.3	25.5	**45.5**	**60.0**
CRENet w/o GSF	2040	31.9	54.7	31.8	23.7	43.7	57.2
CRENet	2337	**33.7**	54.3	**33.5**	**25.6**	45.3	58.7

5 Conclusions

In this paper, we propose a new method CRENet for object detection in aerial images. CRENet using the clustering algorithm can adaptively obtain cluster regions. Then, the difficulty threshold can be used to eliminate the cluster region that can not bring precision gain, and speed up detection. We also propose that the Gaussian scaling function(GSF) can scale the difficult cluster region to reduce the scale difference between objects. Experiments show that CRENet

performs well for small and medium objects in dense scenarios. A large number of experiments have demonstrated that CRENet achieves better performance over the VisDrone [45] dataset.

Acknowledgements. This work was supported by National Natural Science Foundation of China grant 61573266.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection (2020)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-NMS - improving object detection with one line of code. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
3. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
4. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
5. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning RoI transformer for detecting oriented objects in aerial images (2018)
6. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
7. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD : deconvolutional single shot detector (2017)
8. Gao, M., Yu, R., Li, A., Morariu, V.I., Davis, L.S.: Dynamic zoom-in network for fast object detection in large images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
9. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2015
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
11. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
13. LaLonde, R., Zhang, D., Shah, M.: ClusterNet: detecting small objects in large scenes by exploiting spatio-temporal information. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
14. Law, H., Deng, J.: CornerNet: detecting objects as paired keypoints. In: The European Conference on Computer Vision (ECCV), September 2018
15. Li, C., Yang, T., Zhu, S., Chen, C., Guan, S.: Density map guided object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020

16. Li, Y., Huang, Q., Pei, X., Jiao, L., Shang, R.: RADet: refine feature pyramid network and multi-layer attention network for arbitrary-oriented object detection of remote sensing images. Remote Sens. **12**(3) (2020). https://doi.org/10.3390/rs12030389
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
18. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
19. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Lu, Y., Javidi, T., Lazebnik, S.: Adaptive object detection using adjacency and zoom prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
21. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans. Multimed. **20**(11), 3111–3122 (2018)
22. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 850–855 (2006)
23. Newell, A., Yang, K., Deng, J.: Stacked Hourglass Networks for Human Pose Estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
24. Unel, F.O., Ozkalayci, B.O., Cigla, C.: The power of tiling for small object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019
25. Perreault, H., Bilodeau, G., Saunier, N., Héritier, M.: SpotNet: self-attention multi-task network for object detection. In: 2020 17th Conference on Computer and Robot Vision (CRV), pp. 230–237 (2020)
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
27. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
28. Redmon, J., Farhadi, A.: YOLOV3: an incremental improvement (2018)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015). http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
31. Tang, Z., Liu, X., Shen, G., Yang, B.: PENet: object detection using points estimation in aerial images (2020)
32. Uzkent, B., Ermon, S.: Learning when and where to zoom with deep reinforcement learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020

33. Uzkent, B., Yeh, C., Ermon, S.: Efficient object detection in large images using deep reinforcement learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2020
34. Wang, H., et al.: Spatial attention for multi-scale feature refinement for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, October 2019
35. Wu, Z., Suresh, K., Narayanan, P., Xu, H., Kwon, H., Wang, Z.: Delving into robust object detection from unmanned aerial vehicles: a deep nuisance disentanglement approach. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019
36. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
37. Yang, F., Fan, H., Chu, P., Blasch, E., Ling, H.: Clustered object detection in aerial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019
38. Yang, X., Liu, Q., Yan, J., Li, A., Zhang, Z., Yu, G.: R3Det: refined single-stage detector with feature refinement for rotating object (2019)
39. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)
40. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
41. Yu, X., Gong, Y., Jiang, N., Ye, Q., Han, Z.: Scale match for tiny person detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2020
42. Zhang, J., Huang, J., Chen, X., Zhang, D.: How to fully exploit the abilities of aerial image detectors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, October 2019
43. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points (2019)
44. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
45. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q.: Vision meets drones: a challenge (2018)