An Elliptic Centerness for Object Instance Segmentation in Aerial Images

Yixin Luo,1 Jiaming Han,1 Zhou Liu,2 Mi Wang,1 and Gui-Song Xia1,2

1LIESMARS, Wuhan University, Wuhan, China
2School of Computer Science, Wuhan University, Wuhan, China

Correspondence should be addressed to Zhou Liu; liuzhou@whu.edu.cn

Received 15 January 2022; Accepted 15 May 2022; Published 1 June 2022

Copyright © 2022 Yixin Luo et al. Exclusive Licensee Aerospace Information Research Institute, Chinese Academy of Sciences. Distributed under a Creative Commons Attribution License (CC BY 4.0).

Instance segmentation in aerial images is an important and challenging task. Most of the existing methods have adapted instance segmentation algorithms developed for natural images to aerial images. However, these methods easily suffer from performance degradation in aerial images, due to the scale variations, large aspect ratios, and arbitrary orientations of instances caused by the bird’s-eye view of aerial images. To address this issue, we propose an elliptic centerness (EC) for instance segmentation in aerial images, which can assign the proper centerness values to the intricate aerial instances and thus mitigate the performance degradation. Specifically, we introduce ellipses to fit the various contours of aerial instances and measure these fitted ellipses by two-dimensional anisotropic Gaussian distribution. Armed with EC, we develop a one-stage aerial instance segmentation network. Extensive experiments on a commonly used dataset, the instance segmentation in aerial images dataset (iSAID), demonstrate that our proposed method can achieve a remarkable performance of instance segmentation while introducing negligible computational cost.

1. Introduction

Instance segmentation is aimed at predicting both the location and semantic mask of each object instance in an image. Therefore, an intuitive approach to instance segmentation is to detect the bounding boxes of object instances and then perform semantic segmentation in the area of each box. The conventional two-stage instance segmentation methods adopt this detect-then-segment pipeline [1–14]. Mask R-CNN [3] is a state-of-the-art method, which extends Faster R-CNN [15] by adding a mask prediction branch. Based on Mask R-CNN, Mask Scoring R-CNN [6] predicts the IoU between mask and ground-truth and further rescres the confidence of mask through the added mask-IoU branch, acquiring better segmentation results. PANet [4] pays more attention to the process of feature propagation. It proposes bottom-up path aggregation and adaptive feature pooling to merge the features from all levels and finally boost up the performance of instance segmentation. HTC [5] is a cascade architecture for instance segmentation. Different from Cascade R-CNN [16], HTC integrates features from each stage to add complementary information and finally get better mask predictions. PointRend [7] refines the segmentation of objects in a rendering way that is similar to classical computer graphics methods. Methods above all depend on anchor-based detectors. There are also some two-stage methods built upon anchor-free detectors. Extending the detector FCOS [17] by adding a mask branch, CenterMask [8] improves the backbone network [18] and uses Spatial Attention-Guided to focus on important features during the segmentation process. DeepSnake [10] introduces circular convolution, which finishes segmentation by transforming the box detected by CenterNet [19] into a polygon. As for two-stage methods in the aerial scene, most currently extensive works [20–26] are based on Mask R-CNN [3]. Su et al. [20] propose the Precise Roi-Pooling to replace the Roi-Align in Mask R-CNN and further avoids the degeneration of a precision result from the quantization of coordinate. To learn multiscale context information, Feng et al. [21] embed the local context module into the mask branch of [3]. There are also a few works that depend on the rotating object detector, which is one of the characteristics of aerial images. Recently, the rotated object detection task has gained much attention in an aerial scene [27–34], as the oriented bounding boxes enclose the objects.
of aerial images better. For instance, to save the loss caused by extracting features of the neighboring objects, ISOP [22, 23] follow [29] and predict the mask result on the oriented proposal rather than the horizontal one. Even though these two-stage methods can achieve competitive performance, they are usually time-consuming due to the complicated network architectures.

To accelerate the instance segmentation model, one-stage instance segmentation methods [35–49] simplify the pipeline of two-stage methods and reduce the cost of computation. Some one-stage approaches achieve instance segmentation from semantic segmentation results by grouping the pixels that belong to the same object. Using instance-sensitive score maps for generating proposals, InstanceFCN [35] first produces the score maps, then generates object instances by an assembling module. YOLACT [41] linearly combines the proposed prototype masks according to the predicted coefficients and then crops with a predicted bounding box. Some one-stage methods learn to generate the feature point or contour of instances. ExtremeNet [42], with the heavy backbone of HourGlass [50], detects 8 extreme points to generate an octagon, which is a relatively rough mask result. For a faster speed, ESESeg [43] directly regresses the coordinates of contour points through Chebyshev polynomial fitting. Curve-GCN [38] and Point-Set Anchors [46] learn to generate the contour of an instance by the regression way. Poly-YOLO [47] extends YOLOv3 [51] to be able to instance segment in the distance and angle regression way. Most recently, PolarMask [49] utilizes polar masks to represent the contour of instances. It predicts multi-orientation distances from the central area to the contour of the instance and finally decodes the set of distances to the polar mask. As for one-stage aerial instance segmentation, it is not as luxuriant as the two-stage ones. Most of the existing one-stage aerial instance segmentation methods [52–54] group the pixels that belong to the same object on the semantic segmentation result. Audebert et al. [52] utilize a classification network to achieve object-wise classification on the semantic segmentation result. Mou and Zhu [53] learn both semantic segmentation and semantic boundary simultaneously to finish the instance segmentation of the vehicle. Different from the one-stage aerial methods above, Huang et al. [54] follow PolarMask [49] and propose the Polar Template Mask to fit the ship instance in aerial better.

Although these one-stage methods are usually faster than two-stage methods, there is still a performance gap between one-stage and two-stage methods [41, 49]. It is observed that low-quality locations (i.e., locations far away from the centroid of an object instance) often produce low-quality predictions [17], and thus, these low-quality locations result in performance degradation. In order to suppress low-quality locations, centerness [17, 49] is introduced to estimate the quality of locations, which is a branch in parallel with the classification branch. The value of centerness predicted by the network is in the range of [0, 1], and the higher centerness value will be assigned to the location that is closer to the centroid of the instance. Then the predicted centerness value is multiplied by the classification score as the final score, thus reducing the effect of low-quality locations and improving the performance remarkably [17, 49].

However, due to the fact that scale variations and aspect ratios of object instances in aerial images are often larger than those in natural images, most existing centerness methods [17, 49] developed for natural images are not effective in aerial images. Examples are shown in Figure 1. Polar centerness (PC) is a polar representation-based centerness method proposed by the most recent method PolarMask [49], which can achieve state-of-the-art performance. If a location is at the centroid of an object instance, the centerness value of this location should be 1 or approach 1. But the centerness predicted by PC is close to 0 even when the location is at the centroid of instance, as shown in Figure 1.

To address this issue, we propose a novel centerness for object instance segmentation in aerial images, termed elliptic centerness (EC). To be concrete, we introduce ellipses to fit the complex contours of aerial instances. Then we estimate the quality of locations inside ellipses by using two-dimensional anisotropic Gaussian distribution. Considering the information of the whole contour, EC can assign more proper centerness values to aerial objects. As shown in Figure 1, the centerness values at the centroid of objects are 1 strictly, which are not affected by large variations of scale or aspect ratios. Based on EC, we develop a one-stage instance segmentation network for aerial images. Experimental results show that our proposed EC achieves a remarkable performance of instance segmentation in a large-scale aerial image dataset, iSAID [55]. Generally, our contributions are summarized as follows:

(i) We propose an elliptic centerness for aerial instance segmentation, which is a single-layer branch to estimate the quality of locations. Our proposed EC can estimate the appropriate centerness values for the intricate aerial object instance and thus improve the instance segmentation performance

(ii) Experimental results show that our EC outperforms PC by a large margin (8% mAP) of 15 categories in total on iSAID. Furthermore, extensive experiments with the trade-off between accuracy and speed demonstrate that our method has a competitive performance on accuracy and a faster inference speed than state-of-the-arts

The remainder of this paper is organized as follows. In Section 2, we describe in detail our proposed method. Then we show our experimental results in Section 3. We briefly discuss the limitation in Section 4. Finally, a conclusion is given in Section 5.

2. Materials and Methods

2.1. Dataset. iSAID [55] is a large-scale aerial image dataset for instance segmentation which has the same raw images as the DOTA [27] dataset. It contains 2806 images with the size ranging from 800 × 800 to 4000 × 4000, and 655451 instances of 15 common object categories include the following: plane (PL), baseball diamond (BD), bridge
Centerness is firstly proposed in object detection [17], which is designed for estimating the location quality of the anchor point by measuring the distance between the sample point and the box center. As it is defined on the bounding box, we term it BC (box centerness) for short. The BC of one location can be calculated by

\[
BC = \frac{\min(I^*, r^*)}{\max(I^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)},
\]

where \(I^*, r^*, t^*,\) and \(b^*\) represent the distance from the location to the bounding box of 4 directions.

Polar centerness (PC) is one variant of the centerness proposed in [49] for instance segmentation. Since a polar mask \(\mathcal{P}\) of an anchor point \(p\) can be viewed as \(N\) contour points \(\{p_n\}_{n=1}^N\), the PC of \(p\) can be calculated by

\[
PC(p) = \frac{\min_{p_n \in \mathcal{P}} \|p - p_n\|^2_2}{\max_{p_n \in \mathcal{P}} \|p - p_n\|^2_2}
\]

where \(\|p - p_n\|^2_2\) means the distance between point \(p\) and point \(p_n\). In other words, the polar centerness value of an anchor point \(p\) is equal to the ratio of minimum distance and maximum distance from \(p\) to the contour of the polar mask \(\mathcal{P}\).

From the definition, both BC and PC lie in \([0, 1]\). They estimate the quality of localization and consequently suppress the noncentral areas. Specifically, they affect models in two aspects. (1) In the training phase, centerness weights the loss of each positive anchor point on the regression branch according to its value. (2) In the testing phase, centerness is multiplied by the corresponding classification score, and the refined score is subsequently used to sort proposals in postprocess like the nonmaximum suppression (NMS).

However, the two centernesses above have shortcomings individually. As the definition of BC (1), it is designed in object detection tasks and only considers the bounding box without the label of the mask. Moreover, it neglects the multi-orientation of aerial instances.

In Equation (2), PC is defined on the contour of the mask, and it takes the centroid into account. But PC is incompatible with aerial objects and is not able to assign proper values to aerial instances. As shown in Figure 1, we use the PC value at centroid for example. There are mainly two kinds of circumstances: (1) In Equation (2), instance (a) with an extremely large aspect ratio (e.g., bridge, large vehicle) and instance (b) with complex contour (e.g., plane) tend to have lower centerness than round or squared instance (e.g., roundabout, storage tank). (2) Appearing as only a few pixels in aerial images, small instance (c) (e.g., small vehicle) has too few contour points to generate a polar mask target at all orientations, resulting in generating a pretty low centerness target. Because its minimal distance target \(d_{min}\) is close to 0. This condition is also faced when the anchor point is near the contour. In general, PC provides aerial instances with lower centerness targets, which is harmful to the performance of instance segmentation.

Figure 1: Illustrative examples of instances with their PC and our proposed EC values at the location of the centroid. \(d_{max}\) and \(d_{min}\) denote the maximum and the minimum distance from the centroid to the contour. It is noted that \(d_{min}\) of (c) is close to 0, so it is not drawn in the figure. As the centroid is the most central location of an instance, its centerness value is supposed to be the theoretical largest (i.e., 1.0). (a) Instance with a large aspect ratio. (b) Instance with complex contour. (c) Instance with too small scale. Our EC is capable of assigning appropriate values for these instances, but PC fails.
2.3. Elliptic Centerness. To estimate the proper centerness value for aerial instances, we propose a new centerness, named elliptic centerness (EC).

Given a polar mask $\mathcal{P}$ with its $N$ points $\{p_i\}_{n=1}^N$ on contour, the elliptic centerness of a pixel $p = (x, y) \in \mathbb{R}^2$ is measured by a two-dimensional anisotropic Gaussian score:

$$EC(p) = e^{-\alpha^2(p-p_C)(p-p_C)^T},$$

where $\alpha$ is a hyperparameter affects the value of $EC$, $p_C = (x_c, y_c) = 1/K(\sum_{k=1}^K x_k, \sum_{k=1}^K y_k)$ is the centroid, and $C$ is the normalized inertial matrix of the polar mask $\mathcal{P} = \{(x_n, y_n)\}_{n=1}^N$. More precisely, the matrix $C$ is calculated as

$$C = \begin{pmatrix} \eta_{11} & \eta_{10} \\ \eta_{02} & \eta_{11} \end{pmatrix},$$

where $\eta_{pq}$ is the normalized $(p + q)$-order central moments of the polar mask $\mathcal{P}$:

$$\eta_{pq}(\mathcal{P}) = \frac{1}{N} \sum_{(x, y) \in \mathcal{P}} (x - x_c)^p (y - y_c)^q.$$

It is worth noticing that representing the polar mask $\mathcal{P}$ with its normalized inertial matrix $C$ is actually equivalent to fit the polar mask by an ellipse with a semimajor axis $2\sqrt{\lambda_1}$ and a semiminor axis $2\sqrt{\lambda_2}$ oriented to $\theta = 1/2 \arctan((2\eta_{11} - 2\eta_{02})/(2\eta_{02} - 2\eta_{02}))$, where $\lambda_1 \geq \lambda_2$ are the two eigenvalues of $C$ (see Figure 2 for example).

With the definition of $EC$ (Equation (3)), the centerness of the centroid is equal to 1 strictly. And the centerness of other neighboring anchor points depends both on the distance to the center and the shape (aspect ratio and rotation) of the ellipse. Figure 3 visualizes the BC, PC, and EC of aerial instances. The visualization shows that BC is more appropriate than PC, while being surpassed by EC.

2.4. Network Architecture. The architecture with proposed elliptic centerness is illustrated in Figure 4. Our overall pipeline is as simple as PolarMask [49] and the one-stage method [17]. We use a fully convolutional network consisting of a backbone and two task-specific head networks. The backbone network with the following feature pyramid network (FPN) [56] extracts deep features of images and generates the multiscale feature maps. Two branches of the head networks with four stacked convolution layers finish the final prediction. One branch predicts both instance categories and centerness, while the other predicts polar distances of the mask.

In the training phase, the forward propagation has finished here, and the three parts of predictions will be subsequently used to calculate the loss to finish the post propagation. While in the testing phase, the classification score ($H \times W \times C$, $C$ means the number of categories) will firstly be reduced to a 2D map ($H \times W \times 1$) by choosing the highest score of $C$ categories. Then, since the classification score and the centerness have the same size, they will be multiplied by each other to get the refined score, which is ranked later for postprocess (e.g., NMS). At last, the remaining sets of distances will be assembled and generate the final polar mask result. The polar-based mask $\mathcal{P}$ follows the formulation in [49]. It is modeled in polar coordinates (see Figure 4(d)), involving an anchor point $p$ with $N$ contour points $\{p_n(d_n, \theta_n)\}_{n=1}^N$ that have evenly angle intervals. Such polar coordinates can be further decoded to Cartesian coordinates $\{(x_n, y_n)\}_{n=1}^N$. The number of $N$ is set to 20 in the figure.

2.5. Loss Function for Back-Propagation. The loss function consists of three parts according to three tasks, i.e., the losses of classification, mask regression, and centerness, which is defined as

$$\mathcal{L} = \frac{1}{N_{pos}} \left( \sum I_c(l_i, l_i') + \sum I_c(l_i, d_i') + \sum I_{\text{ce}}(c_i, c_i') \right),$$

where $I_c(l_i)$ is an indicator function, $N_{pos}$ is the number of positive samples, and $i$ is the index of a sample in a minibatch. $l_i$, $d_i$, and $c_i$ are the predicted category, distance set, and centerness of the anchor $i$. $l_i'$, $d_i'$, and $c_i'$ are the ground-truth category, distance set, and centerness of the anchor $i$. The focal loss [57], polar IoU loss [49], and cross-entropy loss are adopted as the classification loss $\mathcal{L}_c$, the mask regression loss $\mathcal{L}_r$, and the EC loss $\mathcal{L}_{EC}$, respectively.
3. Results

3.1. Implementation Details. With the superiority of the ResNet [58] compared with the conventional backbone network VGG [59], we adopt ResNet-50 pretrained by ImageNet [60] with FPN [56] as the backbone network by default. In the mask regression branch, 36 rays are adopted to represent a polar mask based on polar if not specified. The hyperparameters of focal loss \( L_c \) are set to \( \alpha = 0.25 \) and \( \gamma = 2.0 \). We adopt the same training schedules as [49]. We adopt the same training schedules as MMDetection [61]. We train all models in 12 epochs on iSAID in the ablation study. The SGD optimizer is adopted with an initial learning rate of 0.0025, and the learning rate is divided by 10 at each decay step. The momentum [62] and weight decay are 0.9 and 0.0001, respectively. We adopt a learning rate warmup for 500 iterations. In the inference phase, we choose 2000 predictions at most in each feature level before NMS and finally retain no more than 2000 predictions in one image. We use a single RTX 2080 Ti GPU with a batch size of 2 for training and 1 for testing.

3.2. Evaluation Indicators. In this section, we briefly introduce some evaluation indicators in the instance segmentation task, i.e., average precision (AP) and frames per second (FPS).

AP is used to measure the accuracy of the prediction result. The higher the AP gets, the better result is predicted. AP considers both the precision and the recall.

\[
AP = \int_0^1 P(R) d(R), \quad (7)
\]

where \( P \) and \( R \) denote the precision and the recall, respectively. More specifically, the precision and the recall can be, respectively, calculated by

\[
P = \frac{TP}{TP + FP}, \quad (8)
\]

\[
R = \frac{TP}{TP + FN}.
\]

TP, FP, and FN denote true positives, false positives, and false negatives, respectively. It is worth noticing that the standard of distinguishing positives and negatives is whether the IoU (mask-IoU in instance segmentation task) between prediction and ground-truth is higher than the threshold. In the experiments below, the AP of each category is under the IoU threshold of 0.5.

The mAP denotes the mean AP of the 15 categories under the series of thresholds (0.50.95, with an interval of 0.05), and mAP \(_t\) denotes the mean AP of the 15 categories under one specific threshold of IoU.

\[
mAP = \frac{1}{N_{ds}} \sum_i AP_i, \quad (9)
\]

Besides, as the settings in Pycocotools, mAP \(_s\), mAP \(_m\), and mAP \(_l\), respectively, denote the mAP of small objects (area < 32 × 32), medium objects (32 × 32 ≤ area ≤ 96 × 96), and large objects (area > 96 × 96).

FPS means the number of images that the model can infer in one second. The whole process contains the forward propagation and the postprocess.

![Figure 3: Visualization of different centerness rendered in RGB; color blue to red denotes centerness value from 0 to 1. First row: heat maps of box centerness (BC). BC offers high targets of location quality to the box centers of instances, without considering the label of the mask. Second row: heat maps of polar centerness (PC). As discussed before, instances with large aspect ratios or complex contours have lower centerness than square instances (i.e., first to third columns). Small instances (i.e., ships in the last column) have rather low centerness, which can be hardly seen in the heat map. Third row: heat maps of elliptic centerness (EC). The irrationalities of BC and PC have been alleviated remarkably. Every instance has been well-assigned to centerness targets, and the EC targets of their centroids are set to 1 strictly (better viewed in color).](image-url)
3.3. Ablation Studies. In this section, we conduct a series of experiments with different settings to validate the effectiveness of our proposed method. ResNet-50 and FPN are used in all experiments, and the AP is tested on the validation set of iSAID.

3.3.1. Verification of Upper Bound. An essential concern about the polar-based mask is that it might not depict the mask precisely. In fact, even the pixel-based mask will be likely to lose some details of the mask as it usually faces the RoI-Pooling operator. As shown in Figure 5, we calculate the pixel-wise IoU of the target mask and ground-truth on the whole validation set and finally verify the upper bound of the polar-based mask in [49] and the pixel-based mask in [3]. It can be seen that the IoU between target and ground-truth is growing higher when the number of rays increases. Most categories remain a negligible gap between the two kinds of masks except for some of them, which shows that the concern about the upper bound of the polar-based mask is unnecessary.

3.3.2. Hyperparameter. As the formulation of EC (Equation (3)) involves one hyperparameter \( \alpha \), which is a part of covariance in Gaussian function and therefore affects values of EC, we first conduct experiments on different \( \alpha \). As shown in Table 1, different values of \( \alpha \) in [3.54,1.41,0.71,0.35,0.18] are used to train the model. We observe that the proposed EC is quite insensitive to the variation of \( \alpha \) from 1.41 to 0.35. Too small \( \alpha \) (i.e., 0.18) will give high scores (i.e., 1.0) to nearly all positive points and make them weigh equally as the centroid point, which violates the original design intention of centerness and causes a slight drop in accuracy. On the contrary, too large \( \alpha \) (i.e., 3.54) will result in a large proportion of low-quality samples that decreases the performance noticeably. Overall, the only hyperparameter \( \alpha \) is quite robust, and the proposed EC can be nearly regarded as hyperparameter-free. We set \( \alpha = 1.41 \) in the remaining experiments.

3.3.3. Effectiveness of Elliptic Centerness. After confirming the setting of the hyperparameter, we compare our EC with box centerness (BC) in [17] and polar centerness (PC) in [49]. Table 2 (a) and (c) are two baselines of BC and PC, respectively. Following the original designs, BC estimates the centerness to the box center, and the PC faces the centroid. Although the PC is specifically designed for the instance segmentation task while the BC serves for detection, we found that BC outperforms the PC in an aerial scene, especially in mAP_5. Such a result indicates that the more proper centerness provided, the better model learned, which is shown in Figure 3.

We firstly conduct a new sample strategy, which regards points inside mask contour as positive points (like Figure 3). While in [49], points only in a small range around the centroid will be set to positive. As shown in Table 2 (b) and (d), the new sample strategy improves about 1% mAP, which shows the effectiveness of this sample strategy. Then, in Table 2 (e), we replace the centerness with the proposed EC. We firstly try to use the box center to replace \( \mathbf{p} \) in Equation (3) and observe EC outperforms BC by nearly 2% in mAP_5, which demonstrates the effectiveness of considering the label of the mask. In the meantime, EC surpasses PC by a large margin (47.9% to 54.9% in mAP_5). This is mainly because our EC is more robust for evaluating the sample quality of aerial instances. Moreover, since PC in [49] is defined according to the centroid, we generate EC target with the centroid of instance as Equation (3). And it (Table 2 (f)) improves about 0.5% in mAP_5 than Table 2 (e). The result shows that the centroid is better than the box center in defining EC. Finally, we use both centroid as instance center and the new sample strategy (Table 2 (g)) and achieve 30.0% in mAP, which improves slightly than (e) while outperforming baselines (a) and (c) obviously.

We also report the results of class-wise accuracy for further analysis. Instead of listing all of the 15 categories, we classify these categories according to their characteristics for briefness. Specifically, the 15 categories are divided into 4 parts, i.e., objects with large aspect ratios (lar), complex contours (cc), small scales (ss), and normal aspect ratios (nar). The lar objects include large vehicle (LV), bridge (BR), and ship (SH). The cc objects include plane (PL), harbor (HA), and helicopter (HC). The ss objects only include small vehicle (SV). The remaining eight categories, i.e., baseball diamond (BD), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), swimming pool (SP), and ground track field (GTF), compose

![Figure 4: The architecture with proposed elliptic centerness.](image)
the last group, whose member is the category with normal aspect ratios (nar). As shown in Table 2, when compared with PC, our EC outperforms it by a large margin in mAP$_{lar}$, mAP$_{cc}$, and mAP$_{ss}$ and also outperforms PC about 2% in mAP$_{nar}$, which verifies that our EC alleviates the irrationality of PC in aerial obviously. While compared with BC, our methods are also better than BC about 2% mAP in each group.

3.3.4. Number of Rays. We also conduct experiments with different numbers of rays for representing the mask. Experiments in [49] verify that the performance of PolarMask which is embedded with PC keeps improving when utilizing more rays of the polar mask and finally becomes saturated. It is intuitive because more rays for representing can produce a more elaborate contour (shown in Figure 5). However, the result above is not accordant with the performance of PolarMask on the iSAID dataset. As shown in the top half of Table 3, the performance of the PC keeps dropping while the rays of the mask keep rising. We argue that the irrationality of polar centerness discussed before is the main reason that causes the consequence, especially for small instances (shown in Figure 1(c)), as the mAP$_S$ drops notably for rays from 20 to 72. As shown in the bottom half of Table 3, for our EC, the irrationality of former centerness has been alleviated obviously since the mAP keeps growing with the rising number of rays, which indicates the effectiveness of the proposed EC.

3.4. Comparisons with the State-of-the-Art. In this section, we compare our proposed EC with other state-of-the-art (SOTA) methods for instance segmentation. The settings have been stated in Sections 2.1 and 3.1.

Before the comparison experiments with SOTA methods, we firstly illustrate the advantages and disadvantages of those methods in the aerial instance segmentation task. As for two-stage methods, we compared our methods with Mask R-CNN [3], Mask Scoring R-CNN [6], and CenterMask [8], whose results tend to be more accurate. However, as they utilize the horizontal bounding boxes to extract the feature of the regional area, the mask results tend to contain patches when objects of the same category gather compactly. Besides, their computation complexity and parameters are larger. As for one-stage methods, we introduce PolarMask [49], Yolact [41], and ExtremeNet [42] for comparison. Their architectures are more concise and therefore have less computation complexity and

![Figure 5: Upper bound analysis. As for the polar-based mask, more rays can model the polar mask of the instance with higher IoU with the ground-truth. Generally, the pixel-based mask represents the mask a little better than the polar-based mask. Most categories remain a negligible gap between the two kinds of masks except for some categories whose contours are complicated (e.g., plane, harbor, and helicopter).](image-url)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Pixel mask</th>
<th>Polar mask (36 rays)</th>
<th>Polar mask (72 rays)</th>
<th>Polar mask (20 rays)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>0.60</td>
<td>0.65</td>
<td>0.70</td>
<td>0.75</td>
</tr>
<tr>
<td>LV</td>
<td>0.70</td>
<td>0.75</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td>PL</td>
<td>0.80</td>
<td>0.85</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>ST</td>
<td>0.90</td>
<td>0.95</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>SP</td>
<td>0.50</td>
<td>0.55</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>HB</td>
<td>0.70</td>
<td>0.75</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td>TC</td>
<td>0.80</td>
<td>0.85</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>GTF</td>
<td>0.20</td>
<td>0.25</td>
<td>0.30</td>
<td>0.35</td>
</tr>
<tr>
<td>SBF</td>
<td>0.10</td>
<td>0.15</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td>BD</td>
<td>0.05</td>
<td>0.10</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td>BR</td>
<td>0.01</td>
<td>0.05</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>BC</td>
<td>0.00</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>RA</td>
<td>0.05</td>
<td>0.10</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td>HC</td>
<td>0.50</td>
<td>0.55</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>AVG</td>
<td>0.80</td>
<td>0.85</td>
<td>0.90</td>
<td>0.95</td>
</tr>
</tbody>
</table>

**Table 1:** Analysis of different values of hyperparameter $\alpha$, which is a component of covariance in a two-dimensional Gaussian kernel. The smaller the value of $\alpha$, the higher the elliptic centerness target is generated.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>mAP</th>
<th>mAP$_S$</th>
<th>AP$_{75}$</th>
<th>mAP$_S$</th>
<th>mAP$_M$</th>
<th>mAP$_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.54</td>
<td>27.7</td>
<td>52.0</td>
<td>26.4</td>
<td>13.9</td>
<td>36.5</td>
<td>40.4</td>
</tr>
<tr>
<td>1.41</td>
<td>29.4</td>
<td>54.5</td>
<td>27.8</td>
<td>15.5</td>
<td>37.8</td>
<td>42.0</td>
</tr>
<tr>
<td>0.71</td>
<td>29.1</td>
<td>55.0</td>
<td>27.3</td>
<td>15.8</td>
<td>36.8</td>
<td>41.3</td>
</tr>
<tr>
<td>0.35</td>
<td>28.8</td>
<td>55.0</td>
<td>26.7</td>
<td>15.7</td>
<td>36.7</td>
<td>41.4</td>
</tr>
<tr>
<td>0.18</td>
<td>28.6</td>
<td>54.2</td>
<td>26.7</td>
<td>15.4</td>
<td>36.1</td>
<td>40.9</td>
</tr>
</tbody>
</table>
parameters. However, the segmentation results of these methods are not satisfactory due to different reasons. PolarMask and ExtremeNet use the polar mask and octagon to represent the mask, respectively, which have a lower upper bound of accuracy. The results of Yolact come from the prototype mask, which is generated directly by the feature map in FPN and lacks regional information.

### 3.4.1. Class-Wise Performances

Tables 3 and 4 show the class-wise AP of each method. We only compare methods whose training schedules are the same for fairness (i.e., the comparisons of the top half and the bottom half of Tables 3 and 4 are independent). In the top half of Tables 3 and 4, we use ResNet-50 as the backbone, and the training epochs are set to 12, except Yolact as the different setting in its original paper. Our EC surpasses PolarMask in every class, especially in the complex instance, e.g., plane (+11.1% AP) and harbor (+14.9% AP), large aspect ratio instances, e.g., large vehicle (+13.2% AP), bridge (+10.4% AP), and small instances, e.g., ship (+12.8% AP) and small vehicle (+16.2% AP), which indicates the effectiveness of EC. The same outperformance in almost all categories also occurs when compared with other one-stage methods like Yolact [41] and ExtremeNet [42]. While comparing with the state-of-the-art method Mask R-CNN [3], our method can still achieve better results in some categories (i.e., vehicle, ship, storage tank, and soccer-ball field). However, as discussed in Figure 5, due to the limitation of the polar-based mask, our method cannot perform desired results compared with Mask R-CNN in categories whose contours are complicated (i.e., plane, harbor, and helicopter). Therefore, we use mAP\_\text{\small{cc}} to represent mAP of the remaining 12 categories and find that only nearly 1% mAP gap between EC and Mask R-CNN. In the bottom half of Tables 3 and 4, we conduct experiments for the extension to validate the potential of our EC. With longer training epochs and a larger training dataset and the ResNet-101 backbone, our EC has less gap with Mask R-CNN in mAP\_\text{\small{cc}} (60.3% vs 61.1%) and even outperforms Mask R-CNN in mAP\_\text{\small{cc}} (62.1% vs 61.0%). Our EC achieves the best results in 10/15 categories.

Qualitative segmentation results of the PolarMask, Mask R-CNN, and our EC are visualized in Figure 6. The series of figures explicitly show how the proposed EC tackle the different kinds of challenge objects in aerial instance segmentation. The first three columns are the results of objects with large aspect ratios. EC detects and segments the typical objects (e.g., large vehicle, ship, and bridge) better than PolarMask and Mask R-CNN. PolarMask and Mask R-CNN fail to detect many objects with large aspect ratios. Besides, Mask R-CNN produces patches between the crowded objects. The fourth column shows the result of objects with complex contours (e.g., plane). It can be seen that the mask results of PolarMask have lower quality than Mask R-CNN and EC. And Mask R-CNN is still trapped in processing the crowded objects. Finally, the last two columns are the segmentation result of small-scale objects (e.g., small vehicle, storage tank). PolarMask is incapable to segment these small objects, and Mask R-CNN is even worse. However, our proposed EC can detect these small objects stably. In conclusion, compared with PolarMask, our method segments better on large aspect ratio instances and small scale instances. While comparing with Mask R-CNN, our method performs better on crowded objects without producing patches like Mask R-CNN and is also more effective in large aspect ratio instances and small scale instances.

### 3.4.2. Trade-Off between Accuracy and Speed

Mask R-CNN has a slightly higher accuracy, but its inference speed is not unsatisfactory due to its two-stage network design. The FPS of our one-stage EC (7.5) is similar to Mask R-CNN (7.8). This phenomenon results mainly from too many proposals produced as their score is rated higher, which costs more time for the postprocess. We conducted...
Table 4: Comparisons with Mask R-CNN on iSAID. † indicates mAP without complex categories (i.e., plane, harbor, helicopter). * means using both training and validation sets for training model.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Epochs</th>
<th>mAP 5</th>
<th>mAP 5†</th>
<th>mAP 5−95</th>
<th>PL</th>
<th>BD</th>
<th>BR</th>
<th>GTF</th>
<th>SV</th>
<th>LV</th>
<th>SH</th>
<th>TC</th>
<th>BC</th>
<th>ST</th>
<th>SBF</th>
<th>RA</th>
<th>HA</th>
<th>SP</th>
<th>HC</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-stage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mask R-CNN [3]</td>
<td>12</td>
<td>60.1</td>
<td>60.4</td>
<td>36.3</td>
<td>82.6</td>
<td>74.0</td>
<td>41.6</td>
<td>46.2</td>
<td>31.9</td>
<td>52.3</td>
<td>81.0</td>
<td>92.3</td>
<td>64.8</td>
<td>57.0</td>
<td>48.5</td>
<td>66.0</td>
<td>63.4</td>
<td>69.2</td>
<td>30.1</td>
<td>7.8</td>
</tr>
<tr>
<td>MS R-CNN [6]</td>
<td>12</td>
<td>59.3</td>
<td>60.0</td>
<td>36.9</td>
<td>81.0</td>
<td>73.6</td>
<td>40.7</td>
<td>45.8</td>
<td>31.8</td>
<td>52.5</td>
<td>80.4</td>
<td>91.3</td>
<td>65.5</td>
<td>56.2</td>
<td>46.8</td>
<td>66.7</td>
<td>62.9</td>
<td>68.6</td>
<td>25.7</td>
<td>7.9</td>
</tr>
<tr>
<td>CenterMask [8]</td>
<td>12</td>
<td>57.0</td>
<td>57.4</td>
<td>33.5</td>
<td>79.7</td>
<td>68.9</td>
<td>39.5</td>
<td>40.8</td>
<td>31.1</td>
<td>51.5</td>
<td>72.3</td>
<td>91.8</td>
<td>61.7</td>
<td>57.1</td>
<td>41.4</td>
<td>63.5</td>
<td>60.7</td>
<td>69.1</td>
<td>26.2</td>
<td>3.0</td>
</tr>
<tr>
<td>One-stage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PolarMask [49]</td>
<td>12</td>
<td>48.9</td>
<td>52.3</td>
<td>27.5</td>
<td>62.5</td>
<td>67.0</td>
<td>28.2</td>
<td>35.0</td>
<td>19.6</td>
<td>40.1</td>
<td>68.7</td>
<td>90.6</td>
<td>61.8</td>
<td>52.6</td>
<td>43.5</td>
<td>60.9</td>
<td>27.8</td>
<td>60.1</td>
<td>15.6</td>
<td>8.2</td>
</tr>
<tr>
<td>Yolact [41]</td>
<td>55</td>
<td>46.3</td>
<td>46.9</td>
<td>23.8</td>
<td>62.6</td>
<td>61.7</td>
<td>31.5</td>
<td>27.0</td>
<td>21.8</td>
<td>26.7</td>
<td>57.9</td>
<td>84.6</td>
<td>49.1</td>
<td>50.0</td>
<td>29.3</td>
<td>61.7</td>
<td>50.1</td>
<td>62.6</td>
<td>18.0</td>
<td>4.0</td>
</tr>
<tr>
<td>ExtremeNet [42]</td>
<td>12</td>
<td>31.3</td>
<td>39.0</td>
<td>11.7</td>
<td>01.0</td>
<td>60.0</td>
<td>15.6</td>
<td>31.5</td>
<td>8.1</td>
<td>6.7</td>
<td>15.2</td>
<td>85.9</td>
<td>62.4</td>
<td>51.6</td>
<td>39.3</td>
<td>66.1</td>
<td>0.8</td>
<td>25.3</td>
<td>1.1</td>
<td>6.2</td>
</tr>
<tr>
<td>EC</td>
<td>12</td>
<td>56.7</td>
<td>59.2</td>
<td>31.0</td>
<td>73.6</td>
<td>71.5</td>
<td>38.6</td>
<td>37.2</td>
<td>35.8</td>
<td>53.3</td>
<td>81.5</td>
<td>91.4</td>
<td>63.8</td>
<td>58.2</td>
<td>49.0</td>
<td>61.4</td>
<td>42.7</td>
<td>69.2</td>
<td>23.6</td>
<td>7.5</td>
</tr>
<tr>
<td>Extension</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EC</td>
<td>24</td>
<td>57.6</td>
<td>59.5</td>
<td>32.3</td>
<td>76.1</td>
<td>70.2</td>
<td>38.2</td>
<td>37.2</td>
<td>37.2</td>
<td>53.7</td>
<td>82.3</td>
<td>92.3</td>
<td>63.3</td>
<td>56.8</td>
<td>49.9</td>
<td>63.0</td>
<td>46.5</td>
<td>69.0</td>
<td>27.3</td>
<td>—</td>
</tr>
<tr>
<td>EC-r101</td>
<td>12</td>
<td>57.2</td>
<td>59.7</td>
<td>31.6</td>
<td>75.0</td>
<td>74.0</td>
<td>39.2</td>
<td>37.9</td>
<td>35.8</td>
<td>53.5</td>
<td>81.7</td>
<td>91.7</td>
<td>64.4</td>
<td>56.9</td>
<td>49.0</td>
<td>62.8</td>
<td>44.2</td>
<td>69.2</td>
<td>22.8</td>
<td>6.9</td>
</tr>
<tr>
<td>EC-r101 *</td>
<td>24</td>
<td>60.3</td>
<td>62.1</td>
<td>34.4</td>
<td>78.1</td>
<td>76.5</td>
<td>41.3</td>
<td>45.6</td>
<td>38.3</td>
<td>55.0</td>
<td>83.4</td>
<td>92.4</td>
<td>66.9</td>
<td>58.4</td>
<td>51.5</td>
<td>64.6</td>
<td>49.6</td>
<td>71.5</td>
<td>30.8</td>
<td>—</td>
</tr>
<tr>
<td>PolarMask-r101 *</td>
<td>24</td>
<td>51.4</td>
<td>53.3</td>
<td>30.1</td>
<td>66.5</td>
<td>72.2</td>
<td>29.9</td>
<td>38.4</td>
<td>19.0</td>
<td>41.9</td>
<td>63.0</td>
<td>90.5</td>
<td>66.1</td>
<td>50.3</td>
<td>49.3</td>
<td>61.9</td>
<td>33.9</td>
<td>65.6</td>
<td>22.1</td>
<td>7.9</td>
</tr>
<tr>
<td>Mask R-CNN-r101 *</td>
<td>24</td>
<td>61.1</td>
<td>61.0</td>
<td>37.8</td>
<td>83.2</td>
<td>76.4</td>
<td>41.3</td>
<td>51.8</td>
<td>31.6</td>
<td>52.8</td>
<td>80.5</td>
<td>91.3</td>
<td>66.1</td>
<td>55.5</td>
<td>51.1</td>
<td>66.0</td>
<td>66.9</td>
<td>68.2</td>
<td>34.2</td>
<td>7.0</td>
</tr>
<tr>
<td>MS R-CNN-r101 *</td>
<td>24</td>
<td>58.8</td>
<td>58.6</td>
<td>38.1</td>
<td>81.6</td>
<td>75.7</td>
<td>40.6</td>
<td>50.4</td>
<td>26.1</td>
<td>49.6</td>
<td>70.7</td>
<td>90.6</td>
<td>65.4</td>
<td>52.2</td>
<td>48.6</td>
<td>67.4</td>
<td>64.8</td>
<td>66.4</td>
<td>32.0</td>
<td>8.0</td>
</tr>
</tbody>
</table>
Figure 6: Comparison of the segmentation results in iSAID with different methods. The first row is the results of PolarMask, the second row is the results of Mask R-CNN, and the third row is the results of EC. Compared with others, our EC segments are better on large aspect ratio instances and small scale instances. Moreover, EC performs better on crowded objects without producing patches like Mask R-CNN.

Figure 7: Results are on the test set of iSAID, and the mAP is reported without complex categories (i.e., plane, harbor, helicopter). our EC obtains 2 times acceleration in FPS with a drop of nearly 1% in mAP.
a trade-off experiment between accuracy and speed by increasing the score threshold of the proposal in the testing stage. As shown in Figure 7, our EC obtains 2 times acceleration in FPS with a drop of nearly 1% in mAP, while Mask R-CNN gains little in FPS, which demonstrates our method has a competitive performance on accuracy and a faster inference speed than state-of-the-arts. PolarMask has a similar trade-off curve with our EC, but it is located totally on our left, which means its accuracy is pretty lower than ours.

3.4.3. Computation Complexity Analysis. Table 5 compares the computation complexity and parameters of different models. Having a similar architecture, the computation complexity and parameters of EC and PolarMask are the same. Note that EC has fewer computation and parameters compared with Mask R-CNN. Even embedded with ResNet-101, our method has a lighter computation than Mask R-CNN.

4. Discussion

There are still some limitations of the EC. The center plays an important role in the definition of EC. However, as shown in Figure 8, when the central area of the object (e.g., harbor) does not lie in the mask, subsequently, the high-quality area defined in EC generated by the mask annotation will not exist in this object. It will harm the both training and testing phases for objects of this kind. The lower AP result for the harbor category of the EC in Tables 3 and 4 validates the issue to some extent. This issue can also be found in other centerness-based methods.

5. Conclusion

In this paper, we propose a novel location-quality estimator, termed elliptic centerness, to alleviate the issues caused by inaccurate centerness for instance segmentation in aerial images. Extensive experiments demonstrate that our EC can achieve competitive performance with other state-of-the-art methods on the accuracy, while it has a faster inference speed in aerial images.

There are still some limitations in centerness-based methods, i.e., EC cannot perform well on objects whose central areas do not lie in the masks, and the extra computation of EC slightly affects the training efficiency. So the future work plan can be focused on these limitations, such as designing a more robust and simple-defined EC. Moreover, since our proposed EC is an estimator of location quality, it is a potential to be integrated into other instance segmentation methods with common CNN architectures, as long as they determine the definition of EC under their architectures and add the module to predict the EC. Therefore, future work can also study how to embed EC in other SOTA CNN architectures.

Data Availability

The data of iSAID used to support this study are publicly available. The iSAID data can be downloaded from the website https://captain-whu.github.io/iSAID/index.html. The code is available upon request.

Conflicts of Interest

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

Authors’ Contributions

Y. Luo proposed the method and implemented the experiments. Y. Luo, J. Han, and Z. Liu participated in the analysis of experimental results. Y. Luo and J. Han wrote the manuscript. Z. Liu, M. Wang, and G. Xia provided supervisory support. All authors read and approved the final manuscript.
Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (grant number: 2042021k0040).

References


[29] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in

