Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.066852

ARTICLE

An Improved YOLO-Based Waste Detection Model and Its Integration to Robotic Gripping Systems

Anjie Wang[1,2], Haining Jiao[1,2,*], Zhichao Chen[1,2,*] and Jie Yang[1,2]

[1]School of Intelligent Manufacturing and Materials Engineering, Gannan University of Science and Technology, Ganzhou, 341000, China

[2]School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, 341000, China

*Corresponding Authors: Haining Jiao. Email: 9320240034@gnust.edu.cn; Zhichao Chen. Email: 7120220036@mail.jxust.edu.cn

Received: 18 April 2025; Accepted: 26 June 2025; Published: 30 July 2025

ABSTRACT: With the rapid development of the Internet of Things (IoT), artificial intelligence, and big data, waste-sorting systems must balance high accuracy, low latency, and resource efficiency. This paper presents an edge-friendly intelligent waste-sorting system that integrates a lightweight visual neural network, a pentagonal-trajectory robotic arm, and IoT connectivity to meet the requirements of real-time response and high accuracy. A lightweight object detection model, YOLO-WasNet (You Only Look Once for Waste Sorting Network), is proposed to optimize performance on edge devices. YOLO-WasNet adopts a lightweight backbone, applies Spatial Pyramid Pooling-Fast (SPPF) and Convolutional Block Attention Module (CBAM), and replaces traditional C3 modules (Cross Stage Partial Bottleneck with 3 convolutions) with efficient C2f blocks (Cross Stage Partial Bottleneck with 2 Convolutions fast) in the neck. Additionally, a Depthwise Parallel Triple-attention Convolution (DPT-Conv) operator is introduced to enhance feature extraction. Experiments on a custom dataset of nine waste categories conforming to Shanghai's sorting standard (7,917 images) show that YOLO-WasNet achieves a mean average precision (mAP_{50}) of 96.8% and a precision of 96.9%, while reducing computational cost by 30% compared to YOLOv5s. On a Raspberry Pi 4B, inference time is reduced from 480 to 350 ms, ensuring real-time performance. This system offers a practical and viable solution for low-cost, efficient automated waste management in smart cities.

KEYWORDS: Waste classification; YOLO; raspberry Pi; resource recycling

1 Introduction

In recent years, the rapid improvement in residents' living standards has led to a substantial increase in household waste. Without proper sorting, recyclable materials, hazardous waste, and kitchen waste are often mixed together, complicating subsequent treatment and potentially releasing harmful substances during landfilling or incineration. Although many regions have introduced waste-sorting regulations and policies [1], the lack of supporting infrastructure and systematic classification mechanisms continues to hinder their effectiveness [2]. Additionally, significant variability in public awareness and participation further increases the difficulty of consistent and accurate classification [3]. These challenges highlight the urgent need for intelligent, efficient, and scalable waste-sorting systems to support sustainable waste management in real-world scenarios.

Recent progress in artificial intelligence and robotics provides promising solutions to these issues [4]. Numerous studies have explored the development of automated waste-sorting systems [5], yielding notable

advances in classification techniques. However, existing systems still suffer from high computational costs, suboptimal performance, and difficulties in deployment on embedded platforms [6]. Additionally, manual intervention remains a dominant part of many waste-sorting processes, resulting in high labor costs [7]. These limitations call for an integrated solution that enables full automation while maintaining high accuracy and low resource consumption.

In this paper, we propose a lightweight and efficient waste-detection and sorting system based on YOLOv5, designed for deployment on embedded platforms. The system integrates a novel waste-aware detection model with a robotic arm guided by a pentagonal grasping algorithm. Our key contributions are summarized as follows:

- A novel model named you only look once with waste-aware sorting network (YOLO-WasNet) is proposed, which substantially reduces both parameter count and computational complexity, thereby lowering hardware investment and meeting the demands of efficient waste-classification systems.
- In mechanical design, a pentagonal grasping algorithm (a control strategy that computes robotic-arm grasp points based on pentagon geometry) is combined with a parallelogram-based palletizing arm structure (a linkage mechanism using parallelogram kinematics) and flexible gripper jaws to accommodate various waste shapes, enabling rapid and precise handling of diverse items.
- The resulting sorting system demonstrates broad applicability-extensible to waste sorting, warehouse order fulfillment, and parcel handling. Experiments show that using MobileNetV3-Small (a compact convolutional neural network architecture for mobile and embedded devices) [8] as the backbone drastically cuts model size; the SCBA module (a fusion of Spatial Pyramid Pooling Fast and Convolutional Block Attention modules) [9] boosts detection accuracy; DPT-Conv (Depthwise Parallel Triple-attention Convolution operator) [10] matches standard convolution performance with fewer parameters; and the C2f (a lightweight feature-fusion block enhancing gradient flow) [11] module provides richer gradient information.

2 Related Works

2.1 Intelligent Waste Recognition

In recent years, Le and Ngo [12] proposed a vision-based deep learning system for waste classification that employs convolutional neural networks for end-to-end feature extraction and classification of waste images, integrating the automatic sorting pipeline into the robotic grasping module. Wahyutama and Hwang [13] developed a YOLO-based system for recyclable collection and bin capacity monitoring, combining real-time object detection with a multi-compartment bin lid mechanism and ultrasonic capacity sensing to synchronize classification results with remaining volume data. Building on these advances, Hu et al. [14] designed a collaborative-robot ACP system that seamlessly links vision-based detection with multi-joint robotic-arm trajectory planning, boosting sorting throughput by over 30% and reducing misclassification rates below 5%, thereby demonstrating high stability and safety in automated waste handling. Lightweight models, with their reduced parameter counts and lower computational overhead, enable real-time, low-power inference on resource-constrained platforms such as Jetson Nano and Raspberry Pi, significantly lowering hardware costs and deployment complexity. Moreover, robotic arms-owing to their high-precision positioning and flexible grasping capabilities-can adapt to a variety of waste shapes, minimizing human intervention and reducing operational errors.

2.2 Object Detection Methods

In recent years, object detection methods have broadly fallen into two categories: Transformer-based detectors and the single-stage YOLO series. UP-DETR [15] employs an unsupervised pre-training task of random patch detection on the Transformer decoder, which significantly accelerates convergence and improves average precision on object detection and panoptic segmentation. However, because it still relies on a one-to-one Hungarian matching strategy to align queries with targets, the query-to-target matching remains inefficient and requires a very long convergence period. Building on this, Group DETR [16] introduces a group-wise one-to-many assignment strategy that indeed shortens training time and speeds up convergence to some degree, but this improvement does not fundamentally reduce the computational and memory overhead of the Transformer architecture, limiting its support for resource-constrained scenarios.

By contrast, the YOLO series, with its single-stage end-to-end regression and lightweight modular design, offers superior real-time performance and deployment flexibility. For example, YOLO-MSA [17] further enhances robustness in complex environments by integrating multi-scale stereoscopic attention with spatial-channel collaboration. A Pi-based YOLO system demonstrates feasibility on a Raspberry Pi, interfacing with a multi-compartment bin lid and capacity sensor for real-time sorting and status synchronization in low-cost hardware settings. YOLO-GD [18] combines the GhostNet module with depthwise-separable convolutions, reaching approximately 97.4% mAP and 32.75 ms inference latency on a Jetson Nano after quantization, while providing precise grasp-point guidance. Finally, YOLOv5, with its PyTorch-based modular architecture, Mosaic augmentation, and CIoU loss, achieves over 30 FPS and excellent mAP on public benchmarks, greatly simplifying both training and edge-deployment workflows.

3 Materials and Methods

3.1 Garbage Sorting System

3.1.1 Introduction of the Garbage Sorting System

In order to design the complex mechanical structure of the robotic arm, SolidWorks, an industry-leading computer-aided design software, was employed to construct the hardware model. Fig. 1 illustrates the overall appearance of the gripping device designed in this study. Specific details regarding the selected vision sensors and actuators and their key characteristics are provided below.

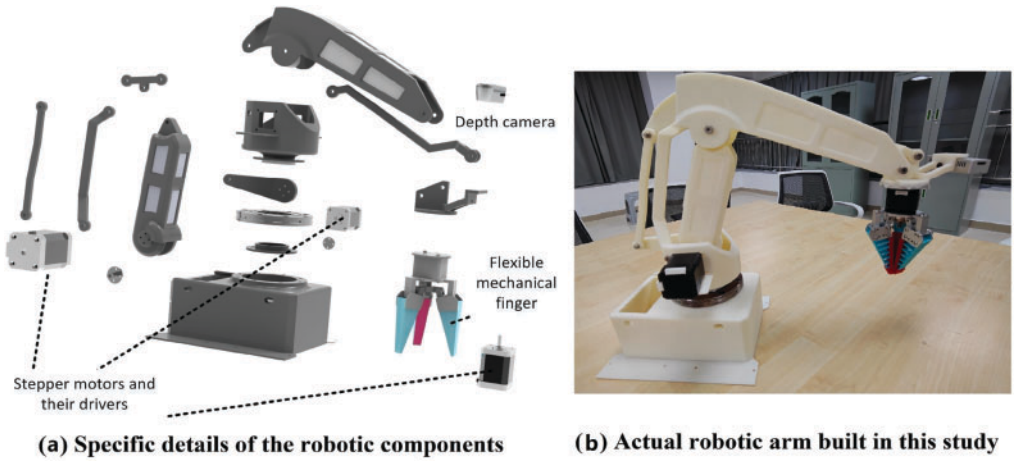

Figure 1: Overall mechanical structure of the designed gripper arm and its components

1) **Stepper motors and drivers:** The robotic arm is equipped with three high-precision stepper motors, comprising one 42BYGH47 primary motor and two 42BYGH60 auxiliary motors.

2) **Camera:** The selected camera is the Intel RealSense D415, which supports a resolution of 1280 × 720. It employs a USB interface for image data transmission and operates over a range of 0.5 to 3 m. The field of view is $65°(\pm 2°) \times 40°(\pm 1°)$.

The actual waste-sorting robotic arm is shown in Fig. 1b, and its operational workflow is as follows. Upon system initialization, the stepper motors perform a homing procedure. The end-mounted depth camera continuously surveys the waste-detection zone and captures real-time images. These images are then processed by the YOLO-WasNet model for object detection and localization. Finally, leveraging the robotic-arm pentagonal algorithm, the arm is precisely actuated to grasp identified waste items and deposit them into their respective collection bins, thereby achieving intelligent waste-sorting management.

3.1.2 Controller Chip

To control the intelligent waste-sorting robotic arm, an efficient control system was designed, with the circuit connections shown in Fig. 2. The system leverages the synergy of two key components:

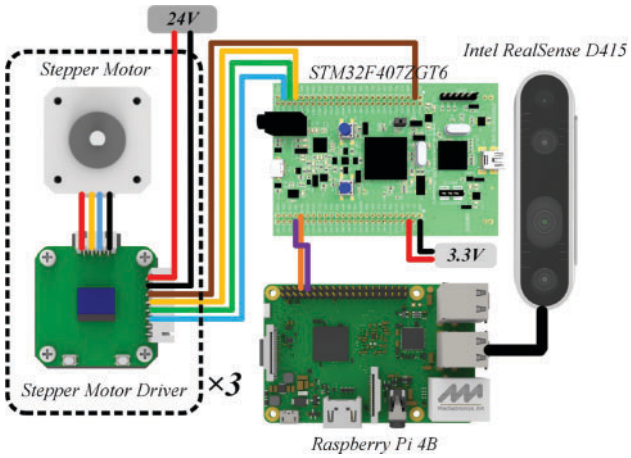

Figure 2: Schematic wiring diagram of the control circuit of the designed robotic arm. Where, ×3 indicates that three stepper motors and drivers were used to control the rotational degrees of freedom of each of the three different joints

1) **Raspberry Pi 4B:** Vision-processing platform with a compact form factor and powerful computing performance; driven a Broadcom BCM2711 SoC with four Cortex-A72 cores for real-time image acquisition and inference.

2) **STM32F407ZGT6:** Central control unit with an ARM Cortex-M4 core clocked at up to 168 MHz; equipped with advanced timers for precise motor speed and position control.

3.2 Robotic Arm Structure and Robotic Arm Pentagon Algorithm

3.2.1 Robotic Arm Gripper

To effectively grasp waste items of various shapes and sizes without employing cost-intensive force-feedback technology [19], a flexible manipulator based on a unique curved-structure design is adopted, as depicted in Fig. 3. The refined internal actuation mechanism not only simplifies the overall assembly but also extends the effective stroke range of the manipulator's flexible fingers.

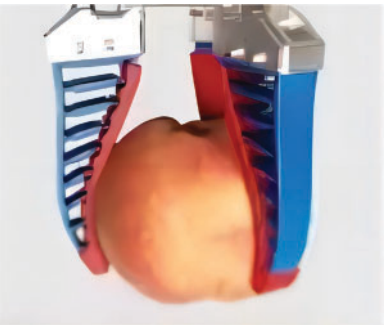

Figure 3: Schematic structure of the adopted flexible gripper. The whole flexible gripper is made of TPU (Thermoplastic Polyurethane) and the inner edge is made of silicone with high friction

3.2.2 Palletizing Robot Arm Structure

To ensure that the depth camera remains parallel to the horizontal plane during operation, the mechanism principle of the palletizing robotic arm is adopted, as shown in Fig. 4. Two four-bar linkage assemblies maintain a constant angle between the arm's end-effector and the horizontal plane, thereby reducing one degree of freedom, minimizing mechanical errors caused by motor backlash, and simplifying the motor control system.

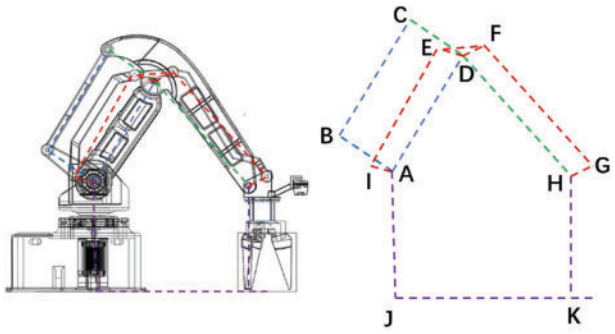

Figure 4: Schematic diagram of the principle of the palletising robot arm. Quadrilateral *AIED*, quadrilateral *HGFD* and quadrilateral *ABCD* are parallelograms, *EDF* is a fixed triangle, and AI is fixed on the rotating base of the robot arm

3.2.3 Robotic Arm Pentagonal Algorithm

The robotic-arm pentagonal algorithm is a technique grounded in mathematical and geometric principles for the precise control of individual joints to achieve designated positional and orientation targets. The algorithm computes the distance from the end effector to the target point and determines the required joint angles to ensure accurate manipulation.

As illustrated in Fig. 5, the initial configuration of the four-axis robotic arm is shown. During system startup, the motors automatically execute a homing routine using the near-zero functionality built into the driver board, returning to their predefined zero positions upon power-on. Consequently, at initialization, the link lengths and their initial inter-joint angles are known. A horizontal reference line, *AB*, is introduced. When the depth camera captures a target object, the distance from the camera to the object centroid is

measured, and hand-eye calibration is used to determine the object's real-world 3D coordinates (x, y, z). From these data, the following parameters can be derived:

$$d = \sqrt{x^2 + y^2} \tag{1}$$

$$\alpha = \arctan\left(\frac{y}{x}\right) \tag{2}$$

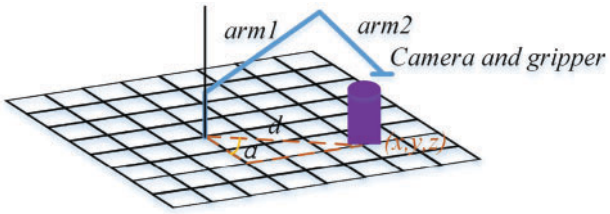

Figure 5: Diagram of the initial state of robotic arm gripping

By rotating the base stepper motor by an angle α, the end effector aligns with the object's profile plane. As depicted in Fig. 6, this configuration represents the ideal posture after grasping the object within the profiling plane. Using the illustrated frame, a pentagonal linkage can be formed comprising arm 1, arm 2, the robotic arm base, the object height, and the horizontal distance from the base to the object centroid, with the two lower vertices being right angles. From this geometry, the following equation can be derived:

$$\theta_1' = \arctan\left(\frac{z-h}{d}\right) \tag{3}$$

$$\theta_2' = \arccos\left(\frac{arm1^2 + d^2 + (z-h)^2 - arm2^2}{2 \times arm1 \times \sqrt{d^2 + (z-h)^2}}\right) \tag{4}$$

$$\theta_3' = 90 - \theta_1' - \theta_2' \tag{5}$$

$$\varphi' = \arccos\left(\frac{arm1^2 + arm2^2 - d^2 - (z-h)^2}{2 \times arm1 \times arm2}\right) \tag{6}$$

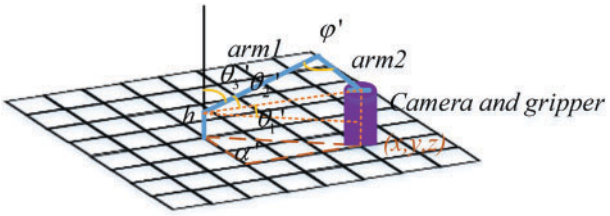

Figure 6: Schematic diagram of the end state of robotic arm gripping

3.3 Visual Algorithm Design

3.3.1 Overall Architecture Schematic

Object detection algorithms enable the robotic arm to precisely localize waste items. YOLO v5 represents a significant advancement in this domain, achieving an optimal balance between speed and accuracy through single-stage detection and end-to-end optimization. Nevertheless, the standard YOLO v5 model relies on the CSPDarknet53 backbone, which incurs substantial parameters and computational overhead due to its Darknet Bottleneck and convolutional layers. Consequently, this study replaces the original backbone with the lightweight MobileNetV3-Small, thereby markedly reducing the model's overall computational effort.

The network architecture employed in this work is depicted in Fig. 7. To counteract accuracy degradation caused by a lightweight backbone, an SCBA module-comprising cascaded SPPF (Spatial Pyramid Pooling-Fast) and CBAM (Convolutional Block Attention Module)-is embedded at the backbone's terminus. Additionally, to mitigate redundant computation in the detection head, a DPT-Conv module with efficient feature operators based on depth-separable convolutions is introduced. Finally, the original C3 module is replaced by the C2f block [20], thereby enhancing gradient flow.

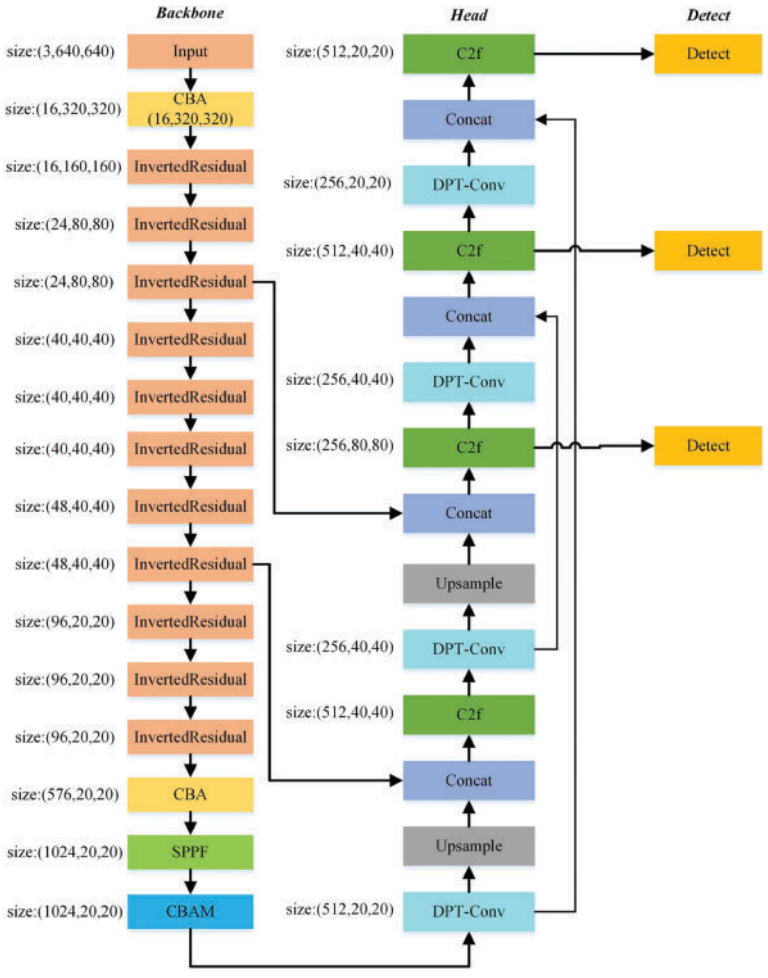

Figure 7: Overall network structure of the proposed YOLO-WasNet

3.3.2 SCBA Feature Enhancement Module

To further optimize the feature maps output by the backbone network, SPPF (Spatial Pyramid Pooling-Fast) and CBAM (Convolutional Block Attention Module) are sequentially integrated at the backbone's terminus to form the SCBA module. The SPPF block retains contextual information by applying three successive 5 × 5 max-pooling kernels-thereby continuously expanding the receptive field-and concatenating the pooled outputs across channels to enhance both global and local feature perception. The concatenated feature map is then fed into the CBAM module: first, channel attention is applied by performing max-pooling and average-pooling on each channel to extract informative signals and suppress redundant features; next, spatial attention assigns weights to different spatial locations, guiding the network to focus on salient regions while ignoring background or irrelevant areas.

SPPF The traditional SPP module employs a parallel architecture, applying three max-pooling layers of differing kernel sizes concurrently and concatenating their outputs. However, this parallel approach incurs substantial computational and memory overhead on resource-constrained platforms such as mobile or embedded devices. To mitigate this issue, the more efficient SPPF module is utilized: it sequentially applies three 5 × 5 max-pooling operations before concatenating the pooled feature maps along the channel dimension. This cascaded design preserves feature representation performance while significantly reducing redundant computation and parameter count, thereby lowering deployment overhead.

The network architecture of the SPPF module used in this study is illustrated in Fig. 8. First, the input feature maps are processed through convolutional layers, batch normalization (BN), and a Leaky ReLU activation. Next, the feature maps undergo multi-scale pooling (e.g. 1 × 1, 2 × 2, and 4 × 4) to capture abstract representations at various receptive fields. The pooled outputs from each scale are concatenated along the channel dimension and then passed through additional convolutional layers, BN, and a SiLU activation. This sequence fuses local and global context at the feature-map level, thereby enhancing the model's ability to represent multi-scale abstract features of waste-category targets.

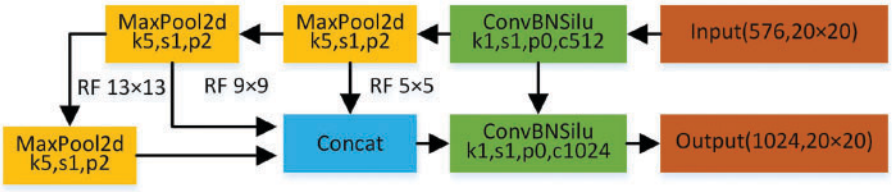

Figure 8: The overall SPPF network structure used in this paper. RF stands for Receptive Field

CBAM CBAM leverages attention mechanisms to enhance the network's capability for salient feature identification. As illustrated in Fig. 9, CBAM employs channel attention and spatial attention modules to recalibrate feature importance without altering the spatial dimensions of the feature maps.

$$F' = \sigma\left(MLP(\text{Avg}(F)) + MLP(\text{Max}(F))\right) \otimes F \tag{7}$$

$$= \sigma\left(W_1 W_0 F^c_{\text{avg}} + W_1 W_0 F^c_{\text{max}}\right) \otimes F \tag{8}$$

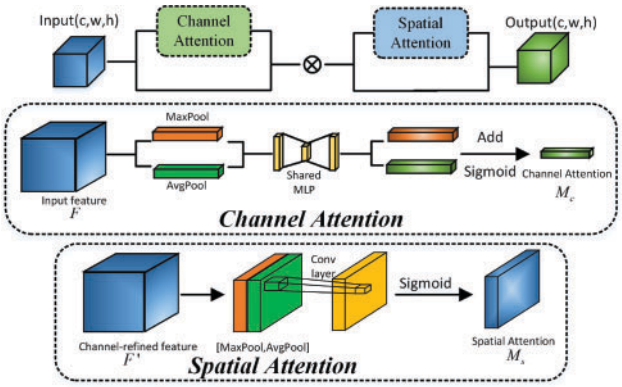

Figure 9: Network structure of CBAM

As described in Eqs. (7) and (8), where F' denotes the output feature map after the channel-attention mechanism, and $\otimes$ represents element-wise multiplication between the importance scores computed by the channel-attention module and the input feature map F. The function σ denotes the activation function-typically sigmoid-ensuring that attention weights lie within the interval $[0,1]$. *MLP* stands for a multi-layer perceptron that transforms channel descriptors obtained from average pooling (F^c_{avg}) and max pooling ($F^c_{\max}$) using weight matrices W_0 and W_1, respectively, to produce the final channel-attention map.

The feature maps refined by the channel-attention module are subsequently processed through a spatial-attention mechanism to strengthen the network's ability to recognize salient spatial features. This operation can be expressed as follows:

$$F'' = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \tag{9}$$

$$= \sigma(f^{7\times7}([F^s_{avg}; F^s_{\max}]) \otimes F' \tag{10}$$

As shown in Eqs. (9) and (10), F'' denotes the final feature map after applying spatial attention, and $\otimes$ represents element-wise multiplication. The function σ-typically a sigmoid activation-ensures that attention weights lie within the interval $[0,1]$. The term $f^{7\times7}$ refers to a convolutional layer with a 7×7 kernel size. Finally, $[AvgPool(F); MaxPool(F)]$ (or equivalently $[F^s_{avg}; F^s_{\max}]$) indicates the concatenation of average-pooled and max-pooled feature maps along the channel dimension.

By combining SPPF and CBAM, the network benefits from rich multi-scale feature representations provided by SPPF and the attention-driven refinement of those features via CBAM. This synergy not only enhances the backbone's capacity for feature extraction but also optimizes feature selection, thereby improving the model's overall performance.

3.3.3 Neck Feature Aggregation

During experiments, it was found that despite the lightweight backbone, the detection head still contains substantial redundant parameters, resulting in a model that remains large in size and exhibits reduced accuracy. Analysis reveals two primary issues:

1) **Inefficiency of traditional convolution operators:** convolutional layers are responsible for feature extraction but larger convolutional kernels significantly increase computational cost and exacerbate the risk of overfitting.

2) **Feature-transfer bottleneck in the C3 module:** the complex "bottleneck" design of the C3 block enhances representational capacity at the expense of increased parameter count and computational overhead.

Aiming at the above problems and analyses, this paper proposes an efficient feature extraction operator, DPT-Conv, which combines deep separable convolution and lightweight attention mechanism to achieve efficient feature extraction with less computational cost. On the other hand, in order to solve the C3 feature transfer problem, the C2f module is employed to ensure sufficient gradient flow and maintain stability during the training process, so as to improve the performance of the model in the target detection task.

In this study, we propose the DPT-Conv architecture, which integrates depthwise-separable convolutions with a novel triple-attention mechanism to enhance feature representation, as illustrated in Fig. 10. Initially, depthwise-separable convolutions are employed to extract low-level features while significantly reducing computational complexity and parameter count. However, to address the limited expressive capacity inherent in such operations, we introduce a multi-path attention strategy that reinforces spatial and channel-wise representations.

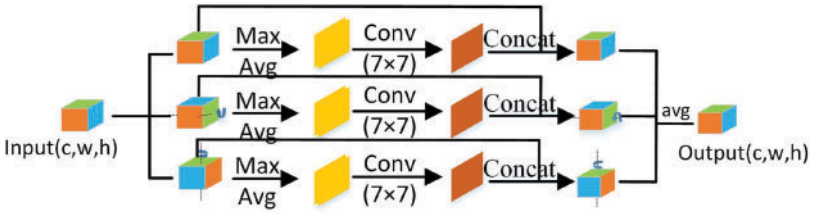

Figure 10: Detailed information on the triple attention structure, Avg and Max stand for average pooling and maximum pooling respectively, and avg stands for the summed average of elements at the same position in each dimension

Given an input feature map U, we first construct three orientation-sensitive variants: the original $U_0 = U$, a version rotated along the horizontal axis $U_1 = \text{Rotate}_x(U)$, and another rotated along the vertical axis $U_2 = \text{Rotate}_y(U)$. Each rotated feature map U_k (where $k \in \{0, 1, 2\}$) undergoes both average pooling and max pooling operations. These pooled maps are then concatenated along the channel dimension to form a joint descriptor:

$$G_k = \begin{bmatrix} \text{AvgPool}(U_k) \\ \text{MaxPool}(U_k) \end{bmatrix} \in \mathbb{R}^{2\times H\times W}. \tag{11}$$

Subsequently, a convolutional layer with a 7×7 kernel is applied to each concatenated feature map G_k, followed by a sigmoid activation to generate the corresponding attention map:

$$A_k = \sigma\left(\text{Conv}_{7\times 7}(G_k)\right) \in \mathbb{R}^{1\times H\times W}. \tag{12}$$

Each attention map A_k is then used to reweight the original input feature map U via element-wise multiplication (Hadamard product), yielding the attention-enhanced outputs V_k:

$$V_k = A_k \otimes U. \tag{13}$$

Finally, the output feature map F_{out} is obtained by averaging the reweighted feature maps from all three attention paths:

$$F_{\text{out}} = \frac{1}{3}\sum_{k=0}^{2} V_k. \tag{14}$$

This attention-augmented depthwise convolution framework allows the model to capture directional and structural patterns more effectively, addressing the expressiveness bottleneck of conventional depthwise convolutions.

C2f After model lightweighting, it becomes crucial to introduce additional gradient pathways to preserve the original performance. In the YOLOv5 architecture, the traditional C3 block adopts the branch-split strategy of CSPNet, integrating residual connections to strike a balance between network size and detection accuracy.

To further enhance gradient propagation and improve feature expressiveness under a lightweight constraint, we introduce the C2f module, illustrated in Fig. 11. This module is an evolution of the C3 architecture, designed with a more efficient channel-splitting and fusion mechanism. It enables flexible channel adjustment and strengthens multi-scale feature interaction, thereby improving object detection accuracy and inference efficiency.

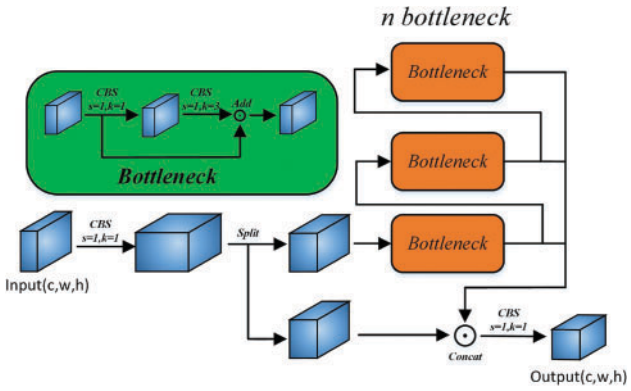

Figure 11: Model diagram of the network structure of C2f, where CBS denotes Convolution, BatchNormalization, SiLu

Formally, let the input feature map be $F \in \mathbb{R}^{C\times H\times W}$. The input is first split along the channel dimension into two parts: a primary stream F_0, and a residual stream F_{res}, defined as

$$F_0 = F[:, 1:\lfloor \alpha C \rfloor, ::], \quad F_{\text{res}} = F[:, \lfloor \alpha C \rfloor + 1 : C, ::], \tag{15}$$

where $\alpha \in (0,1)$ controls the channel split ratio.

Next, a sequential transformation is applied to F_0 using two stacked convolutional blocks. Let $Y_0 = F_0$, and define

$$Y_i = \text{CBS}_i\left(Y_{i-1}\right), \quad i = 1, 2, \tag{16}$$

where CBS_i denotes a composite module consisting of Convolution, Batch Normalization, and the SiLU activation function.

The intermediate outputs Y_1 and Y_2, together with the residual stream F_{res}, are concatenated along the channel axis to form a unified representation:

$$Z = \text{Concat}(Y_1,\ Y_2,\ F_{\text{res}}) \in \mathbb{R}^{C\times H\times W}. \tag{17}$$

Finally, a 1×1 convolution is applied to the fused feature map Z to generate the module output:

$$F_{\text{out}} = \text{Conv}_{1\times 1}(Z). \tag{18}$$

By integrating these operations, the C2f module enables more efficient gradient flow and richer semantic representation, effectively boosting the detection capacity of lightweight networks.

3.4 Dataset

In this study, based on Shanghai's waste-sorting standard, waste is classified into four major categories: hazardous waste, dry waste, recyclable waste, and wet waste [5]. The dataset was constructed by aggregating publicly available images from the Internet, smartphone photographs of everyday waste, and professional images captured with an Intel RealSense camera. A total of nine object classes covering the four primary categories were collected, yielding 7917 images. Table 1 details the mapping between each object class and its corresponding general category.

Table 1: Detail of self-built garbage dataset

Category of waste	Type of waste	Number of waste
Hazardous waste	Waste ointment	1570
	Waste battery	2478
Dry waste	Disposable cultery	1782
Recyclable waste	Paperbark	1709
	Can	1873
	Plastic bottle	1785
Wet waste	Orange	2442
	Apple	1774
	Banana	2037

4 Results and Discussion

4.1 Experimental Environment

To ensure experimental fairness, both the ablation and comparative studies were conducted on a workstation equipped with an NVIDIA RTX 3070 GPU. In the ablation experiments, each newly introduced module was trained using identical hyperparameters (initial learning rate = 0.01; batch size = 16; epochs = 100; input resolution = 640 × 640), and the resulting models were then deployed to a Raspberry Pi 4B for inference.

During deployment, the Raspberry Pi computes the target angle for each stepper motor based on YOLO-WasNet's detection outputs and transmits these commands via a serial interface to an STM32F407ZGT6 microcontroller. Upon receipt, the STM32F407ZGT6 utilizes its advanced timers to precisely drive both the

stepper motors and the TPU-based soft gripper, thus completing the grasping operation and enabling the full waste-sorting workflow.

4.2 Results of Ablation Experiments

Ablation experiments on YOLOv5s were conducted to evaluate the contributions of each enhanced module, including the MobileNetV3-Small backbone, the SCBA (SPPF + CBAM) module, the DPT-Conv operator, and the C2f block, as summarized in Table 2. The baseline YOLOv5s achieved 94% accuracy and 96.0% mAP_{50}. Substituting the backbone with MobileNetV3-Small reduced the parameter count and FLOPs to 3.74 M and 6.5 G, respectively, demonstrating that depthwise-separable convolutions can substantially lower computational overhead with only a marginal accuracy drop. Introducing DPT-Conv further cut parameters to 3.09 M and FLOPs to 5.9 G while improving both accuracy and mAP_{50}. Integrating SPPF and CBAM into the SCBA module added 0.78 M parameters and 0.6 G FLOPs but resulted in a 2.6% accuracy gain, thanks to multi-scale feature extraction and attention refinement.

Table 2: Ablation experiment results on YOLOv5

YOLOv5s	MobileNetV3-s	DPT-Conv	C2f	SCBA	Precision	mAP50	*P*/M	FLOPs/G
✓					94.00%	96.0%	7.03	15.8
✓	✓				91.40%	93.6%	3.74	6.5
✓	✓	✓			95.60%	96.4%	3.09	5.9
✓	✓	✓	✓		94.30%	96.6%	4.11	8.0
✓	✓	✓	✓	✓	96.90%	96.8%	4.89	8.6

4.3 Comparison and Analysis of Algorithms

To accurately assess the performance of YOLO-WasNet, comparative experiments were conducted on a custom waste-classification dataset using several mainstream object detectors, including Sparse R-CNN [21], VarifocalNet [22], SSD [23], CenterNet [24], DETR [25], and EfficientDet [26]. Since Sparse R-CNN, VarifocalNet, SSD, and DETR cannot converge without pretrained weights, all of these models were initialized with ImageNet-pretrained parameters, while the remaining models were trained from scratch. The parameter counts and FLOPs of each model are presented in Fig. 12. The results demonstrate that YOLO-WasNet exhibits a significant lightweight advantage: its parameter count and FLOPs are only marginally higher than those of EfficientDet, yet substantially lower than those of the other baseline models. Compared to Sparse R-CNN, VarifocalNet, SSD, CenterNet, and the Transformer-based DETR, YOLO-WasNet reduces the parameter count by approximately one-seventh of DETR's, thereby greatly facilitating model optimization and deployment.

As shown in Table 3, the effectiveness of the proposed YOLO-WasNet was validated by comparing it against several mainstream object detectors-including CenterNet, DETR, EfficientDet, SSD, VarifocalNet, and Sparse R-CNN on the same custom waste classification dataset. All models employed identical data augmentation and optimization strategies; SSD, VarifocalNet, and Sparse R-CNN were initialized with pretrained weights to accelerate convergence, while the remaining models were trained from scratch. To prevent overfitting, VarifocalNet and Sparse R-CNN were trained for only 12 epochs, whereas the other models were trained for 100 epochs.

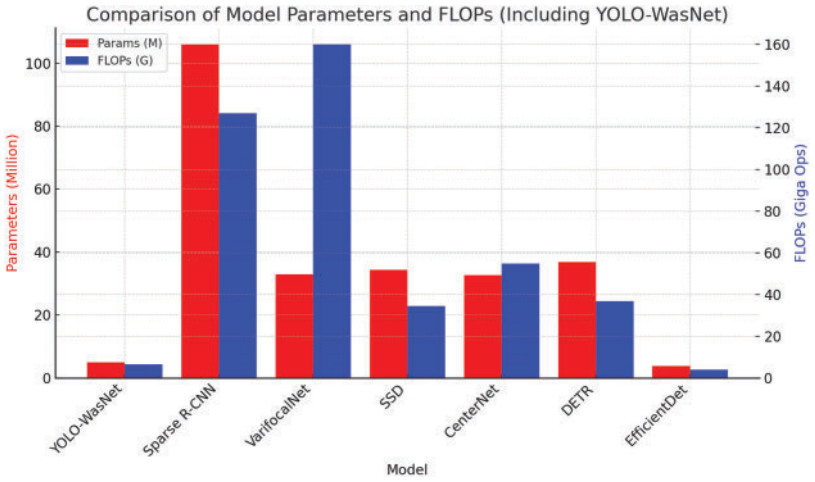

Figure 12: Comparison of FLOPs and param for various models

Table 3: Comparison of mAP_{50}, training epochs, and pretraining status for various models

Model	mAP_{50}	Epochs	Pretrained
YOLO-WasNet	96.80%	100	×
CenterNet	87.87%	100	×
DETR	82.39%	100	✓
EfficientDet	66.76%	100	×
SSD	92.33%	100	✓
Sparse R-CNN	65.30%	12	✓
VarifocalNet	81.50%	12	✓

The experimental results demonstrate that YOLO-WasNet achieved a mAP_{50} of 96.8%, substantially outperforming SSD (92.33%), VarifocalNet (81.50%), and Sparse R-CNN (65.30%), thereby fully validating the significant accuracy gains introduced by the network modifications. Meanwhile, VarifocalNet and Sparse R-CNN reached mAP_{50} scores of 81.50% and 65.30% under short training schedules, showcasing the rapid convergence advantages of pretrained initialization. Compared with ResNet-101 or Transformer-based architectures, YOLO-WasNet reduces both parameter count and FLOPs while maintaining high accuracy, significantly lowering computational overhead.

This superior performance is attributed to four key design innovations: 1) Replacing the CSPDarknet backbone with MobileNetV3-Small based on depthwise separable convolutions to reduce model parameters and computation; 2) Introducing a bi-directional prediction transformation (DPT-Conv) module between the backbone and feature fusion layers to enhance multi-scale feature representation; 3) Adopting the C2f composite fusion structure for cross-layer semantic fusion and channel reorganization; 4) Integrating the SCBA lightweight attention mechanism to strengthen salient features and suppress redundant information.

4.4 System Test

4.4.1 Recognition Speed Test

In this paper, the real-time performance of the model is validated on a Raspberry Pi 4B, as shown in Fig. 13. Although YOLOv5s is a lightweight model designed for mobile and edge devices, YOLO-WasNet

demonstrates superior real-time performance, with a single inference time of approximately 350 ms, significantly outperforming the average 480 ms of YOLOv5s. These results indicate that YOLO-WasNet effectively balances fast inference and detection accuracy on resource-constrained platforms.

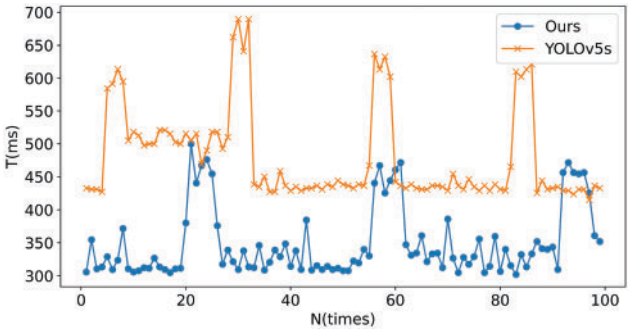

Figure 13: Comparison between YOLOv5s and YOLO-WasNet in real-time performance. Where, N is the number of recognitions, and T is the inference time

4.4.2 Object Detection Correctness Test

To evaluate the performance of the YOLO-WasNet model in practical applications, a set of typical household waste images was selected for testing, with the recognition results presented in Fig. 14. The test outcomes indicate that YOLO-WasNet achieves satisfactory performance, effectively recognizing both single objects and multiple similar objects. In many complex backgrounds, YOLO-WasNet also accurately captures the location and category of waste items, further demonstrating the model's generalization and robustness. Overall, YOLO-WasNet constitutes an efficient and accurate waste-image detection model.

Figure 14: The effect of object detection for various kinds of garbage

4.4.3 Discussions

The robust detection performance of YOLO-WasNet extends beyond technical validation and plays a meaningful role in advancing sustainable waste management practices. Accurate identification of different waste types is a critical step toward enabling intelligent sorting systems, which help reduce environmental pollution, lower landfill usage, and improve recycling efficiency. By applying such AI-driven models in real-world scenarios, it becomes possible to automate the classification process, minimize human intervention, and ensure more consistent and reliable waste sorting outcomes.

Furthermore, the deployment of models like YOLO-WasNet supports the broader goal of building greener urban environments. Its ability to handle complex scenes and recognize multiple objects ensures that recyclable materials can be more effectively separated at the source, laying the groundwork for higher recovery rates and reduced contamination. As societies continue to face growing waste management challenges, integrating efficient object detection technologies into the recycling chain offers a promising path toward a more resource-efficient and environmentally responsible future.

5 Conclusion

With increasing public awareness of environmental protection and the sustainable use of resources, waste separation has become a global and urgent task. However, existing methods for waste classification primarily rely on manual operation, which is inefficient and prone to errors. By integrating robotics and AI-based image processing technologies, automated waste sorting and collection can be achieved, significantly improving efficiency and reducing costs. YOLO-WasNet, a lightweight adaptation of YOLOv5s, incorporates attention mechanisms, an optimized backbone network, a redesigned convolutional structure in the detection head, and an improved up-sampling method. These enhancements effectively meet the requirements of embedded systems and provide an efficient, low-cost automated solution. The specific contributions are as follows:

1) The YOLO-WasNet model is proposed, which significantly reduces parameter count and FLOPs. The resulting YOLO-WasNet ensures real-time performance on embedded devices, thereby reducing hardware costs and fulfilling the market demand for efficient waste-classification systems.

2) In terms of mechanical design, a combination of the robotic-arm pentagonal algorithm and a palletizing-arm structure based on the parallelogram principle is implemented, along with flexible gripper jaws adapted to various waste shapes. The designed mechanism achieves effective gripping and precise placement of diverse waste items, enhancing the applicability and flexibility of the robotic-arm system.

3) The proposed sorting system exhibits broad application prospects, suitable for scenarios such as waste sorting, warehouse sorting, and parcel sorting. Experimental results demonstrate that using MobileNetV3-Small as the backbone substantially reduces model parameters; the SCBA module effectively improves detection accuracy; DPT-Conv achieves comparable performance to standard convolution with lower parameter count; and the C2f module provides richer gradient flow.

Acknowledgement: The authors thank all research members who provided support and assistance in this study.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Research conception and design: Anjie Wang, Zhichao Chen; Data collection: Anjie Wang; Result analysis and interpretation: Anjie Wang, Zhichao Chen; Manuscript preparation: Anjie Wang, Haining Jiao, Zhichao Chen, Jie Yang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Meng X, Tan X, Wang Y, Wen Z, Tao Y, Qian Y. Investigation on decision-making mechanism of residents' household solid waste classification and recycling behaviors. Resourc Conservat Recycl. 2019;140(11):224–34. doi:10.1016/j.resconrec.2018.09.021.
2. Zhao Z, Chen X. Exploring a new way of garbage classification with the dual-carbon target as the traction. Sustain Dev. 2023;13(2):733–9.
3. Zhang M, Wu W. Exploring the motivations and obstacles of the public's garbage classification participation: evidence from sina weibo. J Mater Cycles Waste Manag. 2023;25(4):2049–62. doi:10.1007/s10163-023-01659-y.
4. Chen Z, Yang J, Li F, Feng Z, Chen L, Jia L, et al. Foreign object detection method for railway catenary based on a scarce image generation model and lightweight perception architecture. IEEE Trans Circuits Syst Video Technol. 2025. doi:10.1109/tcsvt.2025.3567319.
5. Chen Z, Yang J, Chen L, Jiao H. Garbage classification system based on improved shufflenet v2. Resour Conserv Recycl. 2022;178(1):106090. doi:10.1016/j.resconrec.2021.106090.
6. Tian X, Shi L, Luo Y, Zhang X. Garbage classification algorithm based on improved mobilenetv3. IEEE Access. 2024;12:44799–807. doi:10.1109/access.2024.3381533.
7. Wang J. Application research of image classification algorithm based on deep learning in household garbage sorting. Heliyon. 2024;10(9):e29966. doi:10.1016/j.heliyon.2024.e29966.
8. Howard A, Sandler M, Chen B, Wang W, Chen LC, Tan M, et al. Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 1314–24.
9. Woo S, Park J, Lee JY, Kweon IS. CBAM: convolutional block attention module. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision–ECCV 2018, 1st ed. Cham, Switzerland: Springer; 2018. p. 3–19. doi:10.1007/978-3-030-01234-2_1.
10. Misra D, Nalamada T, Arasanipalai AU, Hou Q. Rotate to attend: convolutional triplet attention module. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2021 Jan 5–9; Online. p. 3139–48.
11. Sun S, Mo B, Xu J, Li D, Zhao J, Han S. Multi-yolov8: an infrared moving small object detection model based on yolov8 for air vehicle. Neurocomputing. 2024;588(28):127685. doi:10.1016/j.neucom.2024.127685.
12. Le HTN, Ngo HQT. Application of the vision-based deep learning technique for waste classification using the robotic manipulation system. Int J Cogn Comput Eng. 2025;6(5):391–400. doi:10.1016/j.ijcce.2025.02.005.
13. Wahyutama AB, Hwang M. Yolo-based object detection for separate collection of recyclables and capacity monitoring of trash bins. Electronics. 2022;11(9):1323. doi:10.3390/electronics11091323.
14. Hu B, Zhang X, Liu X, Xue R, Cheng T, Qin M, et al. The realization of a collaborative robot ACP system for intelligent garbage classification. In: 2024 IEEE 4th International Conference on Digital Twins and Parallel Intelligence (DTPI); 2024 Oct 18–20; Wuhan, China. p. 338–43.
15. Dai Z, Cai B, Lin Y, Chen J. UP-DETR: unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 1601–10.
16. Chen Q, Chen X, Wang J, Zhang S, Yao K, Feng H, et al. Group DETR: fast DETR training with group-wise one-to-many assignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 6633–42.
17. Yue X, Meng L. YOLO-MSA: a multiscale stereoscopic attention network for empty-dish recycling robots. IEEE Trans Instrum Meas. 2023;72:1–14. doi:10.1109/tim.2023.3315355.
18. Yue X, Li H, Shimizu M, Kawamura S, Meng L. YOLO-GD: a deep learning-based object detection algorithm for empty-dish recycling robots. Machines. 2022;10(5):294. doi:10.3390/machines10050294.
19. Maier J, Perret J, Huber M, Simon M, Schmitt-Rüth S, Wittenberg T, et al. Force-feedback assisted and virtual fixtures based k-wire drilling simulation. Comput Biol Med. 2019;114(1):103473. doi:10.1016/j.compbiomed.2019.103473.
20. Singhania D, Rahaman R, Yao A. C2F-TCN: a framework for semi- and fully-supervised temporal action segmentation. IEEE Trans Pattern Anal Mach Intell. 2023;45(10):11484–501. doi:10.1109/tpami.2023.3284080.

21. Sun P, Zhang R, Jiang Y, Kong T, Xu C, Zhan W, et al. Sparse r-CNN: end-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 14454–63.
22. Zhang H, Wang Y, Dayoub F, Sunderhauf N. VarifocalNet: an IoU-aware dense object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 8514–23.
23. Chen Z, Guo H, Yang J, Jiao H, Feng Z, Chen L, et al. Fast vehicle detection algorithm in traffic scene based on improved SSD. Measurement. 2022;201(7):111655. doi:10.1016/j.measurement.2022.111655.
24. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q. Centernet: keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 6568–77.
25. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. Computer Vision—ECCV 2020. Cham, Switzerland: Springer International Publishing; 2020. p. 213–29. doi:10.1007/978-3-030-58452-8_13.
26. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 10778–87.