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

Beyond Limitations: Omni-DETR for Comprehensive Object Detection in Real-Time Applications

Jiantao Nie[1], Zuohua Ding[1], and Xiao Zhu[2(✉)]

[1] School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310000, Zhejiang, China
2023210602015@mails.zstu.edu.cn,zouhuading@hotmail.com
[2] Zhejiang Petroleum Comprehensive Energy, Hangzhou 310000, Zhejiang, China
875931598@qq.com

Abstract. Within the domain of autonomous driving, the performance of object detectors is crucial, as they must not only provide accurate and reliable environmental perception but also meet the demands for high inference speed and lightweight model requirements. While Transformer-based RT-DETR architecture exhibit notable efficiency and precision in real-time end-to-end detection frameworks, its capability to localize and classify small-scale objects remains limited. To address this issue, we introduce Omni-DETR, an advanced model aimed at enhancing detection accuracy for small objects without compromising efficiency. Omni-DETR incorporates FasterIRANet as its backbone for feature extraction, significantly reducing redundant computations and memory access, thereby achieving high inference speed while enhancing accuracy. In the encoder, our framework introduces the Dimensional Feature Integrator (DFI), a hierarchical feature fusion module that enhances multi-scale representation learning through cross-resolution feature aggregation. To further optimize localization precision, we develop InnerMPDIoU, an advanced bounding box regression loss combining geometric constraints and adaptive scaling mechanisms. Experimental results on the TT100K dataset demonstrate that Omni-DETR achieves an AP of 61.2% and a processing speed of 42.8 FPS on a 3090 GPU, while attaining 53.7% AP on the COCO dataset. Compared to several existing models, Omni-DETR proves its superiority in comprehensive performance.

Keywords: End-to-End Object Detection, Small Object, Transformer.

1 Introduction

Recent advancements in computer vision have significantly enhanced object detection capabilities, with particularly transformative impacts in time-sensitive domains such as autonomous navigation systems and live monitoring scenarios, where it plays a crucial role. These algorithms can identify and track surrounding vehicles, pedestrians, traffic signs, traffic lights, and lane markings in real time, providing autonomous vehicles with accurate perception of complex environments.

Contemporary object detection approaches predominantly fall into two paradigms. The first paradigm leverages convolutional neural networks (CNNs), exemplified by the YOLO series [1-5], renowned in industrial applications for their computational efficiency and detection precision. However, these algorithms often require non-maximum suppression (NMS) for post-processing, which adds computational overhead and introduces hyperparameters that can affect stability. An alternative paradigm employs Transformer architectures [6] to construct fully end-to-end detection systems, with the DETR family of models [7-14] representing prominent implementations of this approach. These models have simplified their architectures by removing post-processing steps like NMS, leading to significant advancements.

RT-DETR [14], an advancement in the DETR series, tackles the computational bottlenecks introduced by multi-scale feature processing by designing an efficient hybrid encoder. This design enhances cross-scale feature integration, substantially accelerating computational processing. The framework implements an uncertainty-aware query selection module that directly minimizes prediction variance during the initialization phase, resulting in more reliable object proposals and improved recognition performance. Notably, RT-DETR supports flexible speed adjustment without retraining, extending DETR's capabilities to real-time detection scenarios and demonstrating exceptional performance.

While RT-DETR achieves notable improvements in speed and accuracy for object detection through its key enhancements, our experiments indicate it still has limitations when it comes to detecting small objects. Specifically, the efficient hybrid encoder in RT-DETR primarily emphasizes the highest-scale features during multi-scale feature processing, while only performing basic fusion for other scales. Such processing risks discarding critical fine-grained information during resolution transitions, potentially compromising detection performance for small-scale targets.

To address this challenge, we have carefully reconsidered and redesigned RT-DETR's feature fusion module. We identified that optimizing the encoder's feature processing is essential for effectively enhancing small object detection. Our architectural enhancements prioritize robust cross-scale feature interactions through advanced fusion methodologies, enabling superior preservation of high-frequency spatial information while optimizing feature discriminability for compact objects.

In this paper, we introduce Omni-DETR, an enhanced and lightweight model based on RT-DETR that excels in small object detection scenarios. The key improvements of our model are as follows:

1. **High-Performance Backbone Network:** We have introduced FasterIRANet as the backbone for Omni-DETR. By streamlining processing steps and memory usage while utilizing Transformer's inherent ability to model distant dependencies, we optimized the extraction of spatial features.
2. **Optimized Multi-Scale Feature Fusion:** We introduce the Dimensional Feature Integrator (DFI), an innovative architectural module specifically developed for hierarchical feature fusion. DFI significantly improves multi-scale information aggregation within the network, enabling more comprehensive contextual representation that enhances detection performance.

3. **Optimized Bounding Box Regression:** To optimize both convergence rate and localization precision in bounding box prediction, we introduce an enhanced loss function, InnerMPDIoU, which combines the concepts of MPDIoU [15] and InnerIoU [16] to improve the model's detection accuracy.

These enhancements boost Omni-DETR's small object detection capability while preserving its computational efficiency and compact architecture, making it suitable for resource-constrained environments. Specifically, Omni-DETR demonstrates outstanding performance on the TT100K [17] dataset, achieving an AP of 61.2%. Experimental results demonstrate significant improvements over RT-DETR, achieving superior accuracy while maintaining real-time performance at 43 FPS (45% faster) on RTX 3090 GPUs. Furthermore, the model attains 53.7% AP on COCO benchmark, confirming its robust generalization capability across diverse datasets.

2 RELATED WORK

2.1 Real-time object detection

Real-time object detection has been a pivotal research area in computer vision, fueled by its extensive applications in autonomous driving, video surveillance, and robotics. The past two years have witnessed remarkable progress in real-time detection architectures, marked by three key developments: (1) compact network design, (2) computationally efficient feature representation, and (3) specialized enhancements for small object recognition. Driven by both hardware advancements and growing application demands, contemporary research has focused on optimizing the accuracy-efficiency trade-off. This trend is exemplified by successive YOLO series iterations (e.g., YOLOv8 [3], YOLOv10 [5]), which demonstrate progressively refined speed-accuracy characteristics through architectural innovations. Additionally, detectors based on the Transformer architecture, like RT-DETR, have garnered considerable attention for their performance in real-time tasks. By integrating dynamic feature enhancement modules and multi-scale feature fusion, these detectors have effectively improved detection efficiency and the recognition capabilities for small objects.

Emerging research has prioritized addressing challenges inherent to complex environments, including scenarios characterized by sparse data distributions and partial target occlusions. Representative works exemplify this progress through hybrid architectures combining parameter-efficient modules (GhostNet [18], FasterNet [19]) with enhanced pyramid networks [20]. These designs simultaneously address computational efficiency and scene complexity challenges. Concurrently, some studies have further optimized model inference speed through knowledge distillation [21], pruning [22], and quantization methods [23], promoting the implementation of real-time object detection technology in applications such as unmanned driving, intelligent security, and mobile devices.

2.2 End-to-End DEtection Transformer

DETR [7] established the first fully Transformer-based paradigm for object detection, significantly streamling the framework of object detection. Its intuitive structure and exceptional performance have garnered DETR considerable attention. However, the standard Transformer attention mechanism shows inherent constraints in handling visual feature representations, leading to DETR's suboptimal training convergence and slower inference speeds. To overcome these limitations, researchers have developed multiple DETR-based variants. For instance, Deformable DETR [8] modifies the attention mechanism to concentrate computations only on relevant sampling regions near reference points, outperforming the original DETR in accuracy while cutting training time significantly. Conditional DETR [11] generates spatially-aware queries from decoder embeddings, optimizing the multi-head cross-attention mechanism to localize classification and regression targets within refined spatial regions. This approach reduces reliance on content embeddings and simplifies the training process. DN-DETR [13] introduces a denoising training strategy where corrupted ground truth boxes are fed into the decoder, enabling the model to recover clean box predictions and achieve faster convergence.

Recently, RT-DETR has resolved the issue of slow inference speed by redesigning the encoder. Through the separation of intra-scale processing and inter-scale feature integration, RT-DETR has notably decreased inference latency and surpassed the YOLO series in precision and inference speed on the COCO dataset [24], representing a substantial leap forward in real-time object detection capabilities.

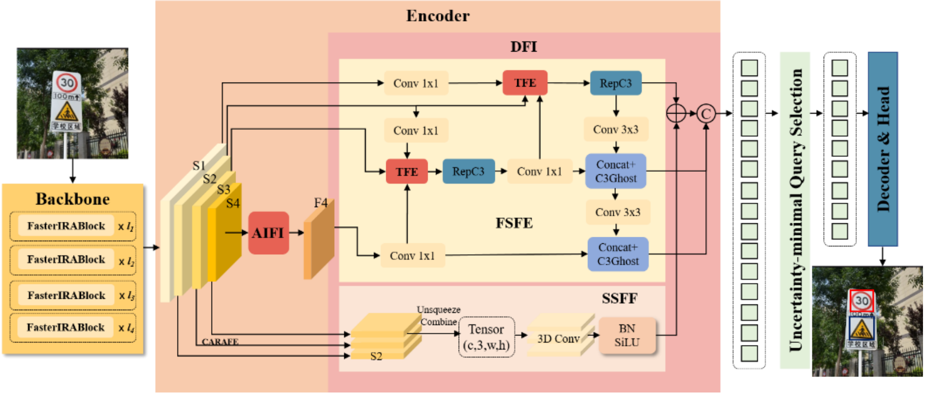

Fig. 1. Overview of Omni-DETR

3 DESIGN OF Omni-DETR

3.1 Model Overview

Omni-DETR's architectural framework integrates three core components: a lightweight FasterIRANet backbone for feature extraction, a hybrid encoder optimized for multi-scale interactions, and a Transformer decoder augmented with auxiliary prediction modules, as shown in Figure 1. Our model takes the output feature maps s1, s2, s3, s4 from the four stages of FasterIRANet as input to the encoder. The top-level feature map S4 is processed in the AIFI module [14] to generate feature map F4. The combined feature set S1, S2, S3, S4, F4 is then fed into the DFI module for further feature fusion. The DFI module integrates two synergistic components: the Scale Sequence Feature Fusion (SSFF) [25] and Fine-Scale Feature Extractor (FSFE) module (Figures 2 and 3). This hierarchical architecture facilitates comprehensive multi-scale feature aggregation, thereby enriching contextual representations. Following feature fusion, a uncertainty-aware query selection mechanism identifies the most informative encoder tokens to initialize decoder queries. Coupled with an auxiliary prediction branch, the decoder progressively refines these queries through iterative updates, ultimately generating both class predictions and bounding box coordinates.

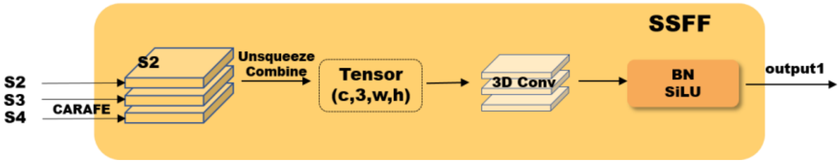

Fig. 2. Scale Sequence Feature Fusion Block

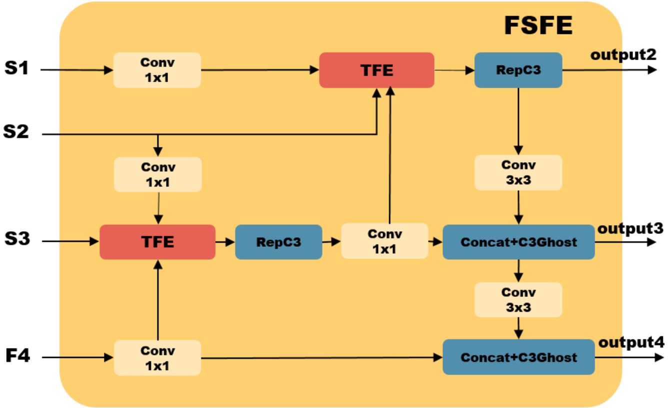

Fig. 3. Fine-Scale Feature Extractor Block

3.2 FastIRANet Backbone

Deploying object detection models on mobile devices typically requires achieving lightweight and high-speed inference under limited hardware conditions. On GPUs, the computational bottleneck often lies in memory bandwidth. High data read/write operations, coupled with the memory bandwidth limitations of GPUs, lead to models spending a significant amount of time on data read/write in video memory. To address these issues, we proposed an efficient backbone network called FasterIRANet. The architecture comprises four hierarchical stages, each preceded by either an embedding layer or merging layer for spatial downsampling or channel dimension expansion. Every stage contains multiple FasterIRABlocks (Figure 4) - each integrating: (1) a Partial Convolution (PConv) [19] layer for local feature extraction and (2) an Inverted Residual Attention Block (IRAB) that synergizes CNN-like efficiency with Transformer-style global modeling capabilities.

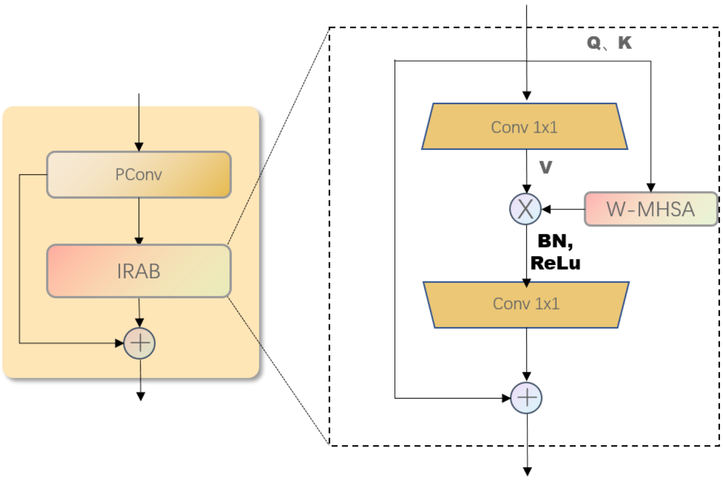

Fig. 4. The structure of the FasterIRABlock

The PConv layer performs standard convolution operations only on a strategically chosen portion of input channels for spatial feature extraction, maintaining the remaining channels in their original state. Compared to standard convolution, it reduces FLOPs and memory access. Following the PConv layer, we integrated the IRAB structure to leverage Transformer long-range interactions.The IRAB consists of a basic inverted residual structure and W-MHSA module [26]. Due to the expansion rate λ of the inverted residual module typically being greater than 1, directly applying W-MHSA would result in a quadratic λ increase in the number of parameters and computational load. Therefore, the IRAB structure calculates W-MHSA using the unexpanded feature maps, and the obtained weights are then applied to the expanded feature maps. Since W-MHSA is more suitable for modeling deeper semantic features, we only apply it in the third and fourth stages of FasterIRANet, which also reduces some parameters and computational load.The formula for Q, K, V is as follows:

$$Q = K = PConv(x)(\in \mathbb{R}^{C\times H\times W}) \tag{1}$$

$$V = MLP_{\lambda}\big(PConv(x)\big)(\in \mathbb{R}^{\lambda\, C\times H\times W}) \tag{2}$$

3.3 Dimensional Feature Integrator

Scale Sequence Feature Fusion. In object detection, a critical challenge is designing models that effectively detect targets of varying sizes, necessitating in-depth multi-scale feature fusion. While RT-DETR employs PANet-style [27] feature integration via basic additive or concatenative fusion, this approach inadequately captures cross-scale feature dependencies. Our solution, the SSFF module, innovatively arranges multi-scale features along the depth dimension and processes them through 3D convolutional filters [28] for enhanced inter-scale relationship modeling. We integrate the CARAFE [29] upsampling operator within the SSFF module to boost performance. CARAFE enhances feature reconstruction by aggregating adjacent channels and dynamically adjusts the upsampling process based on input features, generating more accurate upsampled features. The 3D convolution captures information across three spatial dimensions, aiding in recognizing multi-scale features of small targets and strengthening the model's learning capabilities, thus improving detection accuracy.

The SSFF module particularly emphasizes the importance of the S2 layer, as it has a smaller receptive field and contains information crucial for the detection of small targets. We opt for the S2 layer over the S1 layer due to its higher resolution and computational considerations.

Fine-Scale Feature Extractor. To improve small object detection performance, multi-scale analysis can be performed by examining shape and appearance variations across different resolutions. Conventional Feature Pyramid Networks (FPN) [20] typically neglect the fine-grained details present in high-resolution feature maps, instead relying solely on upsampling lower-resolution features followed by simple concatenation or element-wise addition with coarser layers. To address this limitation, we propose the Triple Feature Encoding (TFE) module, which explicitly processes feature maps at three distinct scales (large, medium, and small) while preserving critical high-frequency details through learned feature amplification. The detailed architecture of TFE is illustrated in Figure 5.

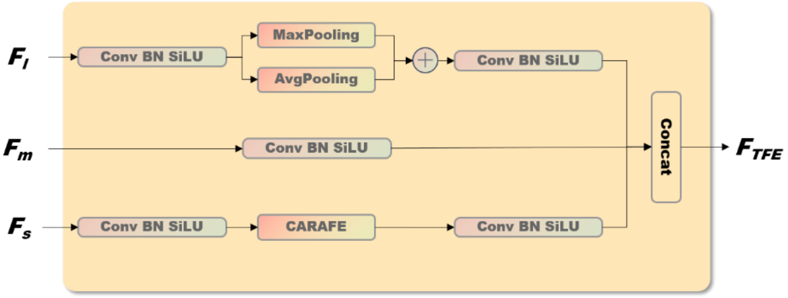

Fig. 5. The structure of TFE module

The TFE module first normalizes channel dimensions across scales to ensure feature compatibility. For high-resolution inputs, a hybrid downsampling strategy (joint max-average pooling) preserves fine-grained structures critical for small objects. Lower-resolution features undergo CARAFE-based upsampling to reconstruct localized detail while minimizing information loss. The module finally generates output $F_{\{TFE\}}$ through concatenation of these adaptively processed multi-scale features.

After the TFE module processes multi-scale features, we introduce two variants of the C3 structure (RepC3 and C3Ghost, as illustrated in Figure 6) for enhanced feature extraction. When the variant of the C3 structure is a RepBlock [30], it forms the RepC3 structure, and when it is a GhostBottleNeck [18], it forms the C3Ghost structure.

The RepBlock is a reparameterization convolution module [30] that enables the network to adopt different structures during training and inference. During training, it comprises multiple branches, including various 1x1 and 3x3 convolutions and identity mappings, to learn diverse features. For inference, the RepBlock's multi-branch structure is streamlined into a single branch, reparameterized as a 3x3 convolutional layer that integrates all feature information from the training phase, enhancing computational efficiency and speed.

The GhostBottleNeck generates additional feature maps through an efficient Ghost module [18], revealing deep information of intrinsic features at a low cost, which aids in constructing a more lightweight and efficient network structure. These extra Ghost feature maps are generated using convolutional kernels of 3x3 or 5x5 sizes with channel-wise operations, similar to DW-Conv [31]. The GhostBottleNeck's overall structure resembles the fundamental residual block found in ResNet [32], where the first Ghost module decreases channel quantity, and the subsequent module acts as an expansion layer to boost the channel count.

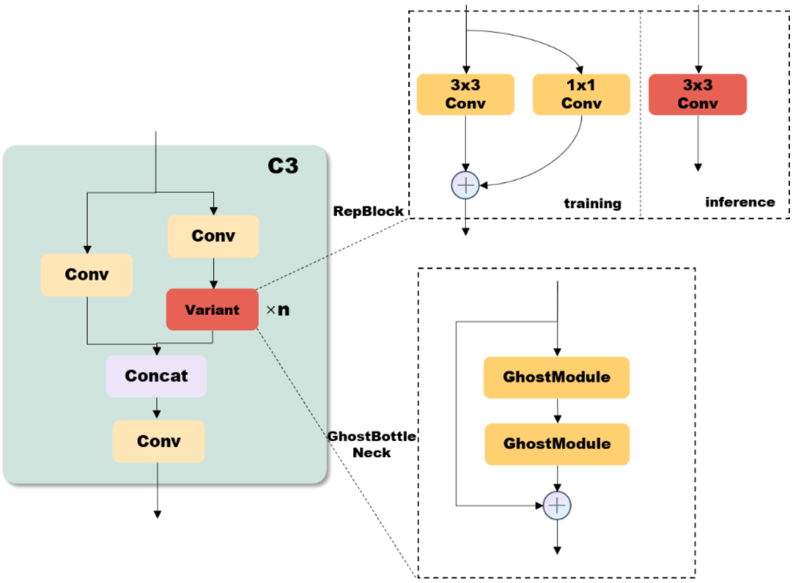

Fig. 6. The structure of C3 block

3.4 Loss Function

Bounding box regression loss functions play a critical role in object detection systems, where optimal design can significantly enhance model performance. Traditional approaches apply homogeneous penalty terms to localization errors regardless of their geometric properties, which may: (1) slow optimization convergence and (2) limit final prediction accuracy due to insufficient spatial discrimination. Therefore, we propose an improved loss function that combines the concepts of MPDIoU and InnerIoU to strengthen model performance. Bounding boxes are conventionally represented through the spatial coordinates of their top-left and bottom-right vertices. Leveraging the geometric properties of bounding boxes, MPDIoU minimizes the distance between the predicted bounding box and the corresponding corners of the Ground Truth box based on the IoU metric. In addition, InnerIoU employs auxiliary bounding boxes of varying scales for distinct regression instances, utilizing a scaling factor to regulate the dimensions of these boxes in the loss computation process. For instances with high IoU, employing a smaller auxiliary box for loss calculation can speed up convergence; conversely, a larger auxiliary box is appropriate for instances with low IoU. The formula for InnerIoU is as follows:

$$inter = B_{inner}^{gt} \cap B_{inner}^{pre} \tag{3}$$

$$union = (w^{gt\times} h^{gt}) \times ratio^2 + (w^{pre\times} h^{pre}) \times ratio^2 - inter \tag{4}$$

$$IoU^{inner} = \frac{inter}{union} \tag{5}$$

where B_{inner}^{gt} represents the auxiliary box obtained by scaling the Target box,B_{inner}^{pre} is the Anchor box after scaling, the scaling factor is $ratio$, typically ranging from [0.5-1.5]. The input image size is $h \times w$, and the corresponding bounding box regression loss is:

$$\mathcal{L}_{Inner-MPDIoU} = 1 - IoU^{inner} + \frac{d_1^2}{w^2+h^2} + \frac{d_2^2}{w^2+h^2} \tag{6}$$

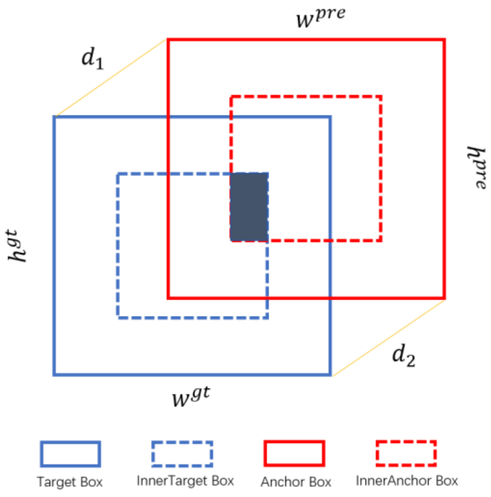

Fig. 7. Factors of Inner−MPDIoU

4 EXPERIMENTS

4.1 Dataset and Environment

To evaluate small object detection performance, we conducted comprehensive experiments on the TT100K benchmark. Furthermore, comparative analysis on the COCO dataset was performed to assess the model's generalization capability across diverse scenarios.

The TT100K dataset, developed collaboratively by Tsinghua University and Tencent Research, comprises 100,000 high-resolution traffic images containing approximately 30,000 annotated traffic sign instances. The TT100K dataset covers 221 unique categories of traffic signs, providing annotation data for 128 categories among them. Over 90% of the targets in this dataset occupy less than 5% of the image width and height, indicating that the dataset primarily consists of small-sized targets. This characteristic strongly corresponds with real-world traffic scenarios where signs typically appear as small objects within complex urban environments.

In terms of experimental environment setup, we used the GeForce RTX 3090 GPU. To ensure the comparability of model performance, the training parameters for all experiments were kept consistent, and no pre-trained weights were employed. Details of the specific parameter configurations are presented in Table 1.

Table 1. Setting table of experimental environment parameters

Item	Value
epochs	100
batch	16
imgsz	640
optimizer	AdamW
Lr0	0.001
momentum	0.9
weight_decay	0.0001
warmup_epochs	5.0
warmup_bias_lr	0.1

4.2 Comparison with Others

We performed comprehensive benchmarking of Omni-DETR against state-of-the-art approaches, including both the YOLO family and contemporary DETR-based detectors, with quantitative results detailed in Table 2 and Table 3.

Table 2 displays the comparative experimental results on the small-object dataset TT100K. The data clearly demonstrates that Omni-DETR achieves a high inference speed of 42.8 FPS, representing a 45% improvement over RT-DETR-1. Compared to existing real-time object detection models, Omni-DETR has achieved the highest scores of 78.6% in mAP50 and 61.2% in mAPval. This result highlights the model's exceptional performance in terms of accuracy in small object detection. When compared to the DETR series models, Omni-DETR has shown significant enhancements across all key metrics.

Table 2. Comparative Results on TT100K

Item	Params(M)	GFLOPs	FPS	mAP50	mAPval
YOLOv5m [2]	25.1	64.2	119.6	73.8	57.0
YOLOv8s [3]	11.2	28.6	126.1	70.2	54.2
YOLOv8m [3]	25.9	78.9	101.6	75.8	58.8
YOLOv9m [4]	20.1	76.8	69.8	75.7	58.6
YOLOv10m [5]	15.4	59.1	95.2	69.7	54.2
YOLOv10b [5]	19.6	91.8	87.5	76.8	60.0
RT-DETR-l [14]	32.1	103.6	29.5	57.5	42.1
RT-DETR-R50 [14]	42.0	125.8	25.1	62.9	46.4
RT-DETRv3-R18 [33]	20.0	60.0	48.2	59.6	43.7
RT-DETRv3-R50 [33]	42.0	134.5	25.7	63.6	46.9
LW-DETR-medium [34]	28.3	42.8	47.3	62.1	46.0
LW-DETR-large [34]	46.8	71.6	28.7	67.8	51.1
Omni-DETR (ours)	27.1	65.7	42.8	**78.6**	**61.2**

The data in Table 3 presents the comparative experimental results on the COCO dataset, showing that Omni-DETR achieves the highest mAPval of 53.7%, demonstrating its broad applicability in general scenarios.

Table 3. Comparative Results on COCO

Model	YOLOv8m	YOLOv10b	RT-DETR-v3-R50	LW-DETR-medium	**Omni-DETR (ours)**
AP	50.6	52.5	53.4	52.5	**53.7**

In summary, the Omni-DETR model not only demonstrates its advantage in computational efficiency but also surpasses several existing state-of-the-art models in detection accuracy. These results substantiate the effectiveness of Omni-DETR in object detection tasks and reveal its potential in application scenarios requiring real-time processing and high precision.

4.3 Ablation Study on Key Components

To systematically evaluate the contribution of each component, we conducted ablation experiments focusing on three critical aspects: (1) backbone architecture, (2) feature integration mechanism, and (3) loss function formulation.

Table 4. Result of the Ablation Experiments on Key Components

Variant	Faster-IRANet	DFI	Inner-MPDIoU	Params	Gflops	FPS	mAP50	mAPval
A(base)	×	×	×	32.1	103.6	27.8	57.5	42.1
B	✓	×	×	27.6	60.0	32.1	68.7	52.3
C	×	✓	×	28.7	80.1	30.7	74.1	56.0
D	×	×	✓	32.1	103.6	27.8	58.8	43.2
E	✓	✓	×	27.1	65.7	42.8	77.3	60.4
F	✓	×	✓	27.6	60.0	32.1	70.1	53.2
G	×	✓	✓	28.7	80.1	30.7	74.9	56.7
H(ours)	✓	✓	✓	27.1	65.7	42.8	**78.6**	**61.2**

Table 4 presents the outcomes of our ablation study, with baseline A representing the RT-DETR-l model, which utilizes HGNetv2 as its backbone network and CCFF as the feature fusion module. The comparison between variants G and H demonstrates that replacing HGNetv2 with FasterIRANet not only reduced the model parameters but also accelerated the inference speed, achieving a 3.7% increase in mAP50 and a 4.5% increase in mAPval. The comparison between variants F and H indicates that the model with the CCFF feature fusion module replaced by our proposed DFI feature fusion module experienced a slight increase in GFLOPs, yet there was an improvement in inference speed, with an 8.5% increase in mAP50 and an 8.0% increase in mAPval. This confirms that our DFI feature fusion module has a distinct advantage over the CCFF module in the domain of small object detection. The results of variant E show a slight enhancement in accuracy after the introduction of Inner-MPDIoU. The experimental outcomes for variants A, B, and C indicate that each module, when applied individually to baseline A, contributes to the enhancement of model performance.

4.4 Ablation Study on Backbone

Table 5 illustrates the impact of two main modules in the backbone network, PConv and IRAB, on model performance. Variant A indicates that each FasterIRABlock is composed of a PConv layer and a basic inverted residual block, while Variant B signifies that each FasterIRABlock consists solely of an enhanced inverted residual block, IRAB. Variant C represents a combination of PConv and IRMB in a single block. When the backbone network employs Variant A, the model still achieves good performance. Furthermore, we observed that when both modules are used together, the highest precision of 61.2% in mAPval is achieved, although the detection speed is somewhat reduced.

4.5 Ablation Study on DFI

Within the FSFE module, we implemented four instances of the C3 structure. The experimental results in Table 6 assess the influence of various C3 structure variations on performance metrics. The subjects of the experiment included RepC3 and C3Ghost, where Variant A indicates that all four positions utilize the RepC3 structure, Variant B indicates that all positions use C3Ghost, and Variant C represents a hybrid structure that employs RepC3 after the TFE module and C3Ghost after the Concat operation. Both Variants A and C achieved high levels of performance, while Variant B (which exclusively used C3Ghost) failed to converge during training. The experimental results demonstrate that the RepC3 structure can achieve high detection accuracy at a lower computational cost, while the C3Ghost structure is suitable for enhancing network non-linearity after feature fusion and is more lightweight compared to RepC3. In the final model, we adopted the configuration of Variant C to achieve optimal performance.

Table 5. Result of the Ablation Experiments on Backbone

Variant	PConv	W-MHSA	FPS	AP
A	✓	×	46.2	59.7
B	×	✓	44.2	50.2
C	✓	✓	42.8	61.2

Table 6. Result of the Ablation Experiments on DFI

Variant	Params	GFLOPs	AP
A	27.9	62.2	60.7
B	26.3	54,8	—
C	27.1	60.7	61.2

4.6 Ablation Study on Loss Function

The result in Table 7 demonstrate that the proposed Inner-MPDIoU loss function in this study achieves the best performance in both mAP50 and mAPval, reaching 78.6% and 61.2%, respectively, significantly outperforming other compared loss functions. Compared to traditional IoU-based loss functions (GIoU, DIoU, CIoU, EIoU), it shows consistent improvements, with the most substantial gain of 1.7% in mAP50 over GIoU. The experimental result validate the effectiveness of introducing Inner-MPDIoU.

Table 7. Result of the Ablation Experiments on Loss

Loss	$\mathcal{L}_{GIoU}$	$\mathcal{L}_{DIoU}$	$\mathcal{L}_{CIoU}$	$\mathcal{L}_{EIoU}$	$\mathcal{L}_{Inner-MPDIoU}$
mAP50	76.9	77.3	77.1	77.3	78.6
mAPval	60.4	60.5	60.5	60.6	61.2

5 CONCLUSION

We present Omni-DETR, an end-to-end object detection framework that achieves superior performance through three key innovations: (1) the FasterIRANet backbone for efficient feature extraction, reducing both parameter count and computational complexity; (2) the Dimensional Feature Integrator (DFI) module that overcomes RT-DETR's small-object detection limitations via advanced cross-scale fusion; and (3) a novel bounding box regression loss that improves localization accuracy.

Ablation experiments validated the effectiveness of each component in Omni-DETR, demonstrating that our model can achieve higher detection precision and speed while reducing computational costs. Compared to existing state-of-the-art object detection models, Omni-DETR has shown superior performance. Overall, Omni-DETR has proven its effectiveness and potential in practical applications through our experiments. We hope that Omni-DETR will be implemented in the field of real-time object detection.

References

1. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified,real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
2. Jocher, G.: YOLOv5 by Ultralytics (May 2020). https://doi.org/10.5281/zenodo.3908559, https://github.com/ultralytics/yolov5
3. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics
4. Wang, C.Y., Yeh, I.H., Liao, H.Y.M.: Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616 (2024)
5. 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)
6. Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
8. 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)
9. 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)
10. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based object detection. arXiv preprint arXiv:2109.07107 3(6) (2021)
11. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3651–3660 (2021)
12. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training.In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6748–6758 (2023)
13. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13619–13627 (2022)

14. 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. pp. 16965–16974 (2024)
15. Ma, S., Xu, Y.: Mpdiou: A loss for efficient and accurate bounding box regression (2023), https://arxiv.org/abs/2307.07662
16. Zhang, H., Xu, C., Zhang, S.: Inner-iou: More effective intersection over union loss with auxiliary bounding box (2023), https://arxiv.org/abs/2311.02877
17. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2110–2118 (2016)
18. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020)
19. Chen, J., Kao, S.h., He, H., Zhuo, W., Wen, S., Lee, C.H., Chan, S.H.G.: Run, don't walk: chasing higher flops for faster neural networks. In: Proceedings of theIEEE/CVF conference on computer vision and pattern recognition. pp. 12021–12031 (2023)
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
21. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015), https://arxiv.org/abs/1503.02531
22. Sun, M., Liu, Z., Bair, A., Kolter, J.Z.: A simple and effective pruning approach for large language models (2024), https://arxiv.org/abs/2306.11695
23. Rokh, B., Azarpeyvand, A., Khanteymoori, A.: A comprehensive survey on modelquantization for deep neural networks in image classification. ACM Transactions on Intelligent Systems and Technology 14(6), 1–50 (Nov 2023). https://doi.org/10.1145/3623402, http://dx.doi.org/10.1145/3623402
24. 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: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
25. Kang, M., Ting, C.M., Ting, F.F., Phan, R.C.W.: Asf-yolo: A novel yolo model with attentional scale sequence fusion for cell instance segmentation. Image and Vision Computing 147, 105057 (2024)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows (2021), https: //arxiv.org/abs/2103.14030
27. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation (2018), https://arxiv.org/abs/1803.01534
28. Singh, R.D., Mittal, A., Bhatia, R.K.: 3d convolutional neural network for object recognition: a review. Multimedia Tools and Applications 78, 15951–15995 (2019)
29. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: Carafe: Content-aware reassembly of features (2019), https://arxiv.org/abs/1905.02188
30. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again (2021), https://arxiv.org/abs/2101.03697
31. Guo, Y., Li, Y., Wang, L., Rosing, T.: Depthwise convolution is all you need for learning multiple visual domains. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8368–8375 (2019)

32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
33. Wang, S., Xia, C., Lv, F., Shi, Y.: Rt-detrv3: Real-time end-to-end object detection with hierarchical dense positive supervision (2024), https://arxiv.org/abs/2409.08475
34. Chen, Q., Su, X., Zhang, X., Wang, J., Chen, J., Shen, Y., Han, C., Chen, Z., Xu, W., Li, F., Zhang, S., Yao, K., Ding, E., Zhang, G., Wang, J.: Lw-detr: A transformer replacement to yolo for real-time detection (2024), https://arxiv.org/abs/2406.03459