Instance segmentation with pyramid integrated context for aerial objects

Juan Wang[1,2], Liquan Guo[1], Minghu Wu[1,2*], Guanhai Chen[1], Zishan Liu[1], Yonggang Ye[1] and Zetao Zhang[1]
[1] Hubei Energy Internet Engineering Technology Research Centre, Hubei University of Technology
Wuhan 430068, China
[e-mail: happywj@hbut.edu.cn]
[2] Hubei Laboratory of Solar Energy Efficient Utilization and Energy Storage Operation Control, Hubei University of Technology, Wuhan 430068, China
[e-mail: wuxx1005@hbut.edu.cn]
*Corresponding author: Minghu Wu

Received August 22, 2022; revised October 21, 2022; revised December 19, 2022; accepted February 12, 2023; published March 31, 2023

Abstract

Aerial objects are more challenging to segment than normal objects, which are usually smaller and have less textural detail. In the process of segmentation, target objects are easily omitted and misdetected, which is problematic. To alleviate these issues, we propose local aggregation feature pyramid networks (LAFPNs) and pyramid integrated context modules (PICMs) for aerial object segmentation. First, using an LAFPN, while strengthening the deep features, the extent to which low-level features interfere with high-level features is reduced, and numerous dense and small aerial targets are prevented from being mistakenly detected as a whole. Second, the PICM uses global information to guide local features, which enhances the network's comprehensive understanding of an entire image and reduces the missed detection of small aerial objects due to insufficient texture information. We evaluate our network with the MS COCO dataset using three categories: airplanes, birds, and kites. Compared with Mask R-CNN, our network achieves performance improvements of 1.7%, 4.9%, and 7.7% in terms of the AP metrics for the three categories. Without pretraining or any postprocessing, the segmentation performance of our network for aerial objects is superior to that of several recent methods based on classic algorithms.

Keywords: Aerial object segmentation, Pyramid integrated context, Local aggregation feature pyramid networks, Context information, Feature enhancement.

1. Introduction

Instance segmentation is a challenging and crucial task in computer vision that seeks to simultaneously handle both object detection problems and semantic segmentation problems. Instance segmentation involves not only predicting and distinguishing between the location and semantic classes of each object in an image but also distinguishing between different instances within the same class. Instance segmentation technology can be commonly employed in autopilot systems, robot control, assisted medical image segmentation and remote sensing imaging.

Instance segmentation methods are currently divided into two categories: two-stage methods and one-stage methods. Recent best-performing instance segmentation methods are usually two-stage methods. Among them, the classic two-stage instance segmentation algorithm Mask R-CNN [1] uses the RoIAlign method of bilinear interpolation by Faster R-CNN [2] to fix the RoI. The region proposal box can still be aligned with the instance in the process of feature map reduction, and a fully convolutional network (FCN) [3] mask branch is added at the end of the process to predict the category of each pixel, which could perform both target detection and segmentation tasks in parallel, but heavily relies on the target detection results. From Mask R-CNN, Liu et al. proposed PANet [4], which enhances the feature pyramid with precise location information existing in low-level features, creates an enhanced bottom-up path, shortens the information path, and avoids arbitrary assignment of feature levels through adaptive feature pooling. Huang et al. presented Mask Scoring R-CNN [5], which solves the problem of high classification confidence by adding the MaskIoU head to score the generated mask, instead of only using the classification score as a measure of model quality. However, the mask quality is not good enough. He et al. proposed PointRend [6], which regards instance segmentation as a rendering problem in image processing. By continuously upsampling and increasing the number of pixels, the points for predicting segmentation labels are adaptively selected in the image plane. The points are classified and predicted to increase the details of the boundary, clarifying the segmentation method and achieving high-quality segmentation, but the process is more complicated for the segmentation results. Cheng et al. proposed BMask R-CNN [7], in which the native mask branch is replaced with the boundary-preserving mask branch and boundary information [8] is introduced to improve the localization accuracy of the network. Recently, Cheng et al. proposed Mask2Former [9-10] and a new unified segmentation method. Mask2Former differs from other methods in the way that it generates binary masks. These binary masks are represented by a set of feature vectors to represent the object query [11-13] so that the transformer decoder can be used to train through a fixed set of queries.

Two-stage instance segmentation algorithms [14-15] usually need to generate proposal regions first and then perform bounding regression and mask segmentation on the proposed regions. Since the RPN increases the inference time, the one-stage instance segmentation algorithm is more dominant in terms of inference speed. To eliminate the limitation of the proposal box and obtain a higher inference speed, a series of one-stage instance segmentation algorithms [16-17] are derived from the one-stage object detection algorithm FCOS [18]. Bolia et al. proposed YOLACT++ [19-20], which simultaneously generates prototype masks and mask coefficients. The prototype mask was fused with mask coefficients to generate masks for each target. Its inference speed reaches an astonishing 33.5 FPS, but its segmentation accuracy is only 29.8% , which could achieve real-time instance segmentation but with much lower accuracy than the two-stage instance segmentation algorithm. Tian et al. proposed CondInst [21], in which a dynamic instance-aware network conditioned on an instance is

adopted without using the ROI of the instance as inputs, so cropping and feature alignment of the ROI are unnecessary. Since the capacity of dynamically generated conditional convolutions is greatly increased, the speed of inference can be substantially improved, and high-speed and high-precision segmentation is achieved, but large-size objects lack segmentation details. Chen et al. proposed BlendMask [22], which mixes top-down and bottom-up segmentation methods and is able to fuse low-level features containing location information and instance-level global information. Tian et al. proposed BoxInst [23] and projection and pairwise affinity mask loss, which changed the instance segmentation algorithm to a semisupervised algorithm using only box annotations. Tian et al. proposed SOLOv2 [24-25], which converts the segmentation problem to a position classification problem so that anchors and bounding boxes are not needed. However, the category is assigned to the pixels of each instance according to the position and size of the instance, so the training time of this network is longer compared to other networks. Although one-stage instance segmentation algorithms have achieved excellent performance in terms of speed, they are still less accurate than the current two-stage instance segmentation network.

Number of detections: 43 Number of detections: 85
Mask R-CNN Ours

Fig. 1. Left: Instance segmentation results of Mask R-CNN for small and dense objects. Right: Context module added to introduce the global information so that the foreground and background could be better distinguished.

Although the strong Mask R-CNN [1] and its series of related variants [4-7] have provided powerful localization and higher-quality mask capabilities, the size of the convolution kernel determines the weighting range of the weights, leaving insufficient receptive field to segment aerial objects with small targets and less texture. Meanwhile, the two-stage instance segmentation algorithm relies heavily on accurate target detection to some extent, and the obtained instance masks have relatively low resolution, which makes it difficult to provide accurate candidate regions when segmenting small-sized and poorly textured objects. Since convolution usually has only a limited perceptual field, it is particularly important to introduce global contextual information to compensate for the lack of texture information and to enhance feature representation for small objects. Although the single-stage instance segmentation algorithm is free from the limitation of detection frame, it still needs to combine the whole picture to segment the scene when facing small-sized objects. During feature extraction, the image traverses a shallower convolutional layer, and its detail information is abundant.

However, the receptive field is too small, and the context information is insufficient. After the image traverses a deeper convolutional layer, its semantic information is rich, and the receptive field is larger. However, the detailed information is lost. Simultaneously, different categories, such as kites and flags, have similar characteristics. Accurately distinguishing between different instances requires a larger receptive field and contextual information between two different convolutional layers. As shown in **Fig. 1**, Mask R-CNN only detects a small number of instances around an object when predicting dense and small object instances. Due to the lack of a comprehensive understanding of the whole scene, it is easily filtered out by the convolution kernel as noise by only local convolution. Due to the pooling operation of ROI Align [1], the size of the feature map decreases, causing the loss of more instances. Compared with Mask R-CNN, our approach captures more small object instances, and the mask confidence is improved.

To enlarge the receptive field of local convolution features, a series of recent works [26-31] propose different methods to introduce contextual information. After summarizing and drawing on the above method of introducing global information, we propose the pyramid integrated context module (PICM) for instance segmentation, which effectively uses multiscale global information to guide local features at different scales. Overall, our main contributions are presented as follows:

·We propose the pyramid integrated context module (PICM) to guide local features by using global information, which enhances the network's comprehensive understanding of an entire image. The PICM compensates for the insufficient texture information of aerial objects and texture similarity between two different categories.

·We propose the local aggregation feature pyramid network (LAFPN). Through the local aggregation of features of adjacent scales, the expression ability of features at different scales is strengthened, and the interference effect of small object features on high-level features is reduced.

·Compared with the classic instance segmentation network in recent years, our method achieves a great improvement in the segmentation performance of small and dense aerial objects, which can effectively alleviate the segmentation difficulty of capturing instances that are too small. Compared with Mask R-CNN, our method achieves performance improvements of 1.7%, 4.9%, and 7.7% in terms of the AP metrics for airplanes, birds, and kites, respectively, on the MS COCO dataset.

2. Related work

2.1. Feature Pyramid Network

To effectively identify objects of different sizes, they are usually placed on feature maps of different scales for prediction. Feature Pyramid Network [32] can fuse feature maps of different scales to enhance their ability to represent features. Liu et al. proposed a simple bidirectional fusion PANet [4], adding another bottom-up feature fusion structure based on FPNs and decreasing the number of layers passed between low-level features and high-level features. Liu et al. proposed ASFF [33], which added an attention module when merging the features of different stages and can effectively control the contribution of the features of different stages to each other during fusion, and uses differential fusion, which allows the weights to be easily learned in back propagation. Ghiasi et al. proposed NAS-FPN [34] to search for the optimal model architecture by using a neural architecture search. The modular search space makes the search pyramid architecture manageable, but the search process

requires a huge amount of computation to find the optimal architecture. Tan et al. proposed BiFPN [35] for fast and efficient fusion of features of different scales through a hybrid scaling method, using learnable weights to assign the contribution of different features to the fusion result through additional feature weighting module. Qiao et al. proposed Recursive-FPN [36], which inputs feature maps after feature fusion into the backbone network for secondary circulation. The abovementioned FPN feature fusion method has numerous parameters and a complex structure, and feature fusion between different scales is prone to information loss and aliasing effects. Also when features are used in the detection process, deep features are used to detect large targets and shallow features are used to detect small targets. Large targets from deep features often require spatial information from shallow features, but these above FPN networks do not take into account the effect of small target features from shallow features on deep features when fusing features. The deep features incorporate both spatial information from the shallow features and unwanted small target features that have been filtered out, thus impacting on the segmentation of the large targets. To mitigate this problem and inspired by the above FPN idea, we propose a simple and efficient LAFPN for the characteristics of aerial objects.

2.2 Contextual Information Extraction

Each pixel of an image is not isolated, and there is a close connection between each pixel. When performing semantic segmentation or instance segmentation, local information and global context information should be fully considered to better process image features. Chen et al. proposed the Deeplab [27] series of semantic segmentation algorithms. The Atros spatial pyramid pooling (ASPP) module was introduced. Parallel sampling of feature maps with dilated convolutions at different sampling rates was conducted to expand the receptive field. Zhao et al. proposed PSPNet [28], which added a pyramid pooling module to the structure to fuse features at different levels for the fusion of semantics and details, and finally, the prior information from the pyramid pooling module and the original feature map are summed and fed to the final convolution module to complete the prediction. Fu et al. proposed DANet [29] to learn the spatial and channel interdependencies of features through a dual-attention network to add rich contextual information to local features. The location attention module selectively aggregates the features at each location by a weighted sum of the features at all locations. The channel attention module selectively emphasizes channel mappings where interdependencies exist by integrating the relevant features between all channel mappings. Yuan et al. proposed OCNet [30] to calculate the similarity between a single pixel and all pixels to obtain the target semantics and the mapping of each pixel. Compared with those of ASPP, which does not differentiate whether there is a relationship between a single pixel and the target semantics, the segmentation results are more accurate. He et al. proposed APCNet [31]. After fusing the three ideal features, the adaptiveness of the context vector, multiscale nature, and global-guided local affinity, an adaptive context module is proposed, which utilizes local and global representations to estimate the similarity of local region weights. Since the perceptual field of convolutional neural networks is much smaller than the theoretical size, especially in deeper networks, this leaves many networks without adequate incorporation of important global scene information. Also in the segmentation of complex scenes, objects at different scales need to be segmented on feature maps of different depths in the backbone network, but the above methods of introducing contextual information all expand the perceptual field on the same feature map, dividing a feature map into different scales to obtain global dependencies. To better extract contextual information for complex scenes, we use feature maps of different scales to compute global information vectors to guide local convolutional features to generate

affinity matrices. The affinity matrix with the region representation is multiplied to generate local convolutional features with global information.

3. Method

Compared with normal objects, aerial objects are smaller and have darker surfaces as they are far from the observation point, and most of the observation angles are upward angles. It is therefore difficult to perform segmentation based only on local features. For small and dense aerial objects, such as flocks of birds and kites, general feature extraction networks tend to aggregate features into a group when extracting them, thus misjudging the flock as birds. To alleviate these two problems, we propose two modules, the LAFPN and PICM, which strengthen high-level features and introduce contextual information.

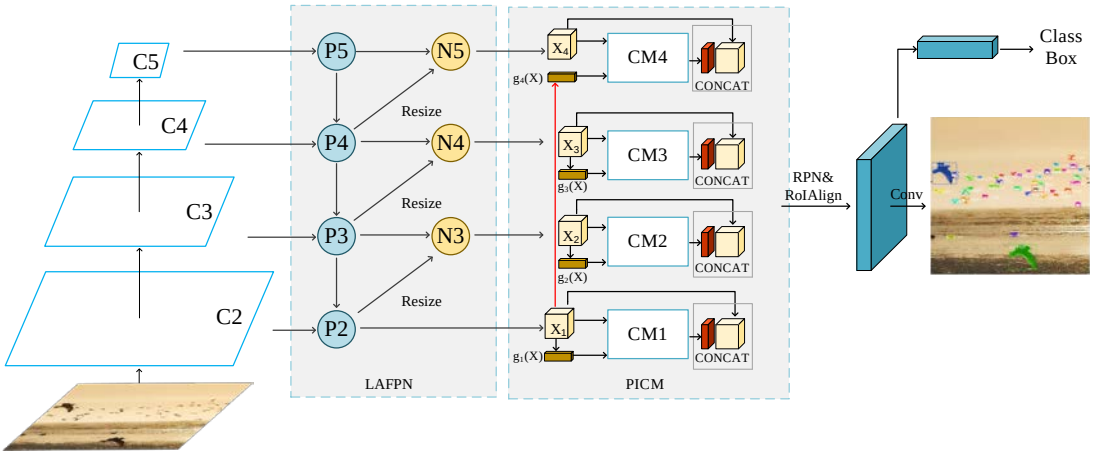


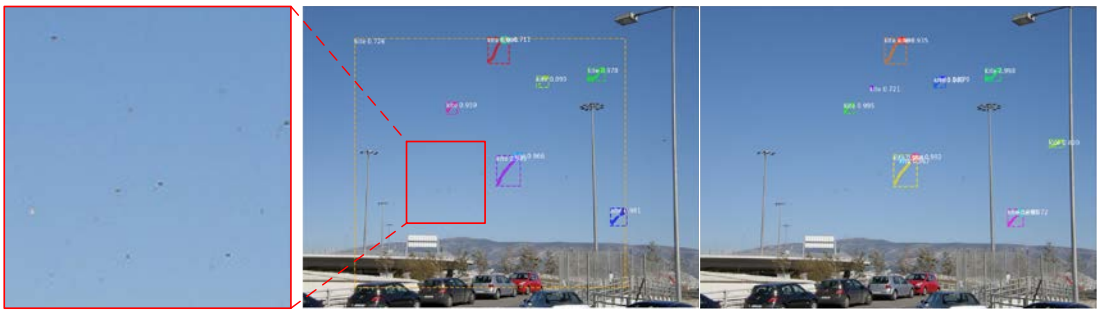
Fig. 2. PIC pipeline.

The main framework of the PIC network for the instance segmentation task is shown in **Fig. 2**. PIC uses ResNet as the backbone network to extract features C_n. To better integrate features of different scales, we design the LAFPN to output $\boldsymbol{X}_n$, which retains more detailed information of features at different scales and has a simple structure with fewer parameters. The network can effectively adapt to the characteristic requirements of aerial objects. The global information $\boldsymbol{g}_n(\boldsymbol{X})$ of different scale features is calculated by global average pooling of $\boldsymbol{X}_n$, and the context vector is constructed by sending $\boldsymbol{X}_n$ and $\boldsymbol{g}_n(\boldsymbol{X})$ to the context module. The context vector is concatenated with $\boldsymbol{X}_n$. We designed the pyramid aggregation context module to introduce global information and to improve the network's comprehensive understanding of an entire scene by using global information at different scales to correct local features, thereby improving the ability to capture small objects.

3.1 LAFPN

In the process of using the backbone network to extract features, the receptive field of the low-level network is smaller, and more fine-grained features, which have more spatial information, can be utilized. As the number of convolutional layers increases, the receptive field increases, and the high-level network has more semantic information. Therefore, the low-level network is generally used to segment small objects, and the high-level network is used to segment medium and large objects. Aerial objects are generally small and clustered in large numbers.

When using the general feature extraction network to extract features, although the high-level network has richer semantic information, it lacks spatial information, and it is easy to misjudge the clustered aerial objects as a single object.



Partial enlargement Mask R-CNN Ours

Fig. 3. Segmentation results for kite groups.

As shown in the segmentation results in **Fig. 3**, Mask R-CNN incorrectly segmented the kite group into one kite as the kites are small and numerous, taking up most of the entire image. In the process of feature extraction, there is not enough spatial information, and only the category of the object, not its position, can be judged. Therefore, we add a bottom-up feature fusion route to the LAFPN, as shown in **Fig. 2**, to strengthen the features of the high-level network. Compared with that of normal objects, the segmentation of aerial objects is more complicated. As shown in **Fig. 4**, due to the large size of normal objects, their features exist in both the high-level network and the low-level network. Bottom-up feature fusion can effectively strengthen the features of high-level networks. Aerial objects are often small and dense, and their features tend to disappear or cluster after being extracted to the high-level network, while small objects only need to be segmented in the low-level network. There are both large objects and small objects in complex scenes. The high-level network has only the features of large objects, whereas the low-level network has the features of both large objects and small objects. Since the high-level network is only used to segment large objects, if the features of C2 are fused with the features of C5, the filtered small object features will be added to the high-level features, thus affecting the segmentation of large objects. Therefore, in the bottom-up feature fusion route, only local feature aggregation is performed on adjacent feature layers.

For a better comparison with other networks, we use ResNet as our backbone network. We do not use the C1 feature map; although its resolution is high enough and it has rich location information, it consumes too much memory. We adopt {P2, P3, P4, P5} to represent the feature level generated by the FPN. P2 is directly input to the next layer. LAFPN conveys the semantic information of large objects from high-level features to low-level features through a top-down feature fusion path, which can alleviate low-level features from incorrectly segmenting prominent parts of large objects into small objects. It also conveys spatial information of large and small objects from neighbouring scales of low-level features to high-level features, helping high-level features to better locate the position of large objects without incorporating too many features of small objects into high-level features, avoiding the segmentation of small objects gathered into clusters into one large object. Other feature fusion networks [4] [33-36], on the other hand, all fuse high-level features and low-level features with each other, so that the small object features of the low-level features are also fused into the high-level features, while the high-level features do not need the small object features, which are noisy for the high-level features to segment the large objects. As shown in **Fig. 5**, a new feature map N_{i+1} is

generated by fusing the high resolution feature map P_i with the rough map P_{i+1}. Each feature map P_i is first bilinearly interpolated to reduce the size of the feature map and then N_{i+1} is generated by adding P_{i+1} to each of the element values in P_i via a lateral join. In the case of fusion of adjacent features, downsampling methods using bilinear interpolation could retain more spatial information about the upper layer of features compared to 3 × 3 convolution and maxpooling. The newly generated feature maps {N3, N4, N5} are convolved with a 3 × 3 convolution kernel to reduce the aliasing effect after feature map fusion.

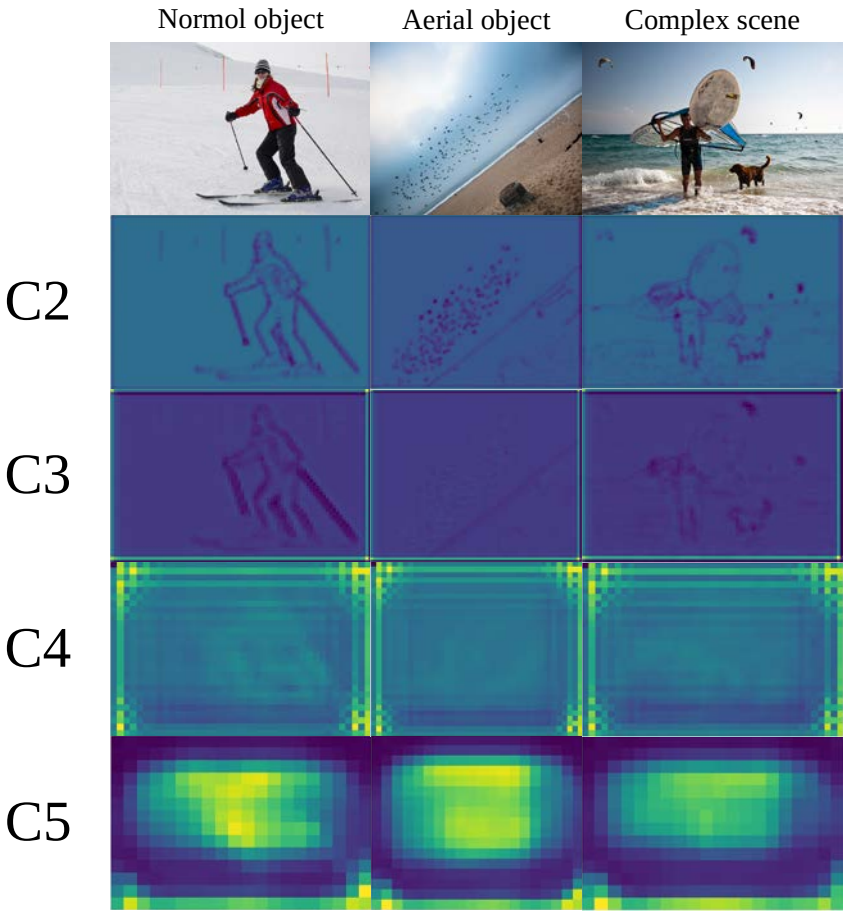


Fig. 4. Comparison of feature maps at different levels in different scenarios for the ResNet101 backbone network.

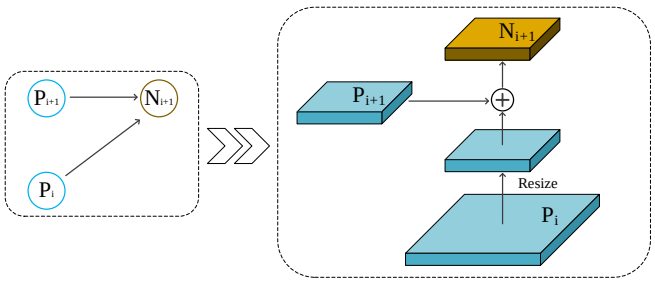


Fig. 5. Illustration of the fusion of different scale features in the local aggregation feature pyramid network.

3.2 PICM

Compared with ordinary objects, aerial objects have smaller sizes, darker surfaces and a lack of texture details owing to the influence of observation distance and observation angle. The bird shown in **Fig. 1** has almost all black surfaces, and its category cannot be judged only from partial observation. Additionally, kites and flags have the same texture. Contextual information is critical for complex scene parsing and instance segmentation. Objects of different scales need to construct context vectors from features of different scales to capture the long-term dependencies of local features and global information. We propose a pyramid aggregation context module to handle objects of different scales and to construct different scales of contextual information.

The instance segmentation task can be simplified to predict instance bounding boxes and pixelwise semantic labels. For this task, the mask can be predicted using only local features of different scales. However, using only local features for prediction disregards the surrounding features of the instance, and some small objects may be missed, which limits the instance segmentation performance of the network. To introduce contextual information and to improve the performance of network segmentation, some previous networks [27-31] applied different methods to introduce global information and achieved good results. The formula for contextual representation can generally be defined as

$$z_i = \rho\left(\frac{1}{|T_i|}\sum_{j\in T_i}\delta(x_j)\right) \tag{1}$$

$\boldsymbol{X}$ and $\boldsymbol{Z}$ are input features and context features, respectively. There are n pixels in $\boldsymbol{T}$. $\boldsymbol{T}_i$ is a subset of the $\boldsymbol{T}$ pixel set. $\boldsymbol{x}_j$ is the j-th element of $\boldsymbol{X}$. $\boldsymbol{z}_i$ is the i-th element of $\boldsymbol{Z}$. $\boldsymbol{x}_j$ is the contextual pixel associated with $\boldsymbol{z}_i$. The above formula means that $\boldsymbol{x}_j$ describes $\boldsymbol{z}_i$ through transformation functions $\boldsymbol{\delta}(\cdot)$ and $\boldsymbol{\rho}(\cdot)$. For ease of description, the above mathematical formulation is based on the one-dimensional case to describe the relationship between contextual pixels and contextual features and can be easily generalized to other dimensions. In this paper, we construct the $\boldsymbol{Z}_n$ function to calculate the context vectors of different scales, and the input variables are the feature maps $\boldsymbol{X}_n$ of different scales and the global information $\boldsymbol{g}_n(\boldsymbol{X})$ of different scales.

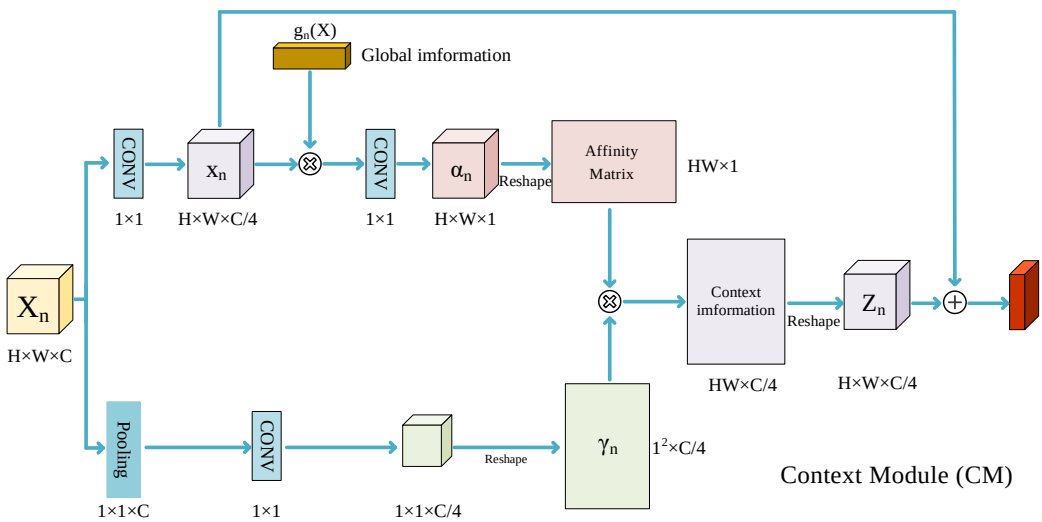


Fig. 6. Context Module.

The pyramid aggregation context module is the core module in the PIC network for introducing contextual information. The module consists of two branches: the first branch computes the affinity coefficient α_n guided by the global information, and the second branch computes the single-scale representation of the convolutional local features. The two matrices that are output are multiplied by the two branches to obtain the context vector. The precise details are described in detail below. The formula for calculating $\boldsymbol{Z_n}$ is expressed as follows:

$$Z_n = f(x_n, g_n(X))\gamma_n \tag{2}$$

$\boldsymbol{x}_n$ is the feature representation resulting from the convolution of $\boldsymbol{X}_n$. $\boldsymbol{g}_n(\boldsymbol{X})$ is the global information at different scales. $\boldsymbol{g}_n(\boldsymbol{X})$ and $\boldsymbol{x}_n$ compute the affinity matrix via function f. $\boldsymbol{\gamma}_n$ is the regional representation of the feature map. We utilize local feature $\boldsymbol{X}_n$ and its associated context vector $\boldsymbol{Z}_n$ to segment objects of different scales.

As shown in **Fig. 6**, the extracted feature maps of size H×W×C are input into two branches, where H, W, and C represent the width, height, and number of channels, respectively. In the first branch, we perform 1×1 convolution processing on the local features X_n output by {P2, N3, N4, N5} of different scales and convert them from original feature maps of size H×W×C to feature maps of size H×W×C/4 through a convolution operation, resulting in simplified feature maps $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$. Next, $\boldsymbol{g}_n(\boldsymbol{X})$ is obtained by performing a 1×1 convolution transformation and spatial global average pooling on $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$. The method is shown in **Fig. 7**. The feature map size of $\boldsymbol{X}_1$ is 256×256×256. After 1×1 convolution transformation and the spatial global average pooling method, as shown in the figure, $\boldsymbol{g}_1(\boldsymbol{X})$ of size 1×1×64 and $\boldsymbol{g}_4(\boldsymbol{X})$ of size 1×1×512 are obtained. $\boldsymbol{X}_2$ of size 128×128×512 is transformed to obtain $\boldsymbol{g}_2(\boldsymbol{X})$ of size 1×1×128. X_3 of size 64×64×1024 is transformed to obtain $\boldsymbol{g}_3(\boldsymbol{X})$ of size 1×1×256. After obtaining the global information representation vector $\boldsymbol{g}_n(\boldsymbol{X})$ of different scales, we multiply local feature $\boldsymbol{x}_n$ and global vector $\boldsymbol{g}_n(\boldsymbol{X})$ and calculate the global guided affinity coefficient α_n through a 1×1 convolution; it is subsequently reshaped into an affinity matrix of size HW × 1.

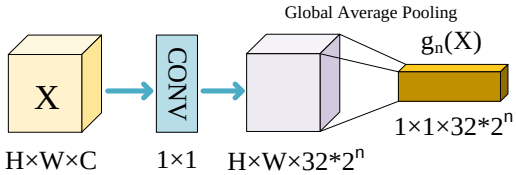


Fig. 7. Illustration of the global information $\boldsymbol{g}_n(\boldsymbol{X})$ calculation for feature maps at different scales.

The second branch transforms $\boldsymbol{X}$ into 1×1×C/4 using spatial global average pooling and a 1×1 convolutional transformation and then reshapes $\boldsymbol{\gamma}_n$ to a size of 1^2×C/4 to match the affinity matrix. Next, the affinity matrix and $\boldsymbol{\gamma}_n$ are multiplied to obtain the context information and are reshaped to obtain context matrix $\boldsymbol{Z}_n$. $\boldsymbol{x}$ is added to $\boldsymbol{Z}_n$ by using a residual connection to simplify the training process. $\boldsymbol{Z}_n$ is concatenated with $\boldsymbol{X}_n$ to introduce contextual information into the local convolutional features. After obtaining feature maps of different scales, we use the RPN to select regions of interest and use RoIAlign to fix the selected regions of interest to feature maps of the same size using bilinear interpolation.

4. Experiment

4.1 Datasets and evaluation metrics

We pick three categories for experiments on the MSCOCO 2017 instance segmentation dataset [37], which includes images of airplanes, kites, and birds. The training dataset contains 8401

images, and the validation dataset contains 312 images. Unless otherwise specified, the average precision values are evaluated using masked IoU. The evaluation metrics are average precision (AP), average precision with an IoU of 0.5 (AP_{50}), average precision with an IoU of 0.75 (AP_{75}), AP_S (AP for small objects: area<32^2), AP_M (AP for medium objects: 32^2<area<96^2), and AP_L (AP for large objects: area>96^2) for objects of different sizes.

4.2 Implementation details

The pretrained weights of the backbone network that we used are publicly available. Unless otherwise stated, we use the following implementation details: Our experimental framework is based on the deep learning framework TensorFlow; its version is 2.4, and the CUDA version is 11.4. The comparison experiment is implemented on the detectron2 platform, and its version is 0.6. All experiments are performed on a server with an Nvidia GTX 3060 graphics card. When we reproduce other models, the hyperparameters in the original network are unchanged or slightly adjusted. We employ ImageNet pretrained with ResNet-101 [38] as our backbone network. The network is trained for 90 K iterations using the stochastic gradient descent (SGD) method with a batch size of 1 and an initial learning rate of 0.001. At 60 K and 80 K iterations, the learning rate is reduced by a factor of 10. The weight decay and momentum are set to 0.00001 and 0.9, respectively. For ablation studies, all networks are trained on the 1× schedule of Mask R-CNN [1].

Table 1. Detailed quantitative results of AP at different IoUs for the three categories in the MS COCO dataset.

Method	**Schedule**	**AP**			**AP_{50}**			**AP_{75}**		
		airplane	**bird**	**kite**	**airplane**	**bird**	**kite**	**airplane**	**bird**	**kite**
Mask R-CNN	1×	59.3	30.1	34.5	82.7	49.7	56.6	67.1	32.0	38.5
CondInst	3×	53.3	30.1	30.6	85.8	56.3	59.7	55.0	27.5	28.9
BlendMask	3×	54.1	32.0	32.8	83.6	55.7	61.7	63.9	30.6	29.1
SOLOv2	3×	55.2	29.1	30.9	83.0	51.1	57.8	65.1	28.3	29.7
BoxInst	3×	44.2	23.8	23.8	82.1	48.7	54.2	40.0	20.4	17.5
Point Rend	3×	54.7	32.5	32.1	82.8	57.2	59.9	65.4	30.8	30.1
BMask	1×	52.3	31.1	31.4	82.8	54.9	60.1	63.6	31.1	27.1
Mask2Former	1×	59.0	33.4	35.8	**89.2**	56.7	63.8	**72.9**	34.3	36.4
Proposed	1×	**61.0**	**35.0**	**42.2**	80.8	**58.3**	**66.0**	67.5	**35.8**	**46.7**

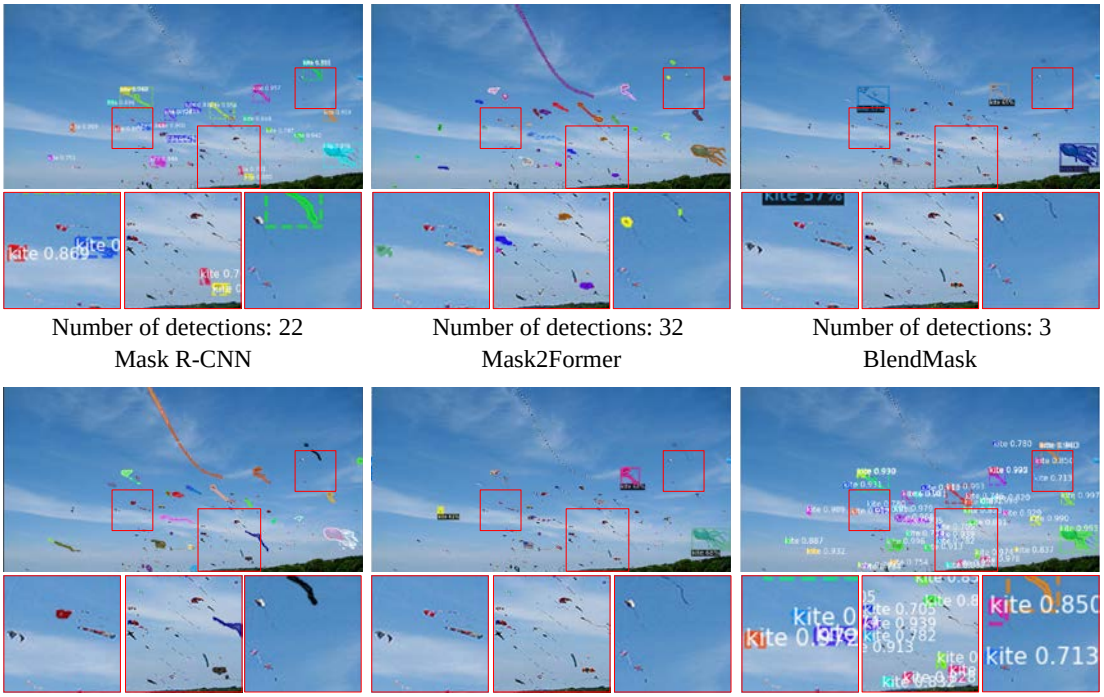
4.3 Experimental results and analysis

We perform a comparison with the state-of-the-art algorithms for instance segmentation, as shown in Table 1, using the same backbone network ResNet-101-FPN. R101 is used below to replace ResNet-101. When reproducing other networks, their source code parameters are set using the original parameters. The algorithms applied for comparison all use three categories, airplanes, kites and birds, from the COCO dataset and use the same evaluation metrics. As shown in **Table 1** and **Table 2**, we conduct a detailed comparison of the APs at different IoUs and scales for three aerial categories, airplane, bird, and kite. Generally, our algorithm exhibits a higher AP than the above algorithms, uses a 1× schedule for training, and has a shorter training time. Compared with the original network, Mask R-CNN, the AP increases by 1.7%, 4.9%, and 7.7% for the categories of airplane, bird, and kite categories, respectively. The AP_S

is improved by 6.9%, 4.4%, and 8.2%. These two indicators are also the highest among those of the above algorithms. According to the experimental results and a comparison of different scale APs for different algorithms, the segmentation performance of different algorithms on objects of different categories and scales are unbalanced, and our proposed network segmentation ability for aerial objects is stronger than that of other algorithms.

Table 2. Detailed quantitative AP results at different scales for the three categories in the MS COCO dataset.

Method	**Schedule**	**AP_S**			**AP_M**			**AP_L**		
		airplane	**bird**	**kite**	**airplane**	**bird**	**kite**	**airplane**	**bird**	**kite**
Mask R-CNN	1×	50.5	20.0	34.9	55.0	46.5	39.5	67.2	68.3	32.5
CondInst	3×	38.5	16.4	26.1	48.8	53.4	40.9	66.2	72.4	51.9
BlendMask	3×	42.4	18.7	28.4	48.8	54.2	40.8	64.1	72.3	56.4
SOLOv2	3×	39.3	15.6	25.1	50.8	51.5	43.2	65.2	70.8	52.9
BoxInst	3×	26.3	10.8	19.2	42.9	44.6	35.5	56.3	68.3	37.3
Point Rend	3×	44.9	19.1	29.8	50.3	51.6	36.9	63.2	73.7	46.6
BMask	1×	41.4	17.8	29.6	49.2	48.1	36.2	60.0	72.9	42.8
Mask2Former	1×	49.5	19.7	30.5	56.3	**58.3**	48.5	**70.8**	73.9	**57.9**
Proposed	1×	**57.4**	**24.4**	**43.1**	**56.5**	53.0	**48.7**	66.0	**74.1**	43.1



Number of detections: 22
Mask R-CNN

Number of detections: 32
Mask2Former

Number of detections: 3
BlendMask

Number of detections: 23
SOLOv2

Number of detections: 3
BoxInst

Number of detections: **47**
Ours

Fig. 8. Visual comparison of different networks for the segmentation of aerial objects.

As shown in **Fig. 8**, we compare the visualization results of some of the above algorithms. The one-stage instance segmentation algorithms BlendMask and BoxInst, which do not use the RPN to pregenerate proposal regions, have difficulty in capturing small objects. In the

visualization results, only a few large objects are detected, and their confidence scores are low. The bottom-up segmentation network SOLOv2, which first performs semantic segmentation at the pixel level and then classifies the pixels of each instance according to the location and size of the instance, can detect some small objects but still leaves a large number of small objects undetected. The grand unified segmentation architecture Mask2Former is surprisingly effective at the instance segmentation task, but the segmentation effect on small targets is still not good enough. Although Mask2Former and SOLOv2 can detect a certain number of instances, there are still many omissions when the object is small, and they do not perform bounding box detection on the instances. Our proposed algorithm detects the most instances, and the confidence score is improved compared with that of the original network.

4.4 Ablation

We performed multiple ablation experiments to analyse the PIC network, which are discussed in detail below. To demonstrate the effectiveness of the proposed module, the performance improvement of the LAFPN and PICM for network segmentation of aerial objects is gradually verified by modifying the backbone of the network model. The experimental results are shown in **Table 3**. After adding the proposed module, the network performance is improved.

Table 3. Comparison of network performances with the addition and absence of the proposed modules.

Backbone	AP	AP_S	AP_M	AP_L
R101+FPN	41.3	35.1	47.0	56
R101+ FPN+PICM	45.5	40.2	52.4	61.0
R101+LAFPN+ PICM	**46.1**	**41.7**	**52.7**	**61.1**

Global guided method. We compare several different ways in which global information guides local features in the PICM module. The different structures are shown in **Fig. 9**. Self-guided: Each scale feature map $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_4\}$ generates its own global information vector $\boldsymbol{g}_n(\boldsymbol{X})$ to guide its own local features. Low-level guided: Only the low-level feature map $\boldsymbol{X}_1$ is used to generate the global information vector $\boldsymbol{g}_n(\boldsymbol{X})$ to guide the local features of each scale. Mixed-guided: $\boldsymbol{X}_1$ generates global information vectors $\boldsymbol{g}_1(\boldsymbol{X})$ and $\boldsymbol{g}_4(\boldsymbol{X})$ to guide $\boldsymbol{X}_1$ and $\boldsymbol{X}_4$, and $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ use self-guided vectors to guide their respective local features. The experimental results are shown in **Table 4**. After adding the self-guided context module, the AP increases by 2.7%, AP_S increases by 2.1%, AP_M increases by 1.5%, and AP_L increases by 3.2%. Afterwards, we changed the global guided method to a low-level guided method. Compared with those of the self-guided method, the AP_M decreased by 0.9%, and the AP_L increased by 2.4%. Although the low-level feature P2 has richer contextual information, it also adds more noise, resulting in the degradation of segmentation performance on mesoscale objects. Since medium-sized instances are mostly segmented on $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, we cancel the guided method by $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ of the pair of $\boldsymbol{g}_2(\boldsymbol{X})$ and $\boldsymbol{g}_3(\boldsymbol{X})$ generated by $\boldsymbol{X}_1$. Using the mixed guided method, AP_L is improved by 3.5% compared with that of the self-guided method.

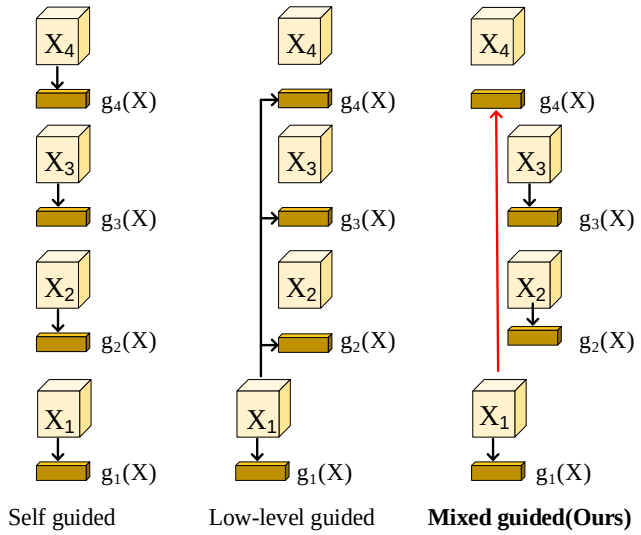


Fig. 9. Comparison of different multiscale global contextual information guidance methods.

Table 4. Different guidance methods for global information.

Backbone	AP	AP_S	AP_M	AP_L
R101+LAFPN	42.6	38.7	50.7	54.4
R101+LAFPN+ self guided	45.3	40.8	52.2	57.6
R101+LAFPN + low-level guided	45.4	40.9	51.3	60.0
R101+LAFPN + mixed guided	**46.1**	**41.7**	**52.7**	**61.1**

LAFPN. We compared several different FPN structures. The FPN reduces the semantic gap between different scale features by fusing the multiscale features. Our LAFPN does not employ a bottom-up feature fusion approach. As shown in **Fig. 10**, after N3 is directly connected to P3, it is only added to the resized result of the previous layer P2. Max pooling can effectively filter noise, but it easily misses details. More detailed features at different scales can be preserved by resizing. Through a nonsimple bottom-up feature fusion method, the loss of high-level semantic information caused by feature aliasing at different scales can also be avoided. The experimental results are shown in **Fig. 11 and Table 5**. In the comparison of visualization results in complex scenarios, PANet uses bottom-up feature fusion to fuse small objects features from the low-level features into the high-level features. To segment large object with high-level features, the small object features distract the segmentation of large object, resulting in the network not being able to segment large objects effectively. In PANet*, the 3×3 convolution downsampling method with stride 2 in the PANet feature fusion module is changed to 3×3 Maxpooling with stride 2. This change filters out noise in the low-level features and improves the network's ability to segment large targets but misses some small targets that could have been segmented. In LAFPN* and the LAFPN, the fusion of features by local aggregation reduces the overfusion of features at different scales, allowing both large and small targets to be segmented effectively, and the LAFPN uses Resize instead of Maxpooling in LAFPN* to retain more information and make the object segmentation profile more detailed. The quantitative results are compared. After adding PANet* and LAFPN*, the network performance did not improve. However, while using our LAFPN structure, the AP metric of the network increased by 0.6%.

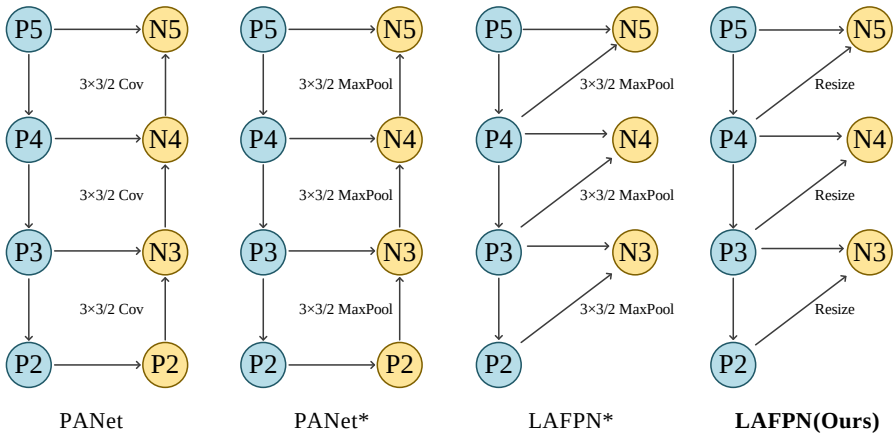


Fig. 10. Comparison of different FPN structures.

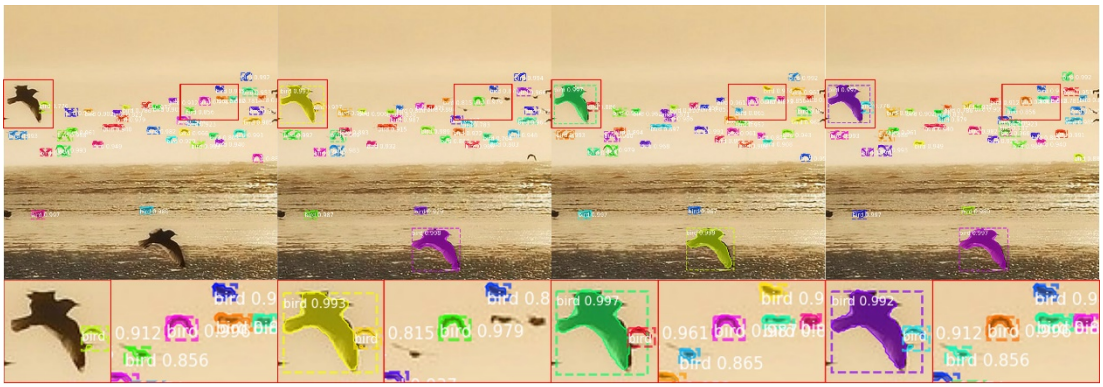


Fig. 11. Comparison of the visualization results of different FPNs in complex scenarios.

Table 5. Feature fusion ability of different FPN structures.

FPN	AP	AP_S	AP_M	AP_L
R101 + PICM+FPN	45.5	40.2	52.4	61.0
R101 + PICM + PANet	45.6	40.9	51.6	60.4
R101 + PICM + PANet*	45.0	40.0	52.6	60.1
R101 + PICM + LAFPN*	44.2	40.2	51.5	56.6
R101 + PICM + LAFPN	**46.1**	**41.7**	**52.7**	**61.1**

Important parameter settings. We compared different settings for the number of channels in the context module and the number of layers for local aggregation in the LAFPN. The experimental results are shown in **Table 6** and **Table 7**. N5 has 2048 channels and P2 has 256 channels. $\boldsymbol{g}_4(\boldsymbol{X})$ is generated by the P2 convolution. When the number of channels in the context module is set to C, the number of channels needs to be expanded from 256 to 2048. Increasing the number of channels too fast would cause a loss of information. Setting the number of channels to C/4 not only reduces the number of parameters but also has been shown to work better in evaluation metrics. When the number of aggregation layers is set to 3, the effect of low-level features on high-level features remains, and the experimental results show no improvement in segmentation ability. Changing the number of aggregation layers to 2

strengthens the features at different scales and reduces the extent to which small object features interfere with high-level features.

Table 6. Different settings for the number of channels in the context module.

Number of channels	AP	AP_S	AP_M	AP_L
C	44.2	40.1	51.0	57.2
C/4	**46.1**	**41.7**	**52.7**	**61.1**

Table 7. Comparison of layers for local aggregation of different scales in the LAFPN.

Number of aggregation layers	AP	AP_S	AP_M	AP_L
3	45.6	41.0	52.0	59.9
2	**46.1**	**41.7**	**52.7**	**61.1**

5. Conclusion

In this work, we propose an instance segmentation framework to alleviate the problem of small and dense aerial objects with unclear textures. This framework can effectively improve the number of detected instances and the quality of segmentation masks for small aerial objects in instance segmentation. Due to the limited receptive field of local convolutional features and the lack of a comprehensive understanding of the entire scene, it is difficult for segmentation networks to capture small objects. To alleviate this problem, PIC enhances the expressive power of local features by adding global information to guide local features. Features at different scales are enhanced by a feature fusion method that is more suitable for aerial objects. With the same experimental setting, PIC outperforms several recent advanced instance segmentation algorithms for aerial object segmentation. However, the segmentation ability for large-scale aerial objects is limited, because for the segmentation of large scale objects, whether they are detected or not is not the difficulty, what is needed for large scale objects is to enhance their edge segmentation details, so further research is needed to improve the segmentation performance of PIC for each scale of aerial objects.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.62006073), Key Research and Development Plan of Hubei Province (No.2021BGD013), Natural Science Foundation of Hubei Province (No.2022CFA007) and Science and Technology Project of Hubei Province (No.2022BEC017).

References

[1] He K, Gkioxari G, Dollár P, et al., "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386-397, Feb. 2020. Article (CrossRef Link)

[2] Ren S, He K, Girshick R, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, Jun. 2017. Article (CrossRef Link)

[3] Long J, Shelhamer E, Darrell T, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-650, Apr. 2017. Article (CrossRef Link)

[4] Liu S, Qi L, Qin H, et al., "Path aggregation network for instance segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 8759-8768, Jun. 2018. Article (CrossRef Link)
[5] Huang Z, Huang L, Gong Y, et al., "Mask Scoring R-CNN," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6409-6418, Jun. 2019. Article (CrossRef Link)
[6] Kirillov A, Wu Y, He K, et al., "PointRend: Image Segmentation as Rendering," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 9799-9808, Jun. 2020. Article (CrossRef Link)
[7] Cheng T, Wang X, Huang L, et al., "Boundary-Preserving Mask R-CNN," in *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 660-676, Nov. 2020. Article (CrossRef Link)
[8] Zhang G, Lu X, Tan J, et al., "Refinemask: Towards high-quality instance segmentation with fine-grained features," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6861-6869, Jun. 2021. Article (CrossRef Link)
[9] Cheng B, Misra I, Schwing A G, et al., "Masked-Attention Mask Transformer for Universal Image Segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 1290-1299, Jun. 2022. Article (CrossRef Link)
[10] Cheng B, Schwing A, Kirillov A, et al., "Per-pixel classification is not all you need for semantic segmentation," in *Proc. of 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Jun. 2021. Article (CrossRef Link)
[11] Fang Y, Yang S, Wang X, et al., "Instances as queries," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6910-6919, 2021. Article (CrossRef Link)
[12] Sun P, Zhang R, Jiang Y, et al., "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 14454-14463, 2021. Article (CrossRef Link)
[13] Dong B, Zeng F, Wang T, et al., "Solq: Segmenting objects by learning queries," *Advances in Neural Information Processing Systems*, Jun. 2021. Article (CrossRef Link)
[14] Cai Z, Vasconcelos N, et al., "Cascade r-cnn: Delving into high quality object detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6154-6162, Jun. 2018. Article (CrossRef Link)
[15] Kai C, Jiangmiao P, Jiaqi W, et al., "Hybrid Task Cascade for Instance Segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 4974-4983, 2019. Article (CrossRef Link)
[16] Xie E, Wang W, Ding M, et al., "Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5385 – 5400, Sep. 2022. Article (CrossRef Link)
[17] Ying H, Huang Z, Liu S, et al., "Embedmask: Embedding coupling for one-stage instance segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, Dec 2019. Article (CrossRef Link)
[18] Li Y, Qi H, Dai J, et al., "Fully Convolutional Instance-Aware Semantic Segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 2359-2367, Jul. 2017. Article (CrossRef Link)
[19] Bolya D, Zhou C, Xiao F, et al., "YOLACT: Real-Time Instance Segmentation," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9157-9166, Nov. 2019. Article (CrossRef Link)
[20] Bolya D, Zhou C, Xiao F, et al., "YOLACT++: Better Real-Time Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1108-1121, Feb. 2022. Article (CrossRef Link)
[21] Tian Z, Shen C, Chen H, "Conditional Convolutions for Instance Segmentation," in *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 282-298, Nov. 2020. Article (CrossRef Link)

[22] Chen H, Sun K, Tian Z, et al., "BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 8573-8581, Jun. 2020. Article (CrossRef Link)
[23] Tian Z, Shen C, Wang X, et al., "BoxInst: High-Performance Instance Segmentation With Box Annotations," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 5443-5452, Jun. 2021. Article (CrossRef Link)
[24] Wang X, Zhang R, Kong T, et al., "SOLOv2: Dynamic and Fast Instance Segmentation," in *Proc. of Advances in Neural information processing systems*, pp. 17721-17732, Oct. 2020. Article (CrossRef Link)
[25] Wang X, Zhang R, Kong T, et al., "SOLO: A Simple Framework for Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8587 - 8601, Nov. 2022. Article (CrossRef Link)
[26] Luo W, Li Y, Urtasun R, et al., "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," in *Proc. of 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Jan. 2017. Article (CrossRef Link)
[27] Chen L C, Papandreou G, Kokkinos I, et al., "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834-848, Apr. 2018. Article (CrossRef Link)
[28] Zhao H, Shi J, Qi X, et al., "Pyramid Scene Parsing Network," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 2881-2890, Jul. 2017. Article (CrossRef Link)
[29] Fu J, Liu J, Tian H, et al., "Dual Attention Network for Scene Segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3146-3154, Jun. 2019. Article (CrossRef Link)
[30] Yuan Y, Huang L, Guo J, et al., "OCNet: Object Context Network for Scene Parsing," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, Mar. 2021. Article (CrossRef Link)
[31] He J, Deng Z, Zhou L, et al., "Adaptive Pyramid Context Network for Semantic Segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 7519-7528, Jun. 2019. Article (CrossRef Link)
[32] Lin T Y, Dollár P, Girshick R, et al., "Feature Pyramid Networks for Object Detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 2117-2125, Apr. 2017. Article (CrossRef Link)
[33] Liu S, Huang D, Wang Y, "Learning Spatial Fusion for Single-Shot Object Detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 2117-2125, Nov. 2019. Article (CrossRef Link)
[34] Ghiasi G, Lin T Y, Le Q V, "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 7036-7045, Jun. 2019. Article (CrossRef Link)
[35] Tan M, Pang R, Le Q V, "EfficientDet: Scalable and Efficient Object Detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 10781-10790, Jul. 2020. Article (CrossRef Link)
[36] Qiao S, Chen L C, Yuille A, "DetectoRS: Detecting Objects With Recursive Feature Pyramid and Switchable Atrous Convolution," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 10213-10224, Nov. 2020. Article (CrossRef Link)
[37] Lin T Y, Maire M, Belongie S, et al., "Microsoft COCO: Common Objects in Context," in *Proc. of the European Conference on Computer Vision(ECCV)*, pp. 740-755, 2014. Article (CrossRef Link)
[38] He K, Zhang X, Ren S, et al., "Deep Residual Learning for Image Recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 770-778, Jun. 2016. Article (CrossRef Link)

Juan Wang received her B.S. degree from Nanyang Normal University in 2007 and M.S. degree from Xihua University in 2010. and her PhD from Tianjin University in 2015. She is currently a faculty member at Hubei University of Technology. Her re-search interests include image processing, pattern recognition, and computer vision.

Liquan Guo received his B.S. degrees in Huazhong University of Science and Technology Wuchang Branch, Wuhan, China, in 2018. He is currently pursuing his master's degree at Hubei University of Technology, Wuhan, China.

Minghu Wu received his B.S. degree from Communication University of China in 1998 and M.S. degree from Huazhong University of Science and Technology in 2003. and his PhD from Nanjing University of Posts and Telecommunications in 2013. His research interests include artificial intelligence and image processing.

Guanhai Chen received his B.S. degrees in Hubei Polytechnic University, Huangshi, China, in 2020. He is currently pursuing his master's degree at Hubei University of Technology, Wuhan, China.

Zishan Liu received her B.S. degrees in Hubei University of Technology, Wuhan, China, in 2019. He is currently pursuing his master's degree at Hubei University of Technology, Wuhan, China.

Yonggang Ye received his B.S. degrees in Wuhan Donghu University, Wuhan, China, in 2021. He is currently pursuing his master's degree at Hubei University of Technology, Wuhan, China.

Zetao Zhang received his B.S. degrees in Hubei University of Arts and Science, Xiangyang, China, in 2021. He is currently pursuing his master's degree at Hubei University of Technology, Wuhan, China.