Aligning Confidence and Localization: An Enhanced DINO Model for Small Object Detection in Robotic Nursing

Yanchen Du[1], Qingzhuo Yuan[2], Shengli Luo[3], XiaoLong Shu[4], Xu Wang[5], Yuheng Jiang[6], Hongliu Yu[7]*
University of Shanghai for Science and Technology, Shanghai, China[1, 2, 3, 4, 5, 6, 7]
Rehabilitation Engineering and Technology Institute, University of Shanghai for Science and Technology, Shanghai, China[1, 2, 3, 4, 5, 6, 7]

Abstract—**In care environments such as nursing homes, robots performing tasks (e.g., feeding assistance) must accurately identify and locate target objects to ensure safe and efficient execution. However, real-world applications face challenges such as numerous small objects, complex backgrounds, and severe occlusions, all of which compromise detection performance. To address these challenges, this study proposes an end-to-end object detection algorithm based on an enhanced DINO framework. Transformer-based DINO is adopted as the baseline, leveraging its global modeling capabilities and avoiding the complex pre- and post-processing required by traditional CNN detectors. In addition, an improved Align-Loss is introduced to enhance small-object detection and address misalignment issues within DINO. Furthermore, a GhostConv module is integrated into DINO's ResNet50 backbone to reduce the computational load of feature extraction and accelerate detection. Finally, multi-scale data augmentation and transfer learning are applied during training to improve detection accuracy and accelerate convergence. To validate the proposed method, experiments were conducted on the augmented MYNursingHome dataset and the COCO dataset. On the MYNursingHome dataset, the proposed approach improved mAP by 3.1% and APs by 2.6% over the DINO baseline, while reducing parameters from 47M to 39.6M and FLOPs from 279G to 243G. On the NVIDIA Jetson Orin Nano Super, inference speed increased from 16.8 FPS to 18.9 FPS (+12.5%). The experimental results demonstrate that the improved DINO detector proposed in this study exhibits a significant advantage in small object detection for nursing scenarios, providing algorithmic support for intelligent and efficient robotic care.**

Keywords*—*Small object detection; matching alignment; robotic object detection; nursing feeding; DINO; GhostConv; edge deployment; transformers

I. Introduction

In recent years, the rapid growth of the ageing population and the increasing number of individuals with disabilities have heightened the demand for intelligent care services. Service robots are becoming essential assistants in daily living, and reliable object detection is a fundamental capability for ensuring safe and effective task execution. Care environments typically contain numerous small objects, complex backgrounds, and frequent occlusions, making robust small-object detection particularly challenging. Existing detection algorithms are mainly divided into CNN-based and Transformer-based approaches.

Early object detection relied heavily on CNN architectures, evolving from two-stage to single-stage frameworks. The R-CNN family [1] [2] introduced deep learning into object detection, while single-stage detectors such as YOLOv3 [3], SSD [4] and RetinaNet [5] improved real-time performance. Subsequent advancements, including multi-scale feature fusion techniques like FPN [6], significantly enhanced detection accuracy, especially for small objects. More recent CNN-based detectors, such as YOLOv8 and YOLOv9, further strengthen multi-scale representation and decoupled head design, achieving consistent improvements on small-object benchmarks [7], while RT-DETR introduces an anchor-free real-time DETR-like detector that significantly improves convergence and small-object accuracy [8]. However, the dependence on anchors and NMS limits their performance in cluttered and highly occluded care scenarios.

The introduction of DETR [9] marked a shift toward Transformer-based detection. By leveraging global self-attention and formulating detection as a set prediction problem with Hungarian matching, DETR removes handcrafted components such as anchors and NMS. Nevertheless, it exhibits slow convergence and struggles with small objects. Improvements such as Deformable DETR [10] and DN-DETR [11] address these issues through deformable attention and denoising training. Building upon these ideas, DINO [12] integrates denoising queries and anchor-based query selection, achieving strong performance across benchmarks. More recently, H-DETR [13] improves cross-scale feature aggregation for small objects, while Group-DETR [14] introduces structured query grouping to enhance training stability and reduce computation. EfficientViT-Det [15] proposes a high-efficiency hybrid architecture that offers improved speed–accuracy trade-offs for edge-device deployment. Align-DETR [16] introduces an alignment-aware loss to mitigate classification–localization inconsistencies. Overall, the emergence of Transformer-based detectors provides novel end-to-end solutions for object detection. TABLE I. compares the performance of the two major algorithmic categories on the same datasets. In accordance with the COCO evaluation protocol, the mean Average Precision (mAP) is further reported at three object scales: AP_s, AP_m, and APl, which measure the detection performance on small, medium, and large objects, respectively. DETR initially showed weak performance, mainly due to slow convergence and long training requirements.

*Corresponding author.
Funding Agency: National Natural Science Foundation of China (General Program), Grant Number: 62473263

However, with continuous optimization, its evaluation metrics now rival those of CNN-based detectors, demonstrating strong potential. Compared with CNN-based approaches, these models not only simplify the detection pipeline but also leverage global modelling to gain clear advantages in challenging scenarios such as complex backgrounds, occlusions, and multi-object interactions. In this work, we build upon DINO, a state-of-the-art DETR-based object detector, as our baseline. However, DINO still suffers from classification–localization inconsistency and high computational cost, which motivates our improvements described in this study. Although the proposed method is built upon the Transformer-based DINO framework, its backbone still uses the CNN-based ResNet50 architecture, within which GhostConv is introduced to improve computational efficiency.

TABLE I. PERFORMANCE METRICS OF DIFFERENT MODELS ON THE COCO DATASET

Model	Epochs	mAP	AP_s	AP_m	AP_l
Faster R-CNN	12	37.9	22.4	41.1	49.1
RetinaNet	12	36.2	14.7	38.5	48.0
DETR	12	15.5	4.3	15.1	26.7
Deformable-DETR	12	47.6	31.2	50.6	62.6
DINO	12	49.0	32.0	53.2	63.0

In summary, current research on object detection in nursing scenarios still faces several challenges: limited accuracy in small object detection, high model complexity, and insufficient validation on real hardware platforms. To address these issues, the main contributions of this study are as follows:

1) Methodological contributions: We propose an enhanced detection framework by introducing an improved Align-Loss that incorporates SIoU and a dynamic τ strategy to alleviate the inconsistency between classification confidence and localisation quality. Furthermore, lightweight GhostConv modules are integrated into the ResNet50 backbone of DINO to reduce redundant feature computations and improve computational efficiency without compromising representational capacity.

2) Empirical contributions: Comprehensive experiments on the extended MYNursingHome+ dataset and the COCO2017 benchmark demonstrate the effectiveness of the proposed components. Ablation studies further verify that each module contributes incremental performance gains, particularly in small-object detection under complex nursing environments.

3) Practical contributions: To assess real-world applicability, the proposed model is deployed and optimized on an NVIDIA Jetson Orin Nano Super embedded platform. The results confirm that the improved detector achieves higher inference speed and reduced computational cost while maintaining competitive accuracy, demonstrating its suitability for deployment in resource-constrained nursing robots.

4) Dataset contributions: An enhanced version of the nursing-scene dataset, namely MYNursingHome+, is constructed by supplementing additional small-object annotations across diverse viewing angles and observation distances. This improved dataset provides a more representative benchmark for evaluating small-object detection in robotic nursing scenarios.

The remainder of this study is organized as follows: Section II reviews the related work on small-object detection, DETR-based models, and lightweight backbones. Section III introduces the improved Align-Loss strategy and enhanced backbone architecture. Section IV presents the experimental platform and design. Section V reports the results and discussion. Finally, Section VI provides conclusions.

II. RELATED WORK

Beyond the evolution of CNN-based and Transformer-based detectors, recent studies have proposed specialized solutions targeting small object detection. One major direction is feature enhancement and multi-scale fusion, which strengthens shallow high-resolution features through improved pyramid structures and fusion modules. Representative work includes ASIF-Net [17] that introduces adaptive scale-interaction fusion for small objects, while SimFusion [18] proposes simplified multi-scale aggregation with reduced computational overhead. Another line of research incorporates attention mechanisms for feature refinement, such as HSF-YOLO, which employs spatial–channel attention and shuffle channel mixing to suppress background noise and enhance small-object features [19].

Knowledge distillation has also been explored, where attention-guided and feature-level distillation improve the representation of small objects in student models under ambiguous environments [20]. A further direction combines super-resolution with detection, exemplified by EFSOD [21], which integrates super-resolution reconstruction to enhance extremely small objects in remote sensing images. In addition, high-resolution inputs with efficient detection heads have been adopted; for instance, Ro-YOLO [22] introduces a Feature Bridge to better connect shallow and deep features while controlling parameter overhead.

In care and medical environments, several works have investigated object detection for assistive scenarios. Mashrur et al. [23] constructed a hospital-facility image dataset and benchmarked mainstream detectors, providing a foundation for healthcare-oriented visual perception, though the targets were primarily large-scale objects. Auweiler et al. [24] improved detection in care homes through Gaussian-based preprocessing, demonstrating the effectiveness of targeted data augmentation in challenging environments. Luo [25] proposed a YOLO-based fall-detection framework, showing that lightweight detectors can achieve real-time performance in smart-care applications. More recently, transformer-based models have also been used in healthcare perception tasks, such as Med-DETR [26], which enhances structured detection for medical instruments. Ismail et al. [27] released the MYNursingHome dataset, but its categories mainly include large furniture and lack fine-grained small-object annotations essential for robotic feeding tasks.

In contrast, this study focuses specifically on small utensil-like objects in nursing scenarios, introducing an improved Align-Loss (Dynamic-τ + SIoU), a lightweight GhostConv-enhanced backbone, and multi-scale augmentation with transfer learning, and further validating deployment feasibility on embedded platforms.

While the aforementioned studies provide valuable insights, several issues remain: First, most research in care settings focuses on facility-level or large-object recognition, with limited attention to small targets, low-sample scenarios, and complex care item interactions. Second, most DETR-based algorithms exhibit mismatches between classification confidence and regression accuracy, which hinder optimal model training. Third, since care robots often operate on resource-constrained edge devices, optimizing computational efficiency while maintaining detection accuracy is essential. Consequently, this study extends the MYNursingHome dataset with scenario-specific samples for feeding assistance and proposes an enhanced DINO model to improve small object detection performance. This provides safe and reliable visual support for robots performing care tasks such as feeding assistance. Unlike prior work that primarily focuses on large objects or facility-level scenes, this study introduces Align-Loss and GhostConv specifically for utensil-like small objects in care environments, addressing classification–localization mismatch and computational constraints not previously explored in DINO-based frameworks.

III. Methods

A. DINO

DINO, as a DETR-based algorithm, represents an end-to-end network architecture. Its architecture consists of four main components: a backbone, a multi-layer Transformer encoder, a multi-layer Transformer decoder, and multiple prediction heads. The overall workflow is shown in Fig. 1. The original DINO uses ResNet50 as its backbone and adopts a multi-scale feature extraction strategy. Features, after positional encoding, are fed into the Transformer encoder, allowing the model to process images at multiple resolutions and thereby improving small object detection performance. After feature enhancement by the encoder, DINO applies a hybrid query selection strategy. This keeps the content query learnable while using the encoder's anchor outputs to initialize the decoder's positional queries. Furthermore, inspired by Deformable-DETR, DINO incorporates a deformable attention mechanism to fuse encoder outputs and perform layer-wise query updates. Subsequently, similar to DN-DETR [10], a DN branch is added to the decoder for denoizing training. Unlike standard DN methods, DINO employs hard negative samples for contrastive denoizing training, thereby improving both training speed and overall performance.

B. Improved Align-Loss

The original DETR uses focal loss for classification optimization, which allows the model to focus on important samples. After the backbone extracts features from the input image, they are flattened into a set of tokens $X=\{x_1,x_2,\ldots,x_m\}$. The transformer encoder-decoder then uses a set of learnable queries $Q=\{q_1,q_2,\ldots,q_n\}$ to extract information from X. These queries are updated layer by layer within the transformer decoder and are finally transformed into predictions $P=\{p_1,p_2,\ldots,p_n\}$ through a prediction head. Given a binary label $y\in\{0,1\}$ and a logit $p\in[0,1]$, the focal loss is defined as Eq. (1):

$$L_{focal}=-y\cdot(1-p)^{\gamma}\cdot logp-(1-y)\cdot p^{\gamma}\cdot\log(1-p) \quad (1)$$

The one-to-one label assignment in DETR leads to low training efficiency. To overcome this, DINO applies denoized contrastive training, which greatly shortens training time and improves efficiency. However, DINO still faces a critical issue: misalignment between predicted classification confidence and anchor box regression accuracy, which severely limits optimization effectiveness.

Cai et al. [16] proposed an Aligned Loss strategy, which can be seamlessly integrated into DETR-based detectors, including DINO, to substantially mitigate the negative impact of classification-regression misalignment on optimization. The approach first introduces a mixed matching and ranking strategy, and then incorporates Align Loss.

1) Mixed matching and ranking strategy: Most DETR variants use Hungarian Matching to learn the mapping between ground truth and predictions. However, previous studies [12][28] show that this method is highly sensitive to instability in the matching process. To address this issue, Align-DETR adopts a gradual transition from positive to negative samples, with the mixed matching and ranking strategy as a prerequisite.

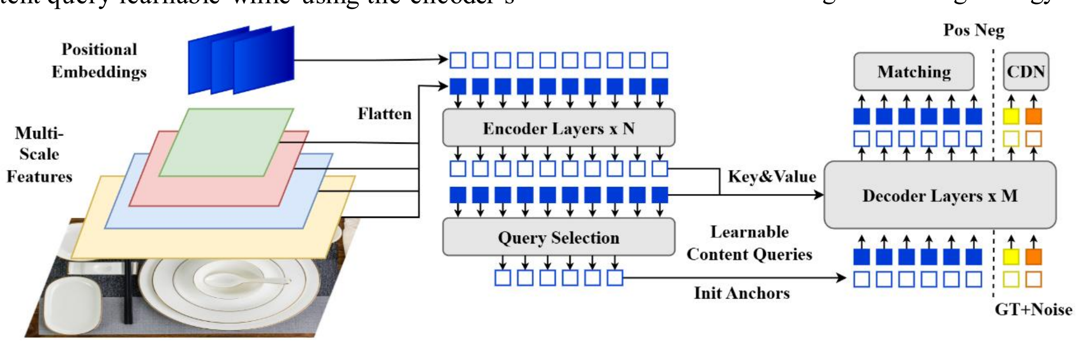

Fig. 1. DINO architecture.

Given predictions P and ground truth G, each with N instances, the enhanced Hungarian Matching algorithm assigns k predictions to each ground truth sample, producing kN candidate pairs. These candidate pairs are ranked according to their distance from the ground truth (GT). As shown in Eq. (2), to jointly use classification and regression accuracy in training, the method defines α quality metric q as the geometric mean of classification accuracy p and regression accuracy μ, defined as:

$$q = p^{\alpha} \cdot \mu^{(1-\alpha)} \tag{2}$$

where, p denotes the binary classification score, μ represents the IoU between the predicted bounding box and the ground truth, and α is a hyperparameter that balances these factors. The ranking of candidate boxes can be represented as $r \in \{0,1,2, \ldots ,k-1\}$. To ensure that the model produces one-to-one predictions, k is set to 1 in the final decoder layer.

2) Align loss: As shown in prior research [28], improving the consistency and efficiency of optimization requires a classification loss that is more adaptive and position-aware. In addition, the transition from positive to negative samples should be smooth. As shown in Eq. (3), the loss function is defined as:

$$\mathrm{L}_{align} = -t_c \cdot (1-p)^{\gamma} \cdot logp - (1-t_c) \cdot p^{\gamma} \cdot \log(1-p) \tag{3}$$

Unlike the original focal loss, which uses a hard label $y$as the optimization target. As shown in Eq. (4), this method introduces a soft target t_c, defined as:

$$t_c = e^{-r/\tau} \cdot q \tag{4}$$

To enable smooth transitions from positive to negative samples during optimization, $t_c$is adjusted from 1 to 0. Here, $e^{-r/\tau}$represents an exponentially decaying weight term, where the decay rate is controlled by the hyperparameter τ. By linking τ to the joint quality q, the loss function enables simultaneous optimization of classification and regression, thus establishing a strong connection between the two tasks. However, treating τ as a fixed hyperparameter has limitations. If τ remains constant during training, the model may rely excessively on noisy predictions in the early phase, favoring boxes with high classification confidence but poor localization accuracy. Conversely, keeping τ unchanged in later stages may overemphasize the balance between classification and regression, limiting the use of high-quality predictions and slowing convergence.

To overcome these limitations, this study proposes a Dynamic-τ strategy. The core idea is to adaptively adjust τduring training: in the early stage, a smaller $\tau$is used to make the consistency score more sensitive to IoU, allowing the model to focus on candidates with higher localization accuracy and reducing noise interference.

As training progresses, τ is gradually increased to balance the influence of classification confidence and localization accuracy, thereby improving the overall stability of matching. As shown in Eq. (5), the method can be formally expressed as:

$$\tau_t = \tau_{min} + (\tau_{max} - \tau_{min}) \cdot \frac{t}{T} \tag{5}$$

where, τ_t denotes the temperature value at iteration t, τ_{min} and τ_{max} represent the initial and final boundary values, and $T$is the total number of training iterations. This linear increment strategy ensures that τ gradually shifts from prioritizing localization accuracy to achieving a balance between classification and localization, thereby improving detection performance and convergence efficiency.

To apply Align Loss to candidate samples while retaining Focal Loss for background samples, as shown in Eq. (6), the classification loss is defined as:

$$\mathrm{L}_{cls} = \sum_{i}^{N_{pos}} \mathrm{L}_{align}(p_i, y_i) + \sum_{j}^{N_{neg}} \mathrm{L}_{focal}(p_j, 0) \tag{6}$$

where, $N_{\text{pos}}$and N_{neg} denote the total number of positive and negative samples, respectively.

In the original Align-Loss design, the localization error between predicted and ground-truth bounding boxes is typically measured by IoU or its variants (e.g., GIoU, DIoU, CIoU). However, these metrics often fail to capture the full geometric relationship between predicted and ground-truth boxes, especially for small objects or those with highly varying aspect ratios. For example, although GIoU alleviates the zero-gradient issue in no-overlap scenarios by incorporating the area of the minimum bounding rectangle, it still shows limitations in convergence speed and directional optimization in detection tasks.

To address this limitation, this study introduces SIoU as an alternative metric for the consistency score in Align-Loss. SIoU considers not only the overlapping area between predicted and ground-truth boxes but also the center-point distance, aspect ratio difference, and angular information, thereby providing a more geometrically meaningful optimization signal. Given a predicted bounding box $b_i$and its ground-truth counterpart $\hat{b}_i$, as shown in Eq. (7), SIoU is defined as:

$$\mathrm{SIoU}(b_i, \hat{b}_i) = \mathrm{IoU}(b_i, \hat{b}_i) - R_{dist} - R_{shape} - R_{angle} \tag{7}$$

where, R_{dist} denotes the center-point distance penalty, R_{shape} measures aspect ratio discrepancy, and R_{angle} reflects angular misalignment between predicted and ground-truth boxes. Compared with standard IoU or GIoU, SIoU provides richer directional gradient information for the regression branch, thereby accelerating convergence in localization accuracy. As shown in Eq. (8), the regression loss is defined as:

$$\mathrm{L}_{reg} = \sum_{i}^{N_{pos}} e^{(-r_i/\tau)} \cdot (\mathrm{L}_{l1}(b_i, \hat{b}_i) + \mathrm{L}_{SIoU}(b_i, \hat{b}_i)) \tag{8}$$

As shown in Eq. (9), the overall model loss can be expressed as:

$$\mathrm{L} = \mathrm{L}_{task}(P, G) + \sum_{l=1}^{L-1} \mathrm{L}_{task}(p_l, G^{(k)}) \tag{9}$$

where, L_{task} is a weighted combination of L_{reg} and L_{cls}, $G^{(k)}$ denotes the augmented ground truth obtained by replication k times, and L represents the number of layers in the Transformer decoder.

In summary, to resolve the inconsistency between classification and regression in DINO, this study refines the Align-Loss function. First, the original static τ parameter is replaced with a dynamic-τ strategy in the joint quality metric q, improving detection performance and convergence efficiency. In addition, SIoU is incorporated into the regression loss L_{reg} to provide a more geometrically meaningful optimization signal. This enhances detection performance for small objects and those with unusual aspect ratios, such as chopsticks, spoons, and other tableware items.

C. Improved ResNet50

ResNet50, used as the backbone network in the original DINO, is structured as shown in Fig. 2. ResNet50 consists of five stages. Stage 0 has a relatively simple structure and mainly performs input pre-processing. It sequentially applies four operations: convolution, batch normalization, ReLU, and max pooling. The following four stages adopt bottleneck structures, each consisting of two basic blocks: the Conv Block and the ID Block, as illustrated in Fig. 3. In the ID Block, the input X has dimensions (C, W, W). It is processed by three convolutional modules (with BN and ReLU) to produce X′. Then, X is added to X′ and passed through a ReLU activation to generate the ID Block's output, maintaining the dimensions (C, W, W). Unlike the ID Block, the Conv Block includes an extra convolutional layer, producing G(X). In the Conv Block, the input channels of X differ from the output channels of the three convolutional modules, preventing direct addition. Therefore, G(X) transforms the channel count of X to enable channel-wise addition. ResNet-50 alleviates the vanishing gradient problem in deep networks through residual connections and provides strong feature representation capabilities. It is considered a classic backbone that balances accuracy and efficiency.

However, in real-world deployment, especially on embedded platforms for care robots (e.g., NVIDIA Jetson series), computational and storage resources are highly constrained. The large number of standard convolution operations in ResNet-50 leads to high parameter counts and computational overhead, limiting efficient operation on resource-constrained platforms. To address this issue, we integrate the GhostConv module into the ResNet-50 backbone to replace part of the standard convolutions. GhostConv generates part of the feature maps through standard convolutions, while the remaining maps are derived from existing features using lightweight linear operations (e.g., per-channel convolutions or linear transformations), thus reducing redundant computation. As shown in Eq. (10), the process is defined as:

$$Y = Conv(X) \,||\, \Phi(Conv(X)) \tag{10}$$

where, Conv(X) denotes standard convolution on input X, $\Phi(\cdot)$ is a lightweight linear transformation, and ‖ denotes channel concatenation. This approach significantly reduces computational complexity and parameter count while maintaining feature expressiveness. In the improved ResNet-50 architecture, standard convolutional layers in each stage are partially replaced with GhostConv, producing a more lightweight backbone. Experimental results in TABLE V. shows that this modification reduces FLOPs and parameters with minimal accuracy loss, thereby enhancing real-time inference performance on embedded nursing robot platforms.

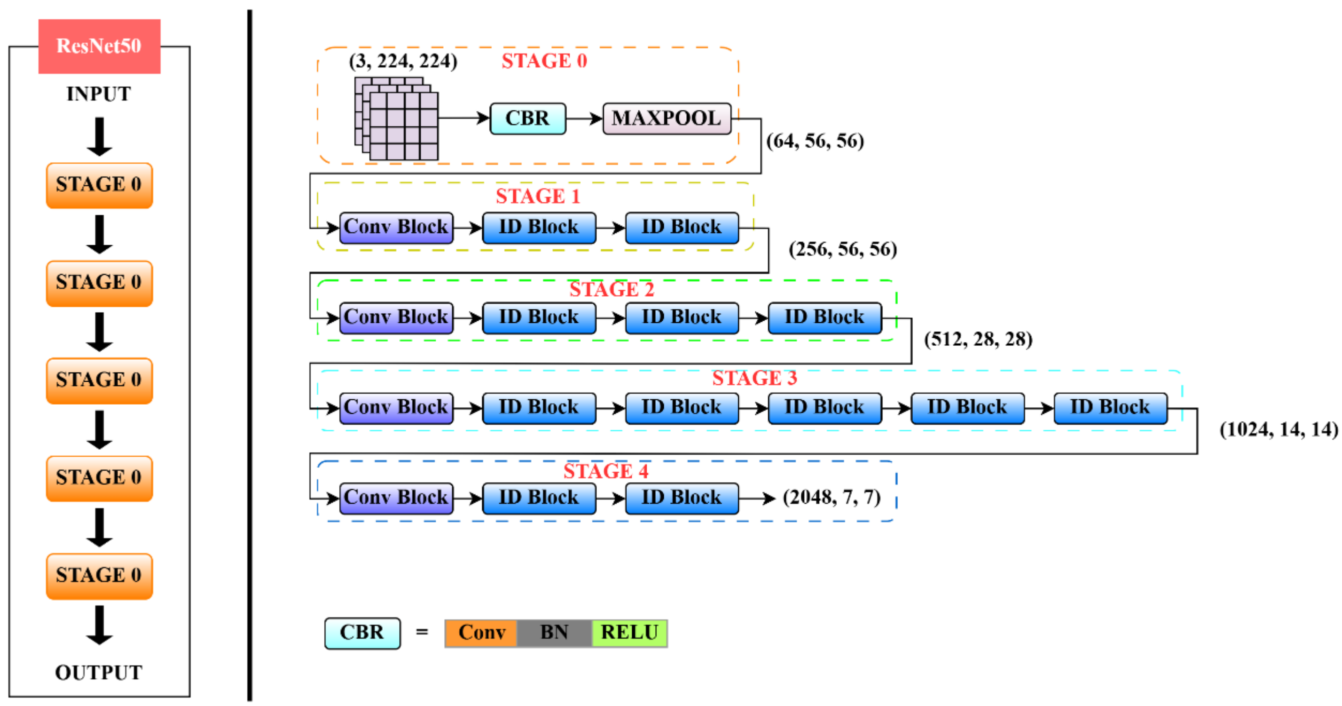

Fig. 2. ResNet50 architecture.

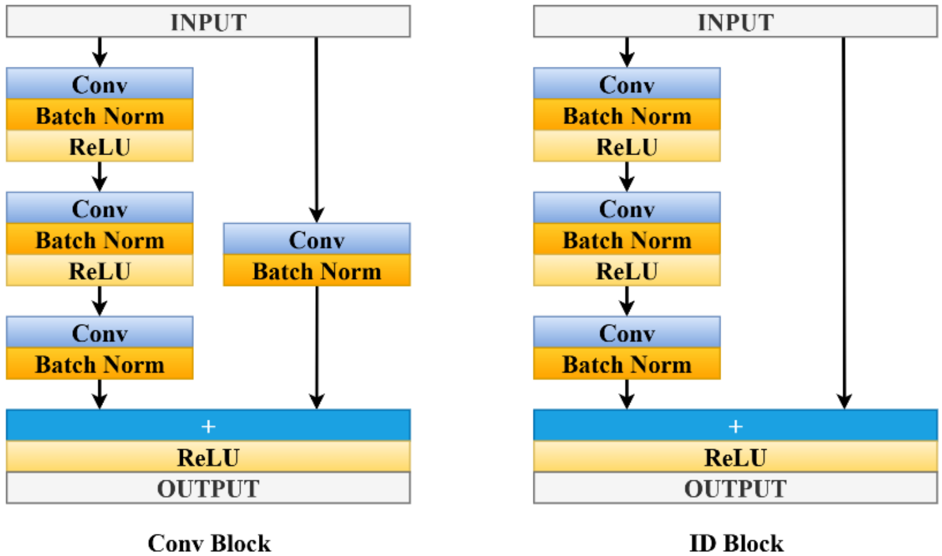

Fig. 3. Conv block and ID block.

Liu et al. [29] proposed an improved YOLOv5 algorithm optimized for robot deployment, demonstrating that lightweight convolutional structures can effectively balance precision and efficiency in embedded systems. Ghostnet [30][29] introduced the GhostConv module, which reduces redundant feature computations in convolutional operations. The GhostConv module consists of three stages: convolution, ghost generation, and feature map concatenation. First, a smaller number of standard convolution kernels generate a subset of base feature maps. Instead of producing a feature maps as in standard convolution, only b feature maps are generated ($b < a$). Second, an inexpensive linear transformation $\phi(\cdot)$ is applied to each of the b feature maps. This generates s new feature maps per original map, resulting in $b \times s$ outputs, with $b \times s = a$, thereby maintaining the same output dimension as a standard convolutional layer. Finally, the feature maps generated by convolution are concatenated with the ghost features produced by the linear transformation to form the complete output. The operational principle of the GhostConv module is illustrated in Fig. 4. Assuming an input feature map of size $h \times w \times c$, an output size of $h' \times w' \times n$, and a convolution kernel of size $k \times k$, as shown in Eq. (11), the computational cost to generate n feature maps using standard convolution is:

$$FLOPs_{conv} = n \times h' \times w' \times c \times k \times k \tag{11}$$

Since n and c are typically large (e.g., 256 or 512), this leads to substantial computational overhead. In GhostConv, partial features are initially generated using n/s standard convolutional kernels, followed by feature expansion through (s-1) inexpensive linear transformations. Consequently, as shown in Eq. (12), the computational cost can be approximated by:

$$FLOPs_{ghost} = \frac{n}{s} \times h' \times w' \times c \times k^2 + \frac{n}{s} \times (s-1) \times h' \times w' \times d^2 \tag{12}$$

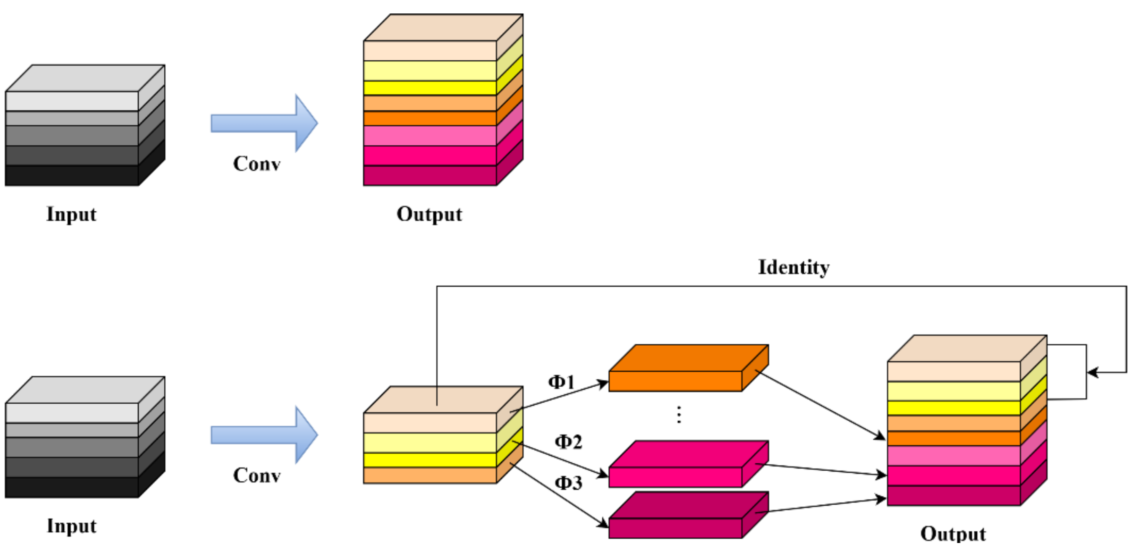

Fig. 4. GhostConv module.

Here, d^2 denotes the kernel size of the linear transformation, which is typically much smaller than k^2, and $s \ll c$. Therefore, GhostConv maintains the same output feature dimensions as standard convolutions while significantly reducing computational complexity. As shown in Eq. (13), its theoretical speedup ratio can be estimated as:

$$\text{Speedup} \approx \frac{\text{FLOPs}_{\text{conv}}}{\text{FLOPs}_{\text{ghost}}} \tag{13}$$

This demonstrates that GhostConv significantly reduces both the number of parameters and FLOPs while preserving feature expressiveness, making it particularly suitable for deployment on resource-constrained embedded platforms.

IV. Experiments

This study uses two datasets to evaluate the performance of our proposed object detection algorithm: the COCO2017 dataset and the MYNursingHome+ dataset, which focuses on nursing care environments. This section describes the experimental setup, datasets, and evaluation metrics.

A. Dataset

1) MYNursingHome dataset⁺: The MYNursingHome dataset is an indoor environment dataset collected from multiple nursing homes in Malaysia. It is specifically designed for visual perception research, including classification and object detection tasks. The publicly available version contains approximately 37,500 images across 25 indoor object categories, including wheelchairs, handrails, cabinets, dining tables, chairs, beds, and walking aids. These categories primarily represent common objects in elderly care settings, providing realistic scenario support for robotic vision tasks in nursing homes. Object categories are shown in Fig. 5. In this study, we augmented and annotated the MYNursingHome dataset, focusing on categories related to tableware and feeding utensils, such as bowls, spoons, chopsticks, and forks. Fig. 5 and TABLE II. show sample images and category distributions for this supplementary dataset, respectively. This subset exemplifies the challenging characteristics of typical care scenarios, including sparse, diverse, and heavily occluded small objects.

2) COCO2017 dataset: To evaluate the generalization capability of the proposed method, we also used the widely adopted COCO2017 object detection dataset, including both training and validation sets. This dataset contains approximately 118,000 training images and 5,000 validation images, covering 80 common object categories, including people, vehicles, tables, cups, and mobile phones. Characterized by its large scale, diversity, and complex backgrounds, this dataset serves as a standard benchmark in the field of object detection. In this study, pre-training is first performed on COCO to establish foundational detection capabilities, followed by fine-tuning and evaluation on MYNursingHome⁺.

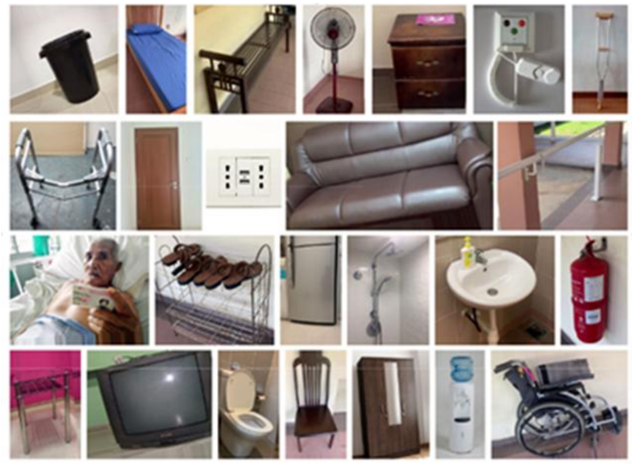

Fig. 5. MYNursingHome dataset.

Fig. 6. Supplementary image data example.

TABLE II. Distribution of Supplementary Data

Labels name	Number of labels
Spoon	1500
Chopstick	1500
Bowl	1500
Knife	1500
Fork	1500
Plate	1500

B. Experimental Settings

Experiments were conducted on two computational devices: a development host (PC) for model training and an edge device (NVIDIA Jetson Nano Super) for deployment and testing. Detailed specifications of both devices are provided in Table III. The experimental workflow, shown in Fig. 7, consists of two distinct phases: the training phase and the testing phase.

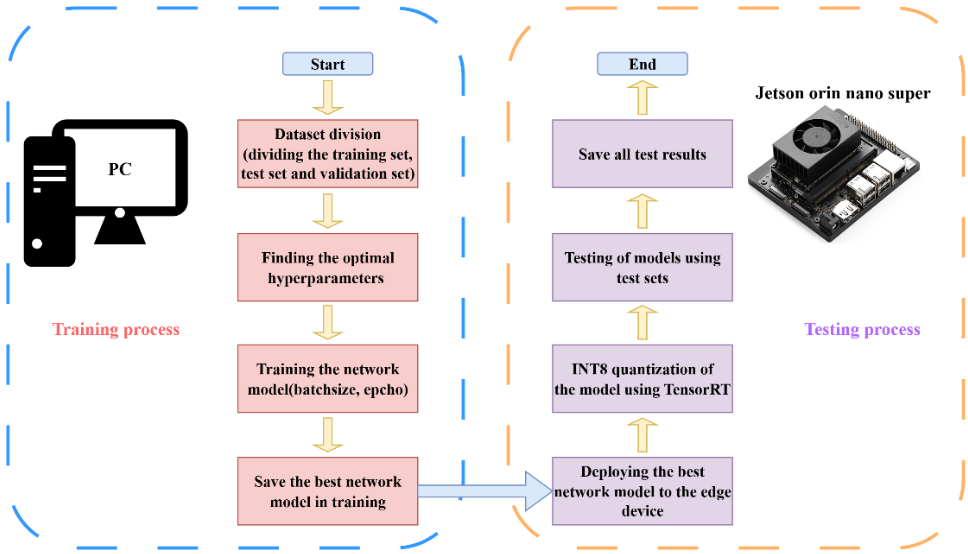

Fig. 7. Experimental workflow.

TABLE III. DEVICE SPECIFICATIONS

Name	Processor	System	Memory	TFLOPS
Jetson orin nano super	NVIDIA Ampere GPU	Ubuntu 22.04	8G	17
PC	GEFORCE RTX 4090	Ubuntu 22.04	24G	190

To improve the detection of small objects (e.g., chopsticks, spoons, bowls) in nursing scenarios, we employ a multi-scale image augmentation strategy during training. In each training iteration, input images are randomly scaled within preset resolution ranges [480, 800], while preserving their aspect ratios. Training across multiple resolutions enables the model to learn scale-invariant feature representations, thereby enhancing its robustness to small objects. Compared with conventional single-scale training, multi-scale training not only enriches the distribution of training samples but also mitigates the issue of small objects appearing disproportionately small at high resolutions, which hinders their detection. Additionally, when images are downsampled, large objects may be compressed into near-small-object representations, further enhancing the model's generalization capability for small object detection. In implementation, multi-scale augmentation is combined with other data augmentation techniques to create a composite augmentation pipeline. Serving as the core strategy, multi-scale augmentation has been shown to significantly improve the model's performance on APs. Experimental results indicate that incorporating multi-scale training increases the model's AP_s on the nursing dataset, effectively validating its advantages for small object detection.

Training Phase: 1) Data Partitioning: The cleaned and annotated nursing dataset is divided into training (70%), validation (20%), and test (10%) sets, ensuring balanced category distributions within each subset. 2) Hyperparameter Search: To obtain stable and high-performing configurations, a genetic algorithm was used to optimize key hyperparameters. These hyperparameters included warmup learning rate parameters, the activation and intensity levels of Mosaic and MixUp [31], the multi-scale range, and the CosineAnnealing [32] learning rate decay schedule. The hyperparameter search used mAP (or AP_s) on the validation set as the fitness function. 3) Metrics: mAP was recorded during training and validation, along with the model's parameters and FLOPs.

Deployment and Testing Phase: 1) Model Export: The optimal weights obtained from training were exported to ONNX format for subsequent quantization and deployment. 2) TensorRT INT8 Quantization Pipeline: The TensorRT build and quantization pipeline involves: a) Using the Builder to parse ONNX and construct the network; b) Configuring dynamic dimensions via Profile and retrieving dimension information from dynamicBinding; c) Building the Calibrator and loading a representative calibration dataset sample using BatchStream for INT8 calibration; d) Configuring the Calibrator and Profile within the Builder's Config to generate the TensorRT engine and calibration parameters; e) Feeding calibrated data into the engine to validate outputs and produce the final INT8 inference engine. Fig. 8 illustrates the workflow.

C. Evaluating Indicators

In object detection tasks, models are required to simultaneously perform object classification and localization. Consequently, evaluation metrics assess both classification accuracy and bounding box regression precision. This study primarily employs the following categories of metrics to evaluate the proposed improved algorithm.

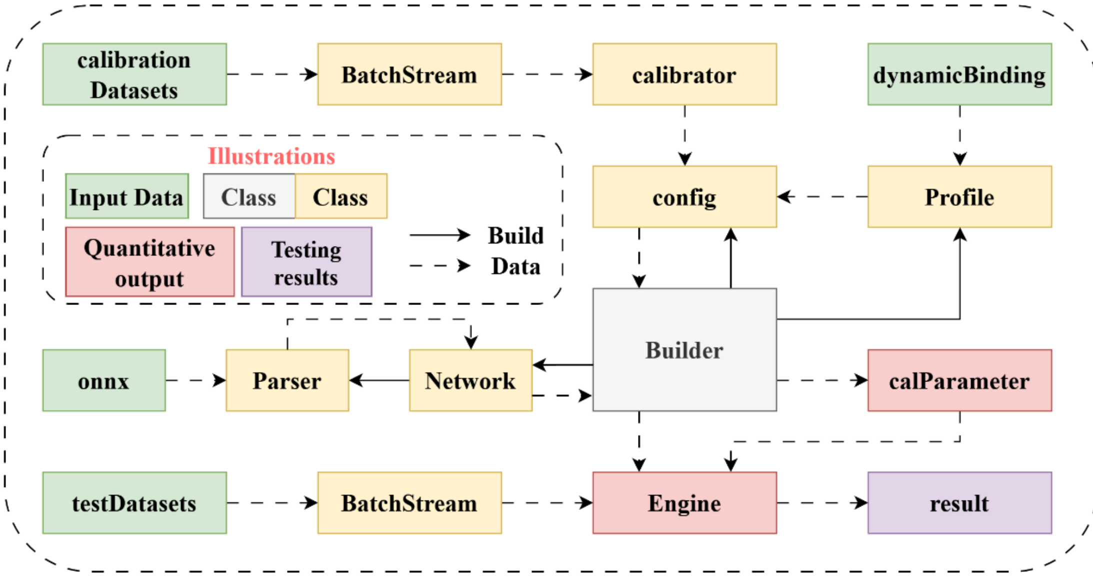

Fig. 8. TensorRT quantization workflow.

Common measures include Precision, Recall, and mean Average Precision (mAP), as shown in Eq. (14) and Eq. (15). It is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Here, TP (True Positive) denotes correctly detected instances where the predicted bounding box has an IoU above the threshold relative to the ground truth; FP (False Positive) denotes predicted boxes with an IoU below the threshold; FN (False Negative) denotes instances where the model fails to detect the ground truth box.

As shown in Eq. (16), Precision is calculated at different recall levels and averaged to obtain the average precision (AP) for a single category:

$$AP = \frac{1}{m}\sum_{i=1}^{m} P_i \tag{16}$$

Here, m denotes the number of samples, and P_i represents the maximum precision at the corresponding recall level. As shown in Eq. (17), averaging the AP values across all categories produces the mean Average Precision (mAP):

$$mAP = \frac{1}{C}\sum_{j=1}^{C} AP_j \tag{17}$$

Here, C denotes the number of classes. The study uses two metrics to comprehensively evaluate detection accuracy: mAP@0.5 (IoU threshold of 0.5) and mAP@0.5:0.95 (IoU thresholds from 0.5 to 0.95 in increments of 0.05).

Regression Metrics: Localization performance is measured by Intersection over Union (IoU), as shown in Eq. (18):

$$IoU = \frac{S_1}{S_2} \tag{18}$$

Here, S_1 denotes the intersection area between the predicted and ground truth bounding boxes, and S_2 denotes the union area. A higher IoU indicates more precise localization.

In nursing robot applications, inference speed and computational resource usage are equally important. The study evaluates model efficiency using the following metrics: Parameters, measuring model size; FLOPs (Floating Point Operations), reflecting computational complexity; FPS (Frames Per Second), indicating real-time inference performance on embedded hardware (NVIDIA Jetson Super Nano).

V. Results and Discussion

A. Ablation Experimental Results

To evaluate the impact of each improvement module on overall performance, ablation experiments were conducted on both the COCO2017 dataset and the nursing dataset. The experimental results are summarized in Table IV. The table shows that replacing the original classification loss with Align-Loss alone increased mAP by 1.5% on COCO and 1.1% on the nursing dataset. A more pronounced improvement was observed in the small object detection metric APS, confirming the effectiveness of classification-regression consistency optimization for small object detection. Further incorporation of SIoU and Dynamic τ led to consistent mAP improvements across both datasets, indicating that the positioning regression and dynamic matching mechanisms effectively enhance detection accuracy. Replacing part of ResNet-50's convolutions with the GhostConv module reduced the number of parameters from 47 million to 45.8 million and decreased FLOPs from 279G to 243G. On COCO, mAP remained largely unchanged (0.0% decrease), while on the nursing dataset, it showed a slight improvement (+0.1%). This demonstrates that GhostConv significantly improves model efficiency while maintaining accuracy. Finally, when combined with multi-scale image augmentation, APS on COCO increased from 32.0% to 35.8% (+2.8%), and on the nursing dataset from 45.2% to 47.8% (+2.6%). This indicates that this strategy significantly enhances the robustness of small object detection.

B. Comparison of Experimental Results

To further evaluate the effectiveness of the proposed method, we compared it with the DINO baseline and other representative DETR-based approaches, including Deformable-DETR, DAB-DETR, and DN-DETR, on the COCO2017 dataset. The results are summarized in Fig. 9 and Table V. Enhanced DINO (ours) demonstrates superior performance, both overall and in small object detection. Specifically, compared with the DINO baseline, our method increases mAP from 49.0% to 51.6%, APS from 32.0% to 35.1%, and small object detection accuracy by 3.1%. Compared with Deformable-DETR (47.0% / 30.1%), our method achieves a 4.6% increase in mAP and a 5.0% increase in APS. Moreover, our approach exhibits advantages in both parameter count and computational efficiency: parameters are reduced from 47.0M to 39.6M, and FLOPs decrease from 279G to 243G, achieving a balanced trade-off between lightweight architecture and performance. Experimental results conclusively demonstrate the effectiveness and practical value of the proposed improvements for small object detection tasks.

C. Result Visualization

Fig. 10 illustrates sample detection results in a nursing scenario. The baseline DINO model tends to miss detections or misclassify objects in complex backgrounds, for example, confusing chopsticks with table edges. In contrast, the proposed improved method achieves more accurate detection of small objects such as chopsticks, bowls, and spoons in the same scenarios, with predicted bounding boxes closely aligned with ground truth. Visualization results indicate that Align-Loss and multi-scale augmentation effectively enhance the model's capability to distinguish small targets.

TABLE IV. ABLATION EXPERIMENT RESULTS

Dataset	Align Loss	SIoU	Dynamic τ	GhostConv	Multi-scale Image augment	mAP	AP_s	Parameters(M)	FLOPs(G)
MYNursing Home+						58.1	45.2	47	279
	√					59.2	46.5	47	279
	√	√				59.8	46.9	47	279
	√	√	√			60.1	47.3	47	279
	√	√	√	√		60.0	47.0	45.8	243
	√	√	√	√	√	60.7	47.8	45.8	243
COCO						49.0	32.0	47	279
	√					50.5	34.7	47	279
	√	√				51.1	35.0	47	279
	√	√	√			51.6	35.2	47	279
	√	√	√	√		51.6	35.1	39.6	243

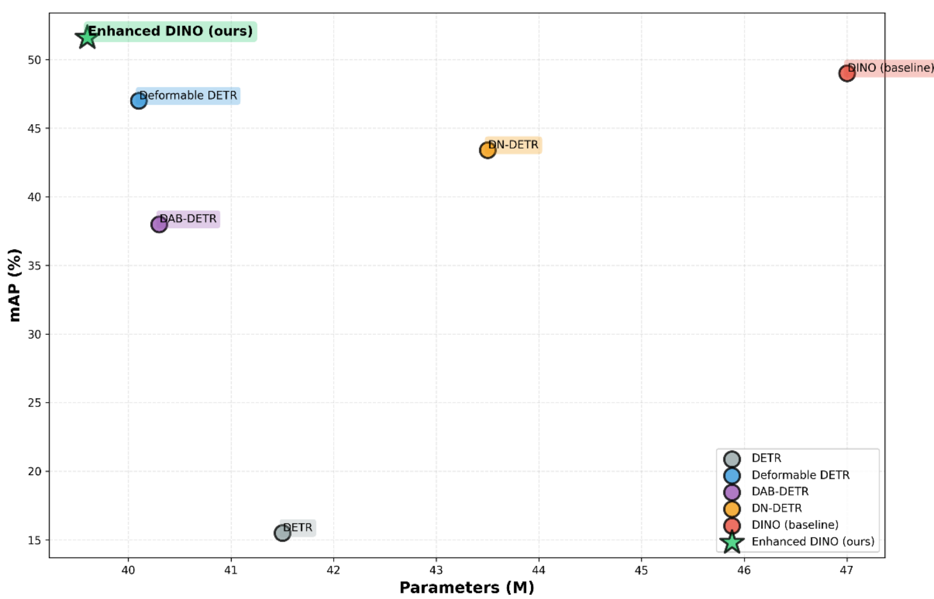

Fig. 9. Comparison of mAP and parameters across different models.

TABLE V. COMPARISON OF TEST RESULTS ACROSS DIFFERENT MODELS

Model	mAP	AP_s	Parameters(M)	FLOPs(G)
DETR	15.5	4.3	41.5	225
Deformable DETR	47.0	30.1	40.1	196
DAB-DETR	38.0	19.2	40.3	256
DN-DETR	43.4	24.8	43.5	265
DINO(baseline)	49.0	32.0	47.0	279
Enhanced DINO(ours)	51.6	35.1	39.6	243

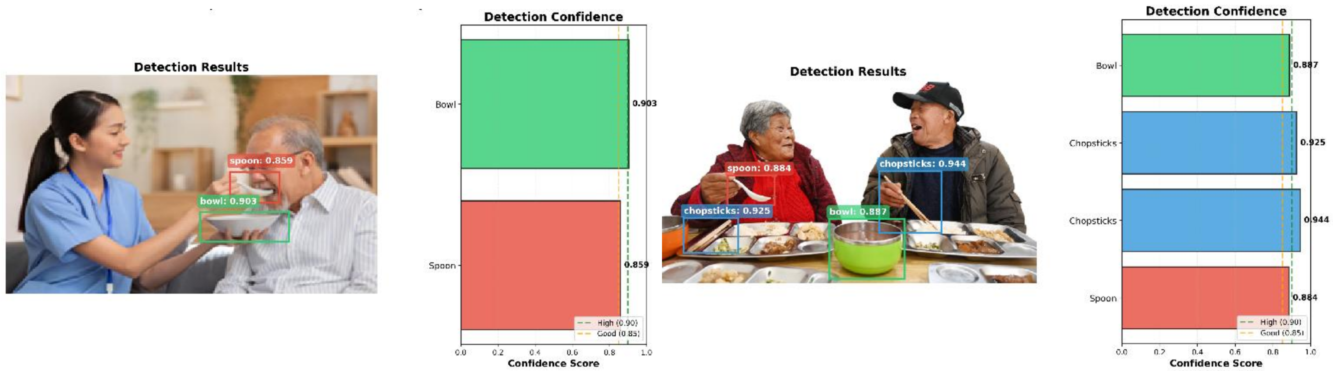

Fig. 10. Example of detection results.

D. Deployment Results on Edge Device

To assess the algorithm's feasibility on an embedded care robotics platform, the optimal model was deployed on the NVIDIA Jetson Orin Nano Super for testing. Results are summarized in Table VI. Without INT8 quantization, Enhanced DINO achieves significant improvements in small object detection compared to the baseline DINO: mAP increases from 48.7% to 51.3%, and APs rise from 32.7% to 35.8%. Concurrently, the number of parameters decreased from 47.0M to 39.6M, FLOPs reduced from 279G to 243G, and inference speed increased from 16.8 FPS to 18.9 FPS, representing an approximately 12.5% improvement. These results demonstrate that the proposed method effectively reduces computational complexity and improves inference efficiency while maintaining or enhancing accuracy, thereby better meeting the real-time requirements of edge devices.

TABLE VI. COMPARISON OF ORIGINAL DINO AND IMPROVED DINO ON NVIDIA JETSON ORIN NANO SUPER

Model	*mAP*	AP_l	AP_m	*AP*	FLOPs(G)	Parameters(M)	FPS
DINO	48.7	62.8	53.0	32.7	279	47.0	16.8
Enhanced DINO	51.3	63.3	53.8	35.8	243	39.6	18.9

E. Discussion

Table V summarizes the impact of the proposed improvements on algorithmic performance. When GhostConv replaces selected convolutional layers in the ResNet50 backbone, both model parameters and FLOPs decrease significantly, indicating that this approach effectively reduces computational overhead. However, due to GhostConv's relatively weaker feature extraction capability, the model's accuracy declines slightly, particularly in complex care scenarios where detecting slender small objects, such as chopsticks, remains challenging.

Introducing Align-Loss alone improves the model's mAP, showing particularly strong performance in detecting small objects on the care dataset. This highlights Align-Loss's significant role in mitigating classification-localization inconsistencies, enhancing robustness against complex backgrounds and occlusions.

Multi-scale augmentation alone yields the most pronounced improvement in small object detection. Experiments confirm that this strategy enables the network to learn object features more effectively across scales, mitigating issues of small object omission and insufficient feature representation. However, multi-scale augmentation slightly increases training time and computational complexity.

When GhostConv, Align-Loss, and multi-scale augmentation are applied simultaneously, the model maintains low computational demands and parameter size while achieving substantial improvements in small object detection accuracy. This demonstrates balanced performance across both the COCO and nursing datasets. This illustrates the complementary nature of the three enhancements: GhostConv provides lightweight processing, Align-Loss ensures consistency between classification and localization, and multi-scale augmentation enhances robustness in detecting small objects.

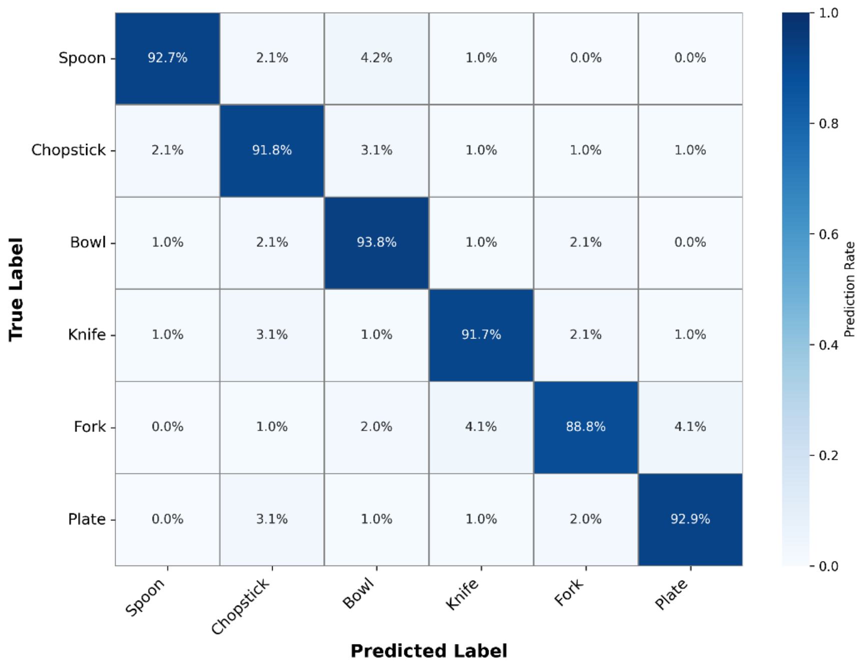

Fig. 11. Confusion matrix results for the improved DINO algorithm, tested with the x-axis representing ground truth labels and the y-axis representing model predictions.

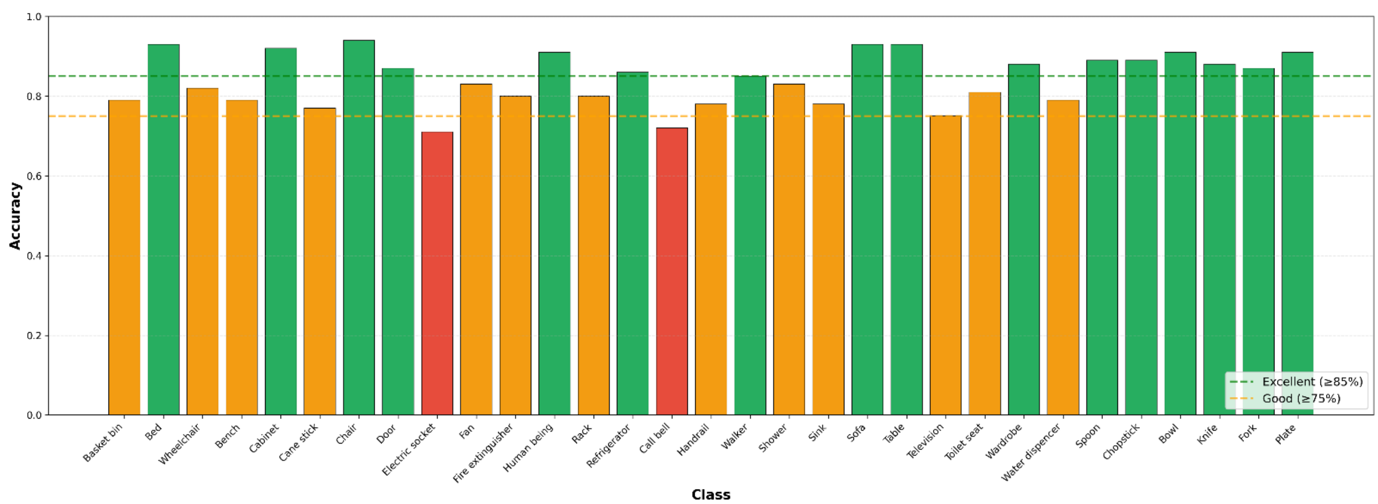

Fig. 12. Object detection accuracy per category.

Table VI compares the proposed method with other DETR-based models, showing superior overall accuracy and small object detection capability compared to the baseline DINO, especially in healthcare scenarios. Visualization results (Fig. 10, Fig. 11, Fig. 12) further substantiate this, showing that the improved model more accurately distinguishes edges of bowls, spoons, and tabletop backgrounds, thereby reducing both false positives and false negatives. Notably, while our approach excels at detecting small objects, accuracy remains insufficient for severely occluded or morphologically complex targets, such as stacked cutlery or partially obscured knives and forks.

Furthermore, the improved model was deployed on NVIDIA Jetson Super Nano edge devices for testing. Experimental results demonstrate that after TensorRT optimization, the model maintains high detection accuracy while significantly reducing latency and energy consumption, enabling near-real-time inference on resource-constrained hardware platforms. Compared to the original DINO, the improved model achieves approximately 12.5% higher frame rates and reduces memory usage by approximately 28 MB on the Jetson Super Nano. These results further validate the feasibility and practicality of our approach for real-world robotic applications. This capability is particularly crucial for care robots, which must operate on low-power, resource-constrained embedded platforms while maintaining detection accuracy to ensure safe task execution.

F. Limitations

Although the proposed enhanced DINO model achieves improved small-object detection performance and demonstrates good deployment feasibility on edge devices, this study still has several limitations:

- First, the experiments are conducted mainly on the COCO dataset and a nursing-oriented dataset, which may not fully represent all real-world robotic nursing scenarios. Further testing on larger and more diverse datasets is needed to improve generalizability.
- Second, the model is evaluated using a ResNet-50 backbone. While GhostConv reduces computational cost, the performance under different backbones (e.g., Swin-T, ConvNeXt-T) remains unexplored.
- Third, the deployment experiments are performed on a single embedded platform (Jetson Orin Nano Super). The latency and energy performance may vary across other hardware configurations.
- Finally, this study focuses primarily on static image detection, and the system has not yet been evaluated in dynamic robotic tasks such as real-time feeding assistance or interaction-heavy nursing actions.

VI. CONCLUSION

To address the challenge of poor small object detection in care settings, this study proposes an end-to-end detection algorithm based on an enhanced DINO framework, which demonstrates excellent performance on edge devices. First, the original loss function in DINO is replaced with an improved Align-Loss function to mitigate inconsistencies between classification and regression. A dynamic τ strategy is also employed to improve training efficiency. Second, lightweight GhostConv modules replace portions of standard convolutions within DINO's ResNet50 backbone, reducing model parameters while preserving accuracy. Subsequently, transfer learning strategies and multi-scale data augmentation are applied to accelerate training and improve detection performance across varying resolutions. Finally, the proposed method is validated on the NVIDIA Jetson Super Nano edge device, with multiple metrics comprehensively compared against various models. Compared to traditional DINO, the improved algorithm achieves 3.1% higher overall accuracy and 2.6% improvement in small object detection. Simultaneously, it reduces model parameters by 2.5% and increases inference speed by 12.5%, while maintaining comparable accuracy. Compared with other algorithms, the proposed method achieves a favorable balance between detection accuracy and inference speed, demonstrating its effectiveness.

This approach improves small object detection accuracy while minimizing computational cost, reducing demands on robotic platform resources. Tailored for feeding robots in care settings, the model was extensively tested on small utensils, including chopsticks, spoons, bowls, and forks, yielding highly satisfactory results. Future research should prioritize improving the model's real-time performance to better meet the online detection requirements of robotic systems. Additionally, expanding the range of detectable objects, such as pharmaceuticals and nursing consumables, is essential to cover broader application scenarios. Another important challenge involves integrating object detection outputs with robotic motion control to realize intelligent nursing systems operable on robotic platforms.

ACKNOWLEDGMENT

We would like to thank the National Natural Science Foundation of China (General Program), having grant number - 62473263 for funding the project titled "Research on Driving Mechanism and Cooperative Control of a Novel Hip–Knee Integrated Intelligent Bionic Prosthetic Joint".

REFERENCES

[1] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. InProceedings of the IEEE conference on computer vision and pattern recognition 2014 (pp. 580-587). Girshick, R. (2015). Fast R-CNN. ICCV.

[2] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence. 2016 Jun 6;39(6):1137-49.

[3] Redmon J, Farhadi A. Yolov3: An incremental improvement. arxiv preprint arxiv:1804.02767. 2018 Apr 8.

[4] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox detector. InEuropean conference on computer vision 2016 Sep 17 (pp. 21-37). Cham: Springer International Publishing.

[5] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. InProceedings of the IEEE international conference on computer vision 2017 (pp. 2980-2988).

[6] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. CVPR.

[7] Wang CY, Yeh IH, Mark Liao HY. Yolov9: Learning what you want to learn using programmable gradient information. InEuropean conference on computer vision 2024 Sep 29 (pp. 1-21). Cham: Springer Nature Switzerland.

[8] Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, Liu Y, Chen J. Detrs beat yolos on real-time object detection. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2024 (pp. 16965-16974).

[9] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. InEuropean conference on computer vision 2020 Aug 23 (pp. 213-229). Cham: Springer International Publishing.

[10] 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 Oct 8.

[11] Li F, Zhang H, Liu S, Guo J, Ni LM, Zhang L. Dn-detr: Accelerate detr training by introducing query denoising. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2022 (pp. 13619-13627).

[12] Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum HY. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arxiv preprint arxiv:2203.03605. 2022 Mar 7.

[13] Jia D, Yuan Y, He H, Wu X, Yu H, Lin W, Sun L, Zhang C, Hu H. Detrs with hybrid matching. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2023 (pp. 19702-19712).

[14] Chen Q, Chen X, Wang J, Zhang S, Yao K, Feng H, Han J, Ding E, Zeng G, Wang J. Group detr: Fast detr training with group-wise one-to-many assignment. InProceedings of the IEEE/CVF international conference on computer vision 2023 (pp. 6633-6642).

[15] Cai H, Li J, Hu M, Gan C, Han S. Efficientvit: Multi-scale linear attention for high-resolution dense prediction. arxiv preprint arxiv:2205.14756. 2022 May 29.

[16] Cai Z, Liu S, Wang G, Ge Z, Zhang X, Huang D. Align-DETR: Enhancing end-to-end object detection with aligned loss. arxiv preprint arxiv:2304.07527. 2024 Aug.

[17] Chen H, Li X, Zhang Z, Zhao R. Research Advanced in Blind Navigation based on YOLO SLAM. InProceedings of the 2nd International

Conference on Computational Innovation and Applied Physics 2023 Mar 25 (pp. 163-171).

[18] Zhang X, Liu A, Yang G, Liu Y, Chen X. SIMFusion: A semantic information-guided modality-specific fusion network for MR Images. Information Fusion. 2024 Dec 1;112:102560.

[19] Zhao Y, Sun H, Wang S. Small Object Detection in Medium–Low-Resolution Remote Sensing Images Based on Degradation Reconstruction. Remote Sensing. 2024;16(14):2645.

[20] Jobaer S, Tang XS, Zhang Y, Li G, Ahmed F. A novel knowledge distillation framework for enhancing small object detection in blurry environments with unmanned aerial vehicle-assisted images. Complex & Intelligent Systems. 2025 Jan;11(1):63.

[21] Zhang H, Hao C, Song W, Jiang B, Li B. Adaptive slicing-aided hyper inference for small object detection in high-resolution remote sensing images. Remote Sensing. 2023 Feb 24;15(5):1249.

[22] Mu X, Chang C, Guo Z, Kan Q, Cheng W, Tian X, Zhou L. Ro-Yolo for Optimized Small Object Detection in Remote Sensing. Available at SSRN 5266679.

[23] Mashrur T, Ghulam Z, French G, Abdullah HA. Assistive feeding robot for upper limb impairment—Testing and validation. International Journal of Advanced Robotic Systems. 2023 Jul 14;20(4):17298806231183571.

[24] Auweiler S, Mueller M, Puhla D, Penava P, Buettner R. A Novel High Performance Object Identification Approach in Care Homes Using Gaussian Preprocessing. IEEE Access. 2025 May 1.

[25] Bo LU. Human fall detection for smart home caring using yolo networks. International Journal of Advanced Computer Science and Applications. 2023;14(4).

[26] Bhat S, Georgescu B, Panambur AB, Zinnen M, Nguyen TT, Mansoor A, Elbarbary KK, Bayer S, Ghesu FC, Grbic S, Maier A. Exemplar Med-DETR: Toward Generalized and Robust Lesion Detection in Mammogram Images and Beyond. InInternational Conference on Medical Image Computing and Computer-Assisted Intervention 2025 Sep 19 (pp. 205-215). Cham: Springer Nature Switzerland.

[27] Ismail A, Ahmad SA, Soh AC, Hassan MK, Harith HH. MYNursingHome: A fully-labelled image dataset for indoor object classification. Data in Brief. 2020 Oct 1;32:106268.

[28] Liu S, Ren T, Chen J, Zeng Z, Zhang H, Li F, Li H, Huang J, Su H, Zhu J, Zhang L. Detection transformer with stable matching. InProceedings of the IEEE/CVF International Conference on Computer Vision 2023 (pp. 6491-6500).

[29] Liu G, Hu Y, Chen Z, Guo J, Ni P. Lightweight object detection algorithm for robots with improved YOLOv5. Engineering Applications of Artificial Intelligence. 2023 Aug 1;123:106217.

[30] Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: More features from cheap operations. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 1580-1589).

[31] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. arxiv preprint arxiv:1710.09412. 2017 Oct 25.

[32] Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts. arxiv preprint arxiv:1608.03983. 2016 Aug 13.