Look Twice and Closer: A Coarse-to-Fine Segmentation Network for Small Objects in Remote Sensing Images

Silin Chen, Qingzhong Wang, Kangjian Di, Haoyi Xiong, *Senior Member, IEEE*, and Ningmu Zou

***Abstract*—Convolutional neural networks (CNNs) are frequently used to analyze remote sensing images and achieve impressive progress. Limited by the receptive field size of CNNs, small objects tended to lack adequate features to obtain more accurate segmentation results. To address this problem, we introduce a novel CNN model for coarse-to-fine segmentation called C2FNet. C2FNet comprises two stages: the coarse network and the fine network. The coarse network identifies the positions and coarse segmentation outcomes of small objects in the input image. The fine network then takes a closer look at the small objects and re-segments the patches using binary segmentation. The fine network distinguishes small objects from the background to refine small object segmentation. Finally, C2FNet employs an aggregation module that merges the binary segmentation maps and coarse outcomes to obtain accurate small object segmentation. We conducted extensive experiments on three widely accepted datasets for remote sensing image segmentation, namely the ISPRS 2-D semantic labeling Potsdam, Vaihingen, and iSAID. Our approach significantly improves the performance of baseline models, achieving a 0.24%–2.83% increase in IoU per small object class on iSAID.**

***Index Terms*—Deep learning, remote sensing images, semantic segmentation, small objects.**

I. Introduction

SEMANTIC segmentation assigns each pixel to a specific category and predicts a semantic mask, which is a fundamental task. To understand more topographic information of objects, semantic segmentation is widely applied to remote sensing images and contributes to land cover mapping, disaster prediction, urban planning [1], [2].

With the rapid development of deep learning, the performance of convolutional neural networks (CNNs) achieved great progress on segmentation in remote sensing images [3], [4], [5]. Fully Convolutional Network (FCN) [6], DeepLab series [7] and PSPNet [8] are three popular CNN-based models for common object segmentation. Recent works have demonstrated the use of Transformer in segmentation, with models such as SegFormer [9] replacing traditional CNN-based feature extraction to generate hierarchical features. Remote sensing images are different from natural images due to their perspective and spectrum. BSNet [4] and HMANet [3] are specialized for remote sensing image segmentation, introducing boundary information and global correlations respectively. However, many objects in remote sensing images only have a few pixels (e.g., vehicle, helicopter, and ship). To some extent, using down-sampling in CNNs loses the information of small objects [10]. CEN [11] enhances the contextual features to distinguish small objects. While in OLCN [12], a low coupling robust regression module was proposed to improve the positional accuracy, alleviating missing small objects. FactSeg [5] employs a foreground activation branch to perform small object mining and a semantic segmentation branch to refine small object segmentation. DRENet [13] designs a degradation reconstruction branch to reconstruct a small-object-aware, blurry version of the input image in the training phase, enhancing the representation of small objects. CIL [14] proposes a twin-auxiliary model, which introduces an auxiliary binary classification task to improve the accuracy of small object segmentation. MTUNet [15] is a multi-task framework for infrared small object detection and segmentation, and the advantages of MTUNet are low computational complexity and high inference speed. CF-Net [2] proposes a cross-fusion block to refine small object features by enlarging the receptive field of the low-level feature map. However, most of these approaches improve small object segmentation accuracy by increasing information or improving the feature representation of small objects, while in this paper, we propose a model that imitates human annotation procedure – first annotating large objects and localizing small objects, and then annotating small objects by zooming in the areas.

Traditional semantic segmentation methods feed the whole images into the model to distinguish the objects, which is limited by the size of the receptive field resulting in inaccurate segmentation of tiny objects. Considering human vision behaviors, when observing tiny objects in remote sensing images, we generally first localize the tiny objects, and then zoom in on the areas of

Received 5 November 2024; revised 31 January 2025; accepted 4 February 2025. Date of publication 10 February 2025; date of current version 20 February 2025. The work of Silin Chen was supported by Baidu reseach under the mentorship of Qingzhong Wang and Haoyi Xiong. The work of Qingzhong Wang and Haoyi Xiong was supported by the National Key R&D Program of China under Grant 2021ZD0110303. The associate editor coordinating the review of this article and approving it for publication was Prof. Yongjie Li. *(Silin Chen and Qingzhong Wang contributed equally to this work.) (Corresponding authors: Haoyi Xiong; Ningmu Zou.)*

Silin Chen is with the School of Integrated Circuits, Nanjing University, Suzhou 210093, China, and also with Baidu Inc., Beijing 100193, China (e-mail: silin.chen@smail.nju.edu.cn).

Qingzhong Wang and Haoyi Xiong are with Baidu Inc., Beijing 100193, China (e-mail: qingzwang@outlook.com; haoyi.xiong.fr@ieee.org).

Kangjian Di is with the School of Integrated Circuits, Nanjing University, Suzhou 210093, China (e-mail: kangjiandi@smail.nju.edu.cn).

Ningmu Zou is with the School of Integrated Circuits, Nanjing University, Suzhou 210093, China, and also with the Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou 210093, China (e-mail: nzou@nju.edu.cn).

Our code is available at https://github.com/PaddlePaddle/PaddleRS/.

Digital Object Identifier 10.1109/LSP.2025.3540374

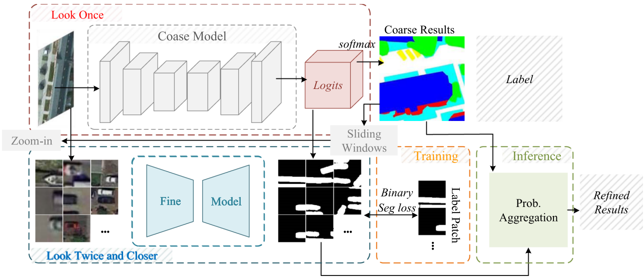

Fig. 1. The structure of our proposed Coarse-to-fine Model. Coarse-to-fine Model contains two stages: Look Once stage and Look Twice and Closer stage.

tiny objects to see more details, such as the boundaries of tiny objects. Motivated by this, we propose a novel coarse-to-fine segmentation network (**C2FNet**) on small objects in remote sensing images. First, we train a coarse segmentation network to take the first look at the entire image in a global view. To localize the small objects in remote sensing images, we then employ a sliding widow on the coarse segmentation map to obtain the local statistics, e.g., the number of pixels that belong to small object categories. In the second look, we feed the patches which contain small objects into the fine segmentation network and apply binary segmentation to train it; hence, small objects are refined. Finally, we aggregate the class-agnostic segmentation maps and the coarse maps to obtain fine-grained segmentation, and more details can be found in Section II.

II. Methodologies

As shown in Fig. 1, we propose a *Coarse-to-Fine Network* (**C2FNet**) for semantic segmentation of small objects in remote sensing images. The pipeline of **C2FNet** consists of three main processes: (1) the coarse semantic segmentation model obtains the positions and *coarse segmentation maps* of small objects from input images; (2) the fine model obtains the *binary segmentation maps* of small objects from the image patches with small objects; and (3) probabilistic aggregation combines the coarse segmentation and binary segmentation maps, yielding find-grained segmentation for small objects.

A. The Coarse Model and Small Object Localization

The coarse model is to provide a segmentation map considering the entire image in a global view. Though the segmentation of small objects is coarse, we can use the segmentation map to localize small objects and then refine them. In this paper, we employ a fully convolutional network (FCN) with HRNet-W18 [16] as the coarse model, since HRNet improves the resolution of feature maps and enhances the extracted feature of multi-scale objects. Note that the proposed plug-and-play **C2FNet** can use any segmentation network as the coarse model. Once we finish training the coarse model, we fix the parameters in the following training and inference phases.

In the look once stage, given an image $\mathrm{I} \in \mathbb{R}^{3\times H\times W}$, where H and W represent the height and width of the input image, we obtain a coarse segmentation map $\mathrm{X^c} \in \mathbb{R}^{N\times H\times W}$ using the well-trained coarse model, where N represents the number of categories. To localize the small objects, we apply a sliding window on the coarse segmentation maps. Specifically, a square window with the size of $k \times k$ slides on the segmentation map X^c, and then the statistics in each position are calculated. In this paper, we use the number of pixels that belong to a specific category in the window to localize small objects. After scanning the coarse segmentation map using a sliding window, we can obtain a set of image patches that contain small objects. Empirically, in high-quality samples, small objects are located in the center of the image patches. Therefore, the stride s of the sliding window should be as small as possible to get sufficient patches. However, a large number of patches requires more time and computational resources during training; hence we need to balance the performance and the number of samples. Alternatively, **C2FNet** sorts the patches based on the number of small object pixels and takes the top n image patches $[p_1, p_2, p_3, \ldots, p_n] \in \mathbb{R}^{3\times k\times k}$ as the final input of the next stage.

B. The Fine Model and Small Object Refinement

In the look-once stage, we obtain the image patches of the small objects that need to be refined, and we can also obtain the ground-truth labels of these patches. In the look twice stage, the fine model takes the image patches as input, referring to a second and closer look. Since we have obtained a coarse segmentation map in the look once stage and localized the small objects in the input images, we can only apply binary segmentation to distinguish small objects from others in this stage. Undoubtedly, any segmentation model can be employed to accomplish this task by simply modifying output classes to 2. In this paper, we use FCN in this stage.

In the training process, **C2FNet** up-samples the small object areas collected in the first stage to obtain more detailed information with reference to human behavior. Given the image patches $[p_1, p_2, p_3, \ldots, p_n]$, **C2FNet** first generates binary mask labels $[y_1, y_2, y_3, \ldots, y_n]$ based on the original ground-truth labels. Each binary label y_i is obtained by an indicator function $\mathbb{1}^c_{(x,y)}$, where $\mathbb{1}^c_{(x,y)} = 1$ if the original label of pixel (x, y) indicates that it belongs to a small object;[1] otherwise, $\mathbb{1}^c_{(x,y)} = 0$.

C. Probabilistic Aggregation

As mentioned above, **C2FNet** firstly obtains the image patches and coarse probability maps X^c of small objects using the coarse model and the softmax function. Then, these patches are fed into the fine model to get binary, fine-grained segmentation results $X^f \in \mathbb{R}^{k\times k}$. Specifically, **C2FNet** uses the two probability distribution maps characterizing segmentation from two distinct domains — X^c obtained by the coarse model reflects class-wise information, e.g., buildings, vehicles, etc., while X^f distinguishes the small objects from the background via binary segmentation.

Specifically, we denote the probability which the category is c at (x, y) in X^c as $p^c_{(x,y)}$. In X^f, $p^f_{(x,y)}$ represents the probability of a pixel that is foreground (small objects) and $1 - p^f_{(x,y)}$ is for the background. First, we consider whether there is a category of objects in the predefined small object list $C = [c_1, c_2, c_3, \ldots, c_x]$ based on X^c. In case there is no small object, *i.e.*, the category of each pixel (x, y) in an image patch is not in C, we return X^c directly. Otherwise, we will calculate a new pixel-wise probability $\tilde{p}^c_{(x,y)}$ for the images that contain

[1] The definition of small objects depends on the statistics of a dataset, which is prior knowledge.

TABLE I
ABLATION STUDY OF PATCH SIZE AND SLIDING WINDOWS' STRIDE ON THE ISAID DATASET (IOU PER CLASS/ F1 SCORE PER CLASS %)

(Size, Stride)	Ship	Large_Vehicle	Small_Vehicle	Helicopter	Swimming_Pool	Plane	Harbor
(32, 16)	69.43/81.96	63.39/77.59	51.21/67.73	24.40/39.22	45.05/62.12	83.43/90.96	60.30/75.23
(64, 32)	69.47/81.98	63.42/77.61	51.54/68.01	24.07/38.80	46.07/63.08	83.91/91.25	60.45/75.34
(128, 16)	69.50/82.00	63.58/77.74	**51.69/68.16**	24.45/39.30	**46.20/63.22**	84.02/91.29	60.47/75.35
(128, 32)	**69.51/82.03**	**63.60/77.75**	51.68/68.15	**24.47/39.31**	46.19/63.20	**84.04/91.32**	**60.55/75.43**
(128, 64)	69.48/81.99	63.52/77.69	51.56/68.04	24.11/38.85	46.18/63.18	84.04/91.32	60.47/75.36
(256, 64)	55.48/71.36	54.84/70.82	39.75/56.88	19.95/36.40	37.46/54.49	67.09/80.30	49.49/66.21

TABLE II
ABLATION STUDY ON ZOOM-IN

Zoom-in	SH	LV	SV	HC	SP	PL	HA	FPS
1×	69.51	63.60	51.68	24.47	**46.19**	84.04	**60.55**	**23.42**
2×	69.56	**63.74**	**52.06**	24.78	46.15	84.05	60.44	12.23
3×	**69.58**	63.73	51.99	**24.80**	46.17	**84.06**	60.51	8.22
4×	68.99	63.52	51.96	24.49	46.05	83.98	60.45	4.41

small objects using the coarse results X^c and fine-grained results X^f. Concretely, if the coarse model classifies the pixel at (x, y) into a non-small-object category, while the fine model treats the pixel as a small object, *i.e.*, $p^f_{(x,y)} \geq 0.5$, then we first find the maximum probability of the category in C at (x, y), *i.e.*, $\hat{p}^c_{(x,y)} = \mathbf{max}(X^c_{(x,y)} | c \in C)$ and then we use the summation of $\hat{p}^c_{(x,y)}$ and $p^f_{(x,y)}$. In contrast, if the coarse model classifies the pixel at (x, y) into a small object category and the fine model considers it as background, *i.e.*, $p^f_{(x,y)} < 0.5$, we use the summation of $1 - p^f_{(x,y)}$ and $p^c_{(x,y)}$. In the cases that there is no conflict between the coarse model and fine model, *i.e.*, the coarse model classifies a pixel as a small object and the fine model also treats it as foreground, or the coarse model classifies a pixel as a no-small-object category and the fine model treats it as background, we directly use $p^c_{(x,y)}$.

$$\tilde{p'}^{c}_{(x,y)} = \begin{cases} 1 - p^f_{(x,y)} + p^c_{(x,y)} & \text{case 1} \\ \hat{p}^c_{(x,y)} + p^f_{(x,y)} & \text{case 2} \\ p^c_{(x,y)} & \text{others} \end{cases} \quad (1)$$

where $\tilde{p'}^{c}_{(x,y)}$ denotes the summation of probability value, "case 1" means the condition when $p^f_{(x,y)} < 0.5$ and the coarse model classifies this pixel into a small object, "case 2" refers to the condition when $p^f_{(x,y)} \geq 0.5$ and the coarse model considers the pixel as a non-small-object category, and "others" refers to there is no small object in the patch or there is no conflict between the coarse model and fine model. Finally, the softmax function is employed to compute the aggregated probability value, denoted as $\tilde{p}^c_{(x,y)}$, which is calculated as $\tilde{p}^c_{(x,y)} = e^{\tilde{p'}^{c}_{(x,y)}} / \sum_{c_i \in C} e^{\tilde{p'}^{c_i}_{(x,y)}}$.

III. EXPERIMENTS AND ANALYSES

A. Datasets

The Potsdam contains 38 images, 24 images for training and a remaining 14 images for testing. The Vaihingen consists of 33 images, 16 images are considered as the training set and 17 images are used for the validation set. iSAID [24] contains 2806 images with 15 categories. We focus on the accuracy of the 7 small object categories: Plane (PL), Small vehicle (SV), Large vehicle (LV), Ship (SH), Harbor (HA), Swimming pool (SP) and Helicopter (HC) in our experiments. To improve the generalization performance of the proposed model, we implement a series of data augmentation techniques during the training phase. First, a multi-step scaling strategy is employed, with a maximum scaling factor of 2.0, a minimum scaling factor of 0.5, and an incremental step size of 0.25. Second, random cropping is applied to resize the input data to a fixed resolution of 512 × 512 pixels [3], [4], [20]. Additionally, random horizontal flipping and distortion are incorporated to further augment the dataset and mitigate the risk of model overfitting to the training data. During the testing phase, the model performance is evaluated using origin data to ensure an unbiased assessment of its generalization capability.

B. Implementation Details

All experiments in this paper are implemented based on the toolkit PaddleSeg.[2] For fair comparisons, all the models are trained using four Tesla V100-16 GB for 80000 iterations with a batch size of 16, and a learning rate of 0.01. The trained optimizer uses a stochastic gradient descent algorithm, configured with a weight decay of 4×10^{-5} and a momentum of 0.9. In our experiments, the fine model is an FCN with HRNetW18, selected for its capability to effectively preserve high-resolution feature representations while maintaining an optimal balance between model complexity and performance. The model is trained on the datasets for 80000 iterations using the full categories available in the dataset.

C. Ablation Studies

1) Patch Size and Sliding Stride: As shown in Table I, we set the initial size to 32 and the stride to 16. Experimental results indicate that excessively small patch sizes negatively impact segmentation performance due to insufficient contextual information. Additionally, while reducing the stride generally improves segmentation granularity, a small stride can result in model overfitting. Our experiments demonstrate that optimal performance is achieved with a patch size of 128 and a stride of 32.

2) The Number of Patches: Fig. 2 presents the impact of the number of patches. We can see that using 32 patches is marginally better than other numbers.

3) Zoom-In Factor: In Table II, we present the comparison of using different zoom-in factors. A larger zoom-in factor may slightly enhance performance but also increases input patch sizes for the fine model, resulting in higher computational costs. Furthermore, higher zoom-in factors introduce a progressive bias in the interpolated labels, which generates erroneous pixel labels. As the zoom factor increases, these errors become more pronounced, ultimately degrading the overall performance of the model.

[2] https://github.com/PaddlePaddle/PaddleSeg

TABLE III
COMPARISONS WITH STATE-OF-THE-ART METHODS ON iSAID (IOU PER CLASS %)

General Model	SH	LV	SV	HC	SP	PL	HA	FPS
UNet [10]	48.68	53.83	36.54	0.00	38.40	75.60	47.38	46.94
OCRNet [17]	52.47	51.53	43.63	0.05	44.09	72.23	46.07	43.33
PSPNet [8]	66.74	62.18	46.35	32.59	47.17	81.87	54.72	33.30
DeeplabV3+ [18]	64.49	61.68	45.77	33.35	49.58	81.65	53.41	24.64
UperNet [19]	67.51	63.69	48.59	35.85	53.34	83.67	57.72	42.59
Remote sensing Model								
HMANet [3]	65.38	59.74	50.28	32.58	51.41	83.79	51.91	-
BSNet [4]	65.30	63.40	46.62	31.80	48.81	81.83	57.30	-
FactSeg [5]	68.34	62.65	49.53	**42.72**	51.47	84.13	55.74	13.99
FarSeg++ [20]	71.70	65.90	53.60	42.70	**53.60**	86.50	62.00	17.40
SatSynth [21]	63.88	62.91	44.70	27.63	50.42	82.75	59.71	-
FCN_HRNetW18 [6]	69.04	62.61	48.75	23.14	44.99	83.35	58.61	32.10
FCN_HRNetW18 with ours	69.51(+0.47)	63.60(+0.99)	51.68(+2.93)	24.47(+1.33)	46.19(+1.20)	84.04(+0.69)	60.55(+1.94)	23.42
HRNetW48 [22]	73.80	66.61	54.27	38.17	52.19	85.51	62.25	29.80
HRNetW48 with ours	**74.32(+0.52)**	**67.56(+0.95)**	**56.46(+2.19)**	38.89(+0.72)	52.78(+0.59)	**86.75(+0.24)**	**63.70(+1.45)**	19.98

TABLE IV
THE FLEXIBILITY OF OUR PROPOSED MODEL

Model	SH	LV	SV	HC	SP	PL	HA
UNet [10]	48.68	53.83	36.54	0.00	38.40	75.60	47.38
UNet with ours	50.54(+1.86)	55.28(+1.54)	42.46(+5.92)	0.00(+0.00)	42.00(+3.60)	77.63(+2.03)	50.02(+2.64)
PSPNet [8]	66.74	62.18	46.35	32.59	47.17	81.87	54.72
PSPNet with ours	67.92(+1.18)	63.94(+1.76)	50.63(+4.28)	33.30(+0.71)	48.85(+1.68)	83.72(+1.85)	56.97(+2.25)
DeeplabV3+ [18]	64.49	61.68	45.77	33.35	49.58	81.65	53.41
DeepLabV3+ with ours	70.37(+5.88)	65.50(+3.82)	51.73(+5.96)	39.04(+5.69)	48.19(-1.39)	84.83(+1.11)	58.13(+4.72)
UperNet [19]	67.51	63.69	48.59	35.85	53.34	83.67	57.72
UperNet with ours	68.57(+1.06)	64.96(+1.27)	51.75(+3.16)	35.92(+0.07)	55.36(+2.02)	83.93(+0.25)	59.14(+1.42)

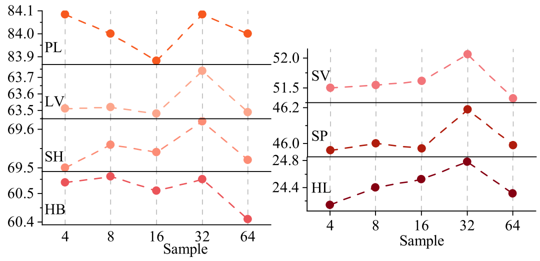

Fig. 2. Ablation Study of the number of the sample on iSAID dataset.

TABLE V
COMPARISONS WITH STATE-OF-THE-ART METHODS ON ISPRS POTSDAM AND VAIHINGEN (IOU PER CLASS%)

	ISPRS Potsdam		ISPRS Vaihingen	
General Model	Car IoU	Car F1 score	Car IoU	Car F1 Score
UNet [10]	88.02	93.62	61.05	75.81
PPLiteSeg [23]	89.08	94.22	67.05	80.27
FCN [6]	90.64	95.08	74.58	85.43
Segformer [9]	90.83	95.19	**75.49**	**85.99**
FCN with ours	**91.21**	**95.40**	75.10	85.78

D. Comparisons With State-of-The-Arts

As shown in Table III, compared with general models, the proposed method shows superiority, e.g., `UperNet` achieves 48.59% IoU on small vehicle (SV), while `HRNetW48 with ours` achieves 56.46% IoU, roughly 8% higher. In addition, the proposed method is superior to existing state-of-the-art remote sensing segmentation models. The refinement stage is added in **C2FNet** compared to the baselines, so the inference speed is slower. The FPS is reduced by 8.68 and 9.82 on `FCN` and `HRNetW48`, which is still nearly real-time.

Table V shows the performance on the ISPRS Potsdam and Vaihingen datasets, where we consider the category "car" as the small object. Compared with CNN-based models, the proposed method achieves better performance on the two datasets. We also list the performance on an advanced transformer-based model — `Segformer`. Our proposed method that is based on FCN achieves competitive performance compared with `Segformer`.

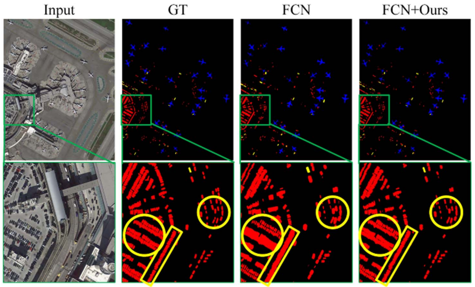

Fig. 3. Visualization of the segmentation maps.

In Tabel IV, we present the performance of the proposed models with different baselines, showing the flexibility of the proposed framework. To further demonstrate the superiority of the proposed model, we visualize the segmentation maps predicted by different models in Fig. 3.

IV. CONCLUSION

In this paper, we proposed a two-stage segmentation network for small object segmentation in remote sensing images. A coarse segmentation model is adopted to accurately segment large objects, as well as to find the locations of small objects. Then we look twice and closer; *i.e.*, a fine network is designed to segment the small objects in the image patches obtained from the coarse segmentation map. A probabilistic aggregation method is used to get the final results. Our proposed method achieves start-of-the-art results on small objects in the iSAID and ISPRS datasets. In the future, we plan to investigate more elegant ways to locate small objects to reduce the cost of the coarse segmentation process.

References

[1] X. He et al., "Semantic segmentation of remote-sensing images based on multiscale feature fusion and attention refinement," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8007105.

[2] C. Peng, K. Zhang, Y. Ma, and J. Ma, "Cross fusion net: A fast semantic segmentation network for small-scale semantic information capturing in aerial scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5601313.

[3] R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu, "Hybrid multiple attention network for semantic segmentation in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5603018.

[4] J. Hou, Z. Guo, Y. Wu, W. Diao, and T. Xu, "BSNet: Dynamic hybrid gradient convolution based boundary-sensitive network for remote sensing image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5624022.

[5] A. Ma, J. Wang, Y. Zhong, and Z. Zheng, "FactSeg: Foreground activation-driven small object semantic segmentation in large-scale remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5606216.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[9] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *NeurIPS*, vol. 34, pp. 12077–12090, 2021.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, Springer, 2015, pp. 234–241.

[11] Y. Chong, X. Chen, and S. Pan, "Context union edge network for semantic segmentation of small-scale objects in very high resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6000305.

[12] Y. Yuan and Y. Zhang, "OLCN: An optimized low coupling network for small objects detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8022005.

[13] J. Chen, K. Chen, H. Chen, Z. Zou, and Z. Shi, "A degraded reconstruction enhancement-based method for tiny ship detection in remote sensing images with a new large-scale dataset," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5625014.

[14] J. Li et al., "Class-incremental learning network for small objects enhancing of semantic segmentation in aerial imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5612920.

[15] Y. Chen, L. Li, X. Liu, and X. Su, "A multi-task framework for infrared small target detection and segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5003109.

[16] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[17] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 173–190.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[19] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

[20] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, "Farseg : Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13715–13729, Nov. 2023.

[21] A. Toker, M. Eisenberger, D. Cremers, and L. Leal-Taixé, "SatSynth: Augmenting image-mask pairs through diffusion models for aerial semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 27695–27705.

[22] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.

[23] J. Peng et al., "PP-Liteseg: A superior real-time semantic segmentation model," 2022, *arXiv:2204.02681*.

[24] 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.