2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Separable Auxiliary Training for Real-Time Small Object Detection

Xinrong Wu[1], Fan Wang[1], Min Nuo[1], Ying Zhou[1] and Xiaopeng Hu[1*]

[1] Dalian University of Technology, Liaoning Province, China
*Corresponding author. Email: huxp@dlut.edu.cn
Contributing authors: {wuxinrong, wangfan,nuomin, zhouying_dlut}@mail.dlut.edu.cn

Abstract. During the training of end-to-end detectors, one-to-one label assignments result in an insufficient number of positive samples, impeding the learning of discriminative features. Existing methods have employed one-to-many label assignments and denoising training strategies to provide additional supervision, thereby increasing the number of positive samples or introducing samples with noise. However, these additional supervisions perform bidirectional feature fusion with the original end-to-end models, increasing the computational costs of the model during inference. In this paper, we propose a Separable Auxiliary Training (SAT) for real-time small object detection to achieve auxiliary supervision without additional inference delay. In SAT, an auxiliary branch supervised by a one-to-many label assignment is adopted to assist a deployment branch during training. To avoid increasing the inference costs, a one-way feature flow from the deployment branch to the auxiliary branch has been designed. The flow ensures that the deployment branch can be deployed independently without sacrificing any accuracy. Extensive experiments demonstrate that SAT can provide additional supervision to enhance performance without increasing computational costs during inference.

Keywords: Small Object Detection, SAT, Separable Auxiliary Supervision, RT-DETR.

1 Introduction

Real-time small object detection involves locating and classifying objects with pixel sizes smaller than 32×32 in images [1], and the inference speed must reach at least 30 frames per second (FPS) [2]. The speed requires the number of parameters and calculation costs. Meanwhile, small objects are difficult to extract effective features due to the limited number of pixels, leading to lower precision and accuracy.

At present, the most popular real-time small object detectors can be divided into two categories: CNN-based and Transformer-based. The CNN-based detectors, such as YOLO series [3–9], adopt an indirect way to predict objects in the images. The entire structure includes many hand-designed components, such as anchor generation, label assignments, and non-maximum suppression (NMS) post-processing [10]. The

Transformer-based detectors, such as DETR series [2, 10–17], employ one-to-one label assignments to supervise the models during training.

DETRs eliminate manual design components and simplify the object detection pipeline. However, their one-to-one label assignments lead to insufficient positive samples for small object detection, impeding discriminative feature learning. Co-DETR [12], Group DETR [14] and MS-DETR [15] combine one-to-one and one-to-many label assignments to conduct additional supervision. DN-DETR [17] utilizes a denoising training strategy for additional supervision to introduce samples with noise. However, these additional supervisions perform bidirectional feature fusion with the original end-to-end models, increasing the computational costs of the model during the inference stage.

To solve the above problem, we propose a Separable Auxiliary Training (SAT) for real-time small object detection to achieve auxiliary supervision during training and separable deployment during inference. In SAT, an auxiliary branch supervised by a one-to-many label assignment is adopted to assist with a deployment branch during the training stage. The auxiliary branch comprises a backbone network with the same backbone as the deployment branch, thereby facilitating the fusion of features at corresponding scales and the provision of auxiliary supervision. To utilize the benefits of auxiliary supervision and circum- vent the limitations of the increased computational costs inherent to previous approaches, we design a one-way propagation of the feature flow from the deployment branch to the auxiliary branch. The flow ensures that the deployment branch is deployed independently without sacrificing any accuracy.

In SAT, the one-to-many label assignment is used to provide additional supervision, but it also introduces a considerable number of ambiguous labels. According to our analysis of the process of label assignments, small objects have less area, leading to many targets being detected within the same anchor range. This makes it difficult for models to accurately learn about dense small objects. Therefore, we employ an Optimal Transport Assignment (OTA) [18] as a label assignment of the auxiliary branch to reduce the number of ambiguous labels and enhance the effectiveness of the auxiliary branch.

In summary, our goal is to build an auxiliary supervision method without additional inference delay. That is, we seek to employ a one-to-many label assignment to improve the performance and develop a one-way propagation of the feature flow for separable deployment. Extensive experiments conducted on VisDrone2021 and AI-TOD datasets have validated the effectiveness of our proposed approach. Our main contributions are as follows:

- We propose a Separable Auxiliary Training (SAT) for real-time small object detection to achieve auxiliary supervision during training and separable deployment during inference. SAT utilizes a one-to-many label assignment to achieve additional supervision without increasing computational complexity during inference.
- We propose a one-way flow approach from the deployment branch to the auxiliary branch to avoid increasing the computational costs during inference. Meanwhile, SAT utilizes the OTA as the label assignment of auxiliary branch to reduce the number of ambiguous labels and enhance the effectiveness of auxiliary branches.
- We verify the SAT on VisDrone2021 and AITOD datasets, and the experimental results show that the auxiliary branch within SAT can provide additional supervision

to enhance the model's accuracy and deploy independently without any compromise in performance.

2 Related Word

2.1 Real-time Small Object Detection

Real-time object detection can be classified into two categories according to the differences in output structures: YOLO series and DETR series. The YOLO series [3–9], generate a considerable number of redundant bounding boxes during the inference phase. The DETR series [2, 10–17], represent an end-to-end approach to object detection, which eliminate the hand-designed anchors and NMS components in the traditional detection pipeline. DETR [10] streamlines detection pipelines and mitigates the performance bottleneck caused by NMS. However, it has a large number of parameters and high computational complexity, which presents a challenge in achieving real-time object detection. RT- DETR [11] has developed an efficient hybrid encoder that effectively processes multi-scale features and has proposed a flexible decoder based on DETR that supports adjustment of the inference speed. At present, both of the YOLO series and the RT-DETR series are capable of real-time object detection.

Real-time small object detection is typically based on normal object detection methods, with the extraction of high-resolution features for detection. TPH-YOLOv5 [19] detects small objects in P2 layer to improve the accuracy and precision of the models. However, the high solution features result in an increased number of prediction boxes and ambiguous labels, which in turn makes it challenging to assign ground-truth labels during training.

2.2 Label Assignment

Label assignments distinguish the positive and negative attributes of each anchor during the training stage. Some anchors may be attributed to multiple ground-truths, resulting in ambiguous anchors, introducing harmful gradients. Therefore, adaptive label assignments are proposed to alleviate this phenomenon. ATSS [20] proposes an adaptive sample selection strategy that adopts mean and standard deviation of IOU values from a set of closest anchors for each ground truth as a position threshold. In contrast, OTA [18] defines the unit transportation cost between each anchor and ground-truth to decode correspondence via Sinkhorn-Knopp iteration. Meanwhile, the method uses the Hungarian algorithm to reduce the number of ambiguous labels of one-to-many label assignment process. According to our analysis of the label assignments of small object detection, the absence of ambiguous labels can effectively improve the small object detection performance of the model. In this paper, we adopt OTA as the one-to-many label assignment strategy of the auxiliary branch to further improve the performance.

2.3 Auxiliary Supervision

The additional supervision of models is achieved through the utilization of diverse loss functions, including collaborative supervision and auxiliary supervision. The cooperative supervision is unable to remove additional modules during the inference phase, due to the bidirectional flow of data between the original structure and the additional structure. Co-DETR [12], Group DETR [14] and MS-DETR [15] integrate one-to-one and one-to-many label assignments to achieve an additional supervision model. However, these methods are difficult to separate additional modules during inference, resulting in a reduction in inference speed. RT-DETRv3 [2] employs a one-to-many separable head to provide additional supervision, thus overcoming the issue of insufficient positive labels. However, the approach employed by RT-DETRv3 results in a relatively long gradient feedback path for auxiliary detection. We found that when the data flow is unidirectional, originating in the original network and terminating in the additional modules, it is possible to separate these modules without affecting the accuracy and precision of the model. Based on the characteristics, we propose a one-way feature flow approach to achieve auxiliary supervision without additional inference latency.

3 Methodology

3.1 Overview

As illustrated in Fig. 1, the architecture of SAT consists of a deployment branch and an auxiliary branch. Both of the branches are worked in different stages. During the training stage, identical inputs and labels are assigned to the deployment branch and auxiliary branch. They independently calculate and propagate the loss after a weighted sum. During the inference stage, the deployment branch is retained solely for the purpose of accelerating the inference speed.

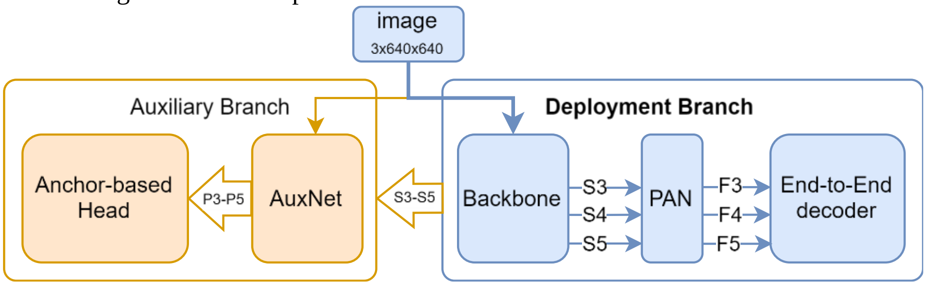

Fig. 1. Overview of Auxiliary Training Framework. The framework consists of a deployment branch and an auxiliary branch. Both of the blue and yellow paths are executed during training, and the blue paths are only executed during inference.

3.2 Separable Auxiliary Training

Deployment Branch. The deployment branch comprises a backbone, a feature fusion neck, and an end-to-end head, as illustrated in Fig. 2. The backbone may be selected from the existing feature extraction networks, including CSPNet [21], YOLO series, and other networks based on transformers, to extract multi-scale features from the input images. The present study employed backbones derived from YOLOv5 [3], YOLOv8 [6] and YOLO11 [9]. The feature fusion neck is responsible for multi-scale feature fusion, utilizing a Path Aggregation Network [22] structure that performs both top-down and bottom-up feature fusion. The end-to-end head adopts the decoder of RT-DETR to predict the bounding box of objects directly without post-processing.

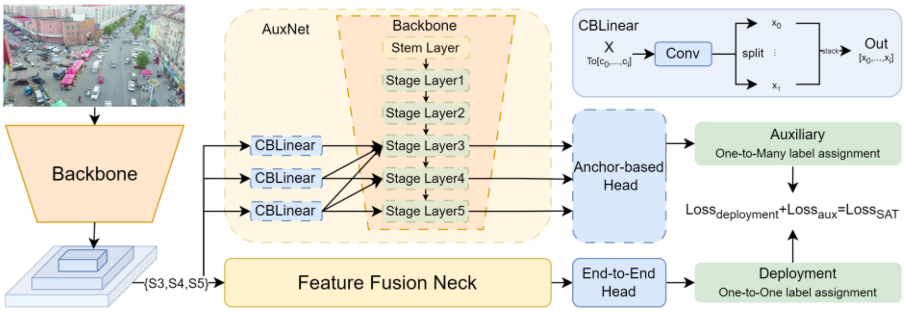

Fig. 2. Details of Auxiliary Training Framework. The solid line modules and the dotted line modules belong to the deployment branch and the auxiliary branch, respectively.

Auxiliary Branch. As illustrated in Fig. 2, the auxiliary branch comprises an auxiliary network (AuxNet) and a one-to-many detection head. The AuxNet receives multi-scale features {S3, S4, S5} from the backbone and generates corresponding scale outputs, which consist of CBLinear [7], stem layer, and stage layer modules. The CBLinear is employed to split and upsample the multi-scale features, which are then concatenated in the subsequent step. The stem layer and the stage layers are identical to the backbone of the deployment branch, and used to extract image features at different scales. The detection head of the auxiliary branch is supervised by the same labels as the deployment branch, while the labels are matched by one-to-many label assignments for loss calculation. In this study, an anchor-based detection head is employed to verify the effectiveness of SAT.

Loss Function. In SAT, the deployment branch and the auxiliary branch employs an end-to-end detection head and an anchor-based detection head, respectively. Because of the disparate output structures, both of branches use different loss functions for calculation. The loss function of deployment branch, based on RT-DETR [11], introduces IoU-aware query selection based on Hungarian algorithm during training to deal with inconsistent distribution of classification score and location confidence. In auxiliary branch, the anchor-based loss function is adapted to simplify the migration of other anchor-based methods. We can formulate the loss of SAT as follows:

$$\mathcal{L}_{SAT} = \beta_1 \mathcal{L}_{\mathrm{deploy}} + \beta_2 \mathcal{L}_{aux}, \tag{1}$$

where $\mathcal{L}_{SAT}$, $\mathcal{L}_{\text{dep}loy}$, and $\mathcal{L}_{aux}$ are the loss of SAT, the deployment branch, and the auxiliary branch, respectively. The $\mathcal{L}_{SAT}$ is the weighted sum of the losses, and the setting of β is set to 1.0.

The $\mathcal{L}_{\text{dep}loy}$ consists of the loss of box and classification, and its hyperparameters of followed by RT-DETR. The loss of the deployment branch is calculated as follows:

$$\mathcal{L}_{\text{dep}loy} = \mathcal{L}_{\text{box}} + \mathcal{L}_{class}, \tag{2}$$

where the $\mathcal{L}_{\text{box}}$ is calculated by GIOU [23],, and $\mathcal{L}_{class}$ is calculated by the BCEloss.

The calculation of $\mathcal{L}_{aux}$ is same as the detection head we used in auxiliary branch, involves three components [3]: loss of location, IoU, and classification. We can formulate the loss of the auxiliary branch as follows:

$$\mathcal{L}_{\text{aux}} = \lambda_{loc}\mathcal{L}_{\text{loc}} + \lambda_{obj}\mathcal{L}_{obj} + \lambda_{cls}\mathcal{L}_{\text{cls}} \tag{3}$$

where the coefficients λ_{loc}, λ_{obj}, and λ_{cls} are hyperparameters that determine the relative importance of each component in the overall loss calculation. The $\mathcal{L}_{\text{box}}$ is calculated by CIoU [24], $\mathcal{L}_{obj}$ and $\mathcal{L}_{cls}$ are calculated by the BCEloss.

3.3 Label Assignment for SOD

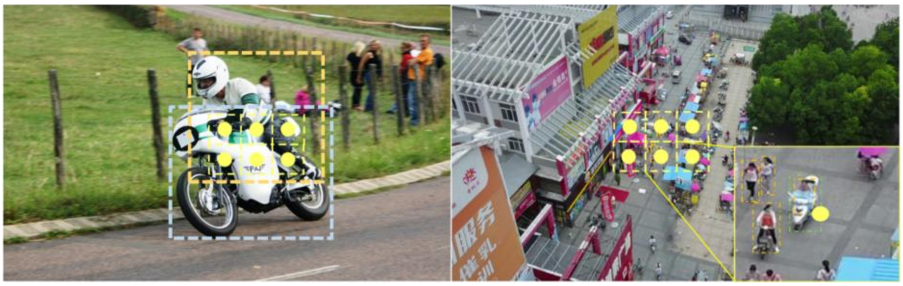

Fig. 3. An illustration of ambiguous anchor points in object detection. Yellow dots show some of the ambiguous anchors in two sample images. The large object (left) is due to large area, resulting in overlap. And the small objects (right) are assigned to the same anchor due to its small area.

A one-to-many label assignment is employed for the purpose of supervising the auxiliary branch. The label assignment distinguishes positive and negative attributes of each anchor during the training process. In some cases, an anchor may be assigned to multiple ground truths, which can result in ambiguous anchors. These anchors can introduce detrimental gradients relative to other ground structures.

It should be noted that all scales of objects have ambiguous anchors, but the reasons for their occurrence are different, as shown in Fig. 3. The presence of large objects is attributable to their extensive surface area, which results in multiple objects overlapping within the same region. This phenomenon gives rise to ambiguous anchor points within the overlapping regions. The situation with small objects is characterized by their limited surface area, which leads to multiple objects appearing within the same anchor range. This situation that small objects cover very few anchors, and ambiguous anchors account for a large proportion makes it difficult for models to learn dense small objects.

To alleviate the problem of ambiguous anchors, the one-to-many label assignment with groups, OTA [18], is used to supervise the auxiliary branch. It should be noted that OTA has the potential to enhance the precision of anchor-based detectors for small objects. Furthermore, this approach is only employed during the training phase, which can improve the accuracy but does not impact the inference speed when deploying models.

4 Experiment

4.1 Experimental Settings

Dataset and Evaluation Measures. We conduct experiments on the VisDrone2021 [25] and AITOD datasets. All models are verified on the test-dev set. We standard mAP, Precision, Recall [3], and COCO style average precision (AP) [1] are used. The AP of small, medium and large objects are also reported, particularly to understand the performance of our method for small object detection.

Implementation Details. The ultralytics toolkit [9] is used to implement our SAT. The backbones used in our study are YOLOv5 [3], YOLOv8 [6] and YOLO11 [9]. During the training phase, to maintain consistency among all of models, pre-trained models are not loaded. We train the proposed model on VisDrone2021 trainset for 300 epochs, and use AdamW optimizer [26]. The learning rates of backbone follow RT-DETR [11]. The coefficients β_1, β_2, λ_{loc}, λ_{obj}, and λ_{cls} of the loss functions are set to 1.0, 1.0, 0.05, 1.0, and 0.625, respectively. During the evaluation phase, confidence and IoU is set to 0.001 and 0.6, respectively.

4.2 Main Results

In this section, the evaluation method of AP is adopted to facilitate comparison with other small object detectors. For real-time detection, the scale of models is set to S, and input size is set to 640×640.

VisDrone. The results are summarized in Table 1, our proposed SAT achieves the best results 35.9% AP compared with other real-time methods, including CNN-based and DETR-like methods. The SAT based on YOLOv8 achieves the optimal performance, surpassing the corresponding real-time detectors with one-to-one label assignments by 1.4%, 1.5%, and 0.7% in terms of AP_{50}, AP_{75}, and AP_L.

Table 1. Comparison of real-time small object detectors on VisDrone.

Model	AP$_{50}$	AP$_{75}$	AP$_{S}$	AP$_{M}$	AP$_{L}$	Params(M)	GFLOPs(G)
YOLOv5s [3]	26.0	12.5	6.0	20.2	30.4	7	15.8
YOLOv6s [4]	29.9	17.7	7.6	26.4	37.5	18.51	45.18
YOLOv7s [5]	24.3	11.5	6.0	18.8	26.7	6.03	13.1
YOLOv8s [6]	30.2	16.9	7.4	26.4	37.2	11.1	28.5
YOLOv9s [7]	31.5	18.0	7.8	28.2	41.4	7.17	26.7
YOLOv10s [8]	31.2	17.6	8.0	27.1	43.1	8.04	24.5
YOLO11s [9]	31.5	17.9	7.8	27.7	39.6	9.4	21.3
RT-DETR(R18) [11]	32.5	18.6	9.9	27.6	35.3	19.9	57.0
RT-DETRv3(R18) [2]	28.3	15.5	8.2	22.7	29.7	20.0	60.0
RT-DETR(YOLOv5)	31.5	17.1	9.7	25.6	32.5	10.9	22.7
RT-DETR(YOLOv8)	32.6	18.1	10.2	26.9	33.0	13.0	27.3
RT-DETR(YOLO11)	32.1	17.6	10.0	26.0	38.0	12.4	25.1
SAT(YOLOv5s)	32.6	18.3	10.3	27.1	33.4	10.9(15.1)	22.7(33.4)
SAT(YOLOv8s)	33.6	19.1	10.3	28.3	30.4	13.0(18.3)	27.3(41.6)
SAT(YOLO11s)	33.0	18.5	10.2	27.3	35.2	12.4(18.3)	25.1(39.8)
YOLOv5s-P2 [3]	29.3	14.1	7.5	22.8	33.1	7.18	18.7
TPH-YOLOv5s-P2 [19]	33.3	17.1	9.4	26.3	35.8	-	-
YOLOv7s-P2 [5]	30.5	15.9	8.8	24.1	31.2	6.46	21.0
YOLOv8s-P2 [6]	30.3	16.1	8.2	25.3	35.7	10.6	36.7
YOLOv10s-P2 [8]	34.1	19.0	10.1	28.1	36.0	8.2	36.6
YOLO11-P2 [9]	35.5	19.8	10.5	29.7	37.0	9.7	31.7
RT-DETR(YOLOv5-P2)	34.3	19.7	11.4	28.4	31.8	11.2	39.9
RT-DETR(YOLOv8-P2)	34.5	19.6	11.3	28.5	36.8	13.1	44.7
RT-DETR(YOLO11-P2)	34.8	19.9	11.9	28.8	36.5	12.6	42.9
SAT(YOLOv5s-P2)	34.9	20.1	11.8	28.8	37.3	11.2(15.4)	39.9(51.3)
SAT(YOLOv8s-P2)	35.9	21.1	12.0	29.9	41.7	13.1(19.2)	44.7(65.0)
SAT(YOLO11s-P2)	35.3	20.7	11.5	29.8	37.8	12.6(18.7)	42.9(59.7)

The optimal, second-best, and third-best results are represented using three different colors, respectively: ■, ▨, and □. The values in parentheses and outside parentheses represent the parameters and GFLOPs during training and deployment, respectively.

Benefiting from the one-way data flow we designed, SAT and RT-DETR share the same FLOPs and the number of parameters during the inference phase. The results demonstrate that SAT can provide auxiliary supervision, thereby enhancing the performance of the model without raising computational costs during inference. And the enhancement is not contingent on the choice of backbone.

AI-TOD. We also conduct experiments on the AI-TOD dataset to demonstrate the effectiveness of our proposed SAT. Table 2 shows our results on the AI-TOD test split. We compare the performance of our SAT with other methods. The SAT achieves the best result 47.6% mAP$_{50}$ compared with other real-time methods, including CNN-based and DETR-like methods. SAT surpasses the RT-DETR with the same FLOPs and the number of parameters by 1.7%, 1.2%, and 1.0% in terms of mAP$_{50}$, mAP$_{75}$, mAP$_{50\text{-}95}$.

Table 2. Comparison of real-time small object detectors on AI-TOD.

Model	mAP_{50}	mAP_{75}	mAP_{50-95}
YOLOv8 [6]	32.5	11.6	14.9
QueryDet [27]	29.3	7.9	12.2
Deformable-DETR [13]	30.5	8.8	10.13
DAB-DETR [28]	30.84	9.87	10.86
RT-DETR(Resnet18)	41.73	-	17.97
RT-DETR(YOLO11)	29.9	17.7	7.6
SAT(YOLO11)	35.3	20.7	11.5

4.3 Ablation

In this paper, we analyzed the improvement of OTA on small objects accuracy, and proposed SAT to migrate OTA to end-to-end detectors. Meanwhile, we also increase the number of query sets and add additional supervision layers to improve the performance of SAT. In this section, we conducted many ablation experiments to verify the correctness of our analysis and the effectiveness of the proposed methods.

OTA For Small Object Detection. To verify the correctness of our analysis about the OTA, we perform ablation studies, and the experimental results are shown in Table 3. We choose the previous version of object detector YOLOv5 as our baselines, and the resolution of the input image is 640×640. The experimental results show that the model with OTA increased by 1.5% in mAP_{50}, and the model with P2 layer increased by 3.5% in mAP_{50}. The OTA is one of the label assignment strategies that only applies to the loss calculation stage and does not increase the number of parameters and computational complexity. These results demonstrate that the reduction of ambiguous labels can enhance the performance of the model in the small object detection.

Table 3. Ablation study of OTA for small object detection.

OTA	P2	mAP_{50}	mAP_{50-95}	Precision	Recall
×	×	27.7	14.6	40.4	30.8
✓	×	29.2(+1.5)	15.8(+1.2)	50.1	31.3
×	✓	31.0(+3.3)	16.4(+1.8)	42.5	33.2
✓	✓	34.5(+6.8)	18.6(+4.0)	44.5	37.5

Speratable Auxiliary Training. To verify the effectiveness of SAT and avoid the influence of the same backbone, we conduct the same experiment on YOLOv5, YOLOv8 and YOLO11. Meanwhile, we also add the P2 layer to the above model for comparison. As presented in Table 4, the SAT with three backbones achieve 34.5%, 35.6% and 34.8% mAP_{50}, increasing 1.1%, 1.0% and 0.9%, respectively. When we add the P2 layer to these models, the SAT also improves the mAP_{50} of the three backbones by 0.6%, 1.3% and 0.6%, respectively. It can be seen that the performance of SAT is not constrained by the backbone.

Table 4. Comparison of different models for small object detection.

Backbone	SAT	P2	mAP_{50}	mAP_{50-95}	Precision	Recall
YOLOv5 [3]	×	×	33.4	18.8	52.3	34.6
	✓	×	34.5(+1.1)	19.7(+0.9)	53.0	36.0
	×	✓	36.2	21.0	55.2	37.1
	✓	✓	36.8(+0.6)	21.3(+0.3)	53.9	38.4
YOLOv8 [3]	×	×	34.6	19.7	52.9	36.3
	✓	×	35.6(+1.0)	20.6(+0.9)	53.8	36.7
	×	✓	36.5	21.0	53.8	37.7
	✓	✓	37.8(+1.3)	22.2(+1.2)	54.7	39.3
YOLO11 [3]	×	×	33.9	19.2	52.7	35.9
	✓	×	34.8(+0.9)	19.9(+0.7)	53.6	36.7
	×	✓	36.7	21.3	54.5	38.3
	✓	✓	37.3(+0.6)	21.9(+0.6)	55.0	38.6

5 Conclusion

In this paper, we propose a Separable Auxiliary Training (SAT) for real-time small object detection to achieve auxiliary supervision during training and separable deployment during inference. To utilize one-to-many label assignments to supervise the end-to-end detector without increasing computational complexity during inference, a one-way propagation of the feature flow from the deployment branch to the auxiliary branch is proposed. Meanwhile, we analyze the process of label assignments and employ OTA to reduce the number of ambiguous labels. The SAT was verified on VisDrone2021 and AI-TOD, and the experimental results demonstrate that the SAT can provide additional supervision to enhance the model's accuracy and deploy the deployment branch independently without any compromise in performance. SAT has explored the potential of one-to-many label assignments based on CNNs to facilitate the supervision of the end-to-end detectors. In the future, we will investigate the possibility of transferring the other CNN-based improvement strategies to supervise different architectural models.

References

1. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the IEEE/CVF Conference on European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
2. Wang, S., Xia, C., Lv, F., Shi, Y.: Rt-detrv3: Real-time end-to-end object detection with hierarchical dense positive supervision. arXiv preprint arXiv:2409.08475 (2024)
3. Jocher Glenn: YOLOv5. https://github.com/ultralytics/yolov5 (2022), https://github.com/ultralytics/yolov5
4. Li, C., Li, L., Geng, Y., Jiang, H., Cheng, M., Zhang, B., Ke, Z., Xu, X., Chu, X.: Yolov6 v3. 0: A full-scale reloading. arXiv preprint arXiv:2301.05586 (2023)

5. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7464–7475 (2023)
6. Ultralytics: YOLOv8. https://github.com/ultralytics/ultralytics (2023), https:// github.com/ultralytics/ultralytics
7. Wang, C.Y., Yeh, I.H., Mark Liao, H.Y.: Yolov9: Learning what you want to learn using programmable gradient information. In: Proceedings of the IEEE/CVF Conference on European Conference on Computer Vision (ECCV). pp. 1–21. Springer (2025)
8. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Real- time end-to-end object detection. arXiv preprint arXiv:2405.14458 (2024)
9. Ultralytics: Yolo11. https://github.com/ultralytics/ultralytics (2024), https:// github.com/ultralytics/ultralytics
10. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the IEEE/CVF Conference on European Conference on Computer Vision (ECCV). pp. 213–229. Springer (2020)
11. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16965–16974 (June 2024)
12. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6748–6758 (2023)
13. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
14. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., Wang, J.: Group detr: Fast detr training with group-wise one-to-many assign- ment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6633–6642 (2023)
15. Zhao, C., Sun, Y., Wang, W., Chen, Q., Ding, E., Yang, Y., Wang, J.: Ms-detr: Efficient detr training with mixed supervision. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17027–17036 (2024)
16. Xia, C., Wang, X., Lv, F., Hao, X., Shi, Y.: Vit-comer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. In: Proceedings of the conference on computer vision and pattern recognition (CVPR). pp. 5493– 5502 (2024)
17. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
18. Ge, Z., Liu, S., Li, Z., Yoshie, O., Sun, J.: Ota: Optimal transport assignment for object detection. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 303–312 (2021)
19. Zhu, X., Lyu, S., Wang, X., Zhao, Q.: Tph-yolov5: Improved yolov5 based on trans- former prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2778–2788 (2021)
20. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9759– 9768 (2020)
21. Wang, C.Y., Liao, H.Y.M., Wu, Y.H., Chen, P.Y., Hsieh, J.W., Yeh, I.H.: Cspnet: A new backbone that can enhance learning capability of cnn. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 390–391 (2020)

22. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8759–8768 (2018)
23. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 658–666 (2019)
24. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
25. Cao, Y., He, Z., Wang, L., Wang, W., Yuan, Y., Zhang, D., Zhang, J., Zhu, P., Van Gool, L., Han, J., et al.: Visdrone-det2021: The vision meets drone object detection challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2847–2854 (2021)
26. Loshchilov, I., Hutter, F., et al.: Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101 5 (2017)
27. Yang, C., Huang, Z., Wang, N.: Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In: Proceedings of the conference on Computer Vision and Pattern Recognition (CVPR). pp. 13668–13677 (2022)
28. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)