GGMIX: GLOBAL-GUIDED MIXING OF SELF-ATTENTION AND CONVOLUTION FOR OBJECT DETECTION

SUPPLEMENTARY MATERIAL

1. RELATED WORK

1.1. Attention-based Module

Some attention-based modules commonly utilize channel, spatial, or a combination of several to enhance feature representations [1, 2, 3, 4]. These modules aim to selectively highlight important features while suppressing irrelevant ones, improving the model's ability to handle complex visual patterns. Due to their effectiveness, such modules are frequently integrated into object detection tasks to refine feature maps and boost detection performance [5, 6, 7, 8, 9, 10]. Other approaches have focused on designing attention-based necks or heads, which directly participate in feature fusion or classification and regression tasks [11, 12, 13, 14]. Unlike methods that enhance feature maps indirectly, these techniques integrate attention mechanisms into the structural processes of object detection, aiming to refine task-specific representations and improve the alignment between features and detection objectives.

1.2. Hybrid Vision Backbone

Many studies suggest that hybrid vision backbones, which integrate self-attention mechanisms with convolutional operations, often outperform architectures based solely on transformers or CNNs in terms of overall performance. Some of them adopt a strategy of alternating self-attention and convolutional operations across different stages [15, 16, 17, 18]. This strategy divides the computational load, allowing convolution to focus on capturing local spatial features while self-attention handles global dependencies. In contrast, other recent state-of-the-art methods employ a more integrated approach, tightly combining convolution and self-attention within each block [19, 20, 21, 22]. This integration enables the simultaneous extraction of local and global features, enhancing feature interaction and alignment.

1.3. Vehicle Detection

Depending on the application context, vehicle detection systems can be categorized based on the source of their visual data. Vehicle detection from the driving view, such as that captured by dashcams or autonomous vehicle cameras, is crucial for advanced driver-assistance systems and autonomous driving [23, 24, 25]. Vehicle detection in aerial images and UAV videos constitutes a distinct subdomain within vehicle detection research [26, 27, 28]. Based on the above discussions, it is clear that vehicle detection from first-person perspectives is mainly suited for autonomous driving research due to its dynamic viewpoints and lack of fixed positions, which limits its applicability for traffic management and vehicle counting. Aerial images and UAV videos, while providing a broad and unobstructed view, come with high costs and generally do not face occlusion challenges. Several previous studies have also attempted vehicle detection using surveillance footage [29]. Segmentation-based methods have the problem that background modeling is difficult in crowded situations or at nighttime [30]. Other approaches use CNN-based vehicle detectors, but most of them are simply applications of general object detectors, such as YOLO [31] and Cascaded R-CNN [32].

2. SUPPLEMENTARY EXPERIMENTS

2.1. Implementation Details and Evaluation Metrics

All the experiments are conducted on an NVIDIA RTX A6000 GPU. The proposed VDC-YOLO and other detectors are implemented using MMYOLO and MMDetection. The Python version is 3.10, the CUDA version is 11.8, the Pytorch version is 2.0.0, and the torchvision version is 0.15.1. The proposed model is trained with SGD optimizer with the learning rate of 0.01, the input image size is 640×640 and the batch size is 12. In addition, we initialize the model with pretrained weights from the COCO dataset, which provided a strong baseline for transfer learning. All other models are evaluated using the same input image size and their default parameters. In our depthwise convolution, the bottleneck ratio is 0.25, group is 8.

In our experiments, we report mean Average Precision (mAP) as the key evaluation metric. Average Precision (AP) provides a measure of the detection accuracy of model by averaging the precision over multiple recall levels, mAP is the AP averaged over classes. We present results for both mAP, which calculates the average precision across IoU thresholds from 0.5 to 0.95, and mAP_{50}, which specifically reports precision at an IoU threshold of 0.5. Besides, considering different categories and and their varying sizes, we also report

Table 1. Comparison of the performance on MLITcctv

Methods	mAP(%)	mAP_{50}(%)	mAP_s(%)	mAP_m(%)	mAP_l(%)	Flops(G)	Parameters(M)	FPS(img/s)
Vfnet	29.6	46.9	15.8	28.6	56.9	48.0	32.7	38.0
Retinanet	29.4	42.0	10.8	25.4	57.2	72.0	55.5	56.4
Faster R-CNN	37.7	53.6	13.9	32.8	57.0	82.2	60.4	48.1
Cascade	35.6	47.8	14.8	29.0	47.0	90.9	69.4	38.2
YOLOv5	32.4	50.4	16.1	29.6	48.0	**8.0**	**7.5**	94.4
YOLOX	38.1	55.4	19.7	33.0	57.4	13.3	9.2	89.9
YOLOv8 (Baseline)	40.5	57.1	16.8	33.9	61.2	14.3	11.2	**106.0**
DINO-swin	41.3	55.3	20.7	39.0	60.1	-	47.0	-
TPH-YOLOv5	37.5	49.8	-	-	-	31.0	27.7	-
HIC-YOLOv5	38.9	53.9	-	-	-	16.6	18.9	-
Ours	**46.9**	**62.5**	**22.9**	**42.8**	**68.2**	17.2	11.9	80.4

Table 2. Comparison of the performance on BitVehicle

Methods	mAP(%)	Bus(%)	Microbus(%)	Minivan(%)	Suv(%)	Sedan(%)	Truck(%)	FPS(img/s)
Faster R-CNN	91.3	90.6	94.4	90.7	91.3	90.6	90.1	14.7
YOLOv2-vehicle	94.8	97.5	93.8	92.2	94.6	**98.5**	92.1	26.3
hog+vsm	92.6	91.7	87.7	89.8	**97.6**	97.6	90.0	-
YOLOv4-AF	83.5	91.6	64.3	65.4	82.3	97.4	96.0	-
Cascade	95.4	97.5	96.4	90.2	96.4	96.7	94.1	23.1
Ours	**97.0**	**98.4**	**97.3**	**96.3**	96.6	97.4	**96.4**	**55.7**

mAP_s, mAP_m, and mAP_l to assess detection performance across small, medium, and large objects.

mAP_s: Object area is less than 32^2 pixels.

mAP_m: Object area ranges between 32^2 and 96^2 pixels.

mAP_l: Object area exceeds 96^2 pixels.

In addition to accuracy, we report Flops (FLoating Point OperationS) and Parameters to assess the computational complexity of the model. Flops measure the total number of floating-point operations required for one forward pass through the model. Parameters include weights and biases that the model learns during training.

In the i2 Object dataset, the training set contains 6,297 images, while the validation set includes 811 images. Additionally, we provide a detailed report on the specific number of categories in the BitVehicle and MLITCCTV datasets in Table 3 and Table 4.

Table 3. MLITcctv Dataset

MLITcctv	Train	Valid	Test
Images	4,458	864	864
Car	9,736	1,896	1,988
Light Cargo	2,116	384	419
Bus	81	14	11
Cargo	2,889	557	558
Special Vehicle	175	32	36
Motorcycle	253	57	51
Bicycle	260	54	53
Person	511	88	81

Table 4. BitVehicle Dataset

BitVehicle	Train	Valid	Test
Images	8,372	1,478	1,478
Bus	481	77	77
Microbus	734	149	149
Minivan	415	61	61
Suv	1,177	215	215
Sedan	5,037	884	884
Truck	699	124	124

2.2. Extended Results

We present the results of our proposed method in comparison to other 10 detectors on MLITcctv in Table 1. For the BitVehicle in Table 2, we additionally report the mAP for each individual vehicle category. To further validate the effectiveness of the proposed method, we add GGMix to the resnet-based model to observe if is effective for resnet in Table 5.

We have discussed the impact of global-guided strategy in the paper. Additionally, we include an extended experimental results to report the individual effects of each guidance signal

Table 5. The performance of GGMix in Resnet-based model

Methods	**mAP**(%)	**mAP**$_{50}$(%)	**mAP**$_{\mathbf{s}}$(%)
Faster R-CNN	37.7	53.6	13.9
+GGMix	**38.4**	**53.8**	**14.5**
Cascade	35.6	47.8	12.7
+GGMix	**37.3**	**50.0**	**13.3**
Retinanet	29.4	42.0	10.8
+GGMix	**34.9**	**47.7**	**17.1**

on the model in Table 6.

Table 6. Global-guided ablation study for GGMix

Methods	**mAP**(%)	**mAP**$_{50}$(%)
w/o global-guided strategy	43.3	59.1
w/o mask	46.5	61.3
w/o offsets	45.9	61.1
w/o channel attention	46.3	61.5
w/o spatial attention	45.9	61.3
ours	**46.9**	**62.5**

2.3. Additional Visualization

We provide additional detection result comparisons in Fig. 1. Meanwhile, we present our detection results in challenging scenarios, including **occlusion**, **scale variation**, and **low-resolution** conditions, as shown in Fig. 2. These cases demonstrate the robustness of our method in handling complex real-world scenarios.

2.4. Detailed Discussion

By substituting the specific values ($C_{in} = 128$, $C_{out} = 128$, $H = W = 40$, bottleneck ratio $= 0.25$, groups $= 8$), the Flops of the three methods are calculated as follows. Compared to depthwise separable convolution, the computational complexity of our method is significantly reduced.

Convolution: Flops $= 40{\cdot}40{\cdot}128{\cdot}3^2 = {\cdot}40{\cdot}40{\cdot}128{\cdot}128{\cdot}9 \approx 23592.96 \times 10^4$

DSConv: Flops $= {\cdot}40 \cdot 40 \cdot 128 \cdot (3^2 + 128) = {\cdot}40 \cdot 40 \cdot 128 \cdot 137 \approx 2805.76 \times 10^4$

Our method: FLOPs $= {\cdot}40 \cdot 40 \cdot \left(128 \cdot 3^2 + 128 \cdot 32 + \frac{32 \cdot 128}{8}\right) = 40 \cdot 40 \cdot \left(1152 + 4096 + 1024\right) = 40 \cdot 40 \cdot 6272 \approx 1003.52 \times 10^4$

Reduction $= \frac{2805.76 - 1003.52}{2805.75} \times 100\% \approx 64\%$

3. REFERENCES

[1] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[2] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 713–13 722.

[3] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 534–11 542.

[4] Y. Liu, Z. Shao, and N. Hoffmann, "Global attention mechanism: Retain information to enhance channel-spatial interactions," *arXiv preprint arXiv:2112.05561*, 2021.

[5] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2778–2788, 2021.

[6] S. Tang, S. Zhang, and Y. Fang, "Hic-yolov5: Improved yolov5 for small object detection," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6614–6619, 2024.

[7] F. Li, H. Yan, and L. Shi, "Multi-scale coupled attention for visual object detection," *Scientific Reports*, vol. 14, no. 1, p. 11191, 2024.

[8] S. Ma, H. Lu, J. Liu, Y. Zhu, and P. Sang, "Layn: Lightweight multi-scale attention yolov8 network for small object detection," *IEEE Access*, 2024.

[9] J. Qu, Z. Tang, L. Zhang, Y. Zhang, and Z. Zhang, "Remote sensing small object detection network based on attention mechanism and multi-scale feature fusion," *Remote Sensing*, vol. 15, no. 11, p. 2728, 2023.

[10] Y. Cao, J. Bin, J. Hamari, E. Blasch, and Z. Liu, "Multimodal object detection by channel switching and spatial attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 403–411.

[11] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7373–7382, 2021.

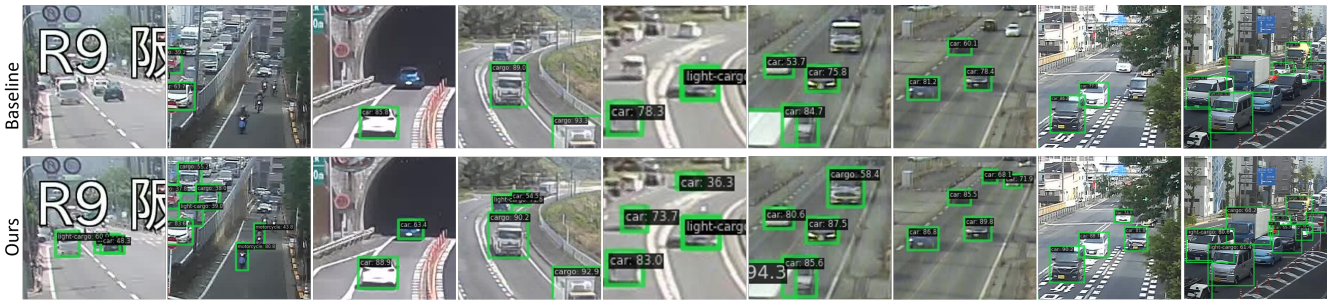

Fig. 1. These comparison shows that our method improved detection accuracy in many challenging scenarios.

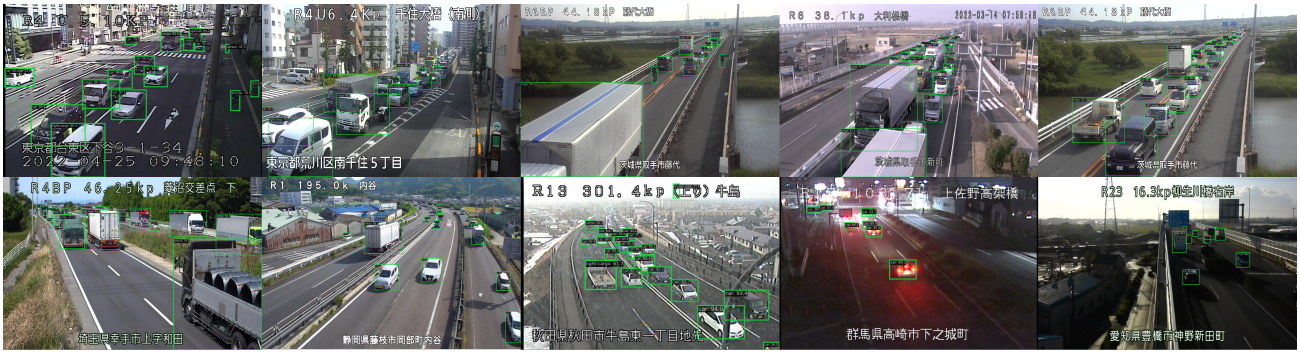

Fig. 2. Common challenges in traffic surveillance for object detection: i) Occlusion, where overlapping vehicles obscure key features; ii) Scale variation, caused by varying distances and angles from the camera; and iii) Low-resolution, particularly under poor lighting conditions, degrading feature quality. These issues significantly complicate accurate detection and classification.

[12] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, and L. Zhang, "Dynamic detr: End-to-end object detection with dynamic attention," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2988–2997.

[13] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos, "Towards universal object detection by domain attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7289–7298.

[14] J. Cao, Q. Chen, J. Guo, and R. Shi, "Attention-guided context feature pyramid network for object detection," *arXiv preprint arXiv:2005.11475*, 2020.

[15] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unifying convolution and self-attention for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12 581–12 600, 2023.

[16] W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6015–6026.

[17] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," in *European conference on computer vision*. Springer, 2022, pp. 459–479.

[18] C. Yang, S. Qiao, Q. Yu, X. Yuan, Y. Zhu, A. Yuille, H. Adam, and L.-C. Chen, "Moat: Alternating mobile convolution and attention brings strong vision models," in *The Eleventh International Conference on Learning Representations*, 2022.

[19] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 815–825.

[20] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 175–12 185.

[21] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 124–12 134.

[22] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 22–31.

[23] Y. Zhang, X. Song, M. Wang, T. Guan, J. Liu, Z. Wang, Y. Zhen, D. Zhang, and X. Gu, "Research on visual vehicle detection and tracking based on deep learning," *IOP Conference Series: Materials Science and Engineering*, vol. 892, no. 1, p. 012051, 2020.

[24] L. Chen, S. Lin, X. Lu, D. Cao, H. Wu, C. Guo, C. Liu, and F.-Y. Wang, "Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3234–3246, 2021.

[25] L. Chen, Q. Ding, Q. Zou, Z. Chen, and L. Li, "Denselightnet: A light-weight vehicle detection network for autonomous driving," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 600–10 609, 2020.

[26] J. Zhong, T. Lei, and G. Yao, "Robust vehicle detection in aerial images based on cascaded convolutional neural networks," *Sensors*, vol. 17, no. 12, p. 2720, 2017.

[27] X. Wang, "Vehicle image detection method using deep learning in uav video," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 8202535, 2022.

[28] S. Srivastava, S. Narayan, and S. Mittal, "A survey of deep learning techniques for vehicle detection from uav images," *Journal of Systems Architecture*, vol. 117, p. 102152, 2021.

[29] A. Boukerche, A. J. Siddiqui, and A. Mammeri, "Automated vehicle detection and classification: Models, methods, and techniques," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–39, 2017.

[30] S. Roy and M. S. Rahman, "Emergency vehicle detection on heavy traffic road from cctv footage using deep convolutional neural network," in *2019 international conference on electrical, computer and communication engineering (ECCE)*. IEEE, 2019, pp. 1–6.

[31] F. Li, Z. Wang, D. Nie, S. Zhang, X. Jiang, X. Zhao, and P. Hu, "Multi-camera vehicle tracking system for ai city challenge 2022," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3265–3273.

[32] X. Yang, J. Ye, J. Lu, C. Gong, M. Jiang, X. Lin, W. Zhang, X. Tan, Y. Li, X. Ye *et al.*, "Box-grained reranking matching for multi-camera multi-target tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3096–3106.