AFSDet: Video Small Object Detection Based on Adaptive Focused Slicing

Kangjian Huang†, Yan Yang*, Yongquan Jiang‡, Xiaobo Zhang§ and Zhuyi Angelina Li¶
* School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu, China
E-mail: yyang@swjtu.edu.cn
† School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu, China
‡ School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu, China
§ School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
¶ Business School, Renmin University of China, Beijing, China

***Abstract*—The detection of video small objects is challenging in images due to their limited presence, which makes the information more susceptible to loss. In this paper, we propose an Adaptive Focused Slicing Detection (AFSDet) method inspired by the way humans observe small objects. AFSDet involves first localizing to a region with potential targets and then scrutinizing the small objects within that region. It is a two-stage approach utilizing the proposed model called Deformable Head YOLO Network (DHYNet) for object detection. The first stage performs a coarse-detection of the original image and utilizes clustering and Adaptive Scale-up Slicing methods to obtain patches, leading to a significant reduction in the number of patches compared to previous slicing-based detection methods.The second stage feeds each patches into the fine-detection model and passes the results to the next frame as a reference to assist coarse-detection. Additionally, taking advantage of the slight variations between consecutive video frames, a superimposing process is employed for coarse-detection, which also contributes to reducing the computational overhead. The results show that the proposed method, tested on the VisDrone-MOT dataset, improves accuracy by 9.6% compared to the baseline model, and improves the accuracy by 3.3 % compared to the previous slicing-based small object detection method.**

I. Introduction

The video small object detection holds a wide array of applications, including people and vehicle recognition and tracking in surveillance videos [1]. While significant research has been conducted on object detection at the image level, video object detection specifically targets objects across continuously captured frames. Video object detection encounters challenges such as motion blur, occlusion, and rare poses, distinct from typical image data [2]. Moreover, consecutive video frames exhibit correlation, enabling the exchange and supplementation of information between them, thereby enhancing detection accuracy. Additionally, these frames demonstrate redundancy, with minimal differences between neighboring frames, which can be leveraged to reduce unnecessary computations and improve detection efficiency. Overall, video-level object detection has new challenges than image-level object detection, but utilizing the characteristics of video can help the model achieve better detection results.

Small objects refer to objects that are small relative to the size of the image or the type of the object, and the definition may vary in different application scenarios. Video small object detection has many challenges [3], such as : (1) Susceptibility of information loss. (2) Vulnerability to noise interference. (3) Limited bounding box tolerance. And the fundamental reason is the small percentage of objects in the image. A straightforward and effective approach involves cropping the original image and enlarging it for detection, which can enhance accuracy. SAHI [4] proposed a method involving sliding traversal of the original image, dividing it into smaller sub-images, which generally improves model accuracy by approximately 5%. However, these methods, as illustrated in Fig. 1, often incur significant computational overhead and may lead to object truncation. Recognizing that small objects tend to cluster in specific regions within an image, this study draws inspiration from human observation techniques for densely packed small objects. Specifically, we first roughly observe the image, lock the region with potential targets, and then carefully identify the small objects within the region.

The Adaptive Focused Slicing Detection (AFSDet) method proposed in this paper aims to only cut the area with potential objects for fine-detection, which reduces the waste of computing resources while improving the detection accuracy. Simultaneously, it utilizes the feature of small changes in consecutive frames of the video data to interact the information between consecutive frames to further enhancing the detection accuracy and efficiency. The network uses the proposed Deformable Head YOLO Network (DHYNet) model as the detection model. As shown in Fig. 2, firstly, a single-class training DHYNet is used to perform coarse-detection on the original image, and the coarse-detection results are adaptively clustered and slicing. Then the patches is sent to the fine-detection model. Finally, the detection results from multiple patches

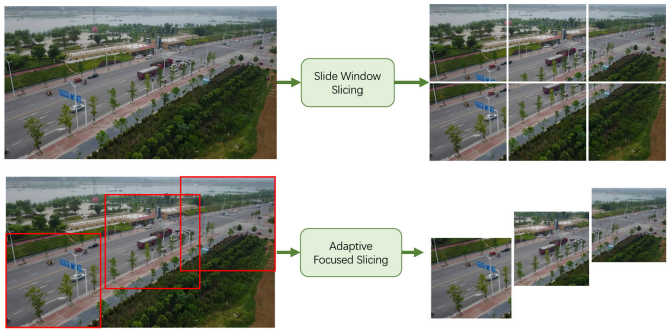

Fig. 1. The first row is the Sliding Window Slicing and the second row is the Adaptive Focused Slicing.

undergo weighted box fusion (WBF) [5], and the results of that frame are passed to the next frame to help the slicing. Meanwhile, since there is little variation between consecutive video frames, this paper proposes that adjacent frames can be fusion and coarse-detection at the same time to further enhance the detection efficiency. The main contributions of this paper are as follows:

- An improved model based on YOLOv8 [6] named DHYNet is proposed. A new module called CSPDarknet53 to FPN with DCNv3(CFD) is integrated into the detection head, fully capturing information in the feature map and alleviating the issue of losing deep semantic information regarding small objects.
- An adaptive focused slicing framework for video small objects detection is proposed, which employs adaptive scale slicing based on clustering to enhance detection. Additionally, the detection results are propagated to the next frame to enhance the slicing effect.
- The method of fusing adjacent frames is proposed, which can be regarded as coarse-detection of multiple frames at the same time to enhance efficiency without sacrificing accuracy.

II. Related Work

A. Video object detection

Due to the temporal dimension inherent in videos compared to 2D images, numerous algorithms have been developed to enhance detection performance by leveraging temporal information. DFF (Deep Feature Flow For Video Recognition) [7] considers that adjacent frames or nearby frames with similar appearances have similar features, and the feature network only acts on the feature extraction of key frames, and the optical flow is utilized to obtain the feature maps of non-key frames. FGFA (Flow-Guided Feature Aggregation For Video Object Detection) [8] considers that the features of some frames in a video sequence are subject to changes in appearance, and these changes can be improved by aggregating the features of adjacent frames or neighboring frames. Trackformer [9] uses Track Query, which is generated by the DETR detector and integrates the position information of the corresponding target over time, and the attention mechanism ensures that the model takes into account the position, occlusion, and recognition features of the object simultaneously.

B. Small object detection

Small objects, due to their limited pixel occupancy, reduced available information, and susceptibility to environmental influences, pose challenges in extracting discriminative features. Moreover, their tendency to aggregate often results in mutual occlusion among targets. To solve the above difficulties, the researcher proposed the following method.

Sample-Oriented Methods: The main problem addressed is how to sample the training data efficiently. Rrnet [10] augmented small targets by copying a small object and pasting it into different locations within the same image with random transformations. Feature-Imitation Methods: Rabbi et al [11] used GAN to perform super-resolution processing on low-resolution remote sensing images, filtering the edge details in order to avoid the loss of high-frequency information during the reconstruction process. Context-Modeling Methods: Objects' semantic or spatial relationships may provide more information when the objects themselves are of poor quality, SINet [12] assumes that primitive RoI pooling operations destroy the structure of small objects, and therefore introduces a context-aware RoI pooling layer to maintain contextual information. Slicing-and-Detect Methods: Akyon [4] et al. proposed the sliding window slicing framework, which can widely improve the detection performance by only simple slicing of the original image, and if fine-tuning on slicing will dramatically improve the performance.

III. Methods

A. Detector Improvement

YOLOv8 developed by Ultralytics in 2023, represents a significant advancement in both detection accuracy and speed compared to its predecessors in the YOLO series. This advanced object detection model has integrated many ideas that favor small objects detection in the design of its network structure. However, YOLOv8 still requires further enhancements. The research indicates that due to the insufficient representation of small object information on the feature map obtained through neck extraction, ordinary convolution is unsuitable. This paper employs deformable convolution [13] to enhance the capability of the detection head in capturing features.

For any given image, let its basic feature be denoted as $x \in R^{C\times H\times W}$, where C represents the number of channels, H and W signify the height and width of the feature map, respectively. The standard 2D convolution can be expressed as follows: Convolution is performed on a point, and the regular grid R is used to sample with as the origin, and then the weight w of the convolution kernel is weighted and summed with the sampling value. R can be expressed as (1):

$$R = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\} \tag{1}$$

Deformable convolution can be computed by (2):

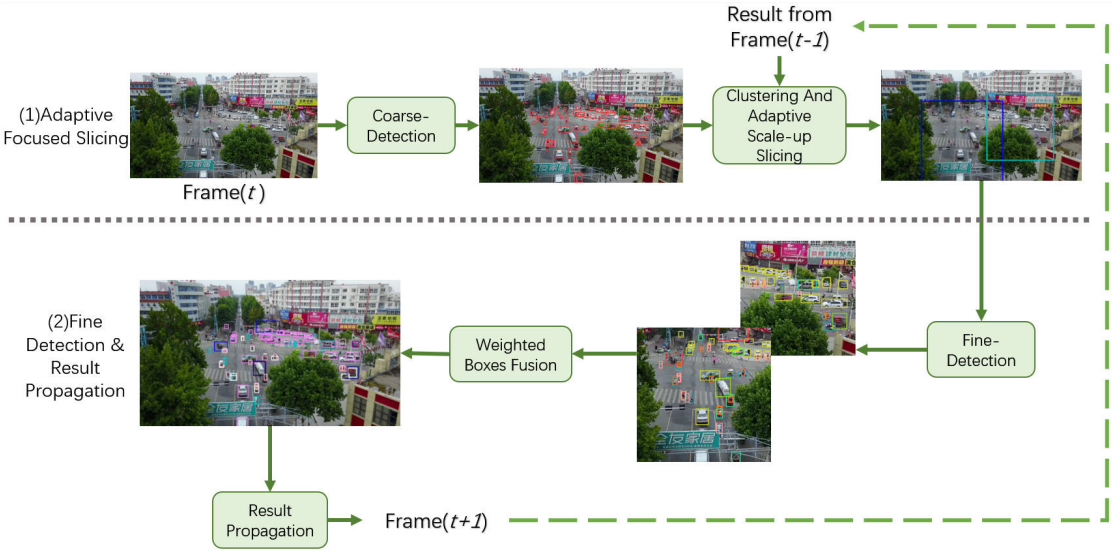

Fig. 2. The framework of AFSDet comprises two phases: (1) Adaptive Focused Slicing and (2) Fine Detection & Result Propagation. In stage 1, the video frames at time t are initially processed by the coarse-detection model. Subsequently, the coarse results at time t are clustered with the final results at time t-1 to obtain the patches through Adaptive Scale-up on the cluster. In stage 2, each patch is sequentially inputted into the fine-detection model. The outputs undergo a Weighted Boxes Fusion (WBF) process to obtain the final results, which are subsequently propagated to the frame at time t+1.

$$y\left(p_0\right)=\sum_{g=1}^{G}\sum_{k=1}^{|R|}w_g m_{gk} x_g\left(p_0+p_k+\Delta p_{gk}\right) \tag{2}$$

where G denotes the division of the spatial aggregation process into G groups, each containing a learnable offset parameter Δp_{gk} and a modulation parameter m_{gk}.

This paper presents a new model named DHYNet that integrates a new module called CFD, as shown in Fig. 3. The CFD module is designed by incorporating the concept of ELAN from YOLOv7 [14] into DCNv3. This module takes the input feature map, processes it through a CONV layer, and then splits it into two parts using the Split operation, The first part of the split feature map undergoes multiple BD modules. The output of the initial BD module, the final BD module, and the features extracted by Split are merged together. Subsequently, the combined features pass through a CONV module with a 1x1 kernel size and a step size of 1. The BD module sequentially integrates a standard convolutional layer with DCNv3, and then performs a residual connect on the result, which is formulized as follows:

$$f=\mathrm{BN}\left(\mathrm{DCNv3}\left(\mathrm{CONV}\left(f_{in}\right)\right)\right) \tag{3}$$

$$f_{out}=\mathrm{CONV}\left(f_{in}\right)\oplus\mathrm{SiLU}\left(f\right) \tag{4}$$

where f_{in} represents the input feature map, while f_{out} represents the output feature map. Additionally, BN refers to batch normalization, and SiLU (Sigmoid Linear Unit) functions as the activation function. The symbol $\oplus$ signifies the connection operation.

The entire process can be expressed as follows:

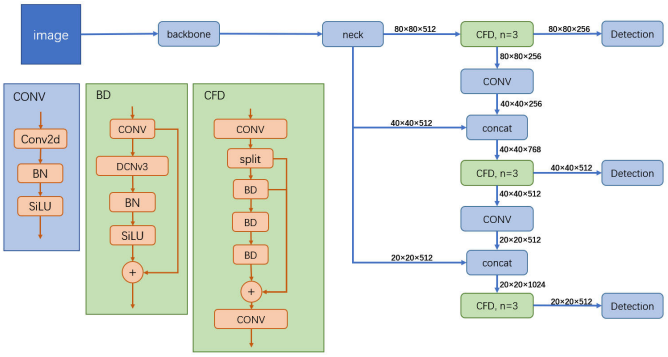

Fig. 3. The network structure of DHYNet.

$$f_a, f_b=\mathrm{split}\left(\mathrm{CONV}\left(f_{in}\right)\right) \tag{5}$$

$$f=\mathrm{BD}\left(\mathrm{BD}\left(\mathrm{BD}\left(f_a\right)\right)\right) \tag{6}$$

$$f_{out}=\mathrm{CONV}\left(f\right)\oplus\mathrm{BD}\left(f_a\right)\oplus f_b \tag{7}$$

where f_a, f_b denote the two feature map components of the original feature map split.

The CFD module harnesses the strong feature-capturing ability of deformable convolution, providing improved computational and memory efficiency through sparse sampling. This overcomes the constraints of standard convolution related to long-range dependencies and adaptive spatial aggregation, resulting in enhanced accuracy and a more thorough gradient flow analysis. By integrating the CFD module into the detection head, it facilitates the retrieval of feature maps at three different scales, effectively reducing the loss of intricate object

characteristics during downsampling and achieving superior object detection performance.

B. Adaptive Focused Slicing

Traditional image slicing methods typically involve dividing the original image into equal parts or using a sliding window approach to extract patches, leading to considerable redundant computations. In this study, we propose a novel slicing method that initially treats all labeled categories in the training set as "foreground" classes and employs the aforementioned DHYNet model for training, thereby yielding a model with improved recall for detecting potential targets.

Affinity Propagation(AP) is an algorithm that obviates the need to specify the number of clusters, instead determining the appropriate number of clusters to assign the samples based on a distance measure from the sample points. Following the extraction of the foreground from the original image in the coarse-detection stage, the centroid of the foreground bounding box serves as a sample point for unsupervised clustering employing the AP algorithm to obtain clusters. Every foreground box is allocated to a cluster, from which the top-left-bottom-right extreme values of all boxes within each cluster are derived to define the initial slicing region. Owing to the aggregation of small objects, this paper contends that there is a significant probability of potential targets being present near the foreground detected during coarse-detection, necessitating an appropriate enlargement of the slicing region.

In this paper, we propose an Adaptive Scale-up Slicing method for scaling the slice region, while mitigating the difficulty of scale variation, i.e., the same image may contain objects of widely varying scales that hinder detection. Denote the width of the original slicing region as w , and express the width of the scaled-up slicing region as in (8):

$$l^{'} = l\frac{w^2}{l^2} + 2\lambda w \tag{8}$$

where w represents the average width of all object boxes within each cluster. Multiply l by the scaling factor w^2/l^2 , and subsequently extend both sides of the slicing region by λw. Then the percentage α of foreground objects in the slicing region can be represented by (9):

$$\alpha = \frac{w}{l^{'}} = \frac{w}{l\frac{w^2}{l^2} + 2\lambda w} = \frac{1}{\frac{w}{l} + 2\lambda} \tag{9}$$

With $w/l \in (0,1)$, we have $\alpha \in (1/(1+2\lambda), 1/2\lambda)$, ensuring that the foreground percentage in the slicing region falls between two constant values. Illustrated in Fig. 4, the approach adjusts slicing regions to accommodate large-scale objects with greater areas and small-scale objects with reduced areas, effectively addressing the issue of scale variation.

C. Fine Detection & Result Propagation

Each patch obtained from the above process is detected utilizing the DHYNet, and the resulting detections are translated into the coordinate system of the original image. Post-processing is necessary to filter out duplicate detection boxes due to overlapping regions in the patches. In this paper, the Weighted Boxes Fusion (WBF) [5] method is employed instead of the conventional NMS method. WBF uses IoU to determine if a detection box corresponds to a specific object and conducts weighted fusion rather than suppression when dealing with multiple detection boxes of the same object.This paper argues that utilizing WBF effectively leverages the results from multiple detections, similar to performing ensemble learning on multiple detectors.

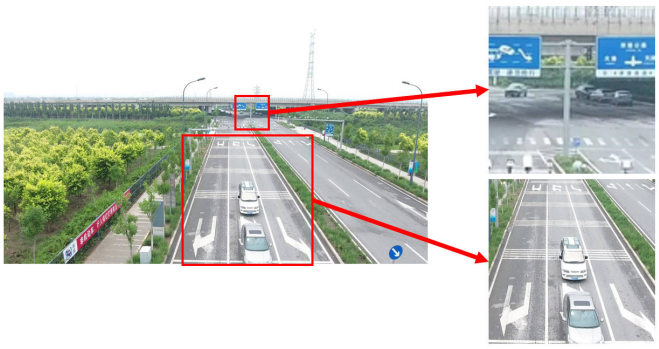

Fig. 4. On the left side, the original image with a notable scale difference between the objects in the two clusters. On the right side, the image after the application of Adaptive Scale-up Slicing, which effectively reduce the scale difference among the foreground objects.

The full detection results of the current frame is acquired and subsequently carried over to the next frame for joint clustering with the coarse-detection output of that frame. This is because the variations between consecutive video frames are usually minimal, and the high-quality detection result from the previous frame can be used as a reference for slicing patches in the next frame. The process of coarse-detection is still maintained, because coarse-detection can detect potential targets outside the existing slicing region.

D. Superimposing Adjacent Frame Detection

Consider using the characteristics of video data to further improve the detection efficiency. Between consecutive frames of a video, there is usually very little change in the background and significant change in the foreground, and it is possible to superimpose the pictures of the two neighboring frames, as shown in Fig. 5.

Consider a video sequence $S = \{(x_t, y_t)\}_{t=1}^{n}$, where x_t represents the image at index t and y_t represents the corresponding label. Then the video sequence $\tilde{S} = \{\tilde{x_t}, \tilde{y_t}\}_{t=1}^{n-1}$ after superimposing of neighboring frames are defined as:

$$\tilde{x_t} = \frac{1}{2}x_t + \frac{1}{2}x_{t+1}, \tilde{y_t} = y_t \cup y_{t+1} \tag{10}$$

the superimposed video sequence $\tilde{S}$ conforms to the distribution P_{fusion}. To ensure that the probability distribution P_{model} of the model closely approximates P_{fusion}, mixup [15] is employed as an augmentation technique during the training of the coarse-detection model.

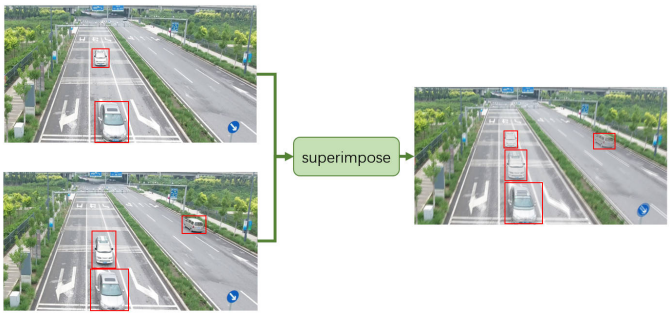

Fig. 5. This figure illustrates the superimposition of adjacent frames. It showcases the rendering of two images with foreground cars into a single composite image, whereby detecting objects in the fused image is tantamount to detecting objects in both original images simultaneously. Notably, the selected images are taken from frames at distinct time intervals rather than adjacent frames in order to show the effect of the method more clearly.

IV. Experiments

A. Datasets and Evaluation Metrics

This paper uses the VisDrone-MOT dataset collected by the AISKYEYE team at the Lab of Machine Learning and Data Mining, Tianjin University, China. The dataset comprises recordings obtained from diverse drone cameras, capturing scenes across varying weather and lighting conditions. It is a small object detection dataset with a high level of difficulty.

This paper adopts the evaluation protocol established by MSCOCO [16]. AP_{50} refers to the AP measured when the IoU threshold is set to 0.5, similarly AP_{75} refers to the AP when the IoU threshold is set to 0.75, and AP refers to mean Average Precision. The $\#img$ is used to record the number of slicing patches, and the FPS indicates how many original images the model can detect per second.

B. Implementation Details

In this paper, AFSDet is implemented with PyTorch 1.7 and the detection model is based on the open source Ultralytics 8.0.195 YOLOv8 framework. The experiments are completed on NVIDIA GeForce RTX 3070, Ubuntu system. In training the coarse-detection model, this paper uses Mosaic, Mixup, Horizontal Flip, Random Crop, and Color Transform data augmentation. The other parameters were kept as the default parameters recommended by the Ultralytics framework. The AP clustering algorithm uses the interface provided by the Scikit-learn library.

C. Comparison Algorithm

This paper compares seven popular and powerful algorithms. YOLOv8 is a highly popular single-stage end-to-end detection algorithm known for its exceptional performance in industrial applications. RT-DETR [17] represents a high-performance Transformer-based object detection model. FCOS [18] is an anchor-free single-stage detection algorithm. Cascade-RCNN [19] is a powerful two-stage object detection algorithm. VFNet [20] is specifically designed for detecting dense small targets. SAHI [4] is a sliding slicing framework, which is applied to the YOLOv8n detector in this experiment.

D. Results

The results on the VisDrone-MOT test set are shown in Table I. The results show that the method presented in this paper provides a large improvement over common object detections, sacrificing detection speed for a larger improvement in detection accuracy relative to the baseline model, YOLOv8, and the sacrificed speed is still usable in real time in some scenarios. In comparison to the previous slicing-based method SAHI, the approach proposed in this study effectively minimizes the number of patches by introducing an additional coarse-detection step, which not only greatly improves the detection speed, but also improves in the detection accuracy.

E. Ablation Experiments

The results of the ablation experiments as shown in Table II demonstrate that the enhanced model achieves superior results as a result of its ability to robustly capture features through the deformable convolutional layer and extract more informative feature maps for the detection head. The proposed adaptive focused slicing framework extracts patches from the original image containing potential targets and conducts separate fine-detection. The experiments demonstrate its significant impact on enhancing detection accuracy, however, it also leads to a decrease in detection efficiency. By combining the DHYNet with the adaptive focused slicing framework, a significant improvement in accuracy is observed on the validation set compared to the baseline model, thus validating the effectiveness of the proposed method in this paper.

V. Conclusion

This paper introduces a novel method called Adaptive Focused Slicing Detection for small object detection in videos. This method slices and scales up the regions in the original video frames containing potential targets, then feeds them into the fine-detection model to aid in small object detection. The DHYNet is proposed, utilizing this model for initial coarse-detection to identify potential targets. Subsequently, the set of targets is clustered and dynamically divided into multiple patches. Each patch is independently processed by the fine-detection model, and the resultant outputs are post-processed. To account for the traits of video data, this paper proposes inter-frame propagation of the results to enhance accuracy, and superimposing adjacent frames to reduce computations. Comparative experiments were conducted on the VisDrone-MOT dataset to showcase the method's superiority over other powerful approaches. Additionally, ablation experiments were performed to validate all components of the method.

References

[1] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-based surveillance analysis: A survey," *IEEE transactions on circuits and systems for video technology*, vol. 29, no. 7, pp. 1985–1997, 2018.

TABLE I
COMPARISON OF OUR METHOD WITH OTHER METHODS FOR OBJECT DETECTION ON VISDRONE-MOT TEST SET.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS	$\#img$
YOLOv8n	8.8	18.6	73	3.6	11.0	18.9	**88.3**	-
RT-DETRn	8.7	18.3	7.4	4.2	9.4	18.2	16.7	-
FCOS	7.8	19.5	4.3	3.8	6.3	15.3	27.1	-
VFNet	10.6	20.6	10.7	4.1	10.1	21.9	22.2	-
Cascade-RCNN	10.9	19.4	11.5	4.8	11.0	18.3	15.9	-
YOLOv8n+SAHI	12.8	24.9	11.9	4.5	16.5	22.5	2.1	46732
AFSDet(Ours)	**14.0**	**28.2**	**12.3**	**6.3**	**17.1**	**22.6**	12.3	**27121**

TABLE II
AN ABLATION EXPERIMENTS ON THE VISDRONE-MOT VALIDATION SET. AFS DENOTES THE PROPOSED ADAPTIVE FOCUSED SLICING FRAMEWORK. DN DENOTES THE PROPOSED DHYNET.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
AFSDet	**18.7**	**35.5**	**17.0**	**15.0**	**30.2**	**40.8**
AFSDet-DN	14.9	30.1	13.0	7.9	24.6	33.4
AFSDet-AFS	8.8	18.6	7.3	6.1	13.4	19.2
AFSDet-AFS-DN	8.7	18.1	7.1	3.6	11.0	18.9

[2] L. Jiao, R. Zhang, F. Liu, *et al.*, "New generation deep learning for video object detection: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3195–3215, 2021.

[3] G. Cheng, X. Yuan, X. Yao, *et al.*, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[4] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *2022 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2022, pp. 966–970.

[5] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104 117, 2021.

[6] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics yolov8*, version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[7] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2349–2358.

[8] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 408–417.

[9] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8844–8854.

[10] C. Chen, Y. Zhang, Q. Lv, *et al.*, "Rrnet: A hybrid detector for object detection in drone-captured images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 100–108.

[11] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network," *Remote Sensing*, vol. 12, no. 9, p. 1432, 2020.

[12] X. Hu, X. Xu, Y. Xiao, *et al.*, "Sinet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 3, pp. 1010–1019, 2018.

[13] W. Wang, J. Dai, Z. Chen, *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 408–14 419.

[14] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[16] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[17] Y. Zhao, W. Lv, S. Xu, *et al.*, "Detrs beat yolos on real-time object detection," *arXiv preprint arXiv:2304.08069*, 2023.

[18] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: A simple and strong anchor-free object detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1922–1933, 2020.

[19] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[20] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8514–8523.