Current Science
CS 5 (2), 1679-1693 (2025)
Received 18 Dec 2024 | **Revised 25 Jan** 2025 | **Accepted 20 Feb**. 2025 | **Online Available 1 April** 2025
https://doi.org/ 10.52845/CS/2025-5-2-31

Current Science

Current Science

OPEN ACCESS

ISSN(O)2795-8639

ORIGINAL ARTICLE

Performance Analysis of Glass Surface Detection Based on the Yolos

Chao Yan[1,2*], Hao Zhang[3], Fei Ye[4], Wei Xu[5]

[1]School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

[2]School of Electrical Engineering and Automation, Suzhou University of Technology, Changshu 215500, China

[3]Henan Airport Group Co., Ltd, Zhengzhou 450000, China

[4]School of Automobile Engineering, Changshu Institute of Technology, Changshu 215500, China

[5]School of Geographic Information and Tourism, Chuzhou University, Chuzhou 239001, China

[*]Corresponding Author: Chao Yan

Abstract

Glass surface detection poses a significant challenge in computer vision due to transparency, reflections, and complex optical effects. This study systematically evaluates the performance of seven YOLO models (YOLOv5 to YOLOv12) for glass surface detection tasks, comparing their detection accuracy, computational efficiency, and real-time capabilities on a unified dataset. Experimental results demonstrate that YOLOv9 achieves the highest detection accuracy with 81.1% precision and 79.6% mean average precision at IoU 0.5 (mAP@0.5). However, its deep architecture (489 layers) leads to lower inference speed (500 FPS) and longer training duration (12.08 hours). YOLOv11 strikes the best balance between efficiency and accuracy, offering the lowest computational cost (6.3 GFLOPs) with competitive recall (71.9%) and F1-score (0.76), making it suitable for edge computing scenarios. YOLOv6 (666 FPS) and YOLOv8 (588 FPS) excel in real-time performance but require trade-offs in terms of missed detection rate (25.2%) and false positive rate (7.3%). This work explores optimization directions for the YOLO series in lightweight architecture design, multimodal fusion, and adaptability to data distributions, providing theoretical support for deploying perception systems in mobile robotics. We have open-sourced our dataset at https://github.com/chaoyanSEU/Glass_surface_detect_dataset to benefit the research community.

Keywords-Glass surface detection, YOLO, Computer Vision, Model Training, Real-time Test

Introduction

In the field of visual perception, glass surface detection is a relatively new and highly challenging task. Since glass surfaces typically lack prominent texture information, detecting them in real-world scenarios is significantly more challenging compared to surfaces of other materials. Meanwhile, the dynamic appearance of glass becomes more complex due to intricate light-material interactions (e.g., reflection, refraction, and transmission) and variations in background environments or object shapes. These factors may lead to catastrophic failures in visual tasks such as semantic segmentation, depth estimation, and instance segmentation in glass surface detection scenarios. Additionally, the transparency of glass often causes visual systems to misinterpret objects behind glass as foreground targets while ignoring the glass itself. Such detection failures may result in real-world misjudgments, posing risks for applications like robotic navigation. The difficulties primarily manifest in three aspects: (1) the optical properties of glass surfaces (reflection, transmission, and scattering) vary significantly under different lighting conditions, leading to insufficient robustness in existing detection models based on

RGB images or depth maps [1]; (2) the transparency and background perspective effects of glass can cause environmental confusion, complicating the distinction between glass and its surroundings [2]; (3) effective glass surface detection typically requires multimodal sensor collaboration, where the high costs of data collection and annotation further increase implementation challenges [3].

With the advancement of object detection technology, algorithm selection is crucial for achieving efficient and accurate glass surface detection. Current studies have explored YOLO algorithm applications in mobile robot perception systems [4–7]. Compared to traditional two-stage detection algorithms (e.g., Faster R-CNN [8], Mask R-CNN [9], Dynamic R-CNN [10]), YOLO treats detection as a single regression problem, performing localization and classification through a single forward propagation. In contrast, two-stage methods require multiple network modules for candidate region generation, classification, and regression operations. This single-stage mechanism enables YOLO to achieve faster and more accurate glass surface detection. Compared to other single-stage algorithms (e.g., SSD [11], EfficientDet [12]), YOLO's single-pass prediction approach maintains high speed while achieving superior accuracy in detection and localization. Therefore, for mobile robot perception systems requiring glass surface detection, YOLO algorithms meet real-time and efficiency demands while enhancing system performance.

This paper evaluates and compares the performance of YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, YOLOv11, and YOLOv12 in glass surface detection tasks, aiming to comprehensively assess their applicability in mobile robot perception systems and reveal their challenges and potential in real industrial scenarios. The contributions are as follows:

(1) A dedicated multi-scene glass surface detection dataset. Addressing the limitations of existing public datasets in diversity and interference coverage, we develop a dataset containing 15,339 images spanning over 10 scenarios (e.g., shopping malls, offices, homes, vehicle glass, industrial equipment protective glass) with varied lighting conditions (strong reflections, low light, frameless interference) and complex factors (dynamic occlusion, background perspective).

(2) First systematic comparison of YOLO series models for glass surface detection. Overcoming previous limitations of evaluating single models or partial versions, we conduct a comprehensive analysis of seven YOLO versions (v5–v12), covering computational efficiency (parameters, GFLOPs, FPS, training time) and detection accuracy (precision, recall, mAP@0.5).

(3) Validation of single-stage detectors for glass surface detection. To address the high cost and deployment complexity of traditional multimodal sensor-based approaches (e.g., polarization/depth cameras), we pioneer the verification of single-stage models (YOLO series) under pure RGB input for glass surface detection.

2 Related Work

2.1 Glass Surfaces Detection

In the field of computer vision, glass surface detection has always been a highly challenging task. The transparency of glass and its dynamic optical phenomena (e.g., reflection, refraction, and transmission) cause glass surfaces to blend with their background environments, complicating accurate detection.

Visual distortion-based methods utilize optical phenomena such as brightness degradation, color shifts, and texture distortion to detect Glass surfaces. Tan et al. [13] proposed a network integrating a multi-scale Visual Distortion Awareness module (VDA) and a Structure Refinement Module (SRM). The VDA extracts distortion differences between glass and non-glass regions through parallel multi-resolution convolutional streams, while the SRM refines edges using structural feature information from classification tasks. This method improves Intersection over Union (IoU) and F1-scores on benchmark datasets while reducing Mean Absolute Error (MAE) and Balanced Error Rate (BER) without requiring additional annotations. However, performance deteriorates under complex occlusion boundaries. Context-based methods detect Glass surfaces by capturing multi-scale context information. Mei et al. [14] introduced a large-scale context feature aggregation module, and Yu et al. [15] proposed a discriminative enhancement module with a focus-exploration feature fusion strategy to combine high-level semantics with low-level details. While

effective in regular scenarios, these methods struggle in complex environments due to inadequate modeling of Glass material properties. Boundary-based methods focus on edge feature extraction techniques. He et al. [16] employed an enhanced boundary learning strategy, extracting boundary features through a differential module and refining edges with graph convolutional networks. However, reliance on explicit boundaries (e.g., window frames) limits generalization, as RGB images alone often lack such references. Reflection- and polarization-based methods enhance detection using auxiliary data. Lin et al. [17] utilized reflective images as additional inputs, and Liu et al. [18] modeled spatiotemporal reflection cues. Mei et al. [19] fused RGB and polarization camera data to improve robustness. Despite improved performance, these approaches require costly sensors or annotations, hindering practical deployment. Visual foundation model-based methods utilize synthetic data and advanced architectures. Hao et al. [20] generated the S-GSD dataset using Stable Diffusion and designed the GEM model, which combines SAM's segmentation capability with a query decoder for efficient detection. The synthetic data demonstrates strong performance in zero-shot and transfer learning, highlighting foundation models' potential for data generation and feature extraction.

Despite progress in accuracy and robustness, glass surface detection still faces challenges: (1) performance degradation under occlusion, lighting variations, or extreme glass sizes; (2) high computational costs from complex modules. Integrating YOLO's single-stage architecture and Feature Pyramid Network (FPN) could optimize real-time performance and multi-scale detection. Additionally, YOLO's lightweight design may enable efficient deployment on mobile devices, facilitating advancements in applications like robot navigation and autonomous driving.

2.2 YOLO Models

Glass surface detection, as one of the core tasks, has driven algorithm evolution with a focus on balancing speed and accuracy. Since the You Only Look Once (YOLO) series was proposed in 2015 [21], it has continuously innovated in real-time detection and become a dominant approach, with the latest version being YOLOv12 [22]. **Figure 1** details the evolution of YOLO models, where each iteration achieves significant leaps in glass surface detection accuracy, computational efficiency, and adaptability to diverse computer vision tasks. This progression underscores rapid advancements in detection technology, with each version introducing innovations and broadening application scenarios.

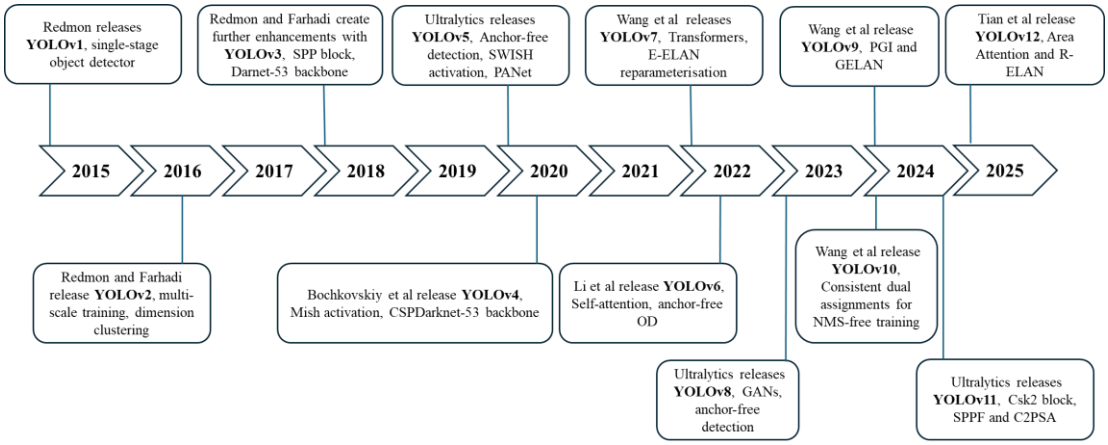


Figure 1 The evolution process of the YOLOv1 model to the latest version YOLOv12

As the pioneering work, YOLOv1 [21] first modeled glass surface detection as a single-stage regression problem. By dividing images into grids and directly predicting bounding boxes and class probabilities, it enabled end-to-end training and established YOLO's "speed-first" philosophy, though its accuracy initially lagged behind two-stage methods. YOLOv2 [23] introduced anchor boxes, multi-scale training strategies, and the Darknet-19 backbone, improving localization and detection accuracy. YOLOv3 [24] adopted Darknet-53 and the Feature Pyramid Network (FPN) for multi-scale prediction, balancing

accuracy and speed.

In the performance breakthrough stage, YOLOv4 [25] proposed CSPDarknet53 with Mosaic augmentation and Complete Intersection over Union (CIoU) loss, reducing redundancy while improving accuracy. YOLOv5 [26] transitioned to PyTorch, integrating adaptive anchor calculation and the Spatial Pyramid Pooling Fast (SPPF) module for industrial deployment. YOLOv6 [27] leveraged RepVGG reparameterization and the Bidirectional Concatenation (BiC) fusion module to enable real-time inference on mobile devices.

The multi-task expansion phase saw YOLOv7 [28] introduce Extended Efficient Layer Aggregation (E-ELAN) for instance segmentation and pose estimation, while YOLOv8 [29] integrated the Cross-stage Feature Fusion (C2f) module for panoramic segmentation and keypoint detection. Recent frontier explorations include: YOLOv9 [30], which proposed the Generalized Efficient Layer Aggregation (GELAN) to mitigate deep network information loss; YOLOv10 [31], pioneering Non-Maximum Suppression (NMS)-free training for edge deployment; YOLOv11 [32], incorporating Cross-3-kernel (C3k2) and Cross-Partial Self-Attention (C2PSA) modules for enhanced feature extraction; and YOLOv12 [22], introducing the Regional Attention Mechanism (A^2) and Residual-ELAN (R-ELAN) to match Convolutional Neural Networks (CNNs) in real-time performance.

Technological evolution trends highlight lightweight architectures, multimodal fusion, innovative training strategies, and hardware adaptability. Future breakthroughs may focus on adaptive multispectral processing, few-shot learning, and 3D glass surface detection.

3 Model Training and Result Analysis

This experiment aims to evaluate and compare the performance of YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, YOLOv11, and YOLOv12 models in glass surface detection tasks. To achieve this goal, these seven models are trained and tested using the same dataset, enabling a direct comparison of their performance. This paper conducts a comparative analysis of the seven models, aiming to reveal their advantages and disadvantages and explore suitable application scenarios in real-world robotic inspection environments.

3.1 Dataset and Evaluation Metrics

Experiments are conducted on widely used glass surface detection datasets, including GDD [14], HSO [15], GSD [17], and Trans10K [33], to validate the proposed method. To align with the target scenarios, the dataset is re-screened, reconstructed, and annotated. This dataset is specifically designed for training and evaluating glass surface detection models in robotic operating environments. The dataset contains 15,339 images, divided into 10,407 training images, 2,932 validation images, and 2,000 test images. This distribution ensures sufficient data diversity for feature learning during training and enables fair evaluation of generalization during validation and testing.

The dataset covers diverse environments, including shopping malls, offices, homes, car windows, industrial machinery, showcases, and countertops. It also includes indoor/outdoor decorative glass under varied weather and lighting conditions, and challenging scenarios (e.g., strong illumination, reflective surroundings, frameless glass, and empty-area interference). These simulate complex glass surface detection scenarios encountered in real-world robotics. **Figure 2** depicts the typical scenarios in the dataset.

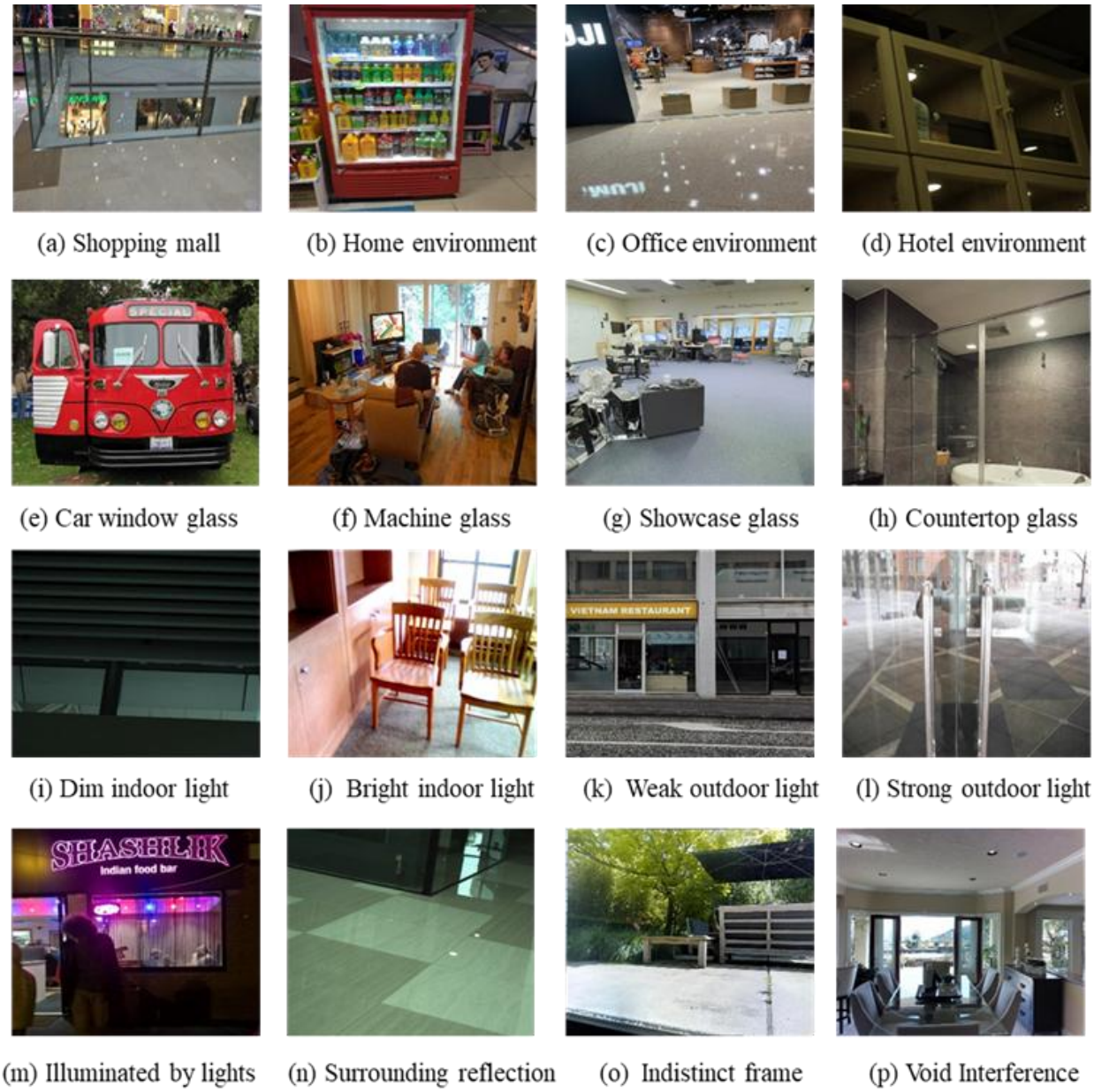


Figure 2 Images of typical scenarios in the dataset

During preprocessing, images are normalized to ensure uniform scale and distribution for model input. To enhance model robustness, data augmentation techniques (e.g., random rotation, flipping, scaling, and color jittering) are applied. To enhance model robustness, data augmentation techniques (e.g., random rotation, flipping, scaling, and color jittering) are applied. **Figure 3** depicts the position and size distribution of Glass surface objects in the images. Most targets are concentrated in the vertical center (y-axis) and evenly distributed horizontally (x-axis) from **Figure 3**. This implies that glass surface detection models should prioritize the central vertical region. Size distribution analysis shows that most Glass surface objects occupy small image areas, suggesting distant positioning. The bounding box heights are more concentrated than widths, potentially due to Glass surface physical properties.

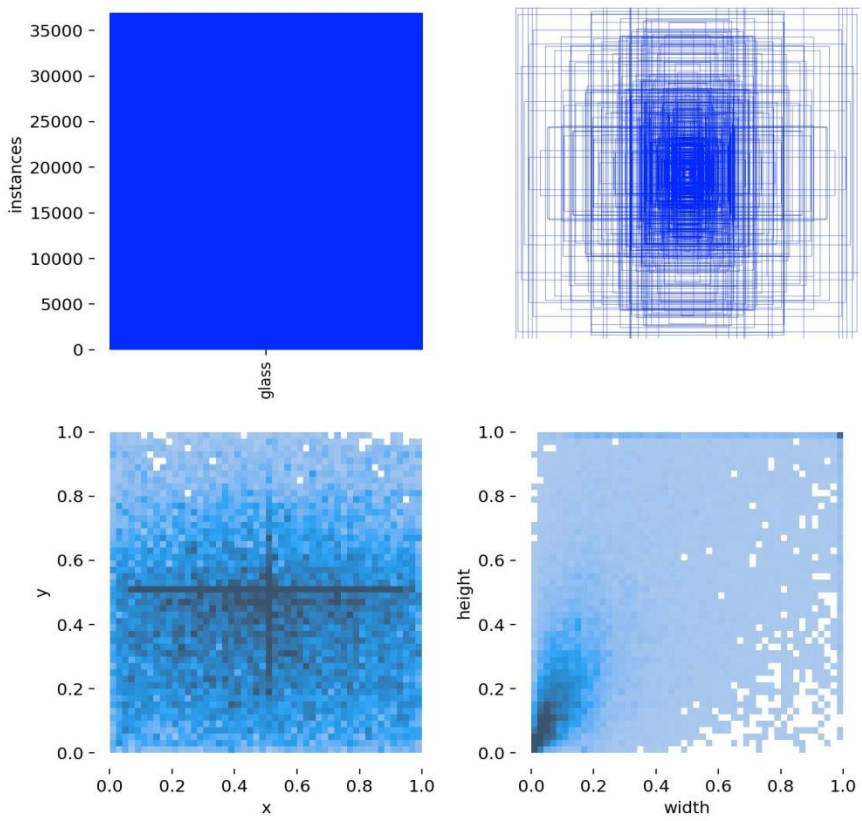


Figure 3. The distribution of objects on glass surfaces in the images

Four evaluation metrics are used: precision (P), recall (R), F1-score, and mean Average Precision (mAP). The formulas for precision (P) and recall (R) are defined:

$$\begin{cases} P = \dfrac{TP}{TP + FP} \\ R = \dfrac{TP}{TP + FN} \end{cases} \tag{1}$$

Where TP (true positive) represents the number of pixels that are predicted to be objects on glass surfaces and are actually objects on glass surfaces; FP (false positive) represents the number of pixels that are predicted to be objects on glass surfaces but are actually objects on non-glass surfaces; FN (false negative) represents the number of pixels that are predicted to be objects on non-glass surfaces but are actually objects on glass surfaces.

Precision and recall are competing metrics, requiring balanced consideration. The F1-score is the harmonic mean of precision and recall:

$$F_1 = \frac{2P \cdot R}{P + R} = \frac{2P \cdot R}{2TP + FP + FN} \tag{2}$$

mAP is the average of AP values across all categories. AP measures per-category accuracy, while mAP averages APs to evaluate overall model accuracy. mAP@0.5 calculates average precision at an IoU threshold of 0.5.

3.2 Model Training

Model training is performed using the PyTorch framework with the Ultralytics YOLO library, on a single NVIDIA GeForce RTX 2080 Super GPU. Input images are cropped and resized to 640×640 pixels during training. The specific settings of the hyperparameters are shown in **Table 1**.

Table 1 YOLOs hyperparameter settings

Hyperparameters	Values	Hyperparameters	Values
Input Size	640*640	Batch Size	30
Epochs	300	Optimizer	SGD
Learning Rate	0.01	Momentum	0.9
Weight Decay	0.00046875	IoU Threshold	0.5

Table 2 summarizes the layers, parameters, GFLOPs (giga floating-point operations), and FPS (frames per second) of YOLOv5 to YOLOv12 models. Layers: Increasing the number of layers enhances complex feature representation, improving model accuracy. However, deeper architectures increase computational complexity, parameters, training time, and resource demands, potentially slowing inference. Parameters: Higher parameter counts improve feature learning capacity, enhancing generalization and accuracy. However, increased parameters raise computational and storage costs. For resource-constrained devices (e.g., mobile/edge devices), models with fewer parameters are preferred for faster inference and lower power consumption. FLOPs (floating-point operations) measures computational complexity, enabling model efficiency comparisons. Lower FLOPs indicate reduced computational complexity. FPS (frames per second) measures processing speed, with higher values preferred.

Table 2. Model parameter statistics

Models	Layers	Parameters	GFLOPs	FPS	Training duration (hours)
YOLOv5	193	2 503 139	7.1	625	10.632
YOLOv6	**142**	4 233 843	11.8	**666**	11.699
YOLOv8	168	3 005 843	8.1	588	10.934
YOLOv9	489	**1 970 979**	7.6	500	12.078
YOLOv10	285	2 694 806	8.2	625	11.696
YOLOv11	238	2 582 347	**6.3**	588	**10.201**
YOLOv12	159	2556923	**6.3**	500	10.487

From **Table 2**, YOLOv6 achieves the highest inference speed (666 FPS) through reduced layers (142) and wider networks, but its high parameters (4.23M) and computational cost (11.8 GFLOPs) indicate hardware-dependent optimization. YOLOv11 and YOLOv12 achieve the lowest GFLOPs (6.3) with moderate parameters (2.58M and 2.56M), demonstrating edge computing potential. YOLOv9 employs depthwise separable convolutions for lightweight design, achieving the fewest parameters (1.97M) and moderate computation (7.6 GFLOPs). However, its depth (489 layers) increases training time (12.08 hours) and limits inference speed (500 FPS), favoring offline high-precision tasks. From YOLOv5 to YOLOv12, computational efficiency improves (GFLOPs decreased from 7.1 to 6.3), with non-monotonic parameter/layer changes reflecting a depth-width-efficiency trade-off in optimization. **Figure 4** shows the training curves of box_loss (bounding box loss), cls_loss (classification loss), and dfl_loss (distribution focal loss) for YOLOv11. box_loss measures the discrepancy between predicted and ground-truth bounding boxes. cls_loss quantifies classification errors. dfl_loss optimizes bounding box probability distributions, enhancing localization accuracy.

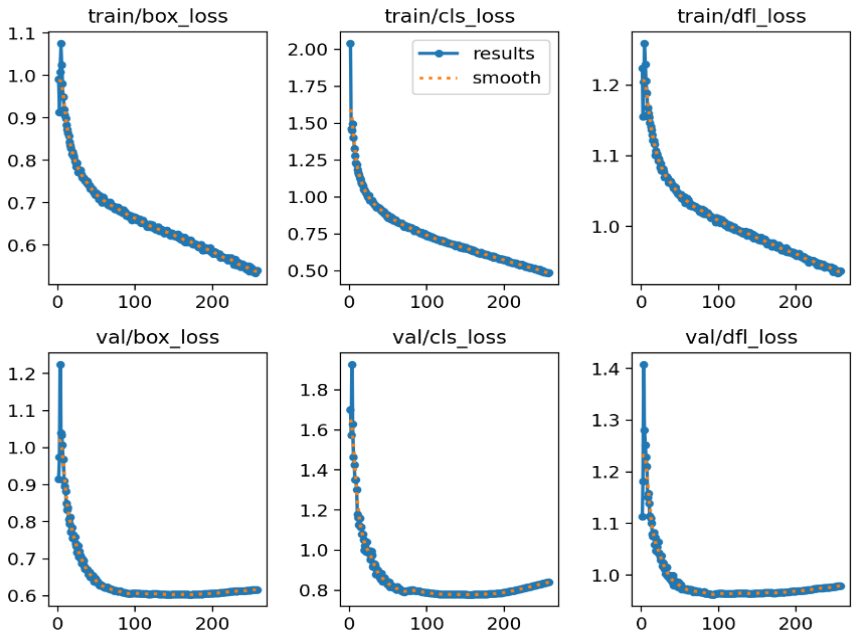


Figure 4. Training results of the YOLOv11

From **Figure 4**, box_loss, cls_loss, and dfl_loss decrease with iterations, showing improved object localization, classification, and confidence prediction. The rapid cls_loss decline indicates strong inter-class discrimination. Validation loss follows a similar trend, confirming generalization on unseen data. Higher validation loss reflects evaluation on non-training data. Other YOLO versions exhibit analogous loss patterns.

3.3 Analysis of Experimental Results

This experiment aims to evaluate and compare the performance of YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, YOLOv11, and YOLOv12 models in glass surface detection tasks. To achieve this goal, these seven models are trained and tested using the same dataset, enabling a direct comparison of their performance. A comparative analysis of the seven models is conducted to reveal their strengths and weaknesses and explore suitable application scenarios in industrial environments. **Table 3** summarizes the performance of YOLOv5–YOLOv12 models in terms of precision, recall, F1-score, and mAP@0.5.

Table 3. Comparison of detection performance of different YOLO models

Models	Precision (%)	Recall (%)	F1-score	mAP@0.5 (%)
YOLOv5	77.6	70.2	0.74	77.0
YOLOv6	78.0	69.0	0.73	76.7
YOLOv8	79.8	71.6	0.75	78.2
YOLOv9	**81.1**	71.5	0.76	**79.6**
YOLOv10	80.0	71.3	0.75	78.7
YOLOv11	80.2	71.9	0.76	78.7
YOLOv12	80.3	**74.1**	**0.77**	68.3

From **Table 1**, **Table 2**, and **Figure** 4 (training dynamics), YOLOv9 achieves the highest precision (81.1%) and mAP@0.5 (79.6%), leveraging its deep architecture (489 layers) for complex feature extraction. However, its slower inference speed (500 FPS) and longer training time (12.08 hours) limit real-time applicability. Although YOLOv12 achieves the highest recall (74.1%) and F1-score (0.77), its lower mAP@0.5 (68.3%) suggests potential bounding box localization errors, requiring further validation of data distribution adaptability and training strategies. YOLOv11 balances efficiency and accuracy, achieving the lowest computational cost (6.3 GFLOPs) with competitive recall (71.9%) and F1-score (0.76), demonstrating strong

potential for edge computing.

The YOLO series exhibits a non-monotonic accuracy-speed trade-off. YOLOv6 achieves the highest speed (666 FPS) via a shallow (142-layer) and wide architecture, but its high parameters (4.23M) and GFLOPs (11.8) indicate hardware-dependent efficiency. YOLOv8 (168 layers) and YOLOv10 (285 layers) balance accuracy (mAP@0.5: 78.2% and 78.7%) and speed (588–625 FPS), validating the depth-width-efficiency dynamic optimization strategy. For high-precision offline glass surface detection, YOLOv9 is preferred. For real-time edge computing, YOLOv11/YOLOv10 (F1-score: 0.75–0.76, GFLOPs: 6.3–8.2) are optimal. YOLOv6 (666 FPS) suits ultra-real-time monitoring but trades off recall (69.0%) and mAP@0.5 (76.7%).

Figure 5 shows ground-truth label of the validation set, while **Figure 6** displays YOLOv9 predictions, with missed/mispredicted labels highlighted in red bounding boxes. Despite promising training results from **Figure 5 and 6**, YOLOv9 exhibits missed and misdetected cases in certain glass surface detection scenarios.

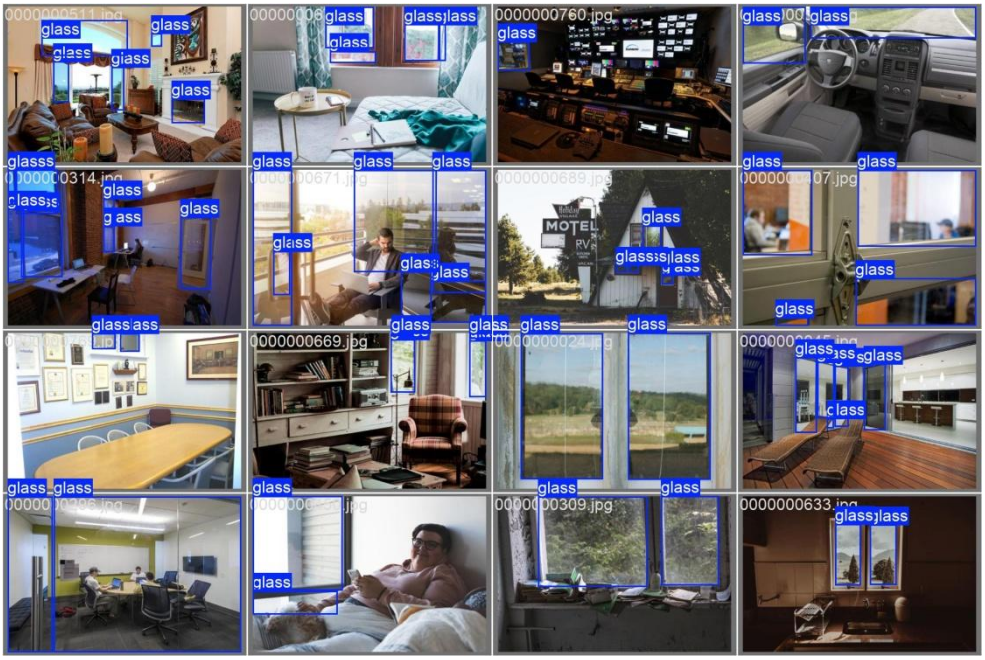


Figure 5. Ground-truth label situation of the validation set

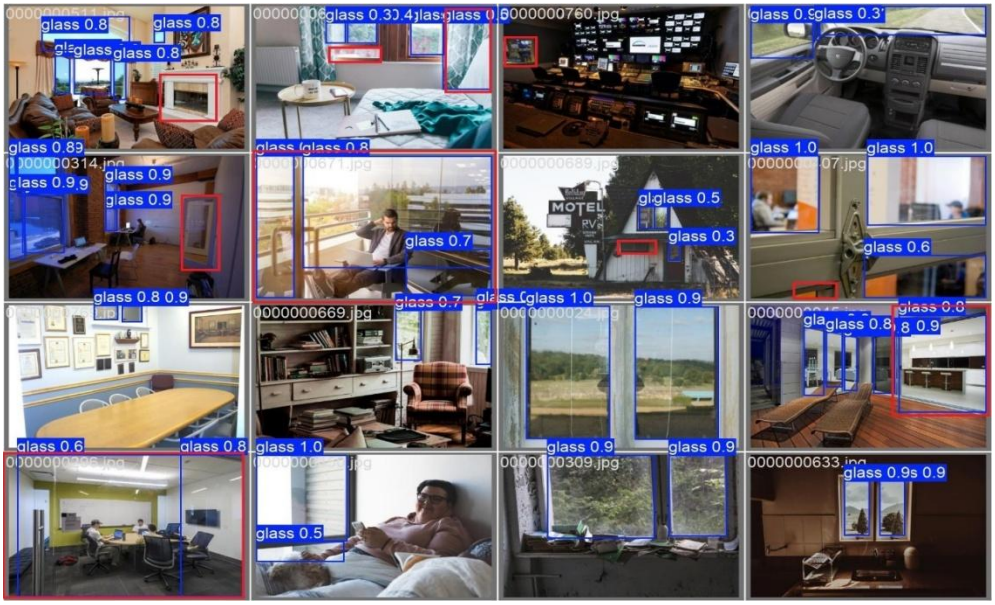


Figure 6. Label situation predicted by the YOLOv9, in which the red bounding boxes are marked for incorrect predictions

4 Real-time Test and Result Analysis

4.1 Mobile Device and Data Collection

This experiment deploys the trained YOLO models on a mobile device for real-time glass surface detection, validating its feasibility and effectiveness in mobile scenarios. The mobile device selected for this experiment is SLAMPlusPro, as shown in **Figure 7.** equipped with an OrinNX processor, a 1024-core NVIDIA Ampere GPU (32 Tensor Cores), and AI performance of 100 TOPS, providing robust acceleration for deep learning inference.

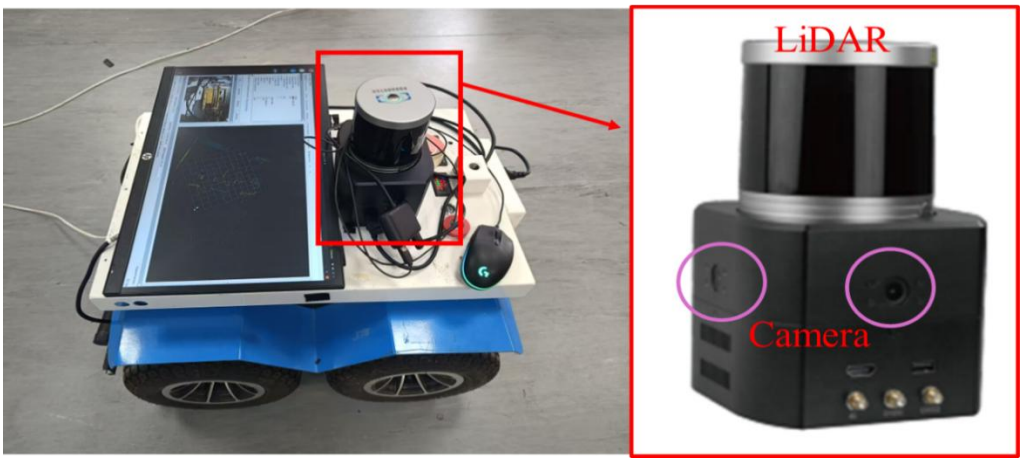


Figure 7 The mobile terminal SLAMPlusPro in the real-time test, which includes 1 RS 16-line Lidar, 4 monocular cameras, 1 IMU, and 1 GNSS receiver.

To evaluate model performance on the mobile device, a dataset of 700 images was compiled from an intelligent factory production line at a university. The dataset spans diverse environments: indoor offices, factory floors, corridors, large glass walls (indoor/outdoor), and outdoor walkways, covering glass surface detection targets such as curtain walls, windows, and machinery protective glass. **Figure 8** illustrates representative scenarios from field testing, including complex glass surface detection challenges.

Figure 8 Typical scenarios in the actual test process

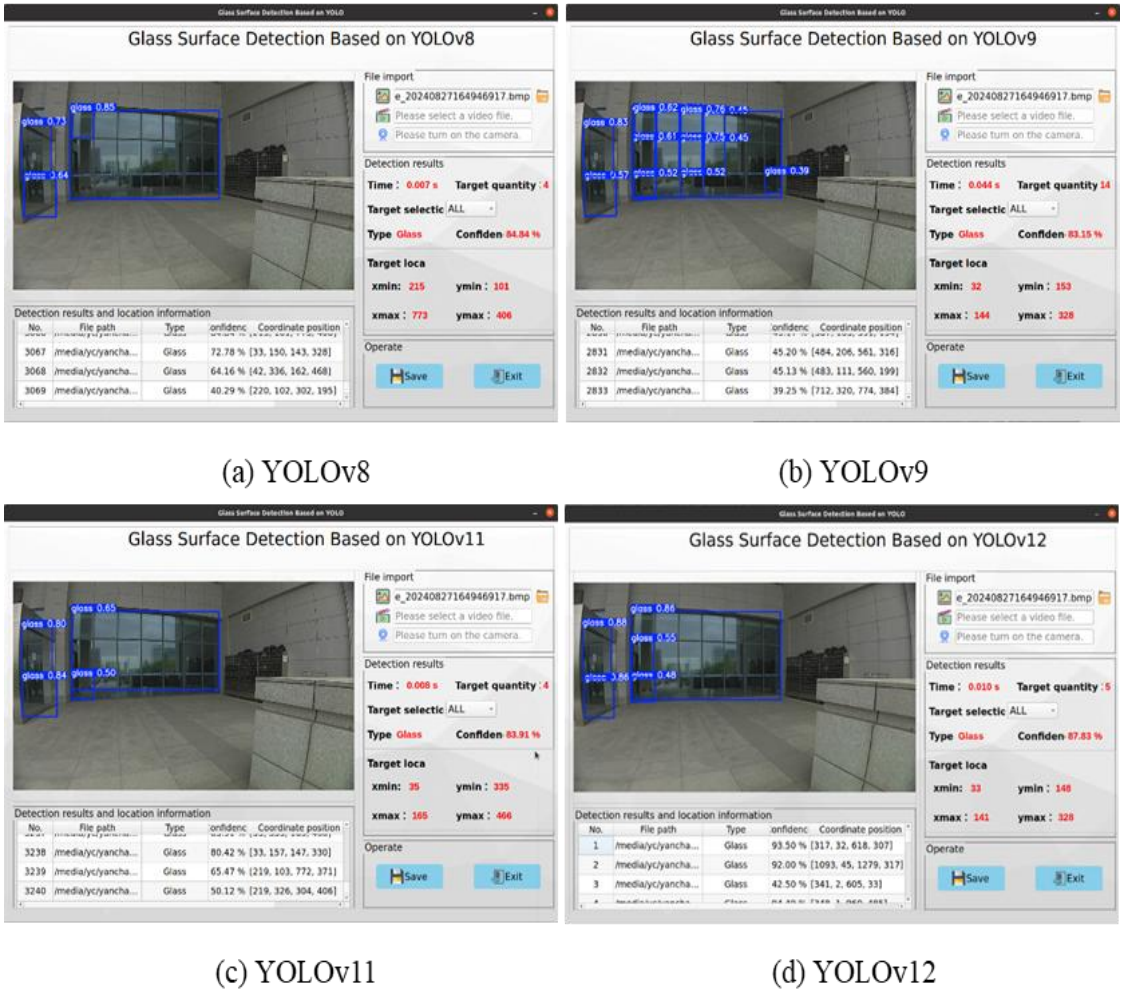
4.2 Analysis of Experimental Results

To better assess real-world performance, this section adopts industrial-relevant metrics: missed detection rate and false positive rate, replacing precision, recall, and mAP. The metrics are defined as:

$$\begin{cases} Ur = \dfrac{B}{A} \times 100\% \\ Mr = \dfrac{C}{A} \times 100\% \end{cases} \tag{3}$$

Where Ur and Mr represent the undetected rate and misjudgment rate respectively; A represents the total number of objects on the glass surface in the test; B represents the number of samples that are actually objects on the glass surface but not detected; C represents the number of samples that are actually not objects on the glass surface but are misidentified.

Figure 9 visualizes the detection process of YOLOv8–YOLOv12, **Figure 10** shows YOLOv12 results, and **Table 4** summarizes missed detection rate, false positive rate, and single-frame detection time of different YOLO models.



(a) YOLOv8 (b) YOLOv9

(c) YOLOv11 (d) YOLOv12

Figure 9 The detection process of YOLOv8, YOLOv9, YOLOv11, and YOLOv12

Figure 10 The detection results of YOLOv12

Table 4 Statistics of the undetected rate and misjudgment rate results of the YOLOs

Models	Undetected rate (%)	Misjudgment rate (%)	Single-frame processing time (s)
YOLOv5	20.7	6.9	**0.007**
YOLOv6	25.2	**5.3**	**0.007**
YOLOv8	18.5	7.3	**0.007**
YOLOv9	21.1	6.2	0.044
YOLOv10	23.6	5.7	0.008
YOLOv11	19.2	8.0	0.008
YOLOv12	**16.3**	7.8	0.010

From **Figures 9–10** and **Table 4**, YOLOv6 achieves the highest theoretical speed (666 FPS), but its actual per-frame latency (0.007 s) matches YOLOv5/YOLOv8, indicating hardware optimization limits. Despite high parameters (4.23M) and computational cost (11.8 GFLOPs), YOLOv6's hardware-specific optimizations underutilize its theoretical potential. YOLOv9's deep architecture (489 layers) results in high latency (0.044 s), making it unsuitable for real-time mobile applications. YOLOv11 (0.008 s) and YOLOv12 (0.010 s) exhibit low computation (6.3 GFLOPs) and fast inference, demonstrating edge computing viability. YOLOv9 achieves superior accuracy (precision: 81.1%, mAP@0.5: 79.6%) due to its deep architecture. However, its high missed detection rate (21.1%) reveals limited generalization in complex industrial environments. YOLOv12 achieves the lowest missed detection rate (16.3%) but suffers reduced mAP@0.5 (68.3%) and elevated false positives (7.8%), suggesting a recall-localization trade-off. YOLOv8 balances performance: missed detection (18.5%), false positives (7.3%), mAP@0.5 (78.2%), and speed (588 FPS), making it ideal for accuracy-speed trade-offs. YOLOv9 suits offline high-precision tasks but is constrained by latency (0.044 s) and training time (12.08 h) for real-time use. YOLOv11/YOLOv12's lightweight design enables mobile deployment, though higher false positive rates (7.8–8.0%) must be accepted.

YOLOv8 optimally balances accuracy, speed, and resource efficiency, fulfilling real-time industrial monitoring demands. On mobile devices, YOLOv8/YOLOv11 excel in real-time applications, whereas YOLOv9 dominates offline high-precision tasks.

5. Discussion

This study systematically evaluates seven YOLO models (v5–v12) in glass surface detection tasks, revealing trade-offs between accuracy, speed, and resource efficiency. The experimental results further verify the following key findings:

(1) Model Architecture-Performance Correlation. YOLOv9's deep architecture (489 layers), leveraging multi-scale feature extraction and parameter sharing, achieves the highest accuracy (mAP@0.5: 79.6%) in complex scenarios. However, its depth reduces inference speed (500 FPS), demonstrating a trade-off between feature representation capacity and computational efficiency. In contrast, YOLOv12 employs a Regional Attention Mechanism (A^2) and lightweight design (6.3 GFLOPs), maintaining high recall (74.1%) with reduced computational demand. However, its lower mAP@0.5 (68.3%) indicates potential localization accuracy degradation, highlighting a feature enhancement vs. localization accuracy trade-off.

(2) Real-Time Performance vs. Accuracy. YOLOv6 (666 FPS) and YOLOv8 (588 FPS) achieve high speeds via shallow architectures and hardware optimizations (e.g., reparameterization), but sacrifice robustness. YOLOv6's high missed detection rate (25.2%) and YOLOv8's false positive rate (7.3%) reflect robustness compromises for speed. YOLOv11 balances low computation (6.3 GFLOPs) and accuracy (F1-score: 0.76), suggesting lightweight designs (e.g., depthwise convolutions) and multi-task strategies (e.g., distribution focal loss) as future directions.

(3) Data Distribution and Generalization Challenges. YOLOv9 exhibits high missed detection (21.1%) under occlusion/lighting variations, while YOLOv12's false positives (7.8%) increase in empty-area scenarios. Training data bias may explain this: small-sized Glass surface objects dominate the dataset (**Figure 3**), limiting distant target feature learning. Synthetic data (e.g., S-GSD [20]) reduce annotation costs, but domain gaps in optical properties require further study.

(4) Multimodal Fusion Potential. Prior work (Lin et al. [17]: reflection images; Mei et al. [19]: polarization data) demonstrates multimodal fusion improves glass surface detection robustness. This study focuses on RGB inputs; future work should integrate YOLO's multi-task capabilities (e.g., YOLOv7 instance segmentation) with cross-modal fusion (e.g., depth/polarization) to address dynamic optical effects.

6. Conclusion

This study systematically evaluates seven YOLO models (v5–v12) in glass surface detection tasks, revealing trade-offs between accuracy, efficiency, and real-time performance. A dedicated multi-scene glass surface detection dataset was constructed, and a comprehensive analysis was conducted under a unified experimental framework. YOLOv9 achieves the highest detection accuracy (mAP@0.5: 79.6%, precision: 81.1%) through its deep architecture (489 layers) and parameter sharing, making it ideal for offline high-precision tasks. However, its deep architecture limits inference speed (500 FPS) and increases training time (12.08 hours). YOLOv11 demonstrates strong edge computing potential with lightweight design (6.3 GFLOPs) and balanced metrics (recall: 71.9%, F1-score: 0.76). YOLOv6 (666 FPS) and YOLOv8 (588 FPS) excel in real-time performance but require balancing missed detection (25.2%) and false positive rates (7.3%).

This study pioneers the feasibility of single-stage YOLO models for glass surface detection using RGB-only input, overcoming traditional multimodal sensor dependencies. The YOLO series, enhanced by Feature Pyramid Networks (FPN) and dynamic optimization, addresses glass transparency, reflections, and background clutter, enabling lightweight perception systems for mobile robots.

Future directions include: Multimodal fusion integrating polarization, depth, or reflection data to tackle dynamic optical effects; Architecture lightweighting via attention mechanisms and reparameterization, inspired by YOLOv11/v12.

Funding: The Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 24KJB420001), and the National Natural Science Foundations of China (No. 42304095). The authors would like to thank the referees or their constructive comments.

Data Availability Statement: Publicly available datasets were analysed in this study. These data can be found here: GDD dataset https://github.com/Charmve/Mirror-Glass-Detection/tree/master/Dataset/train (accessed on 1 August 2024); Trans10K dataset https:// github.com/xieenze/Segment_Transparent_Objects?tab=readme-ov-file (accessed on 1 August 2024).

References

1. Hu R, Rohrbach M, Darrell T. Segmentation from natural language expressions [C]// Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 108-124. https://doi.org/10.1007/978-3-319-46448-0_7
2. Lin J, Yeung Y H, Lau R W H. Depth-aware glass surface detection with cross-modal context mining[J]. arXiv preprint arXiv:2206.11250, 2022.
3. Sajjan S, Moore M, Pan M, et al. Clear grasp: 3d shape estimation of transparent objects for manipulation[C]//2020 IEEE international conference on robotics and automation (ICRA). IEEE, 2020: 3634-3642. https://doi.org/10.1109/icra40945.2020.9197518
4. Wu W, Guo L, Gao H, et al. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint[J]. Neural Computing and Applications, 2022: 1-16. https://doi.org/10.1007/s00521-021-06764-3
5. Yu H, Wang Q, Yan C, et al. DLD-SLAM: RGB-D Visual Simultaneous Localisation and Mapping in Indoor Dynamic Environments Based on Deep Learning[J]. Remote Sensing, 2024, 16(2): 246. https://doi.org/10.3390/rs16020246
6. Li Z, Xu B, Wu D, et al. A YOLO-GGCNN based grasping framework for mobile robots in unknown environments[J]. Expert Systems with Applications, 2023, 225: 119993. https://doi.org/10.1016/j.eswa.2023.119993
7. Cheng S, Sun C, Zhang S, et al. SG-SLAM: A real-time RGB-D visual SLAM toward dynamic scenes with semantic and geometric information[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 72: 1-12. https://doi.org/10.1109/tim.2022.3228006
8. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149. https://doi.org/10.1109/tpami.2016.2577031
9. He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969. https://doi.org/10.1109/iccv.2017.322
10. Zhang H, Chang H, Ma B, et al. Dynamic R-CNN: Towards high quality object detection via dynamic training[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. Springer International Publishing, 2020: 260-275. https://doi.org/10.1007/978-3-030-58555-6_16
11. Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
12. Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790. https://doi.org/10.1109/cvpr42600.2020.01079
13. Tan X, Qi F, Wang N, et al. Glass surface detection method based on visual distortion[J]. Journal of Computer-Aided Design & Computer Graphics, 2023. https://doi.org/10.3724/SP.J.1089.2023-00342.
14. Mei H, Yang X, Wang Y, et al. Don't hit me! glass detection in real-world scenes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 3687-3696. https://doi.org/10.1109/cvpr42600.2020.00374
15. Yu L, Mei H, Dong W, et al. Progressive glass segmentation[J]. IEEE Transactions on Image Processing, 2022, 31: 2920-2933. https://doi.org/10.1109/tip.2022.3162709
16. He H, Li X, Cheng G, et al. Enhanced boundary learning for glass-like object segmentation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 15859-15868. https://doi.org/10.1109/iccv48922.2021.01556
17. Lin J, He Z, Lau R W H. Rich context aggregation with reflection prior for glass surface detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13415-13424. https://doi.org/10.1109/cvpr46437.2021.01321
18. Liu F, Liu Y, Lin J, et al. Multi-view dynamic reflection prior for video glass surface

detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(4): 3594-3602. https://doi.org/10.1609/ aaai.v38i4.28148
19. Mei H, Dong B, Dong W, et al. Glass segmentation using intensity and spectral polarization cues[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 12622-12631. https://doi.org/10.1109/cvpr52688.2022.01229
20. Hao J, Liu M, Yang J, et al. GEM: Boost simple network for glass surface segmentation via vision foundation models[J]. IEEE Transactions on Multimedia, 2025. https://doi.org/10.1109/TMM.2025.3535404.
21. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788. https://doi.org/10.1109/cvpr.2016.91
22. Tian Y, Ye Q, Doermann D. Yolov12: Attention-centric real-time object detectors[J]. arXiv preprint arXiv:2502.12524, 2025.
23. Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271. https://doi.org/10.1109/cvpr.2017.690
24. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
25. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
26. Glenn J, K Nishimura, T Mineeva, et al. yolov5. https://github.com/ultralytics/yolov5/tree, 2, 2020.
27. Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.
28. Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475. https://doi.org/10.1109/cvpr52729.2023.00721
29. Glenn J. Yolov8. https://github.com/ultralytics/ultralytics/tree/main, 2023
30. Wang C Y, Yeh I H, Mark Liao H Y. Yolov9: Learning what you want to learn using programmable gradient information[C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2024: 1-21. https://doi.org/10.1007/978-3-031-72751-1_1
31. Wang, A., Chen, H., Liu, L., Chen, K.: YOLOv10: real-time end-to-end object detection. arXiv:2405.14458,2024.
32. Khanam R, Hussain M. Yolov11: An overview of the key architectural enhancements [J]. arXiv preprint arXiv:2410.17725, 2024.
33. Xie E, Wang W, Wang W, et al. Segmenting transparent objects in the wild[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. Springer International Publishing, 2020: 696-711. https://doi.org/10.24963/ijcai.2021/165