Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083596

ARTICLE

Inference and Performance Analysis of Optimized YOLOv9 for Indoor Object Detection & Classification

Muhammad Asim[1,#], Muhammad Amin Shahid[1,#], Abdullah Khan[1], Muhammad Ishaq[1], Jawad Khan[2,*], Syed Qamrun Nisa[3], Muhammad Amir Khan[4] and Ines Hilali Jaghdam[5]

[1]Institute of Computer Sciences and Information Technology, Faculty of Management and Computer Sciences, The University of Agriculture, Peshawar, Pakistan
[2]School of Computing, Gachon University, Seongnam, Republic of Korea
[3]Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia
[4]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia
[5]Department of Computer Science and Information Technology, Applied College, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

*Corresponding Author: Jawad Khan. Email: jkhanbk1@gachon.ac.kr

[#]These authors contributed equally to this work

Received: 07 April 2026; Accepted: 27 May 2026; Published: 23 July 2026

ABSTRACT: Indoor object detection presents unique challenges such as occlusions, varying lighting conditions, and cluttered environments. While several object detection frameworks, including RetinaNet, Faster R-CNN, SSD, and EfficientDet, have been proposed, they often suffer from high computational cost, reduced inference speed, and limited accuracy in terms of mean Average Precision (mAP), particularly in real-time scenarios. In this study, lightweight YOLO variants, namely YOLOv7, YOLOv8s, YOLOv9s, and a fine-tuned YOLOv9s which considers the optimized training strategy based on albumentations. All the models are evaluated for indoor object detection using the RGB TUT Indoor dataset. The models are assessed using precision, recall, mAP@0.5, and mAP@0.5:0.95. The experimental results demonstrate that the fine-tuned YOLOv9s consistently outperforms the baseline YOLOv9s model across all evaluation metrics, confirming the effectiveness of proposed training optimizations. Specifically, the fine-tuned YOLOv9s achieves a precision of 97.9%, recall of 96.1%, mAP@0.5 of 99.1%, and mAP@0.5:0.95 of 88.7%. These improvements highlight the impact of systematic training refinement beyond standard model configuration. Among the evaluated models, YOLOv8s achieves the highest inference speed of 90 FPS in 11.1 ms, making it suitable for ultra-low-latency applications such as smart homes, assistive systems, and robotics. In contrast, the fine-tuned YOLOv9s provides a superior balance between accuracy and efficiency, making it more suitable for accuracy-sensitive indoor environments where detection reliability is critical. Overall, the study demonstrates that carefully optimized training strategies can significantly enhance the performance of YOLOv9s without architectural modifications, providing practical insights for real-time indoor object detection systems.

KEYWORDS: Image classification; deep learning; indoor object detection; You Only Look Once (YOLO)

1 Introduction

Object detection has significantly evolved from traditional techniques such as Haar cascades [1] and Histogram of Oriented Gradients (HOG) [2] to deep learning-based methods that learn directly from image data. Convolutional Neural Networks (CNNs) enabled the extraction of hierarchical, context-aware features,

greatly enhancing detection accuracy and scalability. Today, object detection is widely applied in surveillance, autonomous systems, healthcare, agriculture, and augmented reality. It enables real-time threat detection, obstacle recognition in self-driving vehicles, diagnostic imaging in healthcare, and early detection of plant diseases in agriculture. Its integration into smart devices underscores its growing practical significance. However, indoor object detection remains challenging due to cluttered scenes, occlusions, inconsistent lighting, and scale variations. These factors often reduce the performance of models trained in Microsoft COCO [3] and PASCAL VOC [4] datasets, which do not fully capture the complexities of indoor environments. In addition, applications such as assistive technologies, service robotics, and smart home systems demand high detection accuracy with real-time performance under constrained computational resources.

This research focuses specifically on the problem of indoor object detection and addresses the limitations of inaccurate detection in indoor scenes by evolving and evaluating the YOLOv9 architecture. Although YOLO models are known for their balance of speed and accuracy, YOLOv9 introduces additional improvements in feature representation, scale awareness, and bounding-box regression, making it a promising candidate for indoor scenarios. To thoroughly assess its performance, this study contrasts YOLOv9 with two other state-of-the-art versions: YOLOv7 and YOLOv8. These comparisons are quantified using established performance indicators, i.e., accuracy, precision, recall, and mean Average Precision (mAP).

The evaluation is carried out using the RGB-based TUT Indoor Dataset, collected and annotated by Tampere University of Technology (TUT), Finland [5]. This dataset includes 2213 RGB images that feature 4599 labeled instances in seven different indoor object classes: fire extinguisher, exit sign, chair, clock, screen, trash bin, and printer. The dataset captures realistic indoor environmental conditions, incorporating occlusions, dynamic lighting variations, and complex scene configurations, making it a robust benchmark for evaluating indoor object detection models.

Previous research, such as that of Adhikari et al. [5], used classical models like Faster R-CNN with ResNet-101, SSD MobileNet, and RetinaNet to improve bounding-box annotations in this dataset. However, these approaches often suffer from slower inference speed and higher computational cost. In contrast, the YOLO family of models, and particularly YOLOv9 [6], provide a real-time alternative with better trade-offs in speed and accuracy.

To our knowledge, this is the first work to conduct a comprehensive model-level comparison of YOLOv7, YOLOv8, and YOLOv9 including a hypertuned version of YOLOv9 on the TUT Indoor dataset. We investigate recent advances in YOLO architectures that effectively address indoor specific challenges, such as occlusion, clutter, and variable illumination, which are critical for real-world deployment.

The primary contributions of this paper are summarized as follows:

- A fine-tuned YOLOv9s model tailored to the given dataset, incorporating systematic training-level optimizations such as refined data augmentation strategies and controlled ablation analysis to enhance indoor object detection performance under challenging conditions.
- A systematic evaluation of YOLOv7, YOLOv8s, and YOLOv9s (baseline and fine-tuned variants) in the TUT Indoor Dataset, which includes 4599 object instances across seven commonly encountered indoor classes.
- We analyze each model performance using a range of quantitative metrics, including accuracy, precision, recall, loss, and mAP.
- We investigate computational efficiency in terms of inference speed and FLOPs that provide insights for real-time intelligent systems for indoor environments.

The remainder of this paper is organized as follows. Section 2 reviews the current literature on object detection with a focus on indoor settings. The Section 3 outlines the dataset, the model architecture, and the

experimental setup. The performance of the used model is evaluated in Section 4 and discusses key findings. Finally, Section 5 presents the contribution and highlights the future research directions.

2 Related Work

Object detection encompasses the identification, localization, and classification of objects within digital image or video frame [7]. Although well-lit outdoor environments facilitate detection and benefit from consistent illumination and clear object boundaries. Indoor scenes present significant challenges due to poor lighting conditions, clutter, and frequent occlusions. Early indoor object detection systems employed hand-crafted features, e.g., SIFT, HOG and conventional machine learning classifiers [8], but these methods struggled with robustness across indoor scenarios.

The advent of deep learning revolutionized object detection, with R-CNN [9] pioneering a two-stage pipeline: (1). region proposals generation via selective search and (2). followed by CNN-based classification. Despite its accuracy, the model was computationally expensive due to repeated CNN forward passes. Fast R-CNN [10] improved efficiency by sharing feature computation across the entire image and introducing Region of Interest (ROI) pooling. Faster R-CNN [11] further optimized the process by replacing the selective search with a Region Proposal Network (RPN), allowing end-to-end training and faster inference.

The single shot multi-box detector (SSD) [12] introduced a single-stage architecture that directly predicts bounding boxes and class scores from multiple feature maps, supporting real-time performance across object scales. RetinaNet [13] addressed the class imbalance problem inherent in single-stage detectors by introducing focal loss, which emphasizes the harder examples. Its use of a Feature Pyramid Network (FPN) also enhanced multiscale object detection.

You Only Look Once (YOLO) models further improved speed and accuracy, making them especially suitable for indoor real-time applications such as assistive technologies and robotics [14]. These models deliver high detection precision with low computational overhead. This shift from traditional shape-based recognition to deep models such as RetinaNet and YOLO, which are increasingly employed to detect and track people with disabilities and their mobility aids. Similarly, Ref. [15] developed an indoor object recognition system for visually impaired users, achieving up to 97% accuracy on benchmark datasets through fine-tuned neural networks.

To mitigate the lack of annotated indoor datasets, Ref. [5] proposed a two-stage annotation method that reduced manual labeling efforts by up to 90%, facilitating efficient model training using datasets such as TUT. YOLO-based frameworks have since gained traction in indoor navigation. For example, Ref. [16] integrated YOLO with depth estimation and spatial audio to assist visually impaired users, while Ref. [17] introduced lightweight YOLO variants optimized for mobile robots, improving detection speed and supporting SLAM-based navigation.

Jiang et al. [18] proposed S-SSD, a lightweight object detection model that combines ShuffleNet with SSD to improve real-time detection performance on indoor mobile robots while maintaining competitive accuracy. Ref. [19] demonstrated the applicability of YOLOv5 to equirectangular panoramas in the interior, supporting smart home systems and digital twin technologies.

Recent advances in indoor object detection have demonstrated significant progress in addressing occlusion and computational constraints. Jia et al. [20] developed a context-aware detection system using Faster R-CNN with ResNet50 backbone, achieving 88.71% accuracy in identifying occluded indoor hazards for the BELUGA robot platform. Building upon this, Zhao et al. [21] proposed LAOF-IPDT, a lightweight anchor-free transformer architecture specifically optimized for indoor personnel detection on edge devices, which achieved real-time inference speeds while maintaining competitive accuracy.

In contrast to these specialized approaches, our study conducts a comprehensive evaluation of YOLO-based models across multiple critical parameters: (1) generalization across diverse indoor object categories e.g., chairs, clocks, screens, (2) robustness to common indoor challenges i.e., occlusion, lighting variations, and (3) computational efficiency for potential deployment in real-time. Table 1 systematically compares these related works with our proposed framework.

Table 1: Summary of object detection techniques and applications.

Reference	Method/Model	Dataset	Performance	Key Remarks
[9]	R-CNN (CNN + region proposals)	ILSVRC2013	–	High accuracy but very slow inference
[10]	Fast R-CNN (ROI pooling)	COCO, ILSVRC	–	Shared computation improves speed
[11]	Faster R-CNN (RPN)	COCO	–	End-to-end training, faster proposals
[12]	SSD (single-shot multi-scale)	COCO	Real-time	Good speed–accuracy trade-off
[13]	RetinaNet (Focal Loss + FPN)	–	High accuracy	Handles class imbalance effectively
[5]	Faster R-CNN (two-stage annotation)	TUT Indoor	–	Reduces manual labeling effort
[15]	RetinaNet (fine-tuned)	IODR	High accuracy	Assistive indoor recognition
[16]	YOLO + depth + audio feedback	Custom (6 classes)	Real-time	Assistive navigation system
[17]	YOLO + SLAM	Custom (6 classes)	–	Mobile robot detection
[18]	S-SSD + ShuffleNet	Custom (9 classes)	Improved accuracy	Lightweight for robotics
[19]	YOLOv5 (panoramic detection)	–	–	Equirectangular indoor scenes
[20]	Faster R-CNN + feature fusion	Custom (7 classes)	88.71%	Context-aware hazard detection
[21]	Transformer-based (LAOF-IPDT)	–	–	Lightweight edge deployment
[22]	SSD + ResNet50 + MCIE + dual NMS	SUN2012 subset	mAP 54.1%	Better occlusion handling; lighting sensitive

(Continued)

Table 1 (continued)

Reference	Method/Model	Dataset	Performance	Key Remarks
[23]	I-DINO (Transformer + BiFPN)	Indoor COCO	mAP 62.3%	Strong multi-scale learning; costly annotation
[24]	RetinaNet (assistive navigation)	Custom (11k images)	mAP 98.75%	High accuracy; domain-specific

3 Method and Materials

Numerous researchers have employed different techniques for the detection and classification of indoor objects. However, many of these approaches demonstrate limited effectiveness in real-time applications due to challenges such as occlusion, varying illumination, and cluttered environments. This study proposes YOLO-based deep learning models, specifically designed for robust and efficient indoor object detection. An overview of the research framework, which illustrates the systematic flow of the proposed approach, is presented in Fig. 1. The proposed methodology is elaborated in the following subsections.

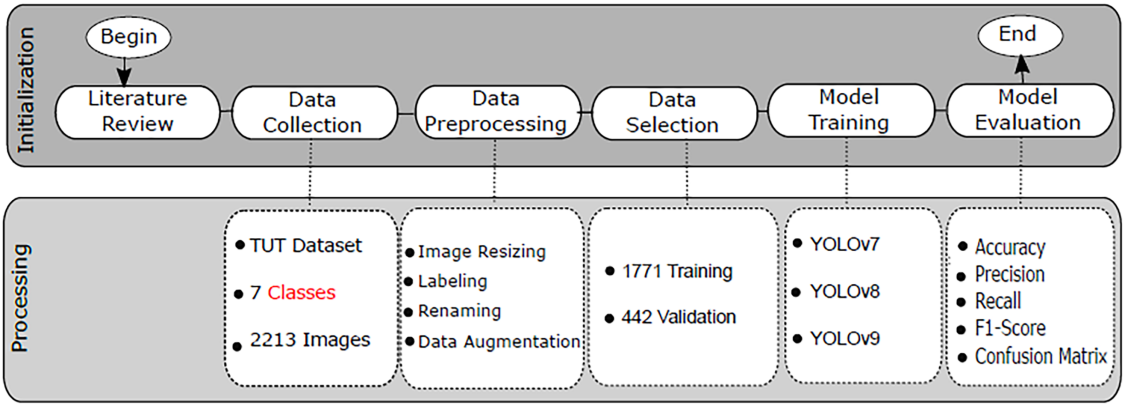


Figure 1: Research methodology for indoor object detection using YOLO models.

3.1 Dataset Collection

This study uses the TUT Indoor Dataset, a publicly available dataset collected by Tampere University of Technology, Tampere, Finland, as described in [5]. This dataset comprises 2213 images with a total of 4599 instances, classified into seven classes: chair, clock, exit sign, fire extinguisher, printer, screen, and trash bin. The distribution of images and instances across each class is presented in Table 2.

3.2 Data Preprocessing

To perform object detection using the YOLO models, we utilized the publicly available TUT Indoor Object Detection Dataset. The YOLO architecture mandates input data to conform to a predefined annotation format, necessitating dataset preprocessing including image resizing, label normalization, and class ID remapping to ensure compatibility.

Table 2: Number of images and instances for each class in the dataset.

Classes	Number of Images	Number of Instances
Clock	277	280
Chair	851	1664
Fire Extinguisher	818	1684
Exit Sign	504	545
Screen	95	116
Printer	81	81
Trashbin	171	229
All	2213	4599

3.2.1 Image Rescaling

The original images in the dataset were extracted from HD video sequences with a resolution of 1280 × 720 pixels. For compatibility and optimal performance with the YOLO models, we resized all images to 640 × 640 pixels. This dimension aligns with the input size used in training recent YOLO models, e.g., those trained on the MS COCO dataset, and it ensures a good balance between accuracy and computational efficiency.

3.2.2 Image Annotation Conversion

YOLO is a supervised object detection algorithm and requires labeled training data to learn and detect objects effectively. The original annotations in the TUT dataset were provided in XML format (PASCAL VOC), which is not directly compatible with YOLO. To convert the annotations to the YOLO format, we used Roboflow [25], a platform that simplifies the preprocessing pipeline for computer vision tasks. Roboflow enables easy conversion from XML to YOLO format and supports exporting annotations in the required.txt files, where each line represents a single object in the image. Each annotation in YOLO format follows the structure:

$$< class_{id} >< x_{center} >< y_{center} >< width >< height >$$

An example annotation can be given as 0 0.1875 0.1667 0.25 0.5833 where 0 represents the class ID, followed by the normalized coordinates of the bounding box. All coordinates are normalized, i.e., relative to the image width and height.

3.3 Training Configurations

The simulation settings are listed in Table 3. All models were trained for 100 epochs based on preliminary experiments showing that the training loss stabilized around epoch 90, with no significant improvement in validation mAP beyond this point. Early stopping was also monitored using validation loss to avoid overfitting. Therefore, 100 epochs were selected as an optimal trade-off between convergence and computational efficiency. Using a fixed image resolution of 640 × 640 on the Kaggle platform equipped with an NVIDIA Tesla P100 GPU. The optimization was performed using stochastic gradient descent (SGD) with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. A OneCycle learning rate scheduling strategy was applied, gradually reducing the learning rate to 1% of its initial value. A warm-up phase of 3 epochs was used to stabilize early training dynamics. Standard YOLO data augmentation

techniques, including mosaic augmentation, mixup, HSV color jittering were consistently applied across all models. To ensure reproducibility, a fixed random seed of 0 was used for all experiments. Close Mosaic is enabled in baseline experiments and selectively disabled in fine-tuning experiments to analyze its effect on convergence stability The dataset was split into training and validation sets using an approximately stratified 80:20 ratio, resulting in 1771 training images and 442 validation images. This split preserves class distribution across subsets and ensures reliable evaluation of unseen data.

Table 3: Training configuration and experimental settings.

Category	Parameter	Value
Epochs	Training epochs	100
Image size	Input resolution	640 × 640
Batch size	Batch size	8
Optimizer	Optimization method	SGD
Learning rate	Initial (lr0)	0.01
Learning rate	Final factor (lrf)	0.01
Momentum	SGD momentum	0.937
Weight decay	Regularization	0.0005
Scheduler	LR policy	OneCycleLR
Warm-up	Warmup epochs	3.0
Random seed	Reproducibility	0
Hardware	GPU	NVIDIA Tesla P100 (Kaggle)
Split ratio	Train/Validation	80/20
Train samples	Dataset split	1771 images
Validation samples	Dataset split	442 images
Augmentation	Mixup	0.15
Augmentation	HSV (H/S/V)	0.015/0.7/0.4
Augmentation	Mosaic	1.0

3.4 YOLO-Based Deep Learning

You Only Look Once (YOLO) is a state-of-the-art, single-stage object detection framework known for its balance between detection and computational efficiency. Unlike two-stage detectors that rely on region proposal generation followed by classification, YOLO formulates object detection as a single regression problem, simultaneously predicting bounding boxes and class probabilities from entire images in one forward pass. This approach enables real-time performance, making YOLO highly suitable for latency sensitive applications. Originally introduced by Redmon et al. in 2015 [26], the architecture has since undergone several significant enhancements that improves both detection precision and inference speed in successive versions. In this article, YOLOv7, YOLOv8, and YOLOv9 are evaluated to exploit their advancements in feature representation, scalability, and real-time object detection capabilities for indoor environments.

3.4.1 YOLOv7

YOLOv7, proposed by Wang et al. in 2023 [27], introduced several architectural and training enhancements aimed at improving both accuracy and computational efficiency. One of the core contributions is the Trainable Bag-of-Freebies (BoF), which comprises a suite of training-time enhancements aimed at increasing detection accuracy while preserving inference efficiency.

YOLOv7 framework also incorporates Extended Efficient Layer Aggregation Networks (E-ELAN) to build a scalable and high-performance backbone, facilitating effective feature extraction with optimized parameter utilization. Furthermore, it introduces compound model scaling, which jointly scales network depth and width to achieve improved accuracy and efficiency across a range of hardware configurations. Another notable component is the Error-Correcting Output (ECO) head, designed to improve detection of overlapping objects while maintaining a lightweight structure. A comprehensive architectural analysis of YOLOv7 and its constituent modules is available in [28]. On the MS COCO benchmark, YOLOv7 achieves a notable 56.8% average precision (AP) while sustaining real-time inference at more than 30 frames per second, establishing it as one of the most accurate and efficient object detectors to date.

3.4.2 YOLOv8

YOLOv8, released by Ultralytics in January 2023 [29], introduces substantial architectural and functional advancements aimed at improving performance in practical object detection applications. The model incorporates an improved backbone and a refined feature pyramid network, resulting in a multiscale feature representation. In particular, it adopts an anchor-free detection head, which simplifies the overall design while preserving high detection accuracy.

To improve localization precision, YOLOv8 integrates the Scaled Intersection over Union (SIoU) loss function, which considers multiple geometric factors, including aspect ratio, distance, and orientation, to improve the bounding box regression. Furthermore, the model employs advanced data augmentation techniques to increase robustness and generalizability across diverse datasets. A detailed analysis of its architecture and component layers is presented in [28]. In addition to object detection, YOLOv8 extends its capabilities to instance segmentation and pose estimation, providing a unified framework for multiple computer vision tasks.

3.4.3 YOLOv9

YOLOv9, introduced by Wang et al. in 2024 [6], addresses the information bottleneck challenge, i.e., where essential feature representations are degraded as they propagate through deep networks. To mitigate, YOLOv9 introduces two key approaches, namely: Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). These mechanisms improve the gradient and feature aggregation, enabling the model to retain critical information while maintaining high performance.

The YOLOv9 architecture is designed for flexibility and efficiency, supporting seamless deployment across a variety of computing environments without compromising speed or accuracy. As shown in Fig. 2, the architecture consists of four principal components: (a) the backbone, (b) the neck, (c) the auxiliary module, and (d) the detection head. Detailed explanations of each component are provided in the subsequent subsections.

Backbone

This component is responsible for extracting and refining features from input images while maintaining computational efficiency. It comprises the following modules:

1. **Silence Block:** Passes the input image directly to the Auxiliary and Backbone branches without modification, ensuring efficient data routing and early feature sharing.
2. **Convolution Blocks:** Use 2D convolutions, Batch Normalization, and SiLU activation to extract low-level features. AutoPad dynamically maintains spatial dimensions during convolution.

3. **RepNCSPELAN Block:** Combines convolutional layers and RepNBottleneck modules to capture multiscale features with reduced computational overhead, drawing on ResNet-inspired design for efficient learning.
4. **ADown Block:** Performs downsampling by splitting features into two parallel paths. Convolution, maxpooling and then concatenating the outputs to preserve both spatial detail and semantic depth.
5. **SPPELAN Block:** Apply Spatial Pyramid Pooling to generate fixed-size feature maps, enabling robust detection of objects at varying scales without resizing the input.

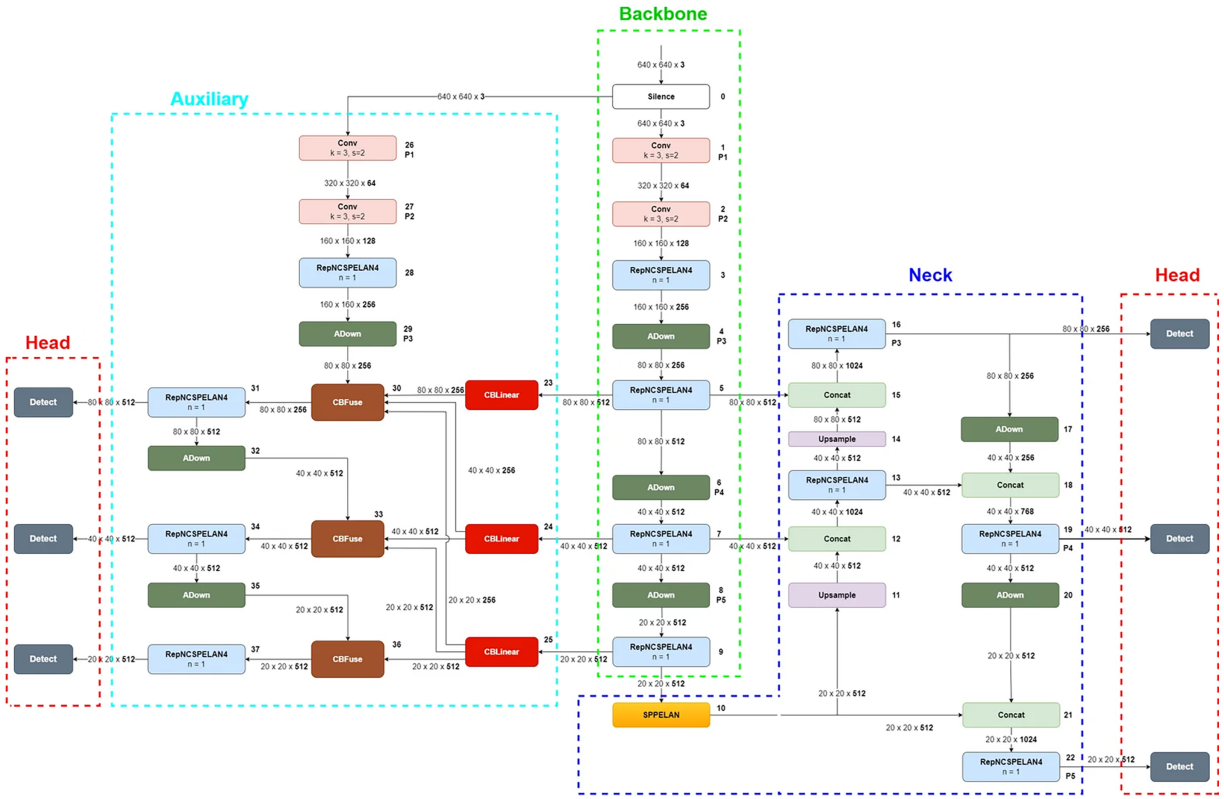


Figure 2: YOLOv9 architecture [30].

Neck

This module integrates multiscale features from the backbone and enhances them for accurate object detection.

1. **Up-sampling:** Increases the spatial resolution of feature maps from the CSPELAN block to match earlier-stage output, allowing effective feature fusion across different scales.
2. **Concatenation:** Combines upsampled features with the corresponding backbone features at the same resolution, preserving fine-grained spatial details essential for precise localization.
3. **RepNCSPELAN:** A multi-branch module composed of parallel RepNCSP blocks, each internally utilizing re-parameterized bottleneck units that combine RepConvN layers with residual shortcuts. The architecture splits the input feature map along the channel dimension and processes them via independent RepNCSP paths for localized and global context extraction, and subsequently fuses them through concatenation followed by pointwise convolution. RepNCSPELAN enhances feature diversity,

gradient propagation, and representational efficiency, significantly improving detection head input quality in GELAN based backbone.

Auxiliary

This module significantly contributes during training by reinforcing deep feature learning and hierarchical representation quality and provides intermediate supervision by generating auxiliary loss signals from early feature maps. Encourages the backbone and neck to learn more discriminative and semantically rich representations. By providing supplementary gradient paths, this branch mitigates vanishing gradients in deep architectures, accelerates convergence, and promotes more effective weight updates. It consists of two additional modules, i.e., CB Linear and CB Fuse blocks discussed in subsections.

1. **CB-Linear:** These blocks produce multiscale pyramid-like feature maps by projecting high-level semantic features from the RepNCSPELAN of backbone. Each CB-Linear block can output multiple feature sets with varying channel dimensions, capturing diverse semantic information across scales.
2. **CB-Fuse:** This block fuses high-level features of the CB-Linear blocks with low-level features of the ADown blocks. By combining semantic richness with fine-grained detail, it improves the network's capacity to learn discriminative features, contributing to better interpretability and detection accuracy, as shown in Fig. 2.

Head

This module is responsible for the final object detection stage, where it outputs bounding box coordinates and class probabilities for indoor objects.

- **Detect Blocks:** Specialized for multi-scale prediction, these blocks process refined features from the neck to detect indoor objects of various sizes such as chair, clock, and screen, etc. Each detect block produces precise bounding boxes and classification outputs, ensuring high accuracy across different spatial resolutions commonly encountered in indoor environments.

This work evaluates two variants of YOLOv9s: (1) a baseline configuration without albumentations and (2) an optimized version incorporating albumentations for enhanced data augmentation. The latter variant, termed fine-tuned YOLOv9 in subsequent sections.

4 Numerical Results

4.1 Evaluation Parameters

In this study, we employ a comprehensive set of performance metrics to evaluate the effectiveness of the proposed methodology. The metrics are selected to quantify both classification performance and operational robustness, as detailed below.

4.1.1 Confusion Matrix

The confusion matrix provides a systematic assessment of the classification performance in all classes. For an n-class detection problem, the matrix $M \in \mathbb{R}^{n \times n}$ quantifies the alignment between predicted and ground-truth labels, where each element M_{ij} represents the count of class-i instances classified as class-j. The diagonal elements (M_{ii}) correspond to correct classifications (true positives for each class), while the off-diagonal elements indicate misclassification patterns.

4.1.2 Accuracy

This metric quantifies global classification performance by measuring the proportion of correct predictions among all instances and is given by

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

4.1.3 Precision

Assess the ability of the model to avoid false alarms, measuring the fraction of correctly predicted positives among all predicted positives. Mathematically,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

In safety-critical applications such as autonomous navigation or medical diagnostics, high precision (low false positive rate) is essential to prevent hazardous misclassifications.

4.1.4 Recall

Also known as the True Positive Rate (TPR), is the ability to correctly identify all relevant instances. A high recall indicates that most of the actual positives were successfully retrieved.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4.1.5 F1-Score

The F1 score provides a balanced evaluation metric by combining precision and recall through their harmonic mean.

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

4.1.6 Mean Average Precision

It is a standard metric for object detection tasks, evaluating both classification accuracy and localization quality, and is given by

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{n} ap_k \tag{5}$$

where ap_k denotes the average precision of class k and n is the number of classes.

4.2 Results & Discussion

4.2.1 Simulation Setup

YOLOv7, YOLOv8, and YOLOv9 models were trained on the Kaggle platform using NVIDIA Tesla P100 GPUs using 16 GB VRAM to perform object detection tasks. The TUT Indoor Dataset served as the training set, and Kaggle's GPU resources enabled efficient model development and rapid experimentation. YOLO repositories were cloned into the Kaggle environment for seamless execution. Model performance was evaluated on 2213 RGB images comprising 4599 annotated instances in 7 object classes, facilitating a comprehensive comparison of the three YOLO architectures.

4.2.2 Performance Evaluation of YOLO Models

1. **Training & Validation Performance of YOLOv7:** Fig. 3 presents the performance of the YOLOv7 model in 100 training epochs. During training, the box loss decreased from approximately 0.06 to 0.02, the objectness loss from 0.014 to 0.004, and the classification loss from 0.025 to 0.002, reflecting progressive improvements in localization, confidence scoring, and classification accuracy.
   The validation metrics showed a similar trend: the box loss dropped from 0.08 to 0.02, the classification loss reduced from 0.025 to 0.005. Although the objectness loss exhibited early fluctuations, it converged to a stable value over time around 0.009, indicating an effective generalization to unseen data.
   Performance metrics also improved substantially. Precision and recall increased from approximately 0.2 and 0.2 to 0.9, respectively. The mean Average Precision (mAP) at IoU threshold 0.5 increased from 0.2 to 0.9, while mAP@0.5:0.95 improved from 0.2 to 0.8, demonstrating strong gains across varying IoU thresholds.
2. **Training & Validation Performance of YOLOv8:** Fig. 4 illustrates the evolution of YOLOv8's training and validation metrics over 100 epochs. During training, box loss decreased from approximately 1.2 to 0.4, classification loss from 1.75 to 0.25, and distribution focal loss (DFL) from 1.2 to 0.8, reflecting steady improvements in detection and localization performance.
   The validation loss showed a consistent downward trend: the box loss decreased from 1.1 to 0.6 classification loss from 1.0 to 0.4, and the DFL from 1.25 to 0.95. These results highlight the strong generalization of the model.
   Performance metrics further reinforce this trend. Precision stabilized around 0.95, recall increased from 0.7 to 0.95, and mAP@0.5 rose from 0.8 to 0.95. mAP@0.5:0.95 improved from 0.55 to 0.85, indicating robust detection performance across multiple IoU thresholds.
3. **Training & Validation Performance of YOLOv9:** Fig. 5 shows the training and validation performance of YOLOv9 in 100 epochs. Training losses decreased consistently, with box loss decreasing from 1.8 to 0.8, classification loss from 5.0 to 0.8, and DFL from 2.0 to 1.2, indicating progressive learning across all components.

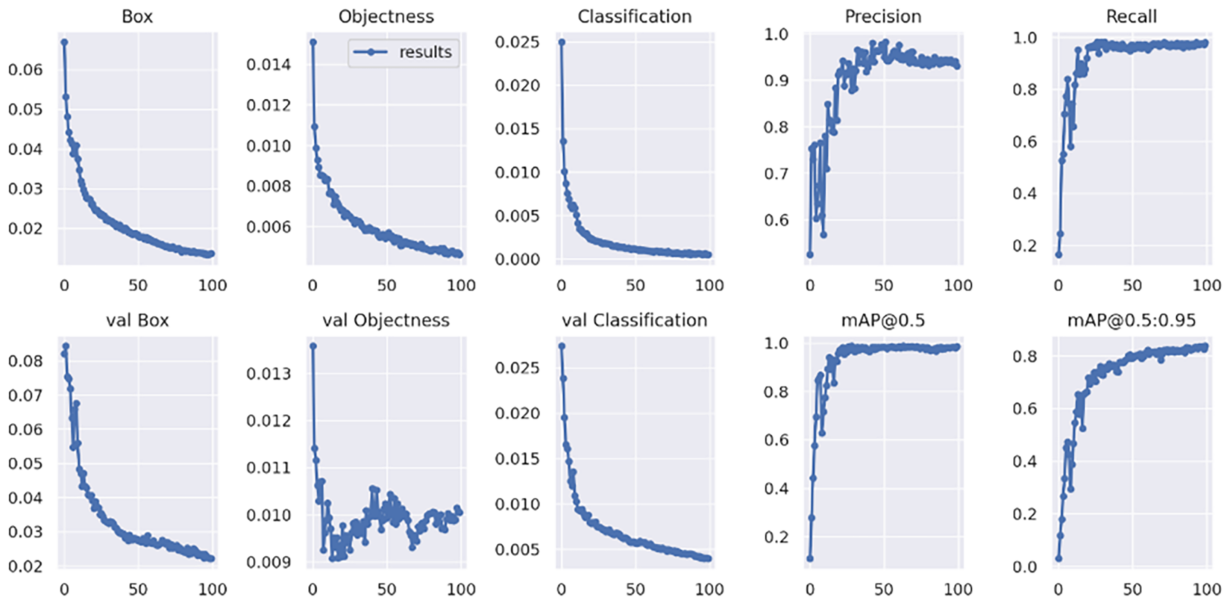


Figure 3: Training and validation performance of YOLOv7.

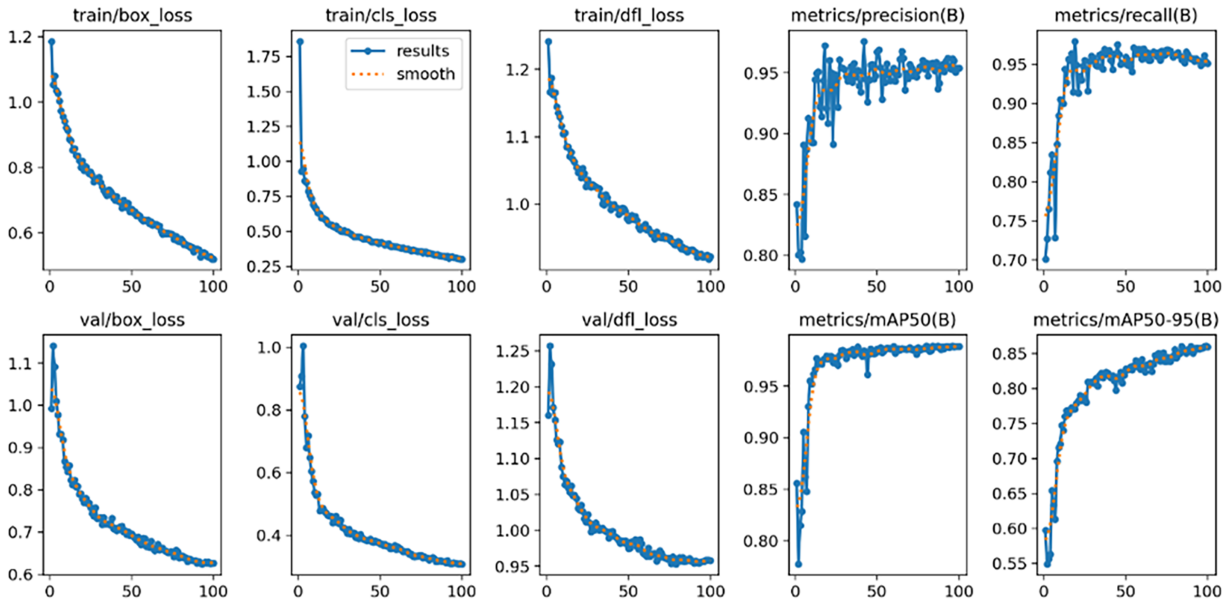


Figure 4: Training and validation performance of YOLOv8.

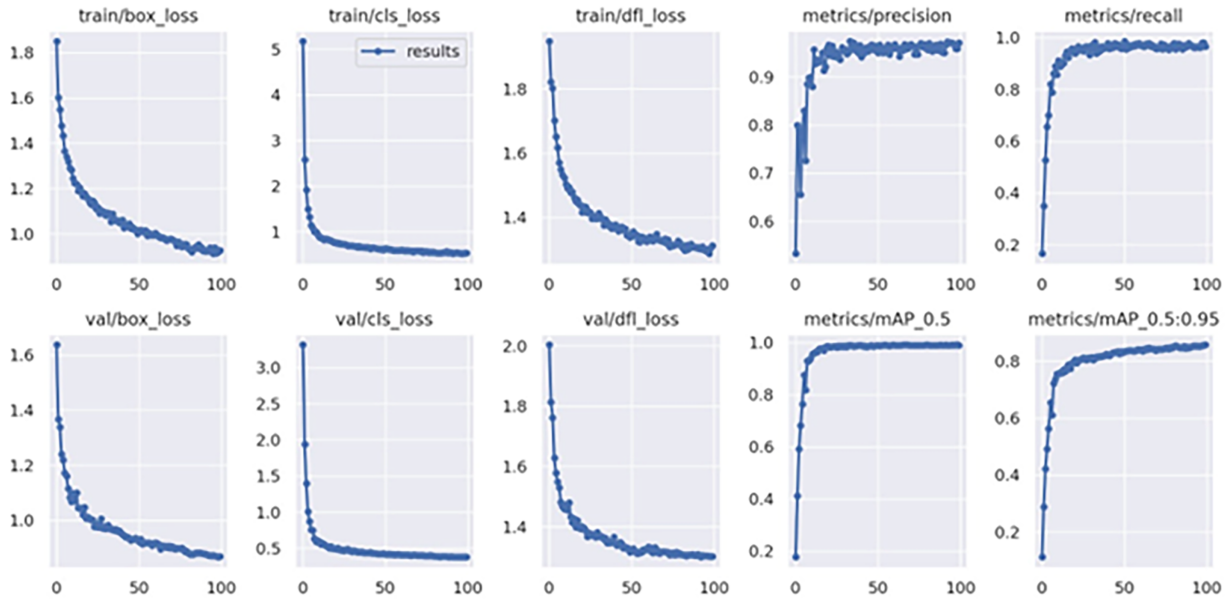


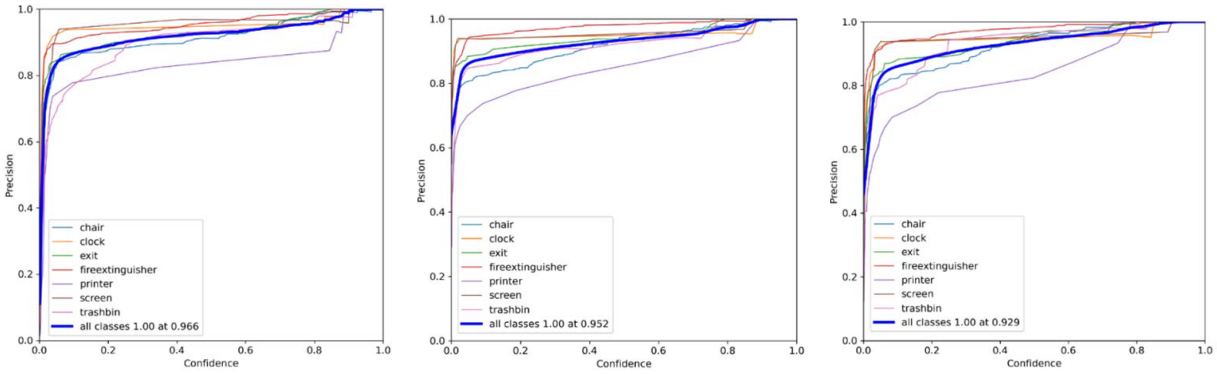
Figure 5: Training and validation performance of YOLOv9.

Interestingly, the validation box loss, classification loss, and DFL loss showed a steady downward trend throughout the training process, with validation box loss decreasing from approximately 1.6 to 0.87, classification loss from 3.3 to 0.4, and DFL loss from 2.0 to 1.3, indicating improved model generalization and stable validation performance.

4.2.3 Precision Performance of YOLO Models

Fig. 6 presents the precision-confidence curves for the YOLO models. Among all object classes, the chair class consistently showed the lowest precision (ranging from 0.925 to 0.983), indicating that it

was the most challenging to detect. Although YOLOv9 generally outperformed the other models, it still struggled with certain classes such as printer and chair. This variation highlights how object characteristics can significantly impact detection performance. However, all YOLO variants demonstrated strong overall precision, underscoring their effectiveness in object detection across diverse categories.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 6: Precision–confidence curve comparison of models: (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.

Table 4 compares the precision performance of YOLOv7, YOLOv8, YOLOv9, and a fine-tuned YOLOv9 model, which incorporates albumentations-based data augmentation and optimized settings. These models were evaluated on a dataset containing 442 images and 1846 labeled objects across seven classes: chair, clock, exit, fire extinguisher, printer, screen, and trash bin. Note that the dataset contains a total of 442 validation images shared across all 7 classes in an object detection setting, and since each image may include multiple objects belonging to different classes simultaneously, it cannot be uniquely assigned to a single class row as shown in Table 4. The precision remained high across all models, with YOLOv7 achieving 93.10%, YOLOv8 at 95.40%, YOLOv9 at 95.20%, and the fine-tuned YOLOv9 leading with 97.90%. The screen class consistently achieved the highest precision (0.968–0.999), reflecting the strong ability of the models to identify screens accurately.

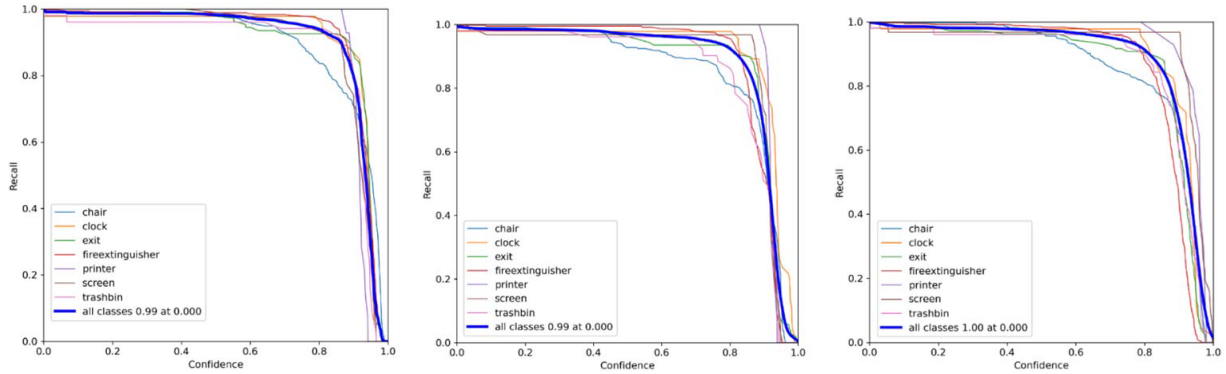
Table 4: Class-wise precision comparison of YOLO models.

Class	Images	Labels	YOLOv7	YOLOv8	YOLOv9	YOLOv9 (Fine-Tuned)
Chair		317	0.925	0.969	0.964	0.983
Clock		46	0.951	0.955	0.952	0.954
Exit Sign		108	0.929	0.953	0.972	0.960
Fire Extinguisher	442	356	0.962	0.991	0.989	0.991
Printer		14	0.845	0.904	0.860	0.985
Screen		31	0.968	0.962	0.957	0.999
Trash Bin		51	0.937	0.940	0.971	0.978
All	442	923	0.931	0.954	0.952	0.979

4.2.4 Recall Performance of YOLO Models

Fig. 7, a higher recall indicates that the model is detecting a higher number of true positive instances. Table 5 summarizes the recall performance of YOLOv7, YOLOv8, YOLOv9 and fine-tuned

YOLOv9 in seven object categories. Among the models, YOLOv7 achieves the highest overall recall of 0.980, demonstrating exceptional detection sensitivity, particularly in classes such as printer achieve a perfect 1.000, fire extinguisher 0.989, and clock 0.978. YOLOv9 and its fine-tuned variant follow closely with overall recall scores of 0.960 and 0.961, respectively. Both versions of YOLOv9 maintain perfect recall in the printer class and perform strongly in other categories such as clock, fire extinguisher, and screen. The fine-tuned YOLOv9 model shows improved consistency, particularly with higher recall score, indicating a more balanced detection performance.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 7: Recall–confidence curve comparison of models: (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.

Table 5: Class-wise comparison of recall.

Class	Images	Labels	YOLOv7	YOLOv8	YOLOv9	YOLOv9 (Fine-Tuned)
Chair	442	317	0.975	0.888	0.933	0.888
Clock		46	0.978	0.978	0.978	0.978
Exit Sign		108	0.973	0.935	0.948	0.954
Fire Extinguisher		356	0.989	0.977	0.981	0.975
Printer		14	1.000	1.000	1.000	1.000
Screen		31	0.984	0.968	0.968	0.968
Trash Bin		51	0.961	0.928	0.961	0.961
All	442	923	0.980	0.953	0.960	0.961

YOLOv8, records the lowest overall recall of 0.953 among the four models. Although it performs well in categories like clock, i.e., 0.978 and printer 1.000, it underperforms in chair 0.888 and trash bin 0.928, which contributes to the lower overall recall. In contrast, YOLOv7 maintains superior recall across nearly all object classes, especially where detection robustness is critical. The comparison reveals that while the YOLOv9 variants approach YOLOv7 in recall performance, YOLOv7 remains the most recall-optimized model across diverse object classes, whereas YOLOv8 trades some recall for higher precision and mAP metrics.

4.2.5 Mean Average Precision

Mean Average Precision (mAP) provides a more comprehensive evaluation by accounting for both false positives and false negatives at varying confidence thresholds. As shown in Table 6, YOLOv9 and its fine-tuned variant achieve the highest mAP@0.5 values of 0.990 and 0.991, respectively, indicating their strong object localization capabilities. YOLOv8 follows closely with a score of 0.988, while YOLOv7 achieves a slightly lower value of 0.985. Although the difference is marginal, these results suggest that the YOLOv9 models offer slightly more accurate detection at higher confidence thresholds.

Table 6: Mean average precision (mAP@0.5) comparison for the YOLO models.

Class	Images	Labels	YOLOv7	YOLOv8	YOLOv9	YOLOv9 (Fine-Tuned)
Chair		317	0.985	0.983	0.990	0.988
Clock		46	0.972	0.980	0.987	0.983
Exit		108	0.990	0.991	0.990	0.991
Fire Extinguisher	442	356	0.994	0.993	0.994	0.994
Printer		14	0.990	0.995	0.995	0.995
Screen		31	0.987	0.993	0.993	0.993
Trash Bin		51	0.978	0.984	0.985	0.989
All	442	923	0.985	0.988	0.990	0.991

When evaluated using the stricter mAP@0.5:0.95 metric as shown in Table 7, which averages precision across a range of IoU thresholds, fine-tuned YOLOv9 again leads with a score of 0.887, outperforming all other models. YOLOv8 achieves the second highest mAP@0.5:0.95 at 0.860, followed by YOLOv9 at 0.860 and YOLOv7 at 0.838. These results indicate that while YOLOv7 is optimized for recall, the fine-tuned YOLOv9 model demonstrates the most balanced performance, combining high precision, strong localization accuracy, and consistent detection across a wide range of IoU thresholds. YOLOv8 also performs competitively, especially in object classes such as chairs and screens, where it achieves some of the highest class-level mAP scores.

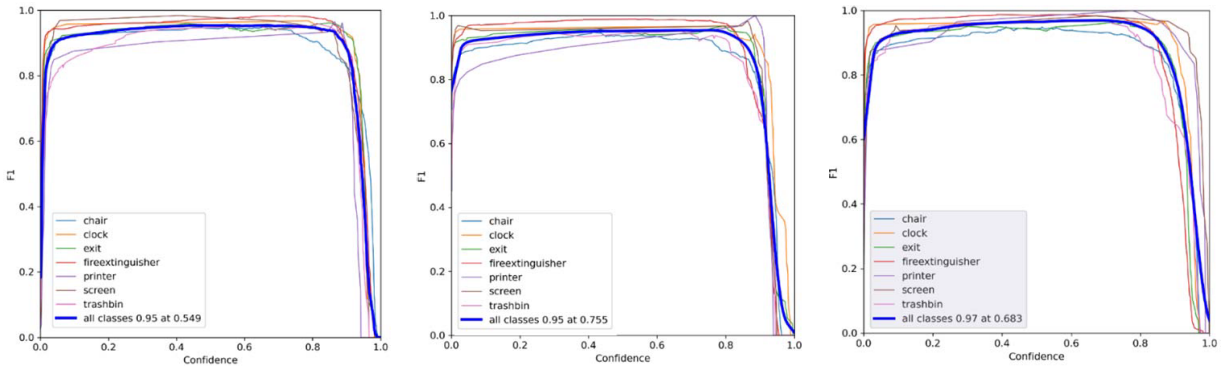
Table 7: Mean average precision (mAP@0.5:0.95) comparison for the YOLO models.

Class	Images	Labels	YOLOv7	YOLOv8	YOLOv9	YOLOv9 (Fine-Tuned)
Chair		317	0.842	0.880	0.873	0.871
Clock		46	0.839	0.866	0.863	0.883
Exit		108	0.836	0.865	0.837	0.880
Fire Extinguisher	442	356	0.842	0.865	0.831	0.887
Printer		14	0.872	0.867	0.888	0.884
Screen		31	0.879	0.879	0.897	0.918
Trash Bin		51	0.757	0.808	0.801	0.881
All	442	923	0.838	0.860	0.860	0.887

4.2.6 F1-Confidence Curve

Fig. 8 illustrates the variation in the F1-score with respect to the confidence threshold in the YOLO models evaluated. The F1-score representing the harmonic mean of precision and recall, serves as a balanced

metric for assessing detection performance. For YOLOv7, the optimal confidence threshold is identified as 0.549, resulting in a maximum F1-score of 0.95, as shown in Fig. 8a. Similarly, YOLOv8 achieves its highest F1-score of 0.95 at a higher threshold of 0.755 (Fig. 8b). In contrast, the fine-tuned YOLOv9 model demonstrates the most favorable performance, achieving a maximum F1-score of 0.97 at an optimal threshold of 0.683, as shown in Fig. 8c.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 8: F1-confidence curve comparison of models: (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.

The superior F1-score of the fine-tuned YOLOv9 indicates an improved balance between precision and recall, reflecting its effectiveness in reducing both false positives and false negatives. Moreover, the model reaches this performance at a moderate confidence level, suggesting greater stability and adaptability under various detection conditions. Compared to earlier versions, the fine-tuned YOLOv9 exhibits notable advancements in detection accuracy and robustness, confirming the benefits of model refinement through targeted fine-tuning.

4.2.7 Precision-Recall Curve

Fig. 9 presents the Precision-Recall (PR) curves for the YOLOv7, YOLOv8, and YOLOv9 models, evaluated at an IoU threshold of 0.5. These curves reflect the trade-off between precision and recall across varying confidence thresholds and are used to compute the mean Average Precision (mAP). The YOLOv9 achieved the highest overall mAP of 99.1%, with class-wise scores of 98.8% for chair, 98.3% for clock, 99.1% for exit sign, 99.4% for fire extinguisher, 99.5% for printer, 99.3% for screen, and 98.9% for trash bin, indicating superior detection accuracy.

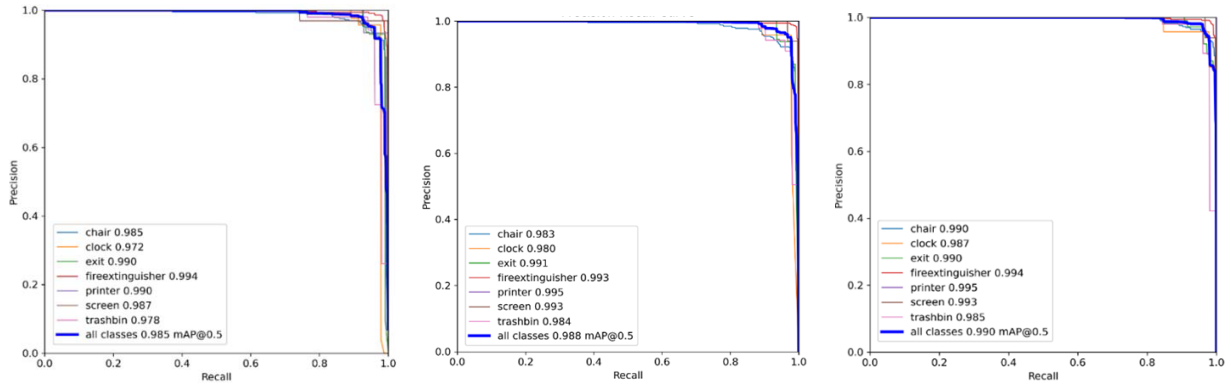
The standard YOLOv9 also performed well, achieving an overall mAP of 99.0% with comparable class-wise performance. YOLOv8 followed with an overall mAP of 98.8%, demonstrating strong consistency, particularly for printer and screen detection. YOLOv7 achieved an mAP of 98.5%, performing especially well in the exit and fire extinguisher categories. Although the fine-tuned YOLOv9 outperforms the others marginally[1], all models exhibit, robust high-precision object detection across categories.

4.2.8 Confusion Matrix

Fig. 10 presents the performance of YOLOv7, YOLOv8 and YOLOv9, which was evaluated at a confidence threshold of 0.25 and an IoU threshold of 0.45, showing high accuracy across object categories. As shown in Fig. 10a YOLOv7 achieved a near-perfect accuracy of 0.99 for most classes, with a flawless

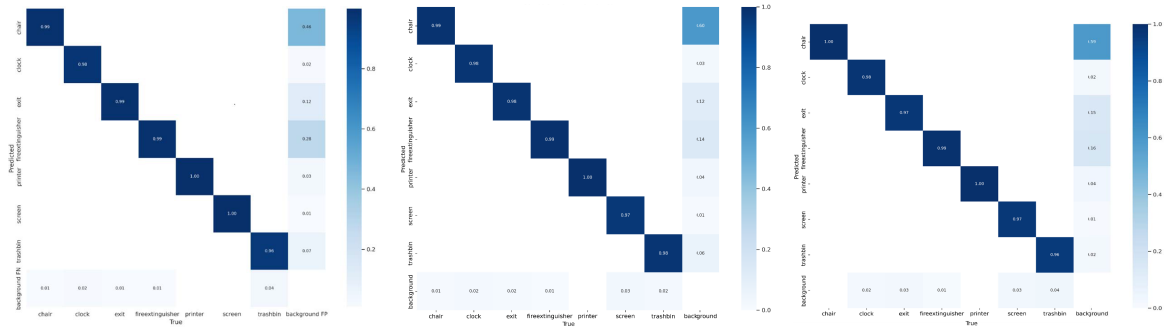
[1]Results not shown here for brevity purpose.

performance of 1.00 for the printer and screen and a slight lower for the trash bin at 0.96. Similarly, Fig. 10b shows the performance of the YOLOv8 model and has a similar accuracy, but the chair error rate increases to 0.60, while the screen and the printer remained nearly flawless at 0.04 and 0.01, respectively. YOLOv9 maintained competitive performance as shown in Fig. 10c, with perfect accuracy for the chair and the printer, i.e, 1.00, and a slightly reduced accuracy of 0.93 for the screen and 0.95 for the trash bin. Overall, all models demonstrate strong classification performance, with YOLOv7 and YOLOv8 outperforming YOLOv9 in accuracy and error rates, although YOLOv9 remained highly reliable.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 9: Precision-recall curve comparison of models: (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

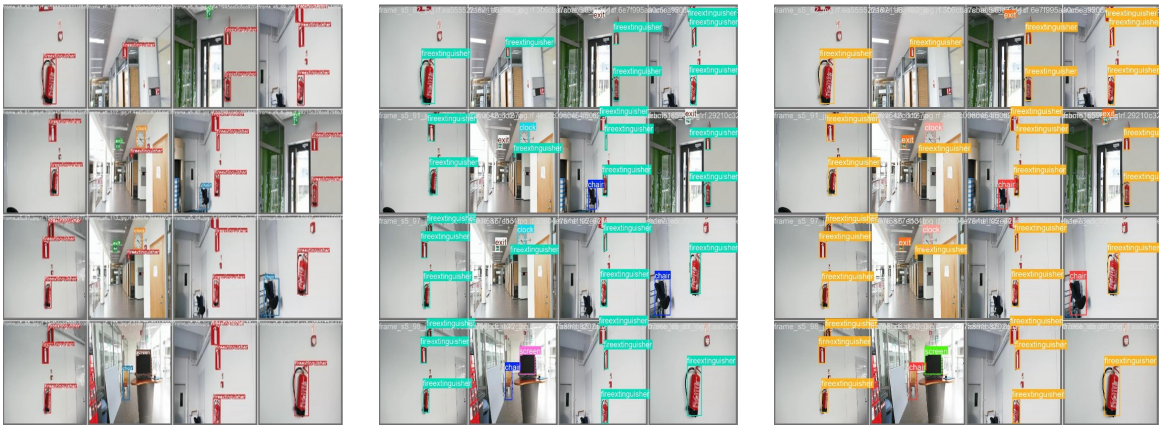
Figure 10: Comparison of confusion matrix for models: (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.

Figs. 11 and 12 illustrate the visual output of the validation and prediction batches, respectively, for the YOLOv7, YOLOv8, and YOLOv9 models applied to the indoor object detection. Batch processing improves computational efficiency and inference speed by enabling the detection of multiple objects across several images simultaneously. The validation batches in Fig. 11 display the original indoor scenes containing various target objects, while the prediction batches in Fig. 12 demonstrate the ability of the models to detect and classify these objects accurately after training. The consistent and precise predictions across all three models confirm the robustness and real-time effectiveness of the YOLO series for indoor object detection tasks.

4.2.9 Detection Time

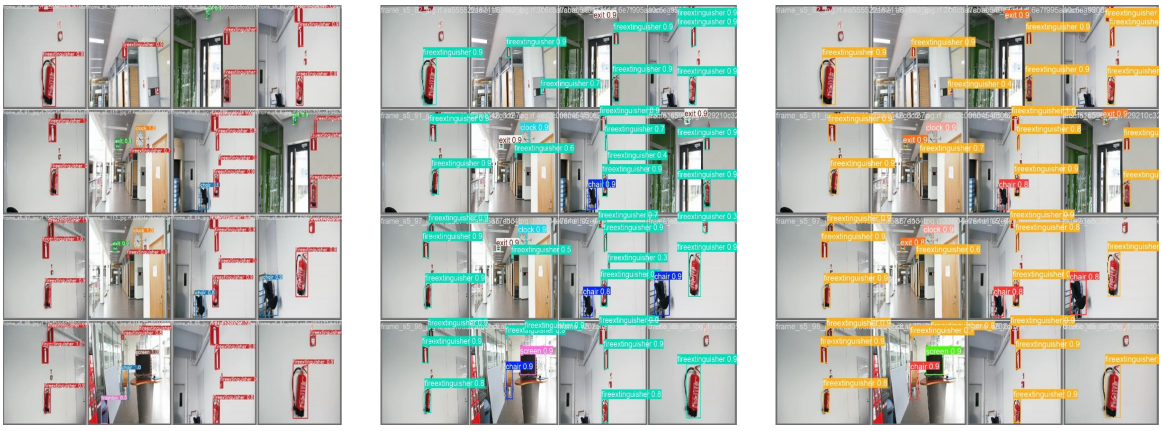
Table 8 presents the inference time and frames-per-second (FPS) performance of the YOLOv7, YOLOv8s, YOLOv9s, and YOLOv9 (fine-tuned) models evaluated on a 217 frame video sequence. Among

the models, YOLOv8s demonstrates the highest processing speed, achieving 90.09 FPS with an inference time of 11.1 milliseconds. YOLOv7 follows with a frame rate of 77.96 FPS at 12.82 milliseconds. The fine-tuned YOLOv9 model processes the video at 40.65 FPS with an inference time of 24.6 milliseconds, while YOLOv9s exhibit the lowest performance, reaching only 35.71 FPS in 28 milliseconds. Note that a higher processing speed does not inherently imply better detection accuracy. In some cases, slower models may yield more accurate results. Therefore, selecting an object detection model requires a balanced consideration of both inference speed and detection accuracy, depending on the application's real-time requirements and performance goals.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 11: Validation batch results for indoor object detection using (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.



(**a**) YOLOv7 (**b**) YOLOv8 (**c**) YOLOv9

Figure 12: Prediction batch results for indoor object detection using (**a**) YOLOv7, (**b**) YOLOv8, and (**c**) YOLOv9.

4.3 Ablation Study & Comparative Analysis

Table 9 presents a systematic ablation study conducted with a fixed input size of 640 × 640 and batch size of 8, to quantify the contribution of individual training components within the YOLOv9 pipeline. Specifically, the analysis isolates the effects of activation functions, optimizer choice, data augmentation strategies, loss functions, and mosaic scheduling under a controlled experimental setting, where only one factor is varied at a time.

Table 8: Detection time of YOLO variants.

Model	Frames	Inference (ms)	FPS
YOLOv5 [31]	x	15	x
YOLOv7 [31]	x	12	x
YOLOv7	217	12.82	77.96
YOLOv8s	217	11.1	90.09
YOLOv9s	217	28	35.71
Fine-Tuned-YOLOv9s	217	24.6	40.65

Table 9: Ablation study of YOLOv9 training configurations.

Experiment	Activation	Opt.	Aug.	Close Mosaic	Loss	Precision	Recall	mAP@0.5	mAP @0.5:0.95
Baseline	SiLU	SGD	Yes	Enabled	CIoU	0.952	0.960	0.990	0.860
ReLU	ReLU	SGD	Yes	Enabled	CIoU	0.948	0.972	0.987	0.830
Adam	SiLU	Adam	Yes	Enabled	CIoU	0.934	0.974	0.985	0.838
No Aug.	SiLU	SGD	No	Enabled	CIoU	0.960	0.961	0.986	0.863
MPDIoU	SiLU	SGD	Yes	Enabled	MPDIoU	0.962	0.974	0.980	0.854
MPDIoU + No Aug.	SiLU	SGD	No	Enabled	MPDIoU	0.961	0.969	0.989	0.863
MPDIoU + No Mosaic	SiLU	SGD	Yes	Disabled	MPDIoU	0.971	0.966	0.990	0.857
Final Model	SiLU	SGD	Yes	Disabled	CIoU	0.979	0.961	0.991	0.887

Replacement of the Sigmoid Linear Unit (SiLU) with the Rectified Linear Unit (ReLU) results in a clear performance degradation, i.e., mAP@0.5:0.95 = 0.86, which was reduced to mAP@0.5:0.95 = 0.83, confirming the benefit of smoother nonlinear activations for stable gradient propagation and feature representation. Similarly, substituting SGD with Adam leads to reduced performance, indicating that SGD offers superior generalization and convergence stability for this detection task.

Interestingly, removing data augmentation yields a marginal improvement, i.e., mAP@0.5:0.95 = 0.86 and improved to mAP@0.5:0.95 = 0.863, suggesting that aggressive augmentation may introduce distributional noise rather than useful variability for this dataset. In addition, MPDIoU loss improves recall in certain configurations but does not consistently enhance mAP@0.5:0.95, revealing a trade-off between localization sensitivity and overall detection precision, as well as an interaction with augmentation strategies.

Finally, modifying mosaic scheduling by disabling *close_mosaic* improves precision and leads to a more stable convergence behavior. Integrating these observations, the final optimized configuration (SiLU + SGD + CIoU + adjusted mosaic scheduling) achieves the best overall performance, i.e., mAP@0.5:0.95 = 0.887. Table 9 presents the summary of the experiments that were conducted to find an optimized fine-tune version of YOLOv9. These results demonstrate that the proposed approach is not defined by a single modification but rather by a systematically optimized combination of training components, where performance gains arise from their interaction rather than architectural changes alone.

Table 10 compares several detection models. Among earlier models, SSD MobileNet led with a mAP@0.5 of 97.73%, closely followed by Faster RCNN with ResNet50 at 96.54%, and Faster RCNN with ResNet101 at 95.06%. RetinaNet with ResNet50 achieved balanced performance with an mAP@0.5 of 96.60%. YOLOv7 attained 98.5% mAP@0.5 and 83.8% mAP0.5-0.95, YOLOv8 improved precision and recall with 98.8% mAP@0.5 and 86% mAP0.5-0.95, and YOLOv9 further advanced results with up to 99.1% mAP@0.5 and 88.7% mAP0.5-0.95 in its optimized version. The optimized YOLOv9 model outperformed all the other models in accuracy and effectiveness.

Table 10: Comparison of the YOLO models.

Model	Precision	Recall	mAP@0.5	mAP@0.5-0.95
SSD MobileNet [5]	x	x	97.73	x
Faster RCNN [5]	x	x	96.54	x
Faster RCNN [5]	x	x	95.06	x
RetinaNet [5]	x	x	96.60	x
YOLOv7	0.931	0.98	0.985	0.838
YOLOv8s	0.954	0.953	0.988	0.86
YOLOv9s	0.952	0.96	0.99	0.86
YOLOv9s (Fine Tune)	0.979	0.961	0.991	0.887

5 Conclusion

Indoor object detection presents critical challenges such as occlusions, varying illumination, and cluttered environments, where real-time performance and high accuracy are essential for applications including assistive navigation, autonomous robotics, and safety monitoring systems. This study provides a systematic evaluation of YOLOv7, YOLOv8s, and YOLOv9s in the TUT Indoor dataset and extends beyond standard benchmarking by introducing fine-tuned YOLOv9s consisting of an optimized training strategy based on albumentations. To our knowledge, this includes one of the first systematic evaluations of YOLOv9 in this dataset under controlled optimization settings. Experimental results demonstrate that fine-tuned YOLOv9s consistently outperform baseline YOLOv9s in all evaluation metrics, achieving a precision of 97.9%, recall of 96.1%, mAP@0.5 of 99.1%, and mAP@0.5:0.95 of 88.7%. These results confirm the effectiveness of systematic optimization in improving detection robustness in complex indoor environments. In terms of computational efficiency, YOLOv8s achieves the highest processing speed 90 FPS, 11.1 ms inference time, making it suitable for ultra-low-latency applications. In contrast, the fine-tuned YOLOv9s achieve a balanced trade-off between accuracy and efficiency with an inference speed of 40.65 FPS, making it more suitable for accuracy-sensitive real-time deployments. The results highlight that carefully designed training strategies can significantly enhance detection performance without architectural modifications and that the proposed fine-tuned YOLOv9s offer a strong balance between accuracy, robustness, and real-time efficiency for indoor object detection systems.

Acknowledgement: The authors gratefully acknowledge the financial and research support provided by Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, under the Princess Nourah bint Abdulrahman University Researchers Supporting Project (Project No. PNURSP2026R845).

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R845), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Muhammad Asim and Muhammad Amin Shahid conceived the study, designed the methodology, and drafted the initial manuscript. Abdullah Khan and Muhammad Ishaq performed the experiments and contributed to data collection and preprocessing. Jawad Khan supervised the research, provided critical revisions, and contributed to the interpretation of results. Syed Qamrun Nisa assisted in data analysis and validation of experimental findings. Muhammad Amir Khan contributed to the implementation of models and technical review of the manuscript. Ines Hilali Jaghdam provided domain expertise, contributed to manuscript editing, and improved the overall structure and presentation. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in the article. The dataset employed is publicly accessible and can also be obtained upon reasonable request from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001; 2001 Dec 8–14; Kauai, HI, USA.
2. Dalal N, Triggs B, Schmid C. Human detection using oriented histograms of flow and appearance. In: Leonardis A, Bischof H, Pinz A, editors. Computer vision—ECCV 2006. Berlin/Heidelberg, Germany: Springer; 2006. p. 428–41.
3. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer vision—ECCV 2014. Cham, Switzerland: Springer International Publishing; 2014. p. 740–55.
4. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vis. 2010;88(2):303–38. doi:10.1007/s11263-009-0275-4.
5. Adhikari B, Peltomaki J, Puura J, Huttunen H. Faster bounding box annotation for object detection in indoor scenes. In: Proceedings of the 2018 7th European Workshop on Visual Information Processing (EUVIP); 2018 Nov 26–28; Tampere, Finland. p. 1–6. doi:10.1109/EUVIP.2018.8611732.
6. Wang CY, Yeh IH, Mark Liao HY. YOLOv9: learning what you want to learn using programmable gradient information. In: Leonardis A, Ricci E, Roth S, Russakovsky O, Sattler T, Varol G, editors. Computer vision—ECCV 2024. Cham, Switzerland: Springer Nature; 2025. p. 1–21.
7. Szeliski R. Computer vision: algorithms and applications. Cham, Switzerland: Springer Nature; 2022.
8. Laidoudi SE, Maidi M, Otmane S. Real-time indoor object detection based on hybrid CNN-transformer approach. In: Proceedings of the 27th European Conference on Artificial Intelligence (ECAI 2024); 2024 Oct 19–24; Santiago de Compostela, Spain. p. 459–66.
9. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014 Jun 23–28; Columbus, OH, USA. p. 580–7.
10. Girshick R. Fast R-CNN. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. p. 1440–8.
11. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49. doi:10.1109/tpami.2016.2577031.
12. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In: Computer Vision—ECCV 2016: 14th European Conference. Amsterdam, The Netherlands: Springer; 2016. p. 21–37.
13. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):318–27. doi:10.1109/tpami.2018.2858826.
14. Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B. A survey of modern deep learning based object detection models. Digit Signal Process. 2022;126(11):103514. doi:10.1016/j.dsp.2022.103514.
15. Afif M, Ayachi R, Said Y, Pissaloux E, Atri M. An evaluation of RetinaNet on indoor object detection for blind and visually impaired persons assistance navigation. Neural Process Lett. 2020;51(3):2265–79. doi:10.1007/s11063-020-10197-9.
16. Davanthapuram S, Yu X, Saniie J. Visually impaired indoor navigation using YOLO based object recognition, monocular depth estimation and binaural sounds. In: Proceedings of the 2021 IEEE International Conference on Electro Information Technology (EIT); 2021 May 14–15; Mt. Pleasant, MI, USA. p. 173–7.

17. Singh KJ, Kapoor DS, Thakur K, Sharma A, Gao XZ. Computer-vision based object detection and recognition for service robot in indoor environment. Comput Mater Contin. 2022;72(1):197–213. doi:10.32604/cmc.2022.022989.
18. Jiang L, Nie W, Zhu J, Gao X, Lei B. Lightweight object detection network model suitable for indoor mobile robots. J Mech Sci Technol. 2022;36(2):907–20. doi:10.1007/s12206-022-0138-2.
19. Pokuciński S, Mrozek D. Object detection with YOLOv5 in indoor equirectangular panoramas. Procedia Comput Sci. 2023;225:2420–8. doi:10.1016/j.procs.2023.10.233.
20. Jia Y, Ramalingam B, Mohan RE, Yang Z, Zeng Z, Veerajagadheswar P. Deep-learning-based context-aware multi-level information fusion systems for indoor mobile robots safe navigation. Sensors. 2023;23(4):2337. doi:10.3390/s23042337.
21. Zhao F, Li Y, Liu H, Zhang J, Zhu Z. Lightweight anchor-free one-level feature indoor personnel detection method based on transformer. Eng Appl Artif Intell. 2024;133(2):108176. doi:10.1016/j.engappai.2024.108176.
22. Ni J, Shen K, Chen Y, Yang SX. An improved SSD-like deep network-based object detection method for indoor scenes. IEEE Trans Instrum Meas. 2023;72(4):1–15. doi:10.1109/tim.2023.3244819.
23. Fan Z, Mei W, Liu W, Chen M, Qiu Z. I-DINO: high-quality object detection for indoor scenes. Electronics. 2024;13(22):4419.
24. Afif M, Ayachi R, Said Y, Pissaloux EE, Atri M. An efficient object detection system for indoor assistance navigation using deep learning techniques. Multimed Tools Appl. 2022;81(12):16601–18. doi:10.1007/s11042-022-12577-w.
25. Ciaglia F, Zuppichini FS, Guerrie P, McQuade M, Solawetz J. Roboflow 100: a rich, multi-domain object detection benchmark. arXiv:2211.13523. 2022. doi:10.48550/arXiv.2211.13523.
26. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 779–88.
27. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 18–22; Vancouver, BC, Canada. p. 7464–75.
28. Thakuria A, Erkinbaev C. Improving the network architecture of YOLOv7 to achieve real-time grading of canola based on kernel health. Smart Agric Technol. 2023;5(12):100300. doi:10.1016/j.atech.2023.100300.
29. Yaseen M. What is YOLOv8: an in-depth exploration of the internal features of the next-generation object detector. arXiv:2408.15857. 2024. doi:10.48550/arXiv.2408.15857.
30. Hidayatullah P, Syakrani N, Sholahuddin MR, Gelar T, Tubagus R. YOLOv8 to YOLO11 performance benchmark and comprehensive architectural comparative review. J RESTI (Rekayasa Sist Dan Teknol Inf). 2026;10(2):341–54. doi:10.29207/resti.v10i2.6598.
31. Deepak GD, Bhat SK. Maximizing YOLOv2 efficiency: a study on multiclass detection of indoor objects. Res Eng. 2025;26:105405.