A Robust RT-DETR-Based Method for Complex Self-Service Buffet Scene Detection

Zhengwang Xu, Hongyang Xiao, Zhou Huang*
School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China

Abstract*—Object detection in buffet-style environments is highly challenging due to densely stacked tableware, frequent occlusions, strong illumination reflections, and substantial visual similarity across categories, all of which undermine the robustness of existing detectors. To address these issues, this paper proposes an improved real-time detection transformer–based model with a lightweight design while significantly enhancing multi-scale feature representation. First, a re-parameterized stem module is introduced to strengthen shallow texture extraction with negligible computational overhead. Second, a dynamic multi-kernel refinement module is developed to enrich directional texture modeling and cross-scale semantic aggregation. Furthermore, a heterogeneous-kernel feature pyramid network is constructed by integrating adaptive multi-scale fusion, multi-kernel fusion nodes, and a lightweight upsampling strategy to improve cross-level feature consistency and mitigate aliasing caused by conventional upsampling. Experimental results on a self-constructed buffet-scene dataset demonstrate that the proposed method improves mAP$_{50}$ and mAP$_{50:95}$ by 2.6% and 1.9%, respectively, while reducing parameters and GFLOPs by 42.6% and 42.3%, and increasing inference speed to 103.1 FPS. On Dota v1.0 and SkyFusion data sets, the small target detection ability has also been improved. The substantial reductions in computation and model size further confirm the effectiveness and practical value of the proposed approach for complex catering scenarios.

Keywords*—*RT-DETR; lightweight object detection; multi-scale feature fusion; attention enhancement; buffet-scene perception

I. Introduction

Object detection in buffet scenarios is a fundamental component of intelligent catering systems, and its results directly affect key processes such as dish consumption analysis, automatic replenishment scheduling, and checkout verification. However, compared with ordinary scenes, buffet environments are characterized by densely stacked tableware, frequent occlusions caused by serving actions, strong reflections from metal or ceramic surfaces, and high visual similarity between categories. These factors make small and overlapping objects highly prone to missed detections; highlighted regions disrupt texture consistency, while visually similar categories lead to an increase in false positives. Therefore, achieving high-precision, robust, and real-time object detection in complex buffet scenarios remains a major challenge for intelligent catering systems.

In recent years, research on object detection in dense scenes, small-object detection, and complex illumination conditions has continued to advance [1]. On the one hand, Wang *et al.* proposed YOLOv10, which removes non-maximum suppression and performs holistic end-to-end architectural optimization, achieving leading accuracy across different model scales on the COCO benchmark [2]. On the other hand, Wang *et al.* introduced RODD, a dedicated benchmark for reflected-object detection, and demonstrated that multiple state-of-the-art detectors still suffer from noticeable performance degradation under reflective-surface scenarios [3]; meanwhile, a NeurIPS 2024 study proposed illumination-invariant feature learning for low-light object detection and reported consistent gains when integrated into existing detection frameworks, indicating that illumination variation remains a non-trivial bottleneck [4]. In addition, two-stage detection frameworks are often limited by their high inference cost in resource-constrained deployments; for example, Liu *et al.* proposed simplification strategies for on-device inference of two-stage detectors to reduce computational complexity [5]. Transformer-based detectors, such as DETR [6] and Deformable DETR [7], model global context via self-attention and thus exhibit stronger robustness in complex scenes. However, standard DETR still suffers from slow convergence and unstable matching. Although Zhao *et al.* proposed RT-DETR, which enables real-time end-to-end transformer detection through structural simplification [8], subsequent studies have continued to improve its training strategy and small-object detection capability. Overall, these advances suggest that, for densely stacked and heavily occluded catering scenarios, there remains a pressing need for new detection solutions that strengthen shallow spatial feature representation and cross-scale feature fusion.

To cope with the above challenges, existing studies have proposed various improvement strategies from the perspectives of backbone enhancement, multi-scale fusion design, and attention mechanism optimization. However, these methods often focus on a single aspect, such as strengthening convolutional representation to improve fine details, enhancing feature pyramids to boost small-object detection, or introducing sparse attention to mitigate background interference. For buffet tableware detection scenarios where density, small-object scale, strong reflections, and occlusions coexist, a systematic solution is still lacking. In particular, under the constraints of lightweight design and real-time inference, how to enhance spatial detail modeling and cross-scale feature consistency within a limited computational budget remains a key bottleneck.

In this work, we propose an improved model based on RT-DETR as the baseline, specifically tailored for complex buffet scenarios. The main contributions of this paper are summarized as follows:

- We propose a lightweight, end-to-end detection framework tailored for complex self-service buffet scenarios, where dense stacking, severe occlusion, specular reflections, and high inter-class visual similarity com-

[*] Corresponding author.

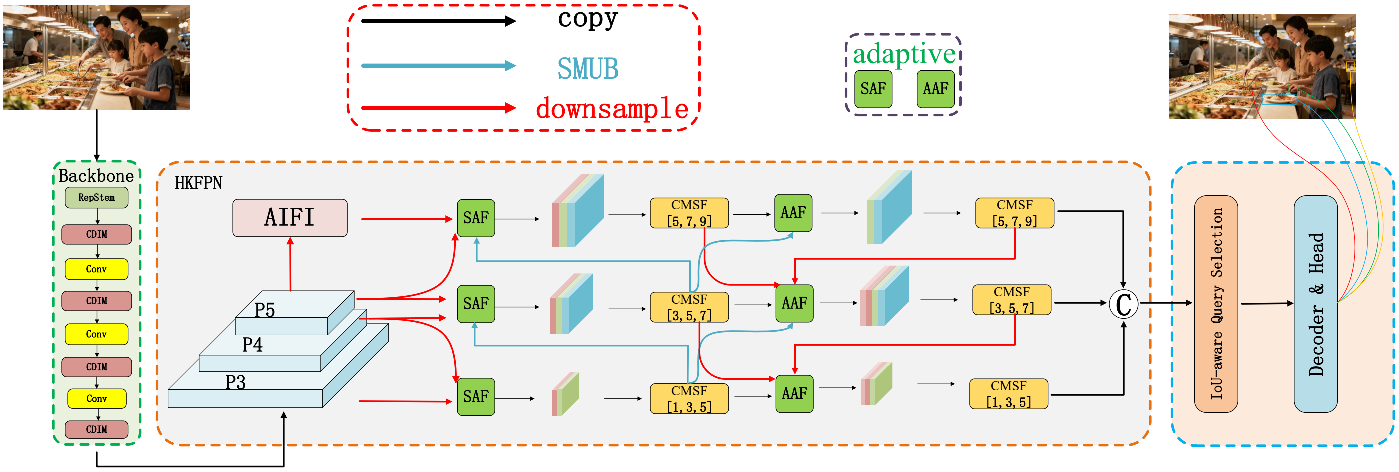


Fig. 1. Network structure of the proposed improved RT-DETR model.

monly coexist. The proposed framework is designed in a system-oriented manner, jointly optimizing feature extraction, multi-scale representation, and fusion efficiency to enable robust real-time detection.

- To enhance fine-grained discrimination and scale-aware representation under dense stacking and reflection interference, we design a lightweight backbone by integrating RepStem [9] with the proposed CDIM (Cross-Scale Fusion Dynamic Dual-Depth Inception Module). As a scene-driven enhancement unit, CDIM combines dynamic dual-path convolution with an adaptive Inception-style kernel selection mechanism, strengthening local texture perception and cross-scale context modeling while maintaining low computational overhead.

- We further construct HKFPN (Heterogeneous Kernel Feature Pyramid Network) as a scale-aware pyramid fusion architecture specifically for buffet scenes. HKFPN introduces heterogeneous large-kernel selection across pyramid levels and incorporates CMSF (Cross-stage Multi-Scale Fusion) and SMUB (Shift-Channel Mixed Upsampling Block) as collaborative fusion nodes together with an adaptive fusion mechanism [10]. This coordinated design mitigates feature aliasing and detail loss induced by conventional upsampling, thereby improving feature consistency and detection robustness for small, medium, and large objects in cluttered environments.

The remainder of this paper is organized as follows. Section II reviews related work on object detection in dense scenes, small-object detection, and transformer-based detectors. Section III describes the proposed method in detail, including the overall architecture and key modules. Section IV presents extensive experimental results and comparisons with state-of-the-art methods. Finally, Section V discusses the limitations of the proposed approach and outlines directions for future work, while Section VI concludes the paper.

II. Related Work

A. Object Detection in Dense and Complex Catering Scenarios

In dense near-field environments such as retail shelf recognition, unmanned vending systems, kitchen scene monitoring, and buffet tableware sorting, objects are often closely arranged, visually similar in shape, and heavily occluded. Traditional detectors tend to degrade significantly under such conditions. Goldman et al. constructed the SKU-110K densely shelf benchmark and demonstrated that mainstream detectors suffer notable performance degradation under homogeneous and crowded target distributions [11]. YOLO-based kitchen perception systems have achieved real-time recognition of cookware and cooking states, validating the applicability of lightweight detectors under strong reflections and complex illumination [12]. In buffet tableware recycling and sorting applications, RGB-D detection has been employed for tableware localization and grasping, providing engineering-oriented solutions to stacking and occlusion problems [13], [14]. For ceramic tableware surface defect detection, domain-specific datasets have been proposed, and YOLOv8-based frameworks have been developed. However, multi-scale small defects remain challenging for detectors, and reflective or patterned surfaces easily cause false detections or missed detections, underscoring the need for stronger fine-grained feature representation and targeted augmentations [15].

These studies collectively show that complex illumination, dense arrangement, and strong visual similarity are the primary challenges in tableware detection tasks, requiring more robust fine-grained feature modeling and cross-scale information fusion.

B. Improvements on RT-DETR

Research on improving RT-DETR mainly focuses on backbone enhancement, multi-scale fusion strengthening, attention optimization, and supervision strategy design. In terms of backbone enhancement, some works incorporate more efficient convolutional structures such as PConv [16] and FasterNet [17] to improve low-level texture and edge modeling, thereby enhancing small-object separability in dense scenes.

TABLE I. Comparison of Related Studies and their Strengths and Limitations

Line of Work	Representative Works	Strengths	Limitations in Buffet Scenes
Dense benchmark analysis	SKU-110K [11]	Quantifies degradation in dense, homogeneous layouts.	Omits buffet-specific factors like specular highlights, fine-grained similarity.
Real-time kitchen perception	Kitchen YOLO [12]	Real-time recognition under complex illumination.	Not optimized for dense stacking and small objects; prone to confusion under occlusion.
RGB-D sorting and grasping	RGB-D pipelines [13], [14]	Depth cues improve robustness to occlusion and stacking.	Requires extra sensors and is less scalable.
Defect inspection on ceramics	Ceramic defect detection [15]	Effective for localized surface anomaly recognition.	Different objective from multi-class instance detection.

For multi-scale fusion, structures such as FPN [18], PANet [19], and BiFPN [20] are widely used to improve cross-layer feature consistency. Meanwhile, ASFF [21] and CARAFE [22] enhance small-object representation through adaptive weighting or content-aware upsampling, effectively mitigating feature aliasing introduced by conventional upsampling.

Regarding attention mechanism optimization, lightweight modules such as CBAM [23], Coordinate Attention [24], and HiLo Attention [25] strengthen spatial feature selection and global contextual representation, helping maintain stable feature modeling under complex background interference.

Despite the encouraging progress, existing approaches still exhibit clear limitations when deployed in buffet-like catering environments. A comparative analysis of their strengths and weaknesses is summarized in Table I.

III. Method

Building upon the original structure of RT-DETR, this work introduces a systematic lightweight design and a series of detection-performance enhancement strategies, resulting in an improved model. Without significantly increasing computational cost, we incorporate lightweight modules to strengthen multi-scale feature extraction and boost fine-grained representation capability. In the backbone stage, a lightweight RepStem module is introduced to reduce redundant parameters while enhancing shallow feature discrimination. For deep feature extraction, the proposed CDIM module reinforces multi-directional texture modeling and cross-scale semantic interaction. Furthermore, to address the limitations of traditional feature pyramids, we design a heterogeneous kernel fusion network, HKFPN, which integrates adaptive multi-scale fusion and ensures stronger cross-layer feature consistency. The overall architecture is illustrated in Fig. 1.

A. Lightweight RepStem Backbone

In restaurant scenarios, images often suffer from uneven illumination, specular reflection, and severe edge blur, which cause the traditional stem layer in RT-DETR to struggle in capturing fine-grained details and maintaining a balance between local and global information. Although the conventional downsampling design is simple and efficient, it provides insufficient shallow feature extraction during the early stages and cannot fully support real-time deployment when deeper layers introduce substantial computational cost. To address this issue, a lightweight RepStem module is introduced at the front end of the network to enhance shallow feature representation while improving inference efficiency. The structural differences relative to the traditional stem layer are shown in Fig. 2.

The core idea of the RepStem design is as follows. During the training stage, a multi-branch structure is adopted for three convolutional layers, maintaining a 4× downsampling rate while dynamically learning multi-branch feature enhancement and high-efficiency fusion. The process is illustrated in Eq. (1) and Eq. (2), where the fused shallow features capture richer edge information and directional detail cues.

$$W_{\text{fused}} = \sum_{i=1}^{n} W_i * Pad_i \tag{1}$$

$$b_{\text{fused}} = \sum_{i=1}^{n} b_i \tag{2}$$

During inference, the multi-branch parameters are equivalently merged into a single 3×3 convolution kernel, enabling improved feature diversity without increasing computational overhead. This mechanism can be viewed as a “train-time over-parameterization, inference-time simplification” design, providing stronger feature modeling during the learning stage while maintaining lightweight computation during deployment. The fused convolutional operation is shown in Eq. (3).

$$y = \text{Conv}(x, W_{\text{fused}}) + b_{\text{fused}} \tag{3}$$

Here, W_i denotes the convolution kernel of the i-th branch, Pad_i is the corresponding padding matrix used to ensure unified kernel sizes, and b_i is the bias term of the i-th branch. W_{fused} and b_{fused} represent the equivalent fused kernel and bias after branch merging.

Consequently, the RepStem module achieves a 4× downsampling ratio by applying two stride-2 convolutions in the first two layers, while the third convolution operates with stride 1 for feature refinement, as shown in Eq. (4) and Eq. (5):

$$x_{\text{stem}} = f_3(f_2(f_1(x))) \tag{4}$$

$$f_i(x) = \sigma(\text{BN}(\text{Conv}_i(x))) \tag{5}$$

where, $f_i(\cdot)$ denotes the convolutional mapping of the i-th layer, σ represents the activation function, and x_{stem} is the final output of the RepStem module.

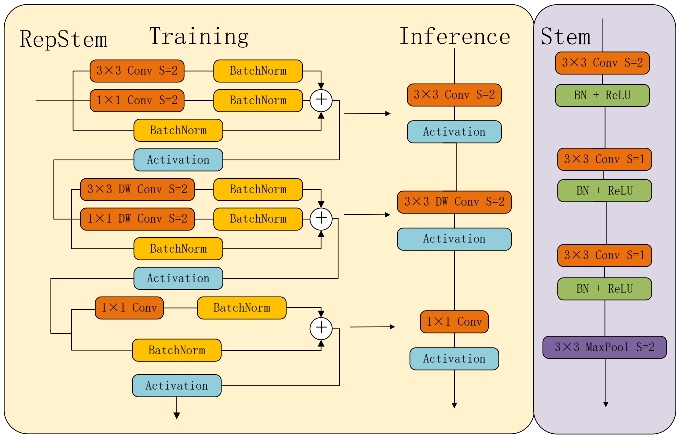


Fig. 2. Comparison between the proposed RepStem module and the original RT-DETR stem block. The left part illustrates the training- and inference-stage architectures of RepStem with multi-branch convolutions, while the right part shows the conventional ConvNorm-based stem composed of three stacked ConvNorm layers followed by a 3×3 max-pooling layer that achieves $4\times$ downsampling.

In RepStem, $f_1(\cdot)$ and $f_2(\cdot)$ are implemented as stride-2 convolutions for spatial downsampling, whereas $f_3(\cdot)$ uses stride 1 to further refine the representation without changing the resolution.

B. Dynamic Multi-Kernel Inception Refinement Block

Convolutional neural networks rely on fixed convolution kernels and static feature extraction mechanisms, which exhibit inherent limitations in many scenarios, particularly in complex restaurant environments where strong reflections, structural inconsistencies, and high inter-class similarity frequently occur. Traditional convolutions extract features using fixed kernel sizes and directions, and these receptive fields and weights remain fixed during inference. Such a static design often ignores the dynamic variations in image content, making it difficult for the model to handle multi-scale objects, strong reflections, occlusions, and fine-grained structures.

To address these challenges, this paper introduces a lightweight and dynamically adaptive feature refinement module, termed CDIM. This module enhances feature representation while preserving real-time performance and computational efficiency by adaptively selecting multi-kernel receptive fields and cross-depth feature fusion., as illustrated in Fig. 3.

At the bottom layer, D3ID designs three parallel depthwise convolution branches. To enable adaptive selection under different spatial contexts, the module introduces dynamic kernel weighting, allowing the network to adjust the contribution of each directional kernel based on the input content. Through lightweight reparameterization, the module generates dynamic selection weights for the three branches, and the final dynamic depthwise convolution is computed as in Eq. (6) and Eq. (7):

$$\alpha = \mathrm{Softmax}\left(W(\mathrm{AvgPool}(x))\right) \in \mathbb{R}^3 \tag{6}$$

$$\begin{aligned}\mathrm{DIDC}(x) = \alpha_s \mathrm{DW}_s(x) + \alpha_h \mathrm{DW}_h(x) \\ + \alpha_v \mathrm{DW}_v(x)\end{aligned} \tag{7}$$

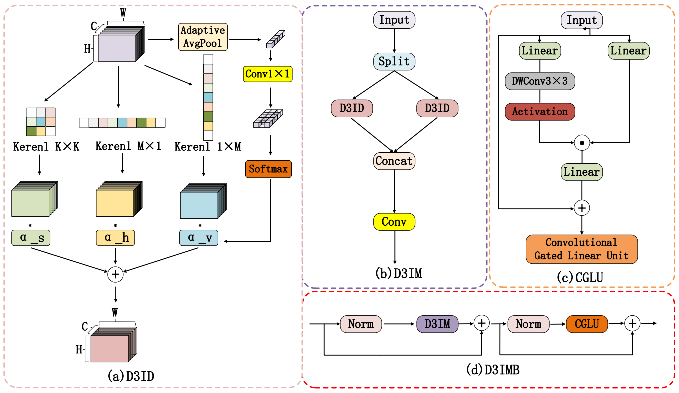


Fig. 3. CDIM is composed of several dynamic branches: Dynamic multi-depth convolution D3ID (Dynamic Dual-Depth Inception DWConv), dynamic mixing unit D3IM (Dynamic Dual-Depth Inception Mixer), CGLU (Convolutional Gated Linear Unit), and D3IMB (Dynamic Dual-Depth Inception Mixer Block). Combined with the CSP partition-and-fusion design, these components are integrated into a unified modular structure.

where, DW_s denotes square depthwise convolution, DW_h horizontal strip depthwise convolution, and DW_v vertical strip convolution. The input tensor $x \in \mathbb{R}^{B\times C\times H\times W}$ represents the given feature map.

On this basis, D3IM further splits the input along the channel dimension into multiple groups, extracting features with dynamic kernels of different receptive field sizes, and subsequently fusing them through a 1×1 convolution. This design aggregates multi-scale upper–lower semantic cues and enhances the model's capacity to jointly encode local and global information.

D3IM then combines the above dynamic paths with a residual connection and a CGLU-based gated dual-branch structure. The first branch employs Mixer to realize dynamic multi-kernel feature aggregation, while the second branch uses CGLU to perform lightweight nonlinear modulation and adaptive feature recalibration. A learnable *LayerScale* parameter is introduced to stabilize the optimization by gradually increasing the residual scaling during training. Additionally, an optional *DropPath* is applied to improve generalization. The fused output in the inference stage is computed as:

$$\begin{aligned}\mathrm{DCMB}(x) = x + \gamma_1 \mathrm{DIM}(\mathrm{BN}(x)) + \gamma_2 \mathrm{MLP}(\\ \mathrm{BN}(x + \gamma_1 \mathrm{DIM}(\mathrm{BN}(x))))\end{aligned} \tag{8}$$

where, γ_1 and γ_2 are learnable scaling coefficients.

C. High-Efficiency Multi-Scale Adaptive Bidirectional Feature Pyramid Network

To address the inconsistency of multi-scale feature representation caused by densely distributed small objects, rapid occlusions, and strong reflections in catering scenarios, and to further enhance cross-scale feature fusion capability while maintaining lightweight design and real-time performance, we propose a high-efficiency multi-branch multi-scale feature pyramid network, termed HKFPN.

The proposed structure is inspired by the overall design philosophy of the MAFPN module and integrates lightweight

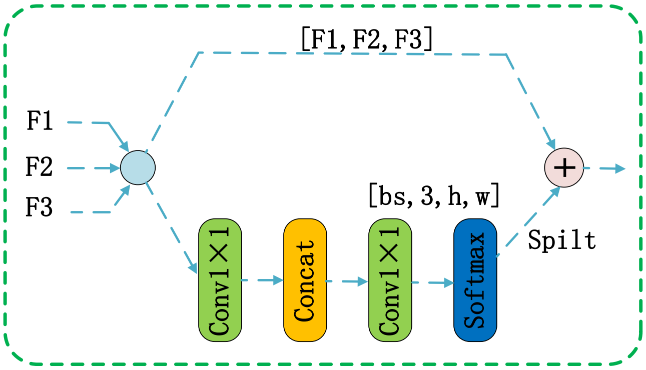


Fig. 4. Features $\{P3, P4, P5\}$ extracted from the backbone are first projected to aligned channel dimensions through pointwise convolutions, forming the three base features $\{F1, F2, F3\}$ that serve as the initial nodes of the HKFPN structure. On top of this, we construct lateral connections and cross-scale fusion pathways for each level, enabling the network to simultaneously receive high-level semantic cues and low-level spatial details.

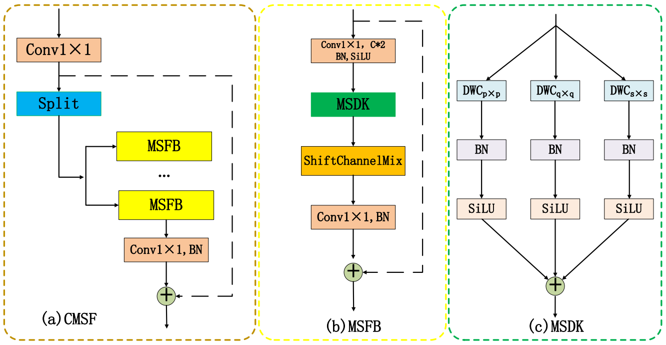


Fig. 5. (a) CMSF adopts parallel convolution branches of different kernel sizes. This design preserves the model's representational capability while reducing computational overhead and mitigating performance degradation in deep networks. (b) MSFB incorporates a Shift-Channel Mix operator to effectively rearrange channel information, achieving efficient feature interaction and cross-scale fusion. (c) MSDK employs multi-branch depthwise convolutions with different kernel sizes at different stages (e.g., {1,3,5}) to enhance scale sensitivity through parallel and depthwise aggregated feature extraction.

principles to perform comprehensive optimization on multi-scale feature fusion and upsampling pathways. HKFPN consists of three major components: an adaptive multi-scale feature fusion module, a multi-kernel aggregation node, and an efficient upsampling module. Together, these components construct a high-performance, scalable, and jointly global–local expressive feature fusion architecture.

1) Adaptive multi-scale feature fusion mechanism: HKFPN achieves bidirectional feature flow across levels P3–P5 through densely connected top-down and bottom-up pathways. Unlike traditional FPN and BiFPN, which rely on fixed-weight fusion for cross-scale aggregation, HKFPN introduces a spatially adaptive fusion strategy that dynamically evaluates feature importance across different scales and semantic levels at each spatial location. Specifically, the fusion weights are predicted per pixel and normalized across scale branches, enabling content-aware integration by assigning higher weights to the most informative scale for a given position. The overall structure is illustrated in Fig. 4.

Let $\{F_i\}_{i=1}^{3}$ denote the aligned multi-scale features to be fused, where $F_i \in \mathbb{R}^{B\times C\times H\times W}$. We first concatenate them along the channel dimension and apply two 1×1 convolutions to obtain the unnormalized fusion logits:

$$\begin{aligned} &S = \mathrm{Conv}_{1\times1}\Big(\mathrm{Conv}_{1\times1}\big(\mathrm{Concat}(F_1, F_2, F_3)\big)\Big), \\ &S \in \mathbb{R}^{B\times 3\times H\times W}. \end{aligned} \tag{9}$$

A spatially adaptive weighting tensor $W \in \mathbb{R}^{B\times 3\times H\times W}$ is then obtained by applying $\mathrm{Softmax}$ along the scale (branch) dimension at each spatial location:

$$W_i(x,y) = \frac{\exp(S_i(x,y))}{\sum_{j=1}^{3}\exp(S_j(x,y))}, \qquad i \in \{1,2,3\}, \tag{10}$$

which enforces $\sum_{i=1}^{3} W_i(x,y) = 1$ for every (x,y).

The final fused feature map $F_{\text{out}} \in \mathbb{R}^{B\times C\times H\times W}$ is computed by a per-location weighted summation:

$$F_{\text{out}}(:,x,y) = \sum_{i=1}^{3} W_i(x,y)\, F_i(:,x,y), \tag{11}$$

where, $W_i(x,y)$ is a scalar weight shared across channels (i.e., broadcast along the channel dimension). This fusion strategy enables the network to dynamically adjust the contributions of different scales at each spatial location, thereby improving the joint encoding of spatial details and semantic representations.

2) Multi-scale large-kernel aggregation node: In the feature refinement stage following cross-scale fusion, we adopt the CMSF module as the basic aggregation unit.

The CMSF structure integrates the CSP paradigm with MSFB (Multi-Scale Fusion Block), enabling efficient multi-scale feature extraction by transmitting partial residual features and performing depth-wise multi-kernel aggregation. The overall structure is illustrated in Fig. 5.

The input features to the CMSF module are first split into two streams via a 1×1 convolution. One stream directly preserves the input features to avoid spatial information loss, while the other passes through a multi-scale MSFB branch to construct multi-scale receptive fields.

Finally, the two processed branches are concatenated channel-wise and subsequently fused by another 1×1 convolution.

In the MSFB module, the core operation is the Multi-Scale Depthwise Kernel Block (MSDK), which consists of three parallel depthwise convolution branches. At levels P3, P4, and P5, different kernel-selection strategies are adopted to adaptively extract multi-scale features. The Shift-Channel Mix operator is subsequently applied to enhance channel interaction, improving the integration of local and global features. The input is first expanded by a 1×1 convolution, followed by multi-scale depthwise convolutions and a final 1×1 compression layer to restore the channel dimension.

The MSFB and CMSF core computations can be formulated as:

$$\mathrm{MSFB}(x) = x + P_2\big(\mathrm{SCM}(\mathrm{MSDK}(P_1(x)))\big) \tag{12}$$

$$\mathrm{CMSF}(x) = \mathrm{Concat}(x_1,\ \mathrm{MSFB}(x_2))\big) \tag{13}$$

where, $P_1(\cdot)$ denotes the $1{\times}1$ convolution for channel expansion, $P_2(\cdot)$ is the $1{\times}1$ projection convolution, $\mathrm{SCM}(\cdot)$ represents the Shift-Channel Mix operator, $[x_1, x_2]$ are the two branches in the CSP structure, and $P(\cdot)$ is the final $1{\times}1$ fusion convolution at the CSP output.

3) SMUB: Lightweight upsampling module-based on channel shifting: In feature pyramid networks, the upsampling operation is a key step for establishing cross-level interactions. However, traditional upsampling methods often suffer from feature misalignment and semantic discontinuity, and their computational cost increases significantly when aiming to preserve high-resolution details. In CMSFB, the SMUB is constructed by improving upon the EUCB module in EMCAD [26] and the ShiftChannelMix operator in BHViT [27], integrating a channel-shifting mixed strategy to achieve a unified design that enhances feature representation while maintaining lightweight computation. Under the constraint of low computational complexity, SMUB effectively alleviates the feature inconsistency introduced during upsampling and improves spatial continuity and semantic coherence in high-resolution features. The overall structure is illustrated in Fig. 6.

Its computation pipeline is formulated as:

$$x_{\mathrm{up}} = \mathrm{DWConv}(\mathrm{Upsample}(x)) \tag{14}$$

$$x_{\mathrm{mix}} = \mathrm{ShiftChannelMix}(x_{\mathrm{up}}) \tag{15}$$

$$x_{\mathrm{out}} = \mathrm{Conv}_{1\times1}(x_{\mathrm{mix}}) \tag{16}$$

The Shift Channel Mix module redistributes information across channels by cyclically shifting subsets of channels, allowing different channel groups to perceive information from different spatial positions. This improves the feature representation ability without introducing computational overhead or learnable parameters. By reordering features via groupwise channel shifting, the module enhances cross-channel interaction. Its computation is expressed as:

$$x_{\mathrm{mix}}(c, h, w) = x\big((c + \Delta_c) \bmod C,\ h,\ w\big) \tag{17}$$

where, C is the total number of channels and Δ_c denotes the shift step size.

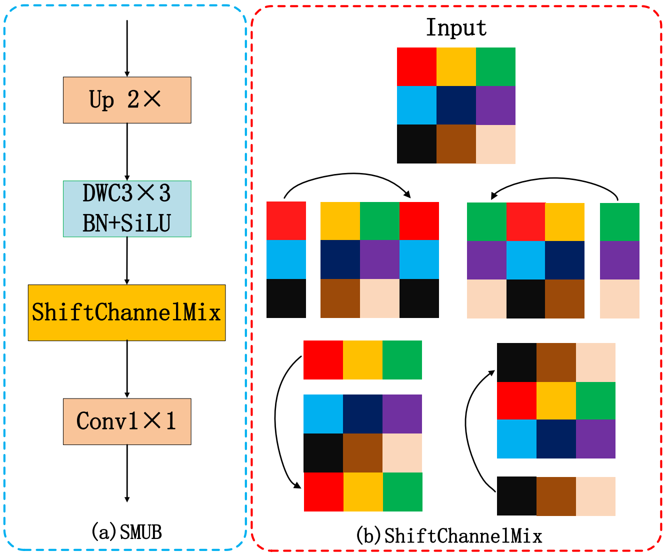


Fig. 6. (a) The upsampling depthwise convolution layer first performs 2× upsampling on the input feature and applies depthwise convolutions for local structure refinement, thereby enhancing spatial consistency under enlarged resolutions. The output is then fed into the Shift Channel Mix module for further feature enhancement, and subsequently compressed by a pointwise convolution for efficient channel reduction. (b) The Shift Channel Mix module divides the input feature into four equal groups along the channel dimension, followed by cyclic shifts in the positive and negative horizontal and vertical directions. The shifted features are finally concatenated and fused to achieve cross-channel information interaction and local feature refinement.

IV. Experimental Results and Analysis

A. Dataset

To support robust object detection in complex self-service buffet scenarios, we construct a composite tableware detection dataset by integrating publicly available data with self-collected samples into a unified benchmark. The dataset focuses on four object categories commonly encountered in buffet environments: cup, hand, plate, and tongs. We collect samples from complementary sources, including publicly released open-source datasets [28], [29], [30], [31], [32], publicly accessible royalty-free images gathered from online resources, and photographs captured by ourselves in real or near-real buffet environments. All self-collected images are manually annotated using LabelImg following the same annotation protocol as the public datasets and are subsequently incorporated into the final dataset.

To ensure consistency across heterogeneous sources, a unified preprocessing pipeline is applied to all data. Label normalization is performed to resolve naming inconsistencies and redundant labels across different datasets, such as *dish* versus *plate* and *hand* versus *human hand*.

All annotations are mapped to the four unified target categories, while labels irrelevant to the detection task are removed. During quality filtering, all images and corresponding annotations are manually inspected and cleaned, and low-quality samples with severe blur or excessive occlusion are discarded. Meanwhile, obvious annotation errors are corrected and invalid annotations are removed, and annotations from

TABLE II. Category-Wise Distribution of the Experimental Data Sets

Category	Number of Images
Hand	2415
Plate	1968
Cup	1756
Tongs	1570
Total	**7709**

heterogeneous formats are uniformly converted into the YOLO format.

The final dataset is split into training, validation, and testing subsets with a ratio of 7:2:1, while maintaining approximately balanced category distributions across splits. In total, the dataset contains 7,709 images, including 5,397 training samples, 1,531 validation samples, and 781 testing samples. The images span diverse viewpoints, brightness levels, and stacking complexities, providing a reliable basis for quantitative evaluation, and the category distribution of the constructed dataset is summarized in Table II.

Due to the wide variation in original image resolutions, all samples are resized to a fixed resolution of 640×640 during training to ensure input consistency and stable optimization while preserving aspect ratios as much as possible. Standard data augmentation techniques, including random flipping, color jittering, and affine transformations, are further applied to enhance model generalization under complex buffet conditions.

B. Experimental Settings

All experiments are conducted on an Ubuntu 22.04 operating system. The hardware configuration includes an Intel(R) Xeon(R) Platinum 8470Q CPU and an NVIDIA GeForce RTX 5090 GPU with 32 GB memory, and 90 GB system RAM. Python 3.12 and PyTorch 2.3.0 are used for deep-learning development, together with CUDA 12.1 as the GPU acceleration platform.

To ensure consistency of the experimental settings, the baseline RT-DETR is trained for 72 epochs following the official training schedule of its original implementation. In contrast, our model incorporates additional modules such as multi-scale feature fusion and adaptive weighting, which increase the optimization difficulty. Empirically, we observe that a longer training schedule is required to reach stable convergence. Therefore, we set the training duration of the proposed model to 300 epochs to ensure sufficient optimization while avoiding evident overfitting. During training, no optimization instability or divergence is observed, and both the training loss and validation performance exhibit smooth convergence trends. As shown in Fig. 7, the proposed method demonstrates stable convergence behavior in terms of mAP$_{50}$ and mAP$_{50:95}$. The batch size is set to 32, and the AdamW optimizer is adopted with an initial learning rate of 0.0001 and a weight decay of 0.0001.

C. Evaluation Metrics

In this study, we adopt several commonly used evaluation metrics, including mean Average Precision (mAP), floating-point operations per second (GFLOPs), number of parameters (Params), and frames per second (FPS). Among them, mAP serves as the primary indicator to evaluate detection performance across all categories and is defined as follows:

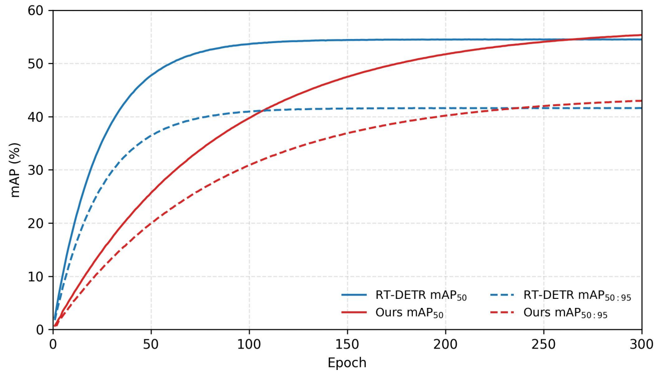


Fig. 7. Convergence curves of the baseline RT-DETR and the proposed method on the buffet tableware dataset.

$$mAP = \frac{\sum_{n=1}^{N} AP_n}{N} \tag{18}$$

D. Ablation Studies

To comprehensively evaluate the contribution of each proposed module to the overall detection performance, RT-DETR-r18 is adopted as the baseline model, upon which lightweight RepStem, the dynamic multi-kernel refinement module CDIM, and the heterogeneous multi-scale fusion network HKFPN are incrementally incorporated. All ablation experiments are conducted under identical training configurations and environmental settings. The results are summarized in Table III, where "✓" indicates that the corresponding module is enabled.

From the first row of results, it can be observed that replacing the original stem with the lightweight multi-branch feature extraction module RepStem leads to a notable improvement. Specifically, mAP$_{50}$ increases to 54.8%, and mAP$_{50:95}$ rises to 42.3%. Meanwhile, the parameter count is slightly reduced to 19.8M, and the inference speed increases to 87.5 FPS.

Introducing the D3IM module further enhances detection performance. With D3IM, mAP$_{50}$ is boosted to 55.1%, and mAP$_{50:95}$ reaches 41.9%. At the same time, the parameter size decreases to 13.1M, while inference speed increases to 96.2 FPS. This demonstrates that the multi-branch dynamic convolution mechanism of D3IM effectively strengthens the model's ability to capture fine-grained spatial cues while maintaining lightweight characteristics, making it particularly effective for small objects and complex fine-structure features.

After integrating the high-efficiency multi-scale fusion network HKFPN, the model achieves its best performance. mAP$_{50}$ increases to 56.9%, and mAP$_{50:95}$ improves to 44.2%. Meanwhile, the total parameters reduce to 11.6M, and the inference speed reaches 103.1 FPS, representing over 18% improvement compared with the baseline. HKFPN enables more adequate semantic communication and cross-scale contextual aggregation through efficient spatial grouping and multi-scale fusion strategies while mitigating the semantic misalignment commonly observed in traditional FPN structures. This results in a more balanced trade-off between accuracy and efficiency.

Overall, the incremental incorporation of each proposed module consistently contributes to the performance gains of the detection model. RepStem primarily enhances shallow feature extraction, D3IM significantly improves multi-scale perceptual capability, and HKFPN further strengthens cross-scale fusion and semantic aggregation, jointly leading to the best accuracy–efficiency balance.

E. Comparative Experiments

To further validate the effectiveness of the proposed method in practical detection tasks, we conduct a comprehensive comparison against various mainstream object detectors under the same hardware platform and dataset. The experimental results are summarized in Table IV.

Compared with conventional two-stage and one-stage detectors such as SSD and Faster R-CNN, our model achieves significantly higher detection accuracy under a lightweight architecture. With only 11.6M parameters and 33.8 GFLOPs, the proposed model attains 56.9% mAP_{50} and 44.2% $\text{mAP}_{50:95}$.

When compared with lightweight models such as MobileNet and EfficientDet, our method still demonstrates a more favorable trade-off between accuracy and efficiency. MobileNet has only 3.2M parameters and 5.4 GFLOPs, offering fast inference but insufficient detection precision. EfficientDet-D0, with 3.7M parameters and 3.9 GFLOPs, slightly improves efficiency, yet its mAP_{50} remains at 51.2%, indicating that the model struggles to simultaneously achieve both lightweight computation and high accuracy.

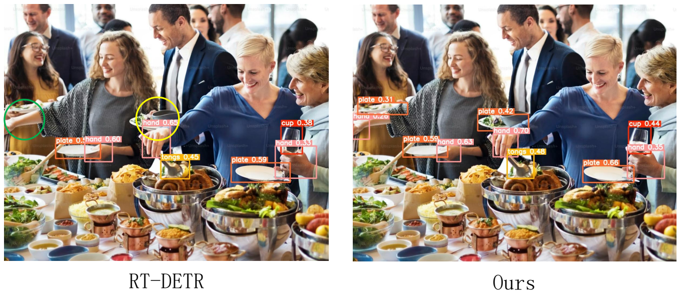
Compared with YOLO-series detectors, the proposed method further exhibits stronger accuracy advantages. YOLOv5s, YOLOv7-tiny, and YOLOv8s are representative lightweight detectors, which achieve mAP_{50} values of 53.1%, 53.4%, and 54.5%, respectively—still lower than the 56.9% achieved by our model. YOLOv8m improves accuracy to 55.6%, but its parameter size of 25.8M results in a slow inference speed of only 75.8 FPS, significantly lagging behind the 103.1 FPS achieved by our method.

It is also noteworthy that although YOLOv9s and YOLOv10s achieve mAP_{50} values of 55.8% and 55.1%, respectively, their inference speeds of 85.2 FPS and 74.9 FPS are not competitive. In contrast, our model maintains the highest inference speed 103.1 FPS while simultaneously achieving 56.9% mAP_{50} and 44.2% $\text{mAP}_{50:95}$, demonstrating a superior balance between accuracy and efficiency.

F. Generalization Experiments

To evaluate the detection performance and cross-dataset generalization ability of the proposed improved RT-DETR model on different types of small-object detection benchmarks, comparative experiments were conducted on the DOTA v1.0 and SkyFusion datasets, and the results are reported in Table V.

On the DOTA v1.0 dataset, compared with the original RT-DETR, the proposed method achieves improvements of 2.3% and 1.4% in mAP_{50} and $\text{mAP}_{50:95}$, respectively. On the SkyFusion dataset, the improved model also demonstrates stable performance gains, with mAP_{50} and $\text{mAP}_{50:95}$ increased by 0.8% and 0.9%, respectively. Although the absolute improvements are relatively modest, the consistent gains across different imaging conditions and data distributions indicate that the proposed method exhibits favorable cross-dataset generalization capability.



RT-DETR

Ours

Fig. 8. Qualitative comparison between RT-DETR (left) and the proposed method (right). Green and yellow circles mark two typical regions where the baseline model fails to detect objects while the proposed method successfully recognizes them with significantly higher confidence.

G. Visualization Analysis

To further verify the effectiveness of the proposed method in complex buffet scenarios, we compare the visual detection results between the baseline RT-DETR and our improved model. The qualitative analysis is shown in Fig. 8. RT-DETR exhibits multiple missed detections in small-object regions and partially occluded areas. For example, the hand holding a plate on the left side of the image is completely undetected by the baseline model, and the hand interacting with the serving tongs in the central region is only partially recognized, resulting in the loss of important semantic cues.

In contrast, our improved model successfully detects all of these targets, demonstrating stronger robustness to occlusion, illumination variations, and fine-grained cluttered environments. More importantly, the confidence scores also increase significantly. These results indicate that the enhanced multi-scale representation and adaptive fusion mechanisms effectively improve the discriminability of subtle structural details.

V. Discussion

Despite the promising performance achieved by the proposed method, several limitations remain. Performance can still degrade under extremely dense stacking or severe occlusion, where overlapping instances cause critical visual cues to be missing in monocular RGB images. Moreover, although robustness to reflective surfaces is improved, extremely strong specular highlights or saturated regions may still obscure textures and boundaries, and such reflection-induced information loss cannot be fully removed by a purely vision-based approach. In addition, the current evaluation is conducted on a specific buffet tableware distribution, and generalization to substantially different catering layouts, novel categories, or unseen material properties requires further validation. Finally, while the framework targets real-time deployment, the added modules inevitably introduce extra computational cost, and

TABLE III. ABLATION STUDY OF EACH PROPOSED MODULE

RT-DETR	RepStem	D3IM	HKFPN	mAP$_{50}$	mAP$_{50\text{-}95}$	GFLOPs	Params(M)	FPS
✓				54.5	41.6	58.6	20.2	86.9
✓	✓			54.8	42.3	51.7	19.8	87.5
✓	✓	✓		55.1	41.9	39.6	13.1	96.2
✓	✓	✓	✓	56.9	44.2	33.8	11.6	103.1

TABLE IV. COMPARISON WITH STATE-OF-THE-ART DETECTORS ON THE BUFFET TABLEWARE DATASET. ALL FPS RESULTS ARE EVALUATED ON AN NVIDIA RTX 5090 GPU FOR FAIR COMPARISON

Model	mAP50	mAP50-95	GFLOPs	Params(M)	FPS
SSD	48.3	28.8	24.3	30.6	52.9
Faster R-CNN	50.5	32.5	41.3	105.0	31.9
MobileNetV3	44.6	25.2	5.4	3.2	92.6
EfficientDet	51.2	35.4	3.9	3.7	87.1
YOLOv5s	53.1	37.6	24.0	9.1	119.3
YOLOv7-tiny	53.4	38.1	13.8	6.2	94.8
YOLOv8s	54.5	42.1	28.5	11.1	115.8
YOLOv8m	55.6	43.1	78.7	25.8	75.8
YOLOv9s	55.8	42.3	26.7	7.2	85.2
YOLOv10s	55.1	41.7	21.5	7.2	74.9
Ours	**56.9**	**44.2**	**33.8**	**11.6**	**103.1**

TABLE V. GENERALIZATION EXPERIMENTS ON DOTA V1.0 AND SKYFUSION DATASETS

Dataset	Model	mAP$_{50}$	mAP$_{50:95}$	Params (M)
DOTA v1.0	RT-DETR	54.3	34.9	20.2
	Ours	56.6	36.3	11.6
SkyFusion	RT-DETR	69.7	38.2	20.2
	Ours	70.5	39.1	11.6

ultra-low-power devices may benefit from further compression or architectural simplification. Future work will investigate stronger occlusion reasoning, improved robustness under extreme reflections, and enhanced cross-domain generalization while maintaining real-time efficiency.

VI. CONCLUSION

This paper investigates the challenge of robust object detection in complex buffet-scene environments, where dense object stacking, severe occlusion, strong specular reflections, and fine-grained visual similarity jointly degrade the performance of existing detectors. To address these issues, an enhanced RT-DETR-based detection framework is proposed, aiming to improve fine-grained feature representation and cross-scale robustness while preserving real-time efficiency. Through the integration of lightweight structural enhancements and adaptive multi-scale feature modeling, the proposed method effectively strengthens discriminative capability under dense and reflective conditions.

Extensive experiments on a self-service buffet tableware dataset demonstrate that the proposed approach achieves consistent performance improvements over the baseline RT-DETR and other representative detectors, particularly in challenging scenarios with heavy clutter and reflection. Additional cross-dataset evaluations further verify the generalization ability of the proposed model on different small-object detection benchmarks. Overall, this work provides a practical and scalable solution for intelligent catering perception, facilitating accurate and efficient object detection in real-world self-service restaurant environments.

Future research will focus on incorporating temporal information and optimizing deployment on edge devices to further enhance robustness and real-time performance in more diverse scenarios.

REFERENCES

[1] M. Nikouei, B. Baroutian, S. Nabavi, F. Taraghi, A. Aghaei, A. Sajedi, and M. E. Moghaddam, “Small object detection: A comprehensive survey on challenges, techniques and real-world applications,” *arXiv preprint arXiv:2503.20516*, 2025.

[2] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han *et al.*, “Yolov10: Real-time end-to-end object detection,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 984–108 011, 2024.

[3] Y. Wu, Z. Wang, Y. Wu, L. Huang, H. Zhou, and S. Li, “Towards reflected object detection: A benchmark,” *arXiv preprint arXiv:2407.05575*, 2024.

[4] M. Hong, S. Cheng, H. Huang, H. Fan, and S. Liu, “You only look around: Learning illumination-invariant feature for low-light object detection,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 87 136–87 158, 2024.

[5] J. Kang, H. Yang, and H. Kim, “Simplifying two-stage detectors for on-device inference in remote sensing,” *arXiv preprint arXiv:2404.07405*, 2024.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[7] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” 2020.

[8] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, “Detrs beat yolos on real-time object detection,” pp. 16 965–16 974, 2024.

[9] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “Repvgg: Making vgg-style convnets great again,” pp. 13 733–13 742, 2021.

[10] J. Doherty, B. Gardiner, E. Kerr, and N. Siddique, “Bifpn-yolo: One-stage object detection integrating bi-directional feature pyramid networks,” vol. 160. Elsevier, 2025, p. 111209.

[11] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, “Precise detection in densely packed scenes,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5227–5236.

[12] I. Azurmendi, E. Zulueta, J. M. Lopez-Guede, J. Azkarate, and M. González, “Cooktop sensing based on a yolo object detection algorithm,” *Sensors*, vol. 23, no. 5, p. 2780, 2023.

[13] D. Zhu, H. Seki, T. Tsuji, and T. Hiramitsu, “Tableware tidying-up robot system for self-service restaurant–detection and manipulation of leftover food and tableware,” *Sensors*, vol. 22, no. 18, p. 7006, 2022.

[14] R. Vermelho and L. A. Alexandre, “Grasping and sorting cutlery in an unconstrained environment with a 6 dof robotic arm and an rgb+ d camera,” in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2022, pp. 3–8.

[15] P. Sun, C. Hua, W. Ding, C. Hua, P. Liu, and Z. Lei, “Ceramic tableware surface defect detection based on deep learning,” *Engineering Applications of Artificial Intelligence*, vol. 141, p. 109723, 2025.

[16] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 85–100.

[17] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," pp. 12 021–12 031, 2023.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[19] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[20] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[21] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," 2019.

[22] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "Carafe: Content-aware reassembly of features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3007–3016.

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

[24] 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.

[25] Z. Pan, J. Cai, and B. Zhuang, "Fast vision transformers with hilo attention," vol. 35, 2022, pp. 14 541–14 554.

[26] M. M. Rahman, M. Munir, and R. Marculescu, "Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 769–11 779.

[27] T. Gao, Y. Zhang, Z. Zhang, H. Liu, K. Yin, C. Xu, and H. Kong, "Bhvit: Binarized hybrid vision transformer," pp. 3563–3572, 2025.

[28] Roboflow Community, "Plate count dataset," https://universe.roboflow.com/roboflow-54rk9/plate-count-g8dz1, October 2024, roboflow Universe, accessed January 14, 2026.

[29] yolov4tiny, "Hand dataset," https://universe.roboflow.com/yolov4tiny-wzb2k/hand-bo3ot, July 2025, roboflow Universe, accessed January 14, 2026.

[30] S. Ayman, "Cup dataset," 2023, kaggle Dataset, accessed January 14, 2026. [Online]. Available: https://www.kaggle.com/datasets/samuelayman/cup-dataset

[31] ICRL, "Cup detection dataset," https://universe.roboflow.com/icrl-jcblb/cup-detection-ne89f, October 2023, roboflow Universe, accessed January 14, 2026.

[32] Hanyang, "Tongs dataset," https://universe.roboflow.com/hanyang-ijbg8/tongs-irjgc, March 2023, roboflow Universe, accessed January 14, 2026.