SOD-YOLOv10: Small Object Detection in Remote Sensing Images Based on YOLOv10

Hui Sun, Guangzhen Yao, *Student Member, IEEE*, Sandong Zhu, Long Zhang, Hui Xu , and Jun Kong

Abstract—YOLOv10, known for its efficiency in object detection methods, quickly and accurately detects objects in images. However, when detecting small objects in remote sensing imagery, traditional algorithms often encounter challenges like background noise, missing information, and complex multiobject interactions, which can affect detection performance. To address these issues, we propose an enhanced algorithm for detecting small objects, named SOD-YOLOv10. We design the Multidimensional Information Interaction for the Transformer Backbone (TransBone) Network, which enhances global perception capabilities and effectively integrates both local and global information, thereby improving the detection of small object features. We also propose a feature fusion technology using an attention mechanism, called aggregated attention in a gated feature pyramid network (AA-GFPN). This technology uses an efficient feature aggregation network and re-parameterization techniques to optimize information interaction between feature maps of different scales. Additionally, by incorporating the aggregated attention (AA) mechanism, it accurately identifies essential features of small objects. Moreover, we propose the adaptive focal powerful IoU (AFP-IoU) loss function, which not only prevents excessive expansion of the anchor box area but also significantly accelerates model convergence. To evaluate our method, we conduct thorough tests on the RSOD, NWPU VHR-10, VisDrone2019, and AI-TOD datasets. The findings indicate that our SOD-YOLOv10 model attains 95.90%, 92.46%, 55.61%, and 59.47% for mAP@0.5 and 73.42%, 66.84%, 39.03%, and 42.67% for mAP@0.5:0.95.

Index Terms—Backbone, feature pyramid network, IoU, remote sensing images, small object detection, YOLOv10.

I. Introduction

With recent advances in the volume and quality of remote sensing imagery, object detection technology becomes central to automated analysis in this domain and is widely applied in both military and civilian fields such as precision agriculture, disaster relief, traffic monitoring, aerospace, urban planning, and geological environmental surveys. As deep neural network technology advances, object detection emerges as an important research area [1]. Deep neural network-based object detection methods are generally divided into two categories: two-stage and one-stage approaches. Two-stage networks like Faster R-CNN and CNN produce proposal regions before classification and localization, suitable for applications requiring high precision. Conversely, one-stage networks such as SSD and YOLO immediately produce classifications and location coordinates, offering rapid processing speeds suitable for real-time detection scenarios. Thus, in the domain of rapid object detection in remote sensing imagery, one-stage networks perform exceptionally well, with the YOLO series being extensively utilized for object detection.

Received 9 December 2024; revised 9 January 2025; accepted 23 January 2025. Date of publication 27 January 2025; date of current version 6 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272096, in part by Jilin Province Education Department under Grant JJKH20241729KJ, and in part by Changchun Humanities and Sciences College under Grant FZKY2024050. *(Hui Sun and Guangzhen Yao contributed equally to this work.) (Corresponding authors: Hui Xu; Jun Kong.)*

Hui Sun, Guangzhen Yao, Sandong Zhu, Long Zhang, and Jun Kong are with the School of Information Science and Technology, Northeast Normal University, Changchun 130000, China (e-mail: sunh333@nenu.edu.cn; yaoguangzhen@nenu.edu.cn; zhusandong@nenu.edu.cn; longzhang@nenu.edu.cn; kongjun@nenu.edu.cn).

Hui Xu is with the Smart Welfare Collaborative Research Center, Changchun Humanities and Sciences College, Changchun 130000, China (e-mail: xuh504@nenu.edu.cn).

Digital Object Identifier 10.1109/LGRS.2025.3534786

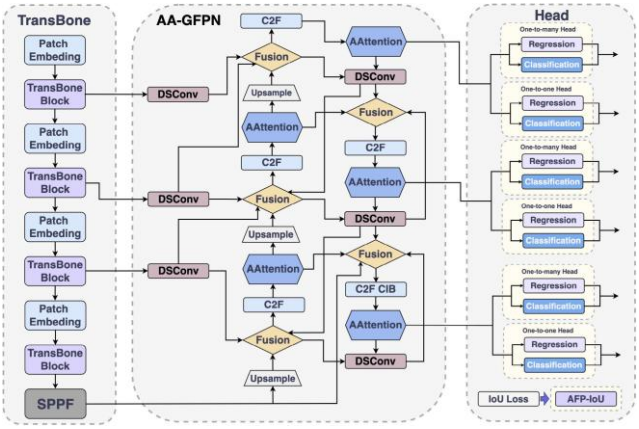

Fig. 1. Structure of SOD-YOLOv10.

Although YOLO algorithms are extensively applied in processing remote sensing images, their performance in detecting small objects against low-quality images, complex backgrounds, diverse object arrangements, and uncertain object orientations is often not ideal. To tackle these issues, as illustrated in Fig. 1, we propose an enhanced algorithm called SOD-YOLOv10, which integrates three core technologies: Transformer Backbone (TransBone) backbone network, aggregated attention in gated feature pyramid network (AA-GFPN) feature fusion technique, and adaptive focal powerful IoU (AFP-IoU) loss function, aimed to enhance the accuracy of detecting small objects in remote sensing imagery.

The primary contributions of this study include the following.

1) We propose a TransBone backbone network, which reorganizes certain channels to be distributed across the batch dimension and divides them into several subfeatures to achieve cross-dimensional interaction, thus capturing pixel-level details accurately. Moreover, the network adjusts the weights of the convolutional kernels, optimizing the integration of local and global information and significantly enhancing the model's capability

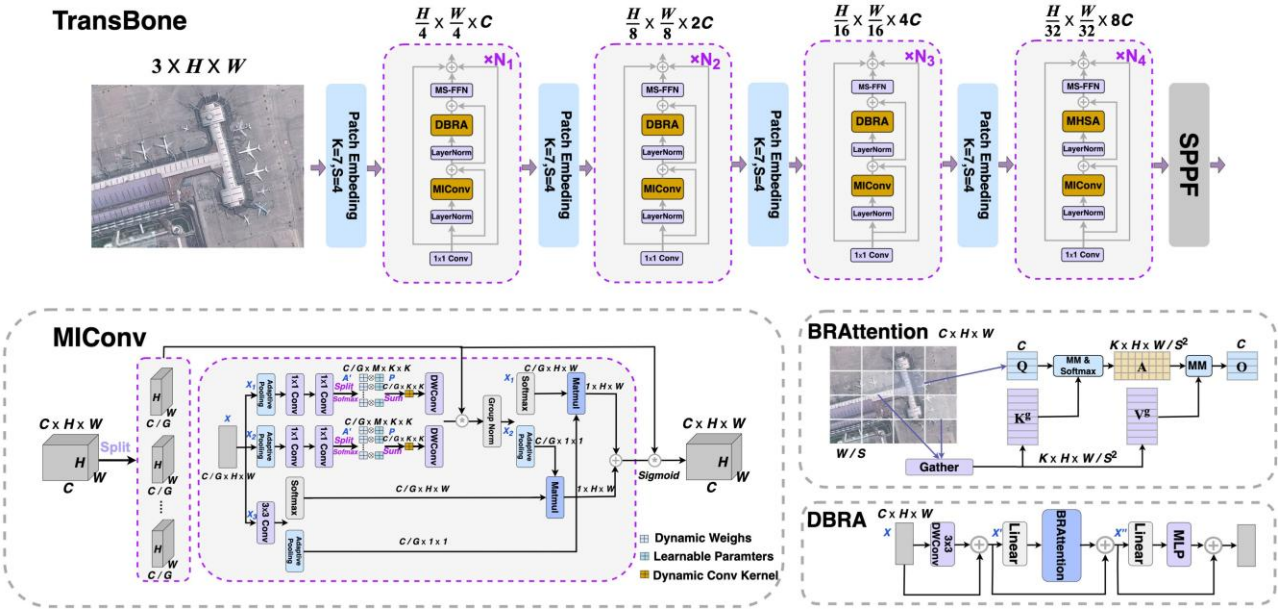

Fig. 2. TransBone backbone network.

to process visual information and extract features in complex situations.

2) We also propose the aggregated attention (AA)GFPN feature fusion technique. This method passes high-level semantic information through a top-down pathway, effectively integrating multilevel features, and significantly enhancing feature discrimination capability. Additionally, by incorporating the AA [2] mechanism, it further enhances the focusing ability on small object features.
3) Furthermore, we propose the AFP-IoU loss function, by introducing an adaptive penalty factor and gradient adjustment mechanism, the AFP-IoU not only effectively prevents excessive expansion of the anchor box area but also significantly speeds up the model's convergence.

II. Methodology

A. TransBone Backbone Network

1) Overall Framework: As illustrated in Fig. 2, our TransBone backbone network employs a four-stage hierarchical structure and an overlapping patch embedding strategy. During the initial stage, the input feature map $X \in R^{C\times H\times W}$ undergoes Patch Embedding processing, dividing the image into several patches of a fixed size. In the main processing loop, the initial features of these patches are first extracted through a 1 × 1 convolution layer and are then normalized using LayerNorm. Following this, the multidimensional information interaction conv (MIConv) module reshapes parts of the channels to the batch dimension to preserve information in each channel and adjust based on the varying characteristics of the input feature map, effectively combining local and global information. Subsequently, the depth perception bi-level routing attention (DBRA) module effectively focuses on key areas of the input features, gradually refining high-level characteristics through a hierarchical structure. Subsequently, the multistage feed-forward network (MS-FFN) processes the data layer by layer. At the last stage, the output feature maps are combined with those from earlier stages, efficiently merging feature information across different stages. This procedure can be performed N_1 times as needed to progressively enhance the feature representation of the data.

In the second and third stages, which are similar to the first stage, each stage progressively decreases the spatial dimensions of the feature maps while expanding the number of channels (e.g., the second stage is $(H/8) \times (W/8) \times 2C$ and the third stage is $(H/16)\times(W/16)\times 4C$). In these stages, the model continues to use the MIConv and DBRA modules to process and refine features. The second and third stages are repeated N_2 and N_3 times, respectively. By the fourth stage, as the dimensions of the feature maps have been reduced to $(H/32) \times (W/32) \times 8C$, traditional feature pooling modules are no longer effective. Therefore, a Multihead Self-Attention layer is added to capture different representations across various heads. This stage is repeated N_4 times to ultimately optimize the diversity and depth of feature representation.

2) Multidimensional Information Interaction Conv: In the MIConv module, by reshaping portions of the channels to the batch dimension and organizing the channel dimensions into multiple subfeatures, this strategy not only preserves the integrity of each channel's information but also significantly reduces computational costs. Additionally, the MIConv module dynamically adjusts the weights of convolution kernels, enabling adaptive adjustments based on the characteristics of the input feature map.

Specifically, the given 2-D input feature map is divided along the channel dimension into G subfeature groups, each of

which has dimensions $X \in R^{((C/G)\times H\times W)}$. Since neurons have a large local receptive field, they effectively gather local multiscale spatial information. Thus, MIConv extracts attention-weight descriptors from grouped feature maps using three concurrent routes, enhancing information extraction capabilities.

In the processing workflow, for the first two subfeature maps X_1 and X_2, spatial context is aggregated through an adaptive average pooling (AvgPool) operation, compressing the spatial dimensions to K^2 effectively encapsulate global information.

Subsequently, these feature maps pass through two successive 1 × 1 convolutional layers to produce the attention map A', enabling the network to dynamically concentrate on crucial regions in the input feature maps and efficiently integrate local information. A' is then reformed into $R^{((C/G)\times M\times K_2)}$, followed by the application of the softmax function across the (C/G) dimension to create attention weights $A \in R^{((C/G)\times M\times K_2)}$. Following this, A is element-wise multiplied by a set of learnable parameters $P \in R^{((C/G)\times K_2)}$, dynamically adjusting the various attributes of the input feature map and to facilitate the integration of both global and local information. Finally, the aggregated data undergoes processing through a depth-wise convolution (DWConv) and the results of these two feature maps are combined with the initial other feature groups through multiplication. This processed result is then normalized by a Group Norm layer. For the subfeature map X_3, it is also divided into two subpaths. The output of the first path is processed by the Softmax function, converted into the format $(C/G) \times H \times W$, successfully boosting the model's capacity to analyze spatial features. The output of the second path is converted through average pooling into the format (C/G)×1×1, focusing on capturing global contextual information.

Subsequently, the output of subfeature map X_1 is subjected to processing via the Softmax function and converted into the format (C/G)×H×W. This result is combined with the output from the second path of the subfeature map X_3 branch through matrix multiplication (Matmul), converting it to $1 \times H \times W$. This fusion of features from various scales greatly enhances the expressiveness of the features. The output of subfeature map X_2 is transformed into $(C/G) \times 1 \times 1$ through global average pooling and, after undergoing Matmul with the output from the first path of the subfeature map X_3 branch, is also converted to $1 \times H \times W$. Adding these two results further enhances the model's ability to integrate information, thereby enhancing its effectiveness.

Finally, this result is subjected to the Sigmoid function and then multiplied by the feature maps from the initial other feature groups to produce the ultimate output feature map.

3) Depth Perception Bi-Level Routing Attention: Due to the severe noise interference and intricate backgrounds in remote sensing imagery, we incorporate the BRA [3] mechanism into the backbone network. This mechanism initially excludes nonessential, large-area features at a broader regional level before honing in on specific tokens with fine granularity, actively choosing the most pertinent key–value pairs for each query. This method substantially conserves memory and computational resources while also boosting the capabilities to detect small objects. Additionally, through our designed hierarchical feature refinement method, the DBRA module effectively handles image tasks in complex scenarios, further enhancing the module's applicability and performance.

Particularly, the input feature X first passes through a 3 × 3 deep DWConv, and the processed output generates X' through a residual connection. Then, X' undergoes a linear transformation, is fed into the BRA mechanism, and then recombined with the original X' via a residual connection to generate X''. Subsequently, X'' is sent through another linear transformation to a multilayer perceptron (MLP) to further extract higher-level feature representations. The final output Y is produced through the residual connection of X''. This process, with continuous residual connections, optimizes gradient flow and information retention, enhancing the model's precision.

In the BRA mechanism, the 2-D input feature map $X \in R^{H\times W\times C}$ is first processed by segmenting it into $S \times S$ nonoverlapping areas, each with (HW/S^2) feature vectors. The feature map X is then reshaped into $X^r \in R^{S_2\times(HW/S_2)\times C}$.

Linear projections generate the query, key, and value tensors $Q, K, V \in R^{S_2\times(HW/S_2)\times C}$. Subsequently, to determine the attention relationships between regions, Q and K undergo regional averaging to generate regional-level queries and keys $Q^r, K^r \in R^{S_2\times C}$. The affinity adjacency matrix $A^r = Q^r(K^r)^T$ is computed by multiplying Q^r and the transpose of K^r.

Based on this, the region-to-region routing index matrix I^r is determined by identifying the top k regions with the highest similarity in the adjacency matrix A^r, using topkIndex(A^r).

After establishing the region-to-region routing index matrix I^r, detailed token-to-token attention is subsequently implemented. Initially, the key and value tensors are retrieved using the index I^r, specifically, V^g = gather(V, I^r) and K^g = gather(K, I^r). Following this, the local context enhancement (LCE) technique is employed, and attention is directed toward the retrieved key–value pairs, producing the result

$$O = \text{Attention}(Q, K^g, V^g) + \text{LCE}(V). \quad (1)$$

B. Structure of AA-GFPN

As depicted in Fig. 1, AA-GFPN optimizes the interaction between feature maps of various scales using a proficient feature aggregation network and reparameterization techniques, enhancing the model's performance to handle multiscale information and successfully integrating high-level semantic information with foundational spatial data. By omitting certain upsampling steps that cause significant delays, AA-GFPN substantially increases processing speed in real-time detection tasks.

In the AA-GFPN framework, we introduce the AA mechanism incorporated before upsampling, following each downsampling C2F module, and before the fusion of features. By applying the AA mechanism before upsampling, the system

can prioritize feature analysis early on, thus allowing for more detailed processing of small object data and markedly improving the model's localization and detection abilities. Additionally, the implementation of the AA mechanism following each C2F module during downsampling ensures that despite the reduction in feature complexity, the model maintains sensitivity to fine details and improves the identification of essential information. Specifically, by integrating the AA mechanism before feature fusion, the network can isolate important regions on a larger scale while concentrating on finer details on a smaller scale, ensuring that crucial image information is emphasized before the features are merged.

To improve small object detection accuracy, we employ the DSConv technique as a replacement for traditional convolution methods. This technique is composed of two primary elements. Depthwise Convolution applies filters independently to each input channel, effectively capturing more detailed spatial features, which is especially important for small objects with

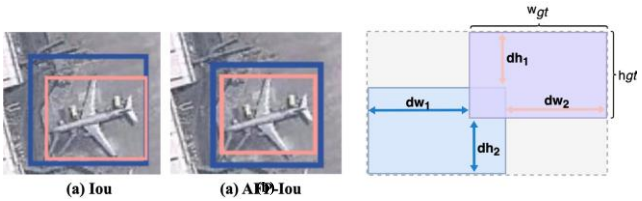

Fig. 3. Comparison of traditional (a) IoU and (b) AFP-IoU.

fewer pixels. Subsequently, Pointwise Convolution combines and reshapes the output of the Depthwise Convolution, not only integrating the captured features but also enhancing the network's nonlinearity. Additionally, DSConv lowers the model's parameter count and computational burden, increasing processing speed. This convolution technique not only improves the model's precision in detecting small objects but also optimizes computational effectiveness, ensuring the model sustains high performance and meets the requirements for realtime processing.

C. Adaptive Focal Powerful-IoU

Traditional IoU loss functions exhibit certain limitations when dealing with sample imbalance and the area inflation problem during anchor box regression, to address the aforementioned issues, we combine the advantages of FocalerIoU [4] and Powerful-IoU [5] and propose AFP-IoU. AFP-IoU aims to combine the sample focusing capability of Focaler-IoU with the anchor box quality adaptive adjustment mechanism of Powerful-IoU, forming a more efficient and robust IoU loss function. As shown in Fig. 3, compared to traditional IoU, AFP-IoU effectively addresses the issues of sample imbalance and anchor box quality differences by incorporating an adaptive adjustment mechanism and sample focusing capability. The formula for this is

$$L_{\text{AFP-IoU}} = 1 - \text{AFP-IoU} \tag{2}$$

the specific definition of AFP-IoU is

$$\text{AFP-IoU} = \begin{cases} 0, & \text{if IoU} < d \\ \dfrac{\text{IoU} - d}{u - d} - f(P), & \text{if } d \le \text{IoU} \le u \\ 1 - f(P), & \text{if IoU} > u \end{cases} \tag{3}$$

where the parameter d is the lower bound of IoU. When the IoU value is less than d, the AFP-IoU output is 0, indicating that the model considers the predicted box to be poorly matched with the target box. The parameter u is the upper bound of IoU. When the IoU value exceeds u, the AFP-IoU output is $1 - f(P)$, meaning the model considers the box to be a good match. We set d to 0.1 and u to 0.8 to penalize low IoU values and increase the attention on high IoU values. P is an adaptive penalty factor, reflecting the geometric alignment between the anchor box and the object box. It is defined as follows:

$$P = -\left(\frac{dw_2}{4 w_{gt}} + \frac{dw_1}{4 w_{gt}} + \frac{dh_2}{4 h_{gt}} + \frac{dh_1}{4 h_{gt}}\right) \tag{4}$$

where $dw_1 = |x_{\text{predleft}} - x_{gt\text{left}}|$, $dw_2 = |x_{\text{predright}} - x_{gt\text{right}}|$, and dh_1 and dh_2 represent the distances between the matching edges of the predicted box and the object box, respectively.

TABLE I

SETTINGS FOR MODEL TRAINING HYPERPARAMETERS

Hyperparameter Options	Setting
Input Resolution	640 × 640 × 3
Batch_size	4
Momentum	0.938
Epochs	200
Learning Rate Float	0.01
Initial Learning Rate	0.01

TABLE II

PERFORMANCE ON THE RSOD DATASETS

Model	Class	Playground	Overpass	Oil tank	Aircraft	Ave.
YOLOv10	F1	96.06	77.62	96.50	93.91	91.03
	AP	98.20	74.96	96.92	95.81	91.48
SOD-YOLOv10	F1	97.15	86.13	96.78	94.41	93.61
	AP	**98.97**	**89.04**	**98.58**	**97.03**	**95.90**

TABLE III

PERFORMANCE ON THE NWPU VHR-10 DATASET. S FOR STORAGE TANK, H FOR HARBOR, T FOR TENNIS COURT, V FOR VEHICLE, B FOR BASKETBALL COURT, A FOR AIRPLANE, BD FOR BASEBALL DIAMOND, SH FOR SHIP, GTF FOR GROUND TRACK FIELD, AND BR FOR BRIDGE

Model	Class	S	H	T	V	B	A	BD	Sh	GTF	Br	Ave.
YOLOv10	F1	88.62	89.93	89.11	77.73	81.03	90.42	93.46	97.63	71.42.	88.60	86.79
	AP	87.92	91.42	87.45	69.02	85.43	92.80	95.83	98.04	66.52	91.90	86.63
SOD-YOLOv10	F1	91.25	98.60	88.75	79.33	90.98	98.04	93.89	98.13	98.24	91.27	92.85
	AP	**90.08**	**95.77**	**91.43**	72.83	**90.64**	**99.47**	**96.72**	**99.21**	**96.47**	**92.04**	**93.17**

The width and height of the object box are denoted by w_{gt} and h_{gt}, respectively. $f(P)$ is an adaptive nonmonotonic attention function designed to modulate the focus on anchor boxes varying in quality. The formulation is as follows: $f(P) = 1 - e^{-P_2}$.

III. Experiments

A. Experimental Setup

Table I presents the hyperparameter settings, providing insights into the model optimization process. We meticulously select four datasets with unique characteristics and advantages—RSOD, NWPU VHR-10, VisDrone2019, and ATTOD—to ensure the research adequately reflects the variety of scenes that may be encountered in remote sensing imagery. The datasets are randomly divided into training, validation, and testing sets in an 8:1:1 ratio.

B. Classification Evaluation of SOD-YOLOv10

As shown in Tables II–V, we conduct extensive experiments on the RSOD, NWPU VHR-10, VisDrone2019, and AT-TOD datasets. The results demonstrate that the SOD-YOLOv10 model exhibits higher accuracy and better balance in Precision, Recall, *F*1 score, and AP metrics. These findings further validate the significant advantages of SOD-YOLOv10 in detecting small-sized and difficult-to-identify objects. As Fig. 4 illustrates, in the VisDrone2019 dataset, the YOLOv10 model fails to detect some small objects that are overlapped or obscured, such as pedestrians and cars, as depicted in Fig. 2(a) and (b). In contrast, Fig. 3(a) and (b) shows that the SOD-YOLOv10 model not only accurately identifies the main objects but also detects obscured small objects, especially excelling in scenarios with overlapping objects.

TABLE IV

Performance on the VisDrone2019 Dataset. AT for Awning Tricycle, Tr for Truck, Ty for Tricycle, B for Bus, M for Motor, P for Person, Bi for Bicycle, C for Car, Pe for Pedestrian, and V for Van

Model	Class	AT	Tr	Ty	B	M	P	Bi	C	Pe	V	Ave.
YOLOv10	F1	14.65	37.00	29.32	58.42	49.05	32.20	12.73	77.02	41.78	45.21	39.73
	AP	16.24	37.54	28.06	58.77	48.72	29.78	12.80	75.28	43.20	44.68	39.50
SOD-YOLOv10	F1	34.84	46.65	41.06	73.06	62.91	50.15	40.68	93.94	63.75	58.80	56.55
	AP	32.39	45.13	39.49	**71.05**	**62.77**	50.34	40.12	**92.58**	**62.83**	59.44	**55.61**

TABLE V

Performance on the AI-TOD Dataset. A for Aircraft, B for Bridge, ST for Storage Tank, Sh for Ship, SP for Swimming Pool, V for Vehicle, P for Person, and W for Windmill

Model	Class	A	B	St	Sh	SP	V	P	W	Ave.
YOLOv10	F1	72.84	33.52	75.81	70.43	29.42	72.10	34.80	19.67	51.07
	AP	70.67	32.85	74.63	70.67	28.98	70.01	34.56	18.04	50.05
SOD-YOLOv10	F1	77.90	48.23	79.52	79.42	51.87	74.83	41.70	30.64	60.53
	AP	**76.65**	47.24	**78.84**	**78.61**	50.60	**74.03**	40.32	29.51	**59.48**

TABLE VI

Comparison With Various Models on RSOD

Model	Params	FPS	P	R	F1	mAP@0.5	0.5:0.95
Faster R-CNN	42.47 M	31.73	87.78	82.39	84.97	85.46	54.45
Cascade R-CNN	70.62 M	26.48	89.54	84.0	86.73	86.21	55.31
Dynamic-RCNN	42.78 M	31.35	87.36	82.88	85.07	85.30	55.86
YOLOv8	44.60 M	75.78	91.38	86.32	88.80	89.82	57.76
YOLOv9	38.78 M	79.64	93.45	89.65	91.53	92.05	60.48
YOLOv10 [6]	28.24M	85.71	93.32	89.05	91.03	91.48	65.82
LAR-YOLOv8 [7]	28.56 M	54.89	93.04	89.95	91.73	90.92	61.55
HP-YOLOv8 [8]	28.52 M	55.46	94.75	91.05	92.65	92.11	70.01
SOD-YOLOv10	**28.00M**	**86.32**	**95.28**	**92.02**	**93.61**	**95.90**	**73.42**

TABLE VII

Ablation Experiment on RSOD Datasets. TB for TransBone, AA for AA-GFPN, and AFP for AFP-IoU

Model				Params	FPS	P	R	F1	mAP@0.5	0.5:0.95
YOLOv10	TB	AA	AFP							
✓				28.24M	85.71	93.32	89.05	91.03	91.48	65.82
✓	✓			28.03M	85.96	94.02	91.42	92.76	93.38	69.66
✓		✓		28.01M	86.06	94.23	91.47	92.30	93.77	70.45
✓			✓	28.24M	85.70	93.62	90.61	91.93	92.81	67.38
✓	✓	✓		28.00M	86.32	95.07	91.88	93.26	95.26	71.53
✓	✓	✓	✓	28.00M	86.32	95.28	92.02	93.61	95.90	73.42

C. Comparison With Other Models

As shown in Table VI, SOD-YOLOv10 achieves a recall rate of 92.02% and a precision of 95.28%, demonstrating its strong capability in detecting small objects and significantly reducing the risk of missing critical small objects. The *F*1 score is 93.61%, indicating that the model maintains good balance and stability under varying conditions. On the mAP@0.5:0.95 and mAP@0.5 metrics, SOD-YOLOv10 reaches 73.42% and 95.90%, the highest among all listed models. This is 20.86% higher than Faster R-CNN, fully demonstrating its precision in detecting extremely small and complex objects.

As Fig. 4 illustrates, in the VisDrone2019 dataset, the YOLOv10 model fails to detect some small objects that are overlapped or obscured, such as pedestrians and cars, as depicted in Fig. (a) (YOLOv10) and (b) (YOLOv10). In contrast, Fig. 4(a) (SOD-YOLOv10) and (b) (SOD-YOLOv10) shows that the SOD-YOLOv10 model not only accurately identifies the main objects but also detects obscured small objects, especially excelling in scenarios with overlapping objects.

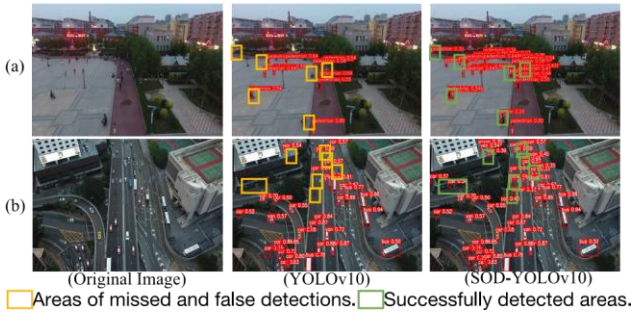

Fig. 4. Detection results are displayed on VisDrone2019.

D. Ablation Experiment

As shown in Table VII, by replacing the original backbone network with TransBone and swapping PANet for BA-FPN, the model sees a slight reduction in parameter count and a significant enhancement in performance. Although the addition of the AFP-IoU optimized loss function does not

change the number of parameters, it does improve the mAP metrics. Ultimately, when the model is fully configured, it demonstrates optimal performance, with all evaluation metrics showing improvement—mAP@0.5 reaches 95.50 and mAP@[0.5:0.95] peaks at 73.42. Additionally, compared to the original version, this model is more lightweight, making it highly suitable for deployment on resource-constrained devices.

IV. Conclusion

This letter presents an enhanced algorithm for detecting small objects in remote sensing imagery, SOD-YOLOv10, which is based on YOLOv10 and incorporates the new Transbone backbone network, AA-GFPN feature fusion technology, and AFP-IoU loss function. We have extensively tested this algorithm on the RSOD, NWPU VHR-10, VisDrone2019, and AI-TOD datasets. Experimental results demonstrate that SOD-YOLOv10 outperforms other detectors in processing speed and precision, especially enhancing the detection performance of small objects in complex environments, significantly increasing the model's mAP.

References

[1] Y. Liu, Q. Li, Y. Yuan, Q. Du, and Q. Wang, "ABNet: Adaptive balanced network for multiscale object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2021.

[2] D. Shi, "TransNeXt: Robust foveal visual perception for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 17773–17783.

[3] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, "BiFormer: Vision transformer with bi-level routing attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10323–10333.

[4] H. Zhang and S. Zhang, "Focaler-IoU: More focused intersection over union loss," 2024, *arXiv:2401.10525*.

[5] C. Liu, K. Wang, Q. Li, F. Zhao, K. Zhao, and H. Ma, "PowerfulIoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism," *Neural Netw.*, vol. 170, pp. 276–284, Feb. 2024.

[6] A. Wang et al., "YOLOv10: Real-time end-to-end object detection," 2024, *arXiv:2405.14458*.

[7] H. Yi, B. Liu, B. Zhao, and E. Liu, "Small object detection algorithm based on improved YOLOv8 for remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 1734–1747, 2024.

[8] G. Yao, S. Zhu, L. Zhang, and M. Qi, "HP-YOLOv8: High-precision small object detection algorithm for remote sensing images," *Sensors*, vol. 24, no. 15, p. 4858, Jul. 2024.