DCTC: Fast and Accurate Contour-Based Instance Segmentation With DCT Encoding for High-Resolution Remote Sensing Images

Zhong Chen, Tianhang Liu, Xueru Xu, *Student Member, IEEE*, Junsong Leng, and Zhenxue Chen

***Abstract*—Instance segmentation in remote sensing images (RSI) poses significant challenges due to the diverse scales of targets, scene complexity, and a high number of targets, making most methods struggle with suboptimal performance and time-consuming computations. To solve those problems, a fast and accurate RSI instance segmentation model (named DCTC) is designed in this article. DCTC transforms classification problem into regression problem to improve the reference speed. DCTC contains two parallel branches. The contour branch performs iterative regression on contours, extracting precise contour information to improve boundary accuracy. Meanwhile, the discrete cosine transformation (DCT) branch refines mask predictions and supplements instance context information, which particularly benefits the segmentation of small targets. DCT encoding is employed in the DCT branch to convert the mask representation into DCT format, aligning the outputs of the contour and DCT branches. Three innovative modules are introduced in the DCT branch: the coarse result generation (CRG) module, iteratively deform and regression (IDR) module, and contour and DCT fusion module (CDF). The CRG module generates coarse DCT vectors and contour coordinates, facilitating information exchange between the contour and DCT branches. The IDR module iteratively refines DCT vectors, enabling DCTC to focus more on small targets and instance details. The CDF module merges DCT vectors and contour coordinates, ensuring effective interaction between boundary and context information, thereby enhancing performance. Extensive experiments demonstrate the superiority of DCTC, which achieves 67.7, 36.3, 67.4, and 55.1AP on NWPU VHR-10, iSAID, synthetic aperture radar (SAR) ship detection dataset, and high-resolution SAR images dataset, respectively, and ranks first among state-of-the-art methods while maintaining real-time processing capability. Furthermore, DCTC exhibits strong performance on both optical and SAR images, and the designed DCT branch can be simply plug into any contour-based method to improve the network performance.**

***Index Terms*—Contour-based method, discrete cosine transformation (DCT) encoding, instance segmentation, remote sensing.**

Manuscript received 23 January 2024; revised 4 March 2024; accepted 3 April 2024. Date of publication 9 April 2024; date of current version 29 April 2024. This work was supported in part by the Major Project of High Resolution Earth Observation System under Grant 30-Y60B01-9003-22/23 and in part by Civil Space Technology Advance Research Program (CSTARP) under Grant D040404. *(Corresponding author: Tianhang Liu.)*

Zhong Chen, Tianhang Liu, Xueru Xu, and Junsong Leng are with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the National Key Laboratory of Science and Technology on Multispectral Information Processing, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: henpacked@hust.edu.cn; m202273310@hust.edu.cn; xuxueru@hust.edu.cn; hzauljs@163.com).

Zhenxue Chen is with the School of Control Science and Engineering, Shandong University, Jinan 250100, China (e-mail: chenzhenxue@sdu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3386754

I. Introduction

THE advancement of satellite technology has facilitated the easy acquisition of high-resolution remote sensing image (RSI) in recent years, thereby making RSI applications, such as object detection [1], [2] and change detection [3], [4], a prominent focus in the field of computer vision. Among these applications, instance segmentation holds particular significance as it not only identifies and classifies objects in each image but also segments each object at the pixel level. Instance segmentation can be viewed as a synthesis of object detection and semantic segmentation, offering substantial utility in both civilian and military domains, such as automatic driving, and national security and defense.

However, the task of instance segmentation in RSI presents formidable challenges, including the following:

1) large variations in target scales;
2) a high number of targets in each image;
3) intricate and complex scenes;
4) the presence of noise in the images [5].

Two-stages methods [5], [6], [7] make improvements based on Mask R-CNN [8] and feature pyramid network (FPN) [9] to make network focus more on small objects and object variations. Some one-stage methods [10], [11], [12], [13] have also refined FPN to extract features at different scales, or incorporated self-attention modules to guide networks toward crucial information. These methods, designed specifically for RSI, have demonstrated improved performance in both object detection and instance segmentation. Nevertheless, most methods utilize fully convolutional networks to generate binary mask of fixed size for each instance. These dense prediction methods are often constrained by low inference speeds due to the large number of objects in RSI. Moreover, the process of mask prediction may encounter challenges from noise and scene complexity, which further compromises the overall performance of these approaches.

Different from those methods, contour-based approaches [14], [15] transform the 2-D mask prediction problem into a 1-D boundary regression problem, resulting in a significant reduction in time consumption. These methods iteratively regress predictions to fit object contours, proving suitable for RSI due to the relatively stable and usually convex contour topology of targets. However, challenges arise due to varying object scales, small object sizes, blurred object boundaries, and similar contours for

different categories (e.g., cars and boats) in RSI, all of which can decrease the segmentation accuracy.

In this article, a novel network named DCTC that combines the advantages of mask-prediction methods and contour-based methods is introduced. Since the boundary refinement is important yet difficult in instance segmentation due to the ambiguity of object boundaries, a contour branch is used to extract boundary features and refine object contour iteratively. To address challenges posed by complex scenes and image noise, a discrete cosine transformation (DCT) encoded mask branch, referred to as the DCT branch, is designed to complement context information. DCT encoding transforms mask classification into DCT vector regression, aligning the DCT branch with the contour branch and reducing time consumption. The DCT branch comprises three innovative modules: the coarse result generation module (CRG), iteratively deform and regression (IDR), and contour and DCT fusion (CDF) module. The CRG module enhances information exchange between the contour and DCT branches, the IDR module iteratively refines DCT vectors to complement and refine context information, and the CDF module fuses results of DCT branch and contour branch, interacts information, and maximizes the advantages of both branches. In DCTC, instance context information is effectively interacted with boundary information, resulting in more accurate instance segmentation results. Besides, the designed modules can work independently, making it easy to apply them to any contour-based methods.

The main contributions are highlighted below.

1) A remote sensing instance segmentation framework that combines the advantages of mask-prediction methods and contour-based methods is designed, which extracts boundary information and context information in parallel, and produces high-quality instance segmentation results with high speed.
2) A DCT branch that includes three novel modules is designed to complement context information and make network focus on small targets as well as instance details. Experiments show that the designed branch can be added to any contour-based method and improve the performance of network greatly.
3) Comprehensive experiments showcase the efficacy of the proposed network in both optical RSI and synthetic aperture radar (SAR) images. It accurately segments complex targets within intricate scenes, even in the presence of strong noise. DCTC outperforms state-of-the-art methods, securing the top position on NWPU VHR-10 dataset, iSAID dataset, SAR ship detection dataset (SSDD), and high-resolution SAR images dataset (HRSID).

The rest of this article is organized as follows. Related works of image instance segmentation are introduced in Section II. The overall architecture and adopted method are described in Section III. Comparison results and ablation experiments are showed in detail in Section IV, as well as the visualization results of different methods. Finally, Section V concludes this article.

II. Related Work

A. Instance Segmentation

Instance segmentation methods can be categorized as two-stages, one-stage, contour-based, attention-based methods, and so on. Two-stages methods follow the framework of "detect and then segment." Mask R-CNN adds a segmentation head to Faster R-CNN [16] to predict instance mask labels, Cascade Mask R-CNN [17] uses a cascaded RPN to improve the quality of input bounding boxes (bbox), while HTC [18] concatenates previous mask predictions and sends to mask head. HTC also uses semantic branch to supervise instance segmentation branch. One-stage methods generate mask predictions while producing object detection bbox. YOLACT [19] adds a mask coefficients prediction branch to RetinaNet [20] and generates prototype using FPN, and the mask coefficients and prototype are then combined to generate binary mask of instance. Some other algorithm adopts different combination methods. BlendMask [21] generates attention maps instead of coefficients, while CondInst [22] adopts dynamic filters for combination. Most two-stages and one-stage methods generate binary mask in bbox, which is especially time consuming. Meanwhile, as the number of instances increases, the memory resources it occupies also increases dramatically.

Contour-based approaches transfer prediction problems into regression problems, which may reduce the dependency of computational resources. Those methods can be further categorized to one-stage methods and two-stages methods. For one-stage methods, PolarMask [23] converts contour coordinates into polar space and adds a prediction head to generate polar coordinates. FourierNet [24] converts coordinates into Fourier space. While one-stage methods predict contour coordinates directly, two-stages methods produce target contour iteratively. DeepSnake [14] uses circular convolution to iterate initial contour. DANCE [25] adopts segmentwise matching to alleviate the problem of correspondence interlacing and learns an edge attention map to enhance the contour deformation. E2EC [15] changes the handcrafted initial contour to learnable initial contour and designs dynamic matching loss to reduce learning difficulty. Two-stages contour-based methods predict contours similar to active contour models [26], which may perform better for instances compared with one-stage methods. Since contour-based methods regress boundary coordinates instead of predict pixel-level masks, they are suitable for resource constrained situations. However, those methods may be overly concerned with contours, but the context information of instance is also important.

Some other methods adopt transformer framework or diffusion framework to achieve better performance. QueryInst [27] uses dynamic mask head to interact information between mask and bbox, and SOLQ [28] concatenates a set of instance mask vectors behind object queries of deformable DETR [29]. Mask2former [30] uses masked attention to mask background when computing attentions to reduce memory usage and exchanges order between self-attention module and cross-attention module. DiffusionInst [31] follows the "noise to filter" pipeline

and adds dynamic filters, such as CondInst to DiffusionDet [32]. Most of them improve from transformer or diffusion detection framework, which makes them especially time consuming.

B. Instance Segmentation in Remote Sensing

Most methods for remote sensing instance segmentation follow the binary mask generation paradigm and focus on improvements of FPN. FB-ISNet [13] adopts BiFPN to improve the performance of multiscale feature fusion. Chen et al. [33] improved the fusion module in FPN and got better performance on detecting multiscale targets. LFG-Net [6] designs LFCP architecture to enhance low-level features information and achieves good performance on SAR images. Furthermore, LFG-Net and IBMG-Net [34] improve RoI module and introduce high-resolution feature interaction. RPFNet [35] enhances low-frequency features and fuses them with high-frequency features, and OSM-Net [36] adopts orientation prediction to improve the precision.

However, those mask generation methods consume more resources at runtime as the number of instances increases, making them unsuitable for situations where are resource constrained or require high reference speed. Furthermore, most remote sensing instance segmentation models are specially designed for specific scenes, such as building segmentation and SAR image ship segmentation, a method satisfied for wide range of remote sensing scenes is imperative.

C. DCT Encoding

Many studies investigate different ways of mask representation to reduce the complexity. MEInst [37] uses PCA to encode instance mask to a compact vector, which is send to one-stage instance segmentation framework. PolarMask transfers contour coordinate to polar coordinate and predicts the distance from boundary to center with fixed angular intervals. Similar to PolarMask, FourierNet transfers boundary coordinates into frequency domain. These studies achieve higher inference speed through converting mask to a more compact representation. However, their mask quality is not ideal enough for high-quality instance segmentation.

DCT is widely used in computer vision field, and the field of instance segmentation based on deep learning in particular. DCTMask [38] improves R-CNN mask head and outputs a set of DCT vectors using fully connected layer. PatchDCT [39] separates image by patches and refines mixed patch using DCT encoding. SOLQ concatenates a set of vectors encoded by DCT, which represents instance mask to object queries containing categories and bbox coordinates. LFG-Net adopts DCT as segmentation representation. Since pixel-based mask representation uses 2-D binary mask to represent instance in low resolution (28×28 for example), which suffers decline of mask quality, DCT encoded mask representation can break the limits while not increasing the complexity, making high-quality mask for instance segmentation become possible.

III. Method

In this section, we introduce DCTC in detail, which includes backbone network, detection network, and instance segmentation network. Three modules of DCTC: CRG, IDR, and CDF are then introduced. Lastly, the loss function used in DCTC is presented.

A. Overall Architecture

The overall architecture of DCTC is illustrated in Fig. 1, which is composed of three basic components: the backbone network to extract features, the detection network to detect locations of objects, and the segmentation network to generate contour coordinates of each instance. Deep layer aggregate (DLA) network [40] is used as the backbone. The backbone network extracts features of the input $X \in \mathbb{R}^{H\times W\times 3}$ and get hierarchical features $B_i \in \mathbb{R}^{\frac{H}{S_i}\times\frac{W}{S_i}\times C_i}$ using deformable convolution, where $i \in \{1, 2, 3, 4, 5, 6\}$, $S_i \in \{2, 4, 6, 8, 16, 32\}$, and $C_i \in \{16, 32, 64, 128, 256\}$. The features of each layer are then aggregated with downsample ratio R, and final features $F \in \mathbb{R}^{\frac{H}{R}\times\frac{W}{R}\times C}$ are output using deformable convolution. The backbone network aggregates features in different scales and sends features to detection network. Since the features have been aggregated, the pipeline of DCTC can discard FPN completely, which is of great importance in remote sensing domain. Thus, the speed of DCTC is greatly improved. In detection network, CenterNet [41] is used to generate bbox of object detection. CenterNet generates a series of heatmap $Y \in [0, 1]^{\frac{H}{R}\times\frac{W}{R}\times C}$, where R is the downsample ratio and C represents the number of heatmap. $Y_{x,y,c} = 1$ represents the detected keypoint, and $Y_{x,y,c} = 0$ represents the background. For each keypoint, CenterNet also predicts four values x, y, w, h that represent the offset, bbox width, and bbox height, respectively. The extracted features and generated heatmap are then sent to segmentation network.

Segmentation network takes the object bbox as input, and iteratively refines both contour coordinates and DCT vectors to generate final segmentation results. It consists of two parallel branches. The contour branch iteratively refines contours to fit target boundaries. Contour branch brings boundary information to the network and focuses on object boundaries. The DCT branch encodes binary masks to DCT vectors and refines DCT vectors iteratively. In DCT branch, masks of small objects are filled to a large size, making DCT branch focuses on details of small targets. Three modules are designed and work sequentially in DCT branch. CRG module produces coarse DCT vectors for mask branch. Coarse DCT vectors are iteratively deformed with IDR module in the following. The contours and DCT vectors are well trained and fused in CDF module to produce final contour of instance.

B. CRG Module

Since the RSI own the characteristic of complexity scenes, instance-awareness-based approaches may fail to differentiate between instance and background. Meanwhile, the boundaries

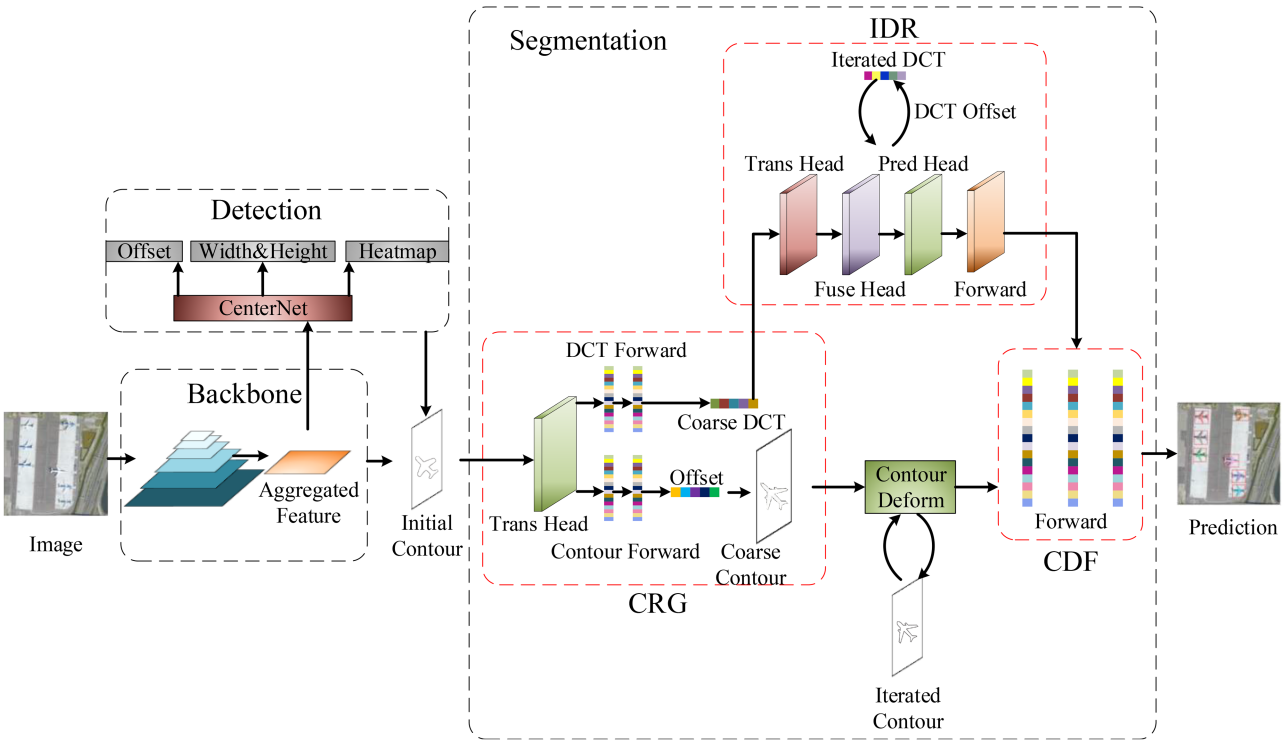

Fig. 1. Proposed framework of DCTC. CRG module takes the initial contour and image features as input and interacts the extracted features to generate coarse DCT vectors and coarse contours. IDR module uses circular convolutions to iteratively refine DCT vectors. CDF module combines the results of two branches by fully connected layers and interacts information of two branches to generate the final prediction.

of the objects in RSI are relatively stable and almost convex. These make contour-based methods fit for RSI. However, some objects of different class may have similar boundary or texture, making it hard for the network to tell them apart. On the other hand, feature of the instance context is particularly helpful for the network to localize objects and segment contours. To make DCTC more concentrates on the instance details and context, a mask branch is designed to complement information of instance and background. In order to harmonize the input form of contour branch and mask branch, as well as reduce the consumption of computing resource, DCT encoding is adopted in mask branch to transform classification problem into regression problem.

Given a $K \times K$ mask, DCT transfers $\text{Mask}_{k\times k}$ into a frequency domain $\text{Mask}^f_{k\times k}$

$$\text{Mask}^f_{k\times k}(u,v) = \frac{2}{K}C(u)C(v) \sum_{x=0}^{K-1}\sum_{y=0}^{K-1} M_{k\times k}(x,y)\cos\frac{(2x+1)u\pi}{2K}\cos\frac{(2y+1)v\pi}{2K} \quad (1)$$

where $C(w) = \frac{1}{\sqrt{2}}$ for $w = 0$, and $C(w) = 1$, otherwise. The top left-hand side corner of $M^f_{k\times k}$ represent low-frequency values that contain most information of the mask. After that, zigzag scanning is used on $\text{Mask}^f_{k\times k}$ to obtain 1-D vectors with length M. In DCTC, $M = 2N$, where N is the number of contour points. The inverse transform equation is as follows:

$$\text{Mask}_{k\times k}(x,y) = \frac{2}{K}C(x)C(y) \sum_{u=0}^{K-1}\sum_{v=0}^{K-1} M^f_{k\times k}(u,v)\cos\frac{(2x+1)u\pi}{2K}\cos\frac{(2y+1)v\pi}{2K}. \quad (2)$$

Therefore, the mask branch is converted to DCT branch.

As the initial contours are generated and sent to CRG module, CRG module produces the coarse DCT vectors for DCT branch and coarse contours for contour branch in parallel. CRG module extracts features of N contour points first, and concatenates initial contours and center point coordinates behind features. The input of CRG module is then sent to feature transmission head consisted of two convolution layers, notice that the output of transmission head is shared in both DCT branch and contour branch. A simple multilayer perceptron (MLP) is used to perceptual the initial vectors. Two fully connected layers are used for DCT branch to obtain coarse DCT vectors, which will be refined in IDR module. CRG module is illustrated in Fig. 2.

C. Iteratively Deform and Regression Module

The contour iteration branch fits boundaries iteratively, while the IDR module regresses the DCT encoded binary mask. Two branches are running in parallel but with same iteration times. Features of N points contours are first extracted. For DCT branch, coarse DCT vectors are combined with features, and circular convolution with different dilation rate is used for further integration of features. Convolution blocks are designed as

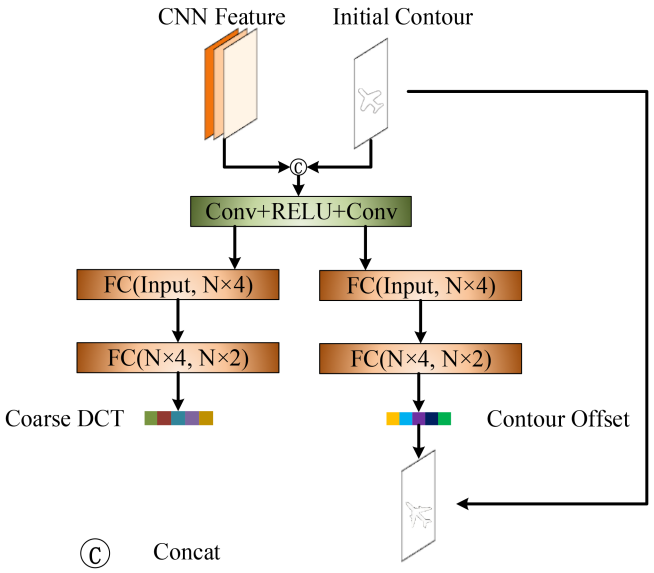

Fig. 2. Proposed CRG module. CRG takes features and initial contours as input and generates coarse DCT vectors and contour offsets in parallel. Two independent MLPs are used for different branches.

residual block [42] format. All the output of convolution blocks are concatenated and fused by convolution layer in fuse head. Max function is used as a substitute for max pooling function. After that, fused features are sent to prediction head for vector generation. In order to fully interact the information in each dimension of the DCT vectors, two fully connected layers are used to further refine the DCT predictions and output final DCT offsets. Fig. 3 illustrates the IDR module. The pipeline of IDR module is as follows:

$$\text{State} = \text{Conv}(\text{Cat}(\text{Res}(\text{Head}(f; v))^{\times 7})) \tag{3}$$

$$\text{Head}(\cdot) = \text{BN}(\text{RELU}(\text{Conv}(\cdot))) \tag{4}$$

$$\text{Res}(x) = \text{BN}(\text{RELU}(\text{Dconv}(x))) + x \tag{5}$$

$$\text{Pred} = \text{Conv}(\text{RELU}(\text{state}; \text{Max}(\text{state})))^{\times 3} \tag{6}$$

$$\text{Offset} = \text{FC}(\text{FC}(\text{Pred})) \tag{7}$$

where f are CNN features of contour points, v are the DCT vectors, Conv represents circular convolution, Dconv represents circular convolution with dilation, $\times n$ means repeat n times with different dilation rates, and Cat is a concatenation operation. For IDR module, the dilation rate of each block is $\{1, 1, 1, 2, 2, 4, 4\}$.

When the offset of DCT branch is got, the offset is simply added to input DCT vector as DCT result.

D. CDF Module

After getting the contour coordinates and DCT vectors, CDF module is used to fuse results of two branches and output final contour coordinates. Since the output of the DCT branch is already aligned with that of the contour branch, DCT vectors and contour coordinates can be simply concatenated together. Besides, the extracted features are also important as they provide sufficient information for supervising the fusion of DCT branch and contour branch. A bottleneck-like multilayer perceptron is

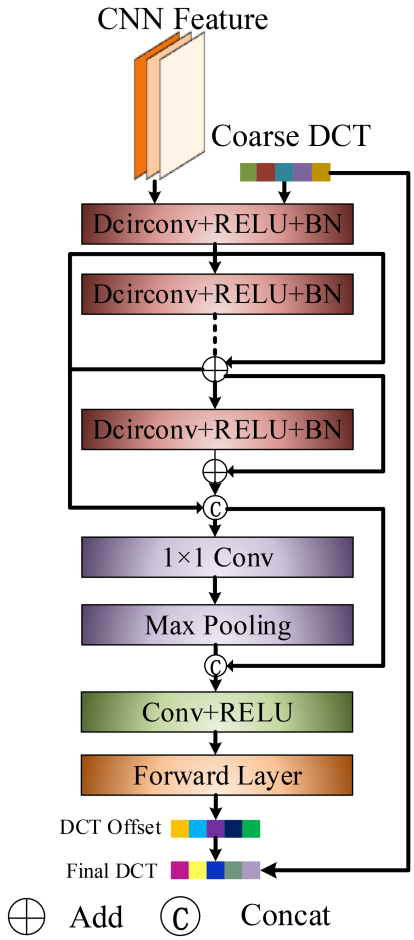

Fig. 3. Proposed IDR module. Circular convolution with dilation is introduced to refine DCT vectors in this module.

designed for this module as follows:

$$\text{Offset} = \text{FC}_{256}(\text{FC}_{1024}(\text{FC}_{1024}(f; c; p; d))) \tag{8}$$

where f are CNN features, and c are center polygons that represent the relative coordinates of contour. p is the contour coordinate, and d are the DCT vectors. FC_N represents fully connected layer with N output neurons. Same as IDR module, DCTC produces the offset of contour coordinates to make learning less difficult. The predicted offset is then added to polygon coordinates from previous contour branch. Therefore, the DCT branch serves as an aid to enhance detailed information. CDF module is illustrated in Fig. 4.

E. Loss Function

The architecture of our method has three subtasks: detection subtask, contour fitting subtask, and mask regression subtask. DCTC has detection loss, contour loss, and mask loss counterpart.

For mask loss, DCTC first transfers groundtruth polygons into binary masks. In order to adapt to the small target scale of RSI, relative coordinates are used of the contour

$$R(x, y) = C(x, y) - (x_m, y_m) \tag{9}$$

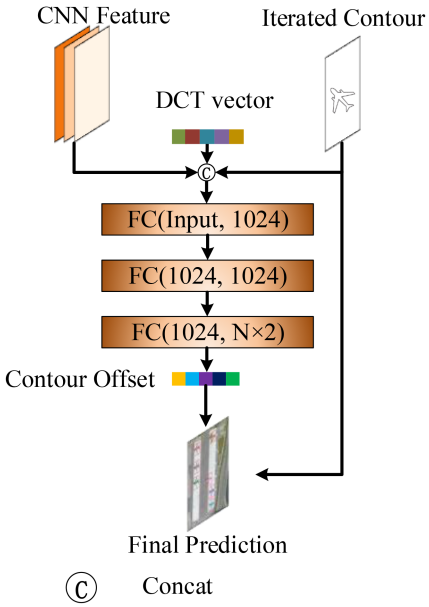

Fig. 4. Proposed CDF module. CDF takes features, DCT vectors, and contour coordinates as input. A small MLP is used to produce the final contour prediction.

where $R(x, y)$ are the relative coordinates, $C(x, y)$ are the initial coordinates, and x_m and y_m represent the minimum value of initial coordinates in x-axis and y-axis, respectively. The binary mask M_{gt} is resized to 128×128 after that to increase mask resolution. By doing this, we hope the DCT branch to focus more on the details of the instance, especially for small targets. Different types of loss function are tried for DCT branch. Contour loss of our algorithm will be applied to the final contour as well.

In the first type of loss function, DCTC inversely transforms the results of DCT branch using (1) to generate binary mask predictions M_{dct} with size of 128×128. Binary cross entropy loss is used to compute loss between M_{gt} and M_{pred}. The overall loss function is as follows:

$$L_{\text{DC}} = \lambda_1 L_{\text{Det}} + \lambda_2 L_{\text{Poly}} + \lambda_3(\text{BCE}(M_{\text{gt}}, M_{\text{iter}}) + \text{BCE}(M_{\text{gt}}, M_{\text{coarse}})) \quad (10)$$

where L_{DC} is the total loss of DCTC, L_{Det} represents the detection loss, L_{poly} represents loss of contour branch, BCE$(\cdot)$ represents binary cross entropy loss, M_{iter} and M_{coarse} are reversed forms of coarse DCT and results of DCT iteration branch, respectively, and λ_1, λ_2, and λ_3 are hyperparameters.

In the other type of loss function, DCTC converts the masks of groundtruth to DCT form using (2) to get DCT vectors of groundtruth V_{gt}, which has the same size of DCT prediction V_{pred}. Smooth L1 loss is used to compute loss between V_{gt} and V_{pred}. The overall loss function is as follows:

$$L_{\text{DC}} = \lambda_1 L_{\text{Det}} + \lambda_2 L_{\text{Poly}} + \lambda_3(\text{SL}(V_{\text{gt}}, V_{\text{iter}}) + \text{SL}(V_{\text{gt}}, V_{\text{coarse}})) \quad (11)$$

where SL represents smooth L1 loss, and V_{iter} and V_{coarse} are coarse DCT vectors and DCT iteration results, respectively.

IV. Experiments

In this section, DCTC is tested on four datasets, including optical images and SAR images. Four datasets are introduced elaborately first, and the result of DCTC will be showed and compared with other state-of-the-art methods subsequently. After that, the efficiency of each module in DCTC will be proved through ablation study as well as the applicability of our branch for different contour-based methods. Finally, the loss functions and hyperparameters of DCTC will be experienced.

DCTC will be tested on NWPU VHR-10 dataset [43], iSAID dataset [44], SSDD [45], and HRSID [46]. The precision results of different methods are listed in the following. Frames per second (FPS) and parameter count are tested on NWPU VHR-10 dataset and SSDD. The NWPU VHR-10 dataset and SSDD will be used for ablation studies and hyperparameters experiments.

A. Datasets

NWPU VHR-10 Dataset is a geo-remote sensing dataset for space object detection, which has 650 images containing targets and 150 background images, totaling 800 images, and the target categories include airplanes, ships, oil tanks, baseball stadiums, tennis courts, basketball courts, track and field stadiums, harbors, bridges, and automobiles, totaling ten categories. We randomly divided the whole dataset, 0.7 for training set and 0.3 for test set.

iSAID Dataset uses the images in the DOTA dataset for pixel-level labeling, and corrected the label errors in the DOTA dataset. Compared with the 188 282 target instances in DOTA, iSAID provides a much larger sample size and a much finer level of detail in the labeling. The target categories in the dataset include: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, swimming pool, and soccer ball field, which basically covers the key targets of urban remote sensing interpretation. iSAID contains 15 classes with 655 451 target instances, the number of images reaches 2806, and the number of instances in a single image can be up to 8000 with an average of 239, which is the first large-scale instance segmentation dataset in the field of remote sensing. The original images in dataset are cropped to size 800×800, and the final number of train set is 28 029, validation set is 9512, and test set is 19 377. As for the groundtruth of test dataset is not provided, we use a validation set for comparison and submit our test result to iSAID online server to get the final precision.

SSDD is the first open dataset widely used to study the deep learning-based ship detection and instance segmentation for SAR imagery. Due to the differences in the scale of the ships in SSDD, ships of different scales or sizes will produce different numbers of contour points. Larger ships provide more points while smaller ships provide fewer points. SSDD contains images with a total number of 1160 and is officially divided to a training set with 928 images and a test set with 232 images.

HRSID is a dataset released by the University of Electronic Science and Technology in January 2020. The dataset contains a total of 5604 high-resolution SAR images and 16 951 ship instances, which includes SAR images of different resolutions.

TABLE I
INSTANCE SEGMENTATION RESULTS (MASK AP) ON NWPU VHR-10 DATASET (%)

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
*YOLACT [19]	ResNet-50-FPN	43.3	77.5	44.0	23.1	40.5	54.3
Mask R-CNN [8]	ResNet-50-FPN	54.9	83.0	65.2	61.3	56.1	37.2
Cascade Mask R-CNN [17]	ResNet-50-FPN	58.9	93.7	65.4	45.1	57.6	69.2
PointRend [47]	ResNet-50-FPN	61.1	90.1	64.7	52.8	59.9	61.0
ARE-Net [48]	ResNet-101-FPN	64.8	93.2	71.5	53.9	65.3	72.9
Kumar [49]	HRNetV2p-W32	65.1	91.9	71.5	49.5	64.7	69.8
Shi and Zhang [10]	ResNet-FPN	65.2	94.9	72.1	49.4	65.7	71.2
*FB-ISNet [13]	DLA-BiFPN	67.0	94.5	72.0	-	-	-
HQ-ISNet [7]	HRFPN-W40	67.2	94.6	74.2	51.9	**67.8**	**77.5**
*YOLOv5s-MLS [11]	-	57.2	**95.5**	-	-	-	-
DCTC	DLASeg	**67.7**	91.8	**75.5**	**56.4**	66.4	69.9

[1] Models with * are one-stage models, while others are two-stages models.
The best results are in bold.

The resolutions of SAR images are: 0.5, 1, and 3 m. The HRSID has been partitioned into a training set that contains 3783 images and a test set that contains 1821 images. Each image of the dataset holds a resolution of 800×800.

B. Evaluation Metrics and Implement Details

All datasets are prepared as COCO format and adopt COCO evaluation metrics to test the performance. COCO tests average precision (AP) with a step size of 0.05, and IoU ranges from 0.5 to 0.95. IoU is a basic evaluation metrics

$$\text{IoU}_{\text{mask}} = \frac{\text{Pred}_m \cap \text{Gt}_m}{\text{Pred}_m \cup \text{Gt}_m} \tag{12}$$

where Pred_m is the mask prediction of models and Gt_m is the corresponding ground truth. The predictions can be categorized into true positive (TP), true negative (TN), false positive (FP), and false negative (FN) with a certain IoU, and the precision (P) can be defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{13}$$

The AP_{thr} is defined as

$$\text{AP}_{\text{thr}} = \int_0^1 P dr \tag{14}$$

Therefore, AP is the mean value of 10 AP_{thr}. Besides, to indicate the performance of objects in scales, AP_S, AP_M, and AP_L are adopted in COCO evaluation metrics. The inference speed of DCTC and other methods is measured using FPS, which indicates the total time consuming for dataset loading, dataset processing, and inference. The parameter count is also figured out, which indicates the size of each model. DCTC and other methods are tested on Ubuntu system and GeForce RTX 3090 GPU. The initial learning rate is selected as 0.0001, and half the amount at 80 and 120 epoch. DCTC is optimized by Adam optimizer. We have fixed a batchsize of 4, and the number of polygon points is 128. The iteration time of the IDR module is 2. There is no data enhancement strategy used during training.

C. Comparison Results

1) Comparison on the NWPU VHR-10 Dataset: Table I presents the instance segmentation results for DCTC and other state-of-the-art methods, as well as some classical methods on NWPU VHR-10 dataset. Models specific to RSI, such as HQ-ISNet [7], Shi and Zhang [10], YOLOv5s-MLS [11], FB-ISNet [13], ARE-Net [48], and Kumar [49], are chosen, as well as some classical models are used in RSI, such as Mask R-CNN [8], Cascade R-CNN [17], YOLACT [19], and PointRend [47]. DCTC achieves best performance on AP, AP_{75}, and AP_{small}, of which the AP indicator is 0.5% higher than the second place, which indicates that DCTC greatly improves the segmentation ability of the model, especially for small-scale objects, which gives DCTC an advantage in testing on the entire dataset. DCTC ranks second place on $\text{AP}_{\text{medium}}$ and AP_{large}, which represents that DCTC owns the ability to segment objects with different scales precisely. Meanwhile, Table II gives that DCTC achieves 23.88fps on the NWPU VHR-10 test set, nearly twice as much as the suboptimal method (YOLACT). It indicates that DCTC improves time consuming without sacrificing segmentation accuracy.

2) Comparison on the iSAID Dataset: Same as the NWPU VHR-10 dataset, Box2Mask-C [55] and Luo et al. [56] are selected as RSI specific methods, while Mask R-CNN [8], Cascade Mask R-CNN [17], YOLACT [19], PointRend [47], SOLO [57], and Mask Scoring R-CNN [58] are chosen as classic models. The results are tested on the iSAID validation set. DCTC gains 0.4% on AP performance compared with the second rank method. Meanwhile, DCTC performs best on AP_{small} and $\text{AP}_{\text{medium}}$, which indicates that the DCT branch can make network focus on small objects and their details, thus improve the segment precision on those objects. DCTC also ranks second place on AP_{50} and AP_{75}, which demonstrate that DCTC achieves good results on both detection and segmentation. In addition, DCTC achieves 21.47fps on the iSAID dataset. Table IV reveals more details.

3) Comparison on the SSDD: For SSDD, models for SAR image instance segmentation specifically are chosen, such as LFG-Net [6], C-SE Mask R-CNN [50], EMIN [51], FL-CSE-ROIE [52], MAI-SE-Net [53], SAR-CNN [54], as well as other

TABLE II
PARAMETER COUNT AND FPS ON NWPU VHR-10 DATASET AND SSDD

Dataset	Model	Backbone	Params(M)	FPS
NWPU VHR-10	*YOLACT [19]	ResNet-50-FPN	**34.83**	14.10
	Mask R-CNN [8]	ResNet-50-FPN	43.82	11.17
	Cascade Mask R-CNN [17]	ResNet-50-FPN	76.85	10.32
	PointRend [47]	ResNet-50-FPN	56.27	7.87
	ARE-Net [48]	ResNet-101-FPN	65.78	-
	DCTC	DLASeg	51.25	**23.88**
SSDD	*YOLACT [19]	ResNet-50-FPN	**34.83**	30.12
	Mask R-CNN [8]	ResNet-50-FPN	43.82	19.71
	Cascade Mask R-CNN [17]	ResNet-50-FPN	76.85	16.13
	PointRend [47]	ResNet-50-FPN	56.27	18.97
	DCTC	DLASeg	51.25	**34.0**

[1] Models with * are one-stage models, while others are two-stages models.
The best results are in bold.

TABLE III
INSTANCE SEGMENTATION RESULTS (MASK AP) ON SSDD (%)

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
*YOLACT [19]	ResNet-50-FPN	57.8	91.4	70.9	58.4	56.5	58.6
Mask R-CNN [8]	ResNet-50-FPN	64.8	94.3	81.7	66.7	59.0	19.4
Cascade Mask R-CNN [17]	ResNet-50-FPN	65.5	94.3	82.3	66.7	62.2	40.1
PointRend [47]	ResNet-50-FPN	65.6	94.5	82.3	**67.0**	62.2	16.8
C-SE Mask R-CNN [50]	-	58.6	89.2	71.2	58.3	60.7	26.7
EMIN [51]	-	61.7	94.3	76.8	62.1	61.3	61.3
FL-CSE-ROIE [52]	ResNet-101-FPN	62.6	93.7	78.3	63.3	61.2	75.0
MAI-SE-Net [53]	ResNet-101-FPN	63.0	94.4	77.6	63.3	62.5	47.7
HQ-ISNet [7]	HRNetV2-W40	57.6	86.0	72.6	56.7	61.3	50.2
*SA R-CNN [54]	ResNet-50-GCB-FPN	59.4	90.4	77.6	63.3	62.5	47.7
LFG-Net [6]	ResNeXt-64×4d	64.2	**95.0**	81.1	63.1	68.2	43.1
DCTC	DLASeg	**67.4**	93.9	**83.8**	64.4	**76.6**	**80.2**

[1] Models with * are one-stage models, while others are two-stages models.
The best results are in bold.

TABLE IV
INSTANCE SEGMENTATION RESULTS (MASK AP) ON iSAID VALIDATION SET (%)

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
*YOLACT [19]	ResNet-50-FPN	22.3	43.3	19.9	8.4	31.8	40.4
*SOLO [57]	Inception-ResNet-V2	24.7	44.9	24.2	7.9	32.7	47.8
Box2Mask-C [55]	ResNet-101-FPN	26.6	50.6	23.8	10.6	33.6	47.4
*Luo *et al.* [56]	ResNet-50-FPN	29.4	54.5	27.8	15.5	37.8	42.0
Mask R-CNN [8]	ResNet-50-FPN	34.8	57.4	37.0	20.5	43.2	50.3
Mask Scoring R-CNN [58]	ResNet-50-FPN	35.9	57.7	**38.4**	20.8	44.3	51.5
PointRend [47]	ResNet-50-FPN	35.6	**59.0**	37.3	20.3	44.8	**52.9**
Cascade Mask R-CNN [17]	ResNet-50-FPN	35.6	57.8	38.0	20.8	44.3	52.7
DCTC	DLASeg	**36.3**	58.2	38.0	**21.2**	**45.2**	47.3

[1] Models with * are one-stage models, while others are two-stages models.
The best results are in bold.

classic methods. Table III exhibits the comparison results. It can be seen that DCTC performs best on SSDD and achieves a performance gain of 3.2% compared with LFG-Net algorithm. Besides, DCTC performs well on AP_{75}, AP_{medium}, and AP_{large}, which demonstrates that DCTC gains improvement on segmentation when dealing with different scales, especially for large ships. DCTC achieves 34.0fps on SSDD, much higher than the compared methods.

4) Comparison on the HRSID: For HRSID, DCTC is compared with one-stage models, such as YOLACT [19] and SOLO [57], and two-stages models, such as Mask R-CNN [8], Cascade R-CNN [17], and so on. From the comparison results in Table V, it can be find that DCTC is tied for first place with PANet [59] on AP. Meanwhile, DCTC performs best on AP_{75}, AP_{small}, and AP_{large}, which further indicates that DCTC owns the ability to segment targets with different scales, and the designed branch helps network to segment objects precisely.

D. Ablation Study

In this section, details about ablation experiments will be showed. In the first section, each module of DCTC is sequentially removed, and different types of combination are experienced to prove the effectiveness of the modules. The modules

TABLE V
INSTANCE SEGMENTATION RESULTS (MASK AP) ON HRSID (%)

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
*YOLACT [19]	ResNet-101-FPN	39.6	71.1	41.9	39.5	46.1	7.3
*SOLO [57]	Inception-ResNet-V2	13.8	27.8	13.8	14.2	13.6	3.8
Mask R-CNN [8]	ResNet-101-FPN	54.8	85.7	65.2	54.3	62.5	13.3
Mask Scoring R-CNN [58]	ResNet-101-FPN	54.9	85.1	65.9	54.5	61.5	12.9
PANet [57]	ResNet-101-FPN	**55.1**	**86.0**	**66.2**	54.7	**62.8**	17.8
Cascade Mask R-CNN [17]	ResNet-101-FPN	52.8	83.4	62.9	52.2	62.2	17.0
DCTC	DLASeg	**55.1**	84.4	**66.2**	**54.8**	61.5	**19.8**

[1] Models with * are one-stage models, while others are two-stages models.
The best results are in bold.

TABLE VI
ABLATION EXPERIMENTS OF EACH MODULE ON NWPU VHR-10 DATASET

CRG	IDR	CDF	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
			64.8	91.2	70.2	51.9	63.8	72.0
		✓	65.7	90.9	71.9	52.2	64.4	71.5
	✓	✓	66.9	91.6	73.1	57.5	65.4	72.5
✓		✓	66.5	91.2	73.2	56.4	64.0	69.8
✓	✓	✓	67.7	91.8	75.5	56.4	66.4	69.9

TABLE VII
ABLATION EXPERIMENTS OF EACH MODULE ON SSDD

CRG	IDR	CDF	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
			65.3	93.9	81.9	62.8	72.4	67.6
		✓	66.1	93.3	82.4	63.9	72.6	80.2
	✓	✓	66.6	94.1	85.3	64.3	72.9	80.2
✓		✓	67.0	94.0	83.3	64.2	74.8	75.2
✓	✓	✓	67.4	93.9	83.8	64.4	76.6	80.2

TABLE VIII
COMPARISON OF LOSS FUNCTION

Loss function	AP	AP_{50}	AP_{75}
BCE Loss	66.9	91.6	74.6
Smooth L1 Loss	67.7	91.8	75.5

TABLE IX
EXPERIMENTS ON DIFFERENT MODELS

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DeepSnake	63.4	91.0	68.3	51.1	62.1	63.1
DeepSnake+Ours	**66.9**	**91.7**	**73.5**	**54.2**	**65.2**	**75.3**
E2EC	64.8	91.2	70.2	51.9	63.8	**72.0**
E2EC+Ours	**67.7**	**91.8**	**75.5**	**56.4**	**66.4**	69.9

The best results are in bold.

are also attached to other contour-based method to show the applicability. In the second section, the experiments on the loss functions as well as the hyperparameters will be showed in order to get the best learning result. All the ablation studies and experiments are tested on NWPU VHR-10 dataset.

1) Ablation Study on Modules. CDF Module: Tables VI and VII give the comparison of framework with and without CDF module. If CDF module is ablated, the whole DCT branch will not work. To serve as a comparison, we change the input of CDF module, simply concatenate polygon features and coordinates. Tables VI and VII give that CDF module slightly increases the AP. It is intuitive since the CDF module without DCT branch can be viewed as a different form of contour iteration module, thus can regress and refine the contour prediction solely.

a) IDR Module: Since the IDR module could not exist on its own, it is tested together with CDF module. Tables VI and VII give the comparison results. As the CRG module is not added, the input of IDR module is represented as random vectors in this experiment. It can be find out that DCT module improves the performance greatly from 65.7 to 66.9 on NWPU VHR-10 dataset, from 66.1 to 66.6 on SSDD, and is of great importance for DCTC.

b) CRG Module: Same as the IDR module, the CRG module could not stand alone as well. Therefore, the performance of CRG and CDF modules are tested together. CDF module takes the coarse result generated by CRG module as input, instead of iteration result. Tables VI and VII indicate that CRG module also enhanced the learning ability of model, as the AP increased from 65.7 to 66.5 on NWPU VHR-10 dataset and from 66.1 to 67.0 on SSDD. All the modules contribute to the performance improvement of the network.

c) Loss Function: Since the binary mask is encoded into DCT format, two different types of supervisory signals can be set up, as described in Section III-E. In Table VIII, the AP of the final result is compared using different loss functions. The loss functions will act on both coarse DCT vectors and iterative DCT vectors. It can be seen that BCE loss for masks is less effective for DCTC than that of smooth L1 loss for DCT vectors. We hypothesize that this is because the mask branch does not act directly on the final output in the form of binary masks. As the CDF module takes DCT vectors as input, a supervised signal acting on DCT vectors directly can assist better for multitask training.

2) Performance on Different Models: To further demonstrate the effectiveness of the designed modules, our branch is added on different contour-based methods, including E2EC and DeepSnake. Table IX expresses the experiment details on NWPU VHR-10 test set. Notice that all the models are tested with two iteration times. For DeepSnake, the designed branch gains 3.5 AP improvement from 63.4 to 66.9. For E2EC, the designed branch gains 2.9 AP improvement from 64.8 to 67.7. It can be seen that our modules dramatically enhance the models' ability to segment small objects, since AP_{small} is vastly improved. In addition, the modules contribute to the segmentation performance of objects at other scales to varying degrees. Visualization results showed in Fig. 5 also demonstrate that DCT branch improves the mask precision, especially for boundaries.

3) Experiments on Hyperparameters: Since the DCT branch is parallel to the contour branch, the weight settings of DCT branch can greatly affect the performance of the model. The

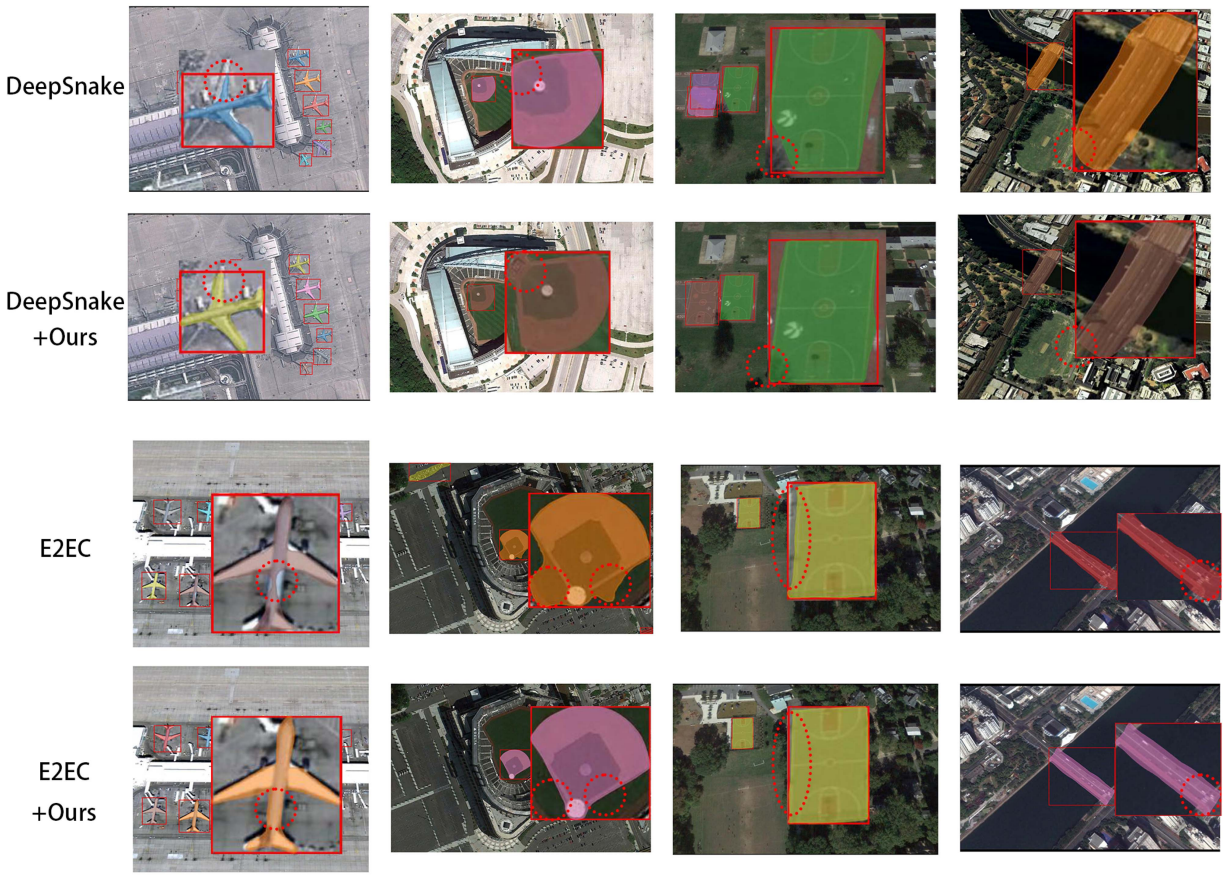

Fig. 5. Comparison of different contour-based models with/without our branch.

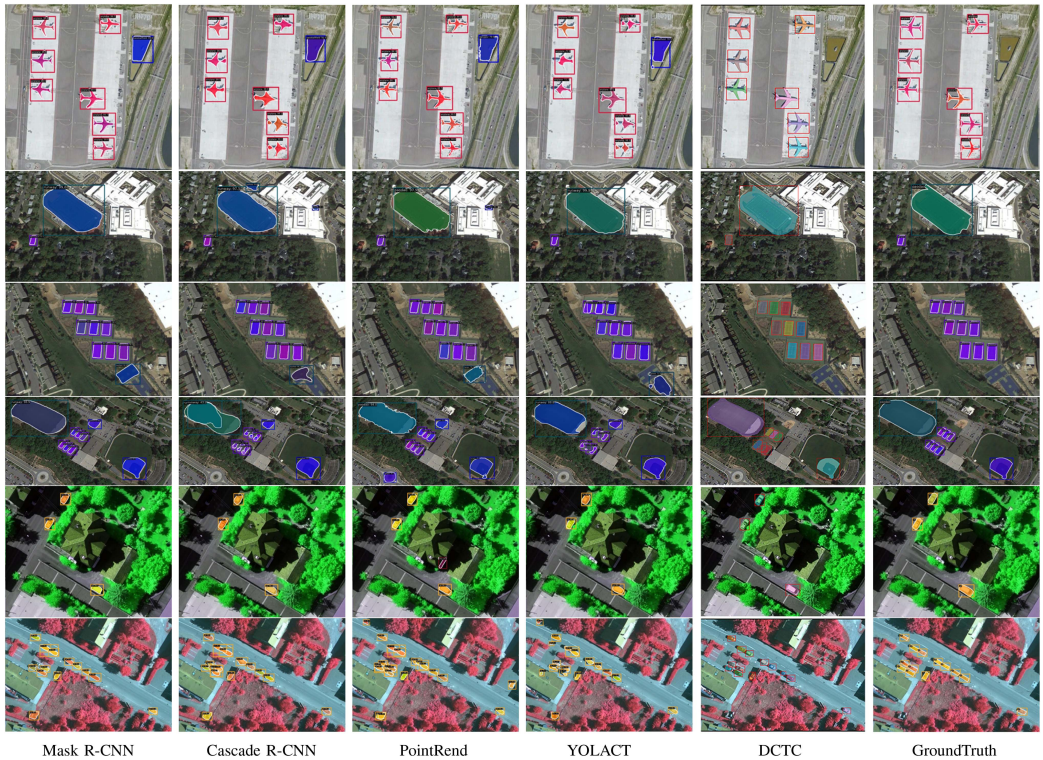

Fig. 6. Visualization results on NWPU VHR-10.

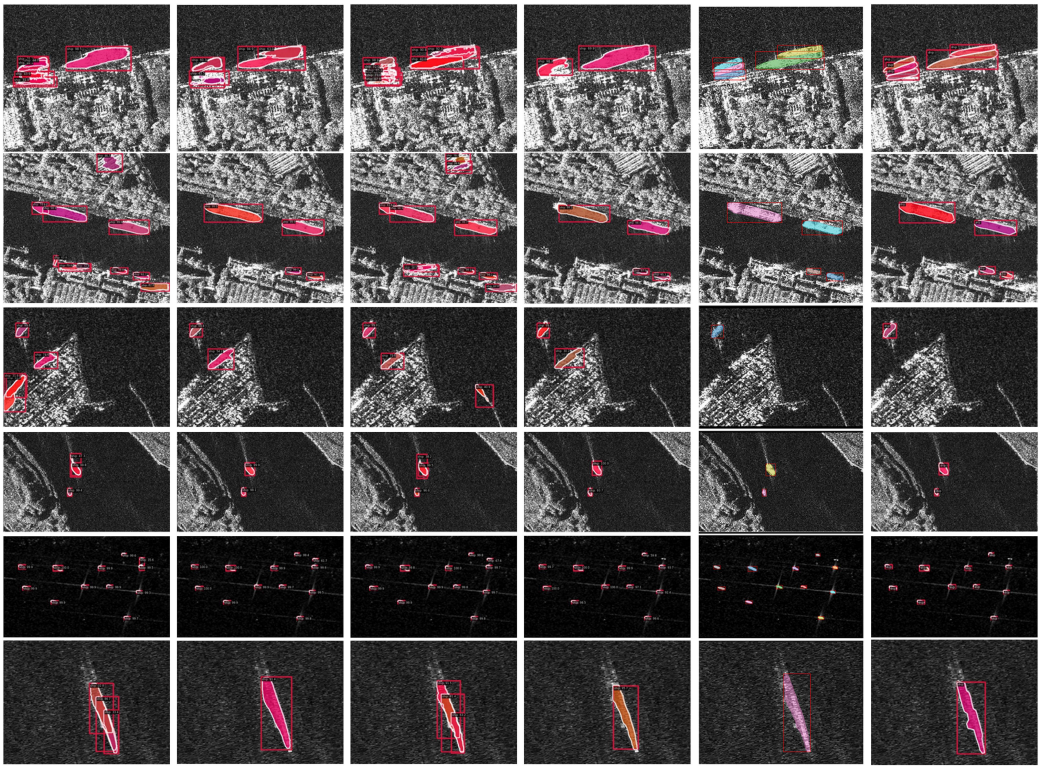

Fig. 7. Visualization results on SSDD.

TABLE X
EXPERIMENTS ON MODULE HYPERPARAMETERS

Module	Weight	AP	AP_{50}	AP_{75}
IDR	0.5	67.4	91.5	75.8
	1.0	67.7	91.8	75.5
	2.0	67.4	92.8	73.7
CDF	0.5	66.4	90.4	71.5
	1.0	67.7	91.8	75.5
	2.0	66.9	90.5	73.5

TABLE XI
EXPERIMENTS ON HYPERPARAMETERS OF DIFFERENT TASKS

$\lambda_1 : \lambda_2 : \lambda_3$	AP	AP_{50}	AP_{75}
1:2:1	65.8	90.7	71.0
2:1:2	67.4	92.4	74.8
2:1:1	66.1	91.3	72.3
1:2:2	67.2	91.8	74.7
1:1:1	67.7	91.8	75.5

weight of the contour branch is fixed as 1.0, and different weight settings for IDR and CDF modules are tried from 0.5 to 2.0. For IDR module, DCTC obtains 67.4, 67.7, and 67.4 AP with weight of 0.5, 1.0, and 2.0. For CDF module, DCTC obtains 66.4, 67.7, and 66.9 AP with weight of 0.5, 1.0, and 2.0. Due to their optimal performance, a weight of 1.0 for the IDR module and 1.0 for the CDF module is finally chosen. Table X gives more details.

As a multitask model, the weight ratio of different tasks may also influence the performance of the model. Table XI gives the experiment result. Different weight ratios for detection task, polygon regression task, and DCT vector regression task have been experienced. $\lambda_1 : \lambda_2 : \lambda_3 = 1 : 1 : 1$ is selected for its best performance.

E. Visualization

The visualization results of DCTC are presented on NWPU VHR-10 dataset and SSDD in order to exhibit the performance advantages of DCTC intuitively. Results show that DCTC obtains better visual effect on both datasets.

1) NWPU VHR-10: Fig. 6 presents the experiment results on NWPU VHR-10 dataset. Row 1 reveals that other methods tend to confuse background and real objects, which demonstrates that RSI has complex scenes. DCTC, on the other hand, do well in distinguishing the fake object. Rows 2–4 showcase that DCTC produces better object contours, while other methods perform bad when handling boundaries. Rows 5 and 6 prove the ability to handle tense and small targets of DCTC. Comparing across different columns, DCTC improves mask precision significantly, and greatly enhances segmentation accuracy of boundaries, especially for small targets.

2) SSDD: The experiment results on SSDD are presented in Fig. 7. Visual inspection reveals that most methods perform well when predicting noncrowded offshore ships. However, due to the inshore ships are extremely similar to the context, other methods have higher false alarm rates, as showed in Rows 1–3. Meanwhile, due to the characteristics of SAR images, other methods tend to truncate large objects, while DCTC predicts large objects as a whole. Rows 5 and 6 showcase this situation. In brief, DCTC has lower false alarm rate and misses less objects, while segments more complete boundaries.

F. Discussion

Although DCTC achieves good balance between speed and accuracy, there are still some mispredictions or missed detections during inference. Most missed detections are due to the similarity between targets and background and the ambiguity of target boundaries. Bad cases of segmentation are generated mostly because the complex and nonconvex outlines. Besides, all methods perform unsatisfactory on iSAID dataset, we assume that it is due to the large number of targets in each image and large-scale variation of same class, which increase the difficulty of detection as well as small targets segmentation.

V. Conclusion

The primary objective of this work is to design a faster and more efficient instance segmentation framework for RSI. Starting from the challenges of the RSI, we optimize contour-based method and design a DCT encoded mask branch including three modules to enhance the detail learning ability for DCTC. Numerous experiments have demonstrated that modules we designed are fit for contour-based method and DCTC performs well in RSI, since the segmentation performance improved significantly, especially for small objects. Meanwhile, DCTC dramatically increases the inference speed. The model is validated on four popular datasets: NWPU VHR-10, iSAID, SSDD, and HRSID datasets. Quantitative and qualitative analyses prove that DCTC obtains high segmentation quality as well as high inference speed. Moreover, experiments showcase that the branch we designed can be easily added to any contour-based method and improve segmentation performance. However, the performance of DCTC on large datasets is not very satisfactory, to solve this problem, the fusion of features with different scales or the spacewise attention may be good ideas since they have been used in some instance segmentation methods for RSI and achieved good results. Moreover, the enhancement of low-level features is also important for RSI instance segmentation, because the scale of target is often small. The fusion strategy of DCT branch and contour branch can be further designed in DCTC to obtain better performance. We are confident that this problem will be solved in the near future.

References

[1] Y. Zhang et al., "RoI fusion strategy with self-attention mechanism for object detection in remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5990–6006, 2023.

[2] T. Zhang et al., "Object-centric masked image modeling based self-supervised pretraining for remote sensing object detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5013–5025, 2023.

[3] C. Han, C. Wu, H. Guo, M. Hu, J. Li, and H. Chen, "Change guiding network: Incorporating change prior to guide change detection in remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 8395–8407, 2023.

[4] Y. Deng et al., "Feature-guided multitask change detection network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9667–9679, 2022.

[5] W. Ye, W. Zhang, W. Lei, W. Zhang, X. Chen, and Y. Wang, "Remote sensing image instance segmentation network with transformer and multi-scale feature representation," *Expert Syst. Appl.*, vol. 234, 2023, Art. no. 121007.

[6] S. Wei, X. Zeng, H. Zhang, Z. Zhou, J. Shi, and X. Zhang, "LFG-Net: Low-level feature guided network for precise ship instance segmentation in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5231017.

[7] H. Su et al., "HQ-ISNet: High-quality instance segmentation for remote sensing imagery," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 989.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[10] F. Shi and T. Zhang, "An anchor-free network with box refinement and saliency supplement for instance segmentation in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6516205.

[11] L. Gong, X. Huang, J. Chen, M. Xiao, and Y. Chao, "Multiscale leapfrog structure: An efficient object detector architecture designed for unmanned aerial vehicles," *Eng. Appl. Artif. Intell.*, vol. 127, 2024, Art. no. 107270.

[12] J. Wang, S. Ji, and T. Zhang, "From image transfer to object transfer: Cross-domain instance segmentation based on center point feature alignment," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4407011.

[13] H. Su, P. Huang, J. Yin, and X. Zhang, "Faster and better instance segmentation for large scene remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 2187–2190.

[14] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, "Deep snake for real-time instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8533–8542.

[15] T. Zhang, S. Wei, and S. Ji, "E2EC: An end-to-end contour-based method for high-quality high-speed instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4443–4452.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[17] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[18] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4974–4983.

[19] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9157–9166.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[21] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 8573–8581.

[22] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[23] E. Xie et al., "PolarMask: Single shot instance segmentation with polar representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12193–12202.

[24] H. U. M. Riaz, N. Benbarka, and A. Zell, "FourierNet: Compact mask representation for instance segmentation using differentiable shape decoders," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 7833–7840.

[25] Z. Liu, J. H. Liew, X. Chen, and J. Feng, "DANCE: A deep attentive contour model for efficient instance segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 345–354.

[26] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[27] Y. Fang et al., "Instances as queries," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6910–6919.

[28] B. Dong, F. Zeng, T. Wang, X. Zhang, and Y. Wei, "SOLQ: Segmenting objects by learning queries," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 21898–21909.

[29] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.

[30] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1290–1299.

[31] Z. Gu, H. Chen, Z. Xu, J. Lan, C. Meng, and W. Wang, "DiffusionInst: Diffusion model for instance segmentation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 2730–2734.

[32] S. Chen, P. Sun, Y. Song, and P. Luo, "DiffusionDet: Diffusion model for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19830–19843.

[33] S. Chen, Y. Ogawa, C. Zhao, and Y. Sekimoto, "Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach," *ISPRS J. Photogrammetry Remote Sens.*, vol. 195, pp. 129–152, 2023.
[34] J. Peng, X. Sun, H. Yu, Y. Tian, C. Deng, and F. Yao, "An instance-based multitask graph network for complex facility recognition in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5615015.
[35] Z. Wang, S. Zhang, C. Zhang, and B. Wang, "RPFNet: Recurrent pyramid frequency feature fusion network for instance segmentation in side-scan sonar images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, early access, Apr. 11, 2023, doi: 10.1109/JSTARS.2023.3266383.
[36] Z. Huang and R. Li, "Orientated silhouette matching for single-shot ship instance segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 463–477, 2021.
[37] R. Zhang, Z. Tian, C. Shen, M. You, and Y. Yan, "Mask encoding for single shot instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10226–10235.
[38] X. Shen et al., "DCT-Mask: Discrete cosine transform mask representation for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8720–8729.
[39] Q. Wen, J. Yang, X. Yang, and K. Liang, "PatchDCT: Patch refinement for high quality instance segmentation," in *Proc. 11th Int. Conf. Learn. Representations*, 2022.
[40] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.
[41] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[43] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogrammetry Remote Sens.*, vol. 98, pp. 119–132, 2014.
[44] S. W. Zamir et al., "iSAID: A large-scale dataset for instance segmentation in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 28–37.
[45] T. Zhang et al., "SAR ship detection dataset (SSDD): Official release and comprehensive data analysis," *Remote Sens.*, vol. 13, no. 18, 2021, Art. no. 3690.
[46] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi, "HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation," *IEEE Access*, vol. 8, pp. 120234–120254, 2020.
[47] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9799–9808.
[48] X. Zeng, S. Wei, J. Shi, and X. Zhang, "A lightweight adaptive RoI extraction network for precise aerial image instance segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5018617.
[49] D. Kumar, "Accurate object detection & instance segmentation of remote sensing, imagery using cascade mask R-CNN with HRNet backbone," *Authorea Preprints*, 2023.
[50] T. Zhang, X. Zhang, J. Li, and J. Shi, "Contextual squeeze-and-excitation mask R-CNN for SAR ship instance segmentation," in *Proc. IEEE Radar Conf.*, 2022, pp. 1–6.
[51] T. Zhang and X. Zhang, "Enhanced mask interaction network for SAR ship instance segmentation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 3508–3511.
[52] T. Zhang and X. Zhang, "A full-level context squeeze-and-excitation ROI extractor for SAR ship instance segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4506705.
[53] T. Zhang and X. Zhang, "A mask attention interaction and scale enhancement network for SAR ship instance segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4511005.
[54] F. Gao, Y. Huo, J. Wang, A. Hussain, and H. Zhou, "Anchor-free SAR ship instance segmentation with centroid-distance based loss," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11352–11371, 2021.
[55] W. Li et al., "Box2Mask: Box-supervised instance segmentation via level-set evolution," 2022, *arXiv:2212.01579*.
[56] Y. Luo, J. Han, Z. Liu, M. Wang, and G.-S. Xia, "An elliptic centerness for object instance segmentation in aerial images," *J. Remote Sens.*, vol. 2022, 2022, Art. no. 9809505.
[57] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 649–665.
[58] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6409–6418.
[59] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

Zhong Chen received the Ph.D. degree in cartography and geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2006.

He is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include computer vision and deep learning, with a focus on remote sensing image processing.

Tianhang Liu received the B.S. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the M.S. degree in control engineering.

His research interests include instance segmentation in remote sensing applications and time series forecasting.

Xueru Xu (Student Member, IEEE) received the B.S. and M.S. degrees in safety science and engineering from Chongqing University, Chongqing, China, in 2016 and 2019, respectively. She is currently working toward the Ph.D. degree in control science and engineering with the Huazhong University of Science and Technology, Wuhan, China.

Her research interests include object detection and fine-grained recognition in remote sensing applications.

Junsong Leng received the B.S. degree in automation and the M.S. degree in mechanical engineering from Huazhong Agricultural University, Wuhan, China, in 2019 and 2022, respectively. He is currently working toward the Ph.D. degree in control science and engineering with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan.

In 2022, he joined the State Key Laboratory of Multispectral Information Processing Technology, Huazhong University of Science and Technology. His current research interests include computer vision, domain adaptation, and remote sensing image processing.

Zhenxue Chen received the Ph.D. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, in 2007.

He is currently a Professor with the School of Control Science and Engineering, Shandong University, Jinan, China. His research interests include computer vision and deep learning, image processing, biometrics, and information fusion.