

Review

# The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection

Momina Liaqat Ali <sup>1</sup>  and Zhou Zhang <sup>2,\*</sup> <sup>1</sup> Department of Computer Science, Middle Tennessee State University, 1301 E Main St., Murfreesboro, TN 37132, USA; ma2mp@mtmail.mtsu.edu<sup>2</sup> Farmingdale State College, State University of New York, 2350 NY-110, Farmingdale, NY 11735, USA

\* Correspondence: zhangz@farmingdale.edu

**Abstract:** This paper provides a comprehensive review of the YOLO (You Only Look Once) framework up to its latest version, YOLO 11. As a state-of-the-art model for object detection, YOLO has revolutionized the field by achieving an optimal balance between speed and accuracy. The review traces the evolution of YOLO variants, highlighting key architectural improvements, performance benchmarks, and applications in domains such as healthcare, autonomous vehicles, and robotics. It also evaluates the framework's strengths and limitations in practical scenarios, addressing challenges like small object detection, environmental variability, and computational constraints. By synthesizing findings from recent research, this work identifies critical gaps in the literature and outlines future directions to enhance YOLO's adaptability, robustness, and integration into emerging technologies. This review provides researchers and practitioners with valuable insights to drive innovation in object detection and related applications.

**Keywords:** YOLO; single stage detection; YOLOv10; YOLOv11; performance evaluation; deep neural network; real-time object detection



**Citation:** Ali, M.L.; Zhang, Z. The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. *Computers* **2024**, *13*, 336. <https://doi.org/10.3390/computers13120336>

Academic Editors: Selene Tomassini and M. Ali Akber Dewan

Received: 21 October 2024

Revised: 8 December 2024

Accepted: 11 December 2024

Published: 14 December 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Object detection, a core task in computer vision, has seen remarkable advancements in recent years due to the ongoing development of more efficient and accurate algorithms [1,2]. One of the most significant breakthroughs in this field is the You Only Look Once (YOLO) framework, a pioneering one-stage object detection algorithm that has drawn widespread attention for its ability to achieve real-time detection with high precision [3,4]. YOLO's approach simultaneously predicting bounding boxes and class probabilities in a single forward pass sets it apart from traditional multi-stage detection methods [5–7]. This capability makes YOLO exceptionally well-suited for applications requiring rapid decision-making, such as autonomous driving, medical diagnostics, and surveillance systems [8,9]. The evolution of object detection methods has paved the way for YOLO, offering a novel solution to the longstanding challenge of balancing speed and accuracy in detection tasks [10,11]. Its real-time performance, coupled with its flexibility across a wide range of domains, has cemented YOLO as a leading algorithm in both academic research and practical applications. As object detection continues to evolve, a deeper understanding of YOLO's architecture and its extensive applicability becomes increasingly important, particularly as newer versions introduce significant architectural improvements and optimizations [12].

This review provides a comprehensive exploration of the YOLO framework, beginning with an overview of the historical development of object detection algorithms, leading to the emergence of YOLO [13]. The subsequent sections delve into the technical details of YOLO's architecture, focusing on its core components, the backbone, neck, and head, and how these elements work in unison to optimize the detection process. By examining these components, we highlight the key innovations that enable YOLO to outperform

many of its counterparts in real-time detection scenarios. A central theme of this review is the versatility of YOLO across diverse application domains. From detecting COVID-19 in X-ray images to enhancing road safety under adverse weather conditions [14], YOLO has demonstrated its ability to address complex challenges across fields such as medical imaging, autonomous vehicles, and agriculture. In each of these domains, YOLO's ability to process high-resolution images quickly and accurately has driven substantial improvements in detection efficiency and accuracy.

Throughout this review, we address several key research questions, including the major applications of YOLO, its performance compared to other object detection algorithms, and the specific advantages and limitations of its various versions. We also consider the ethical implications of using YOLO in sensitive applications, particularly regarding privacy concerns, dataset biases, and broader societal impacts. As YOLO continues to be deployed in applications such as surveillance and law enforcement, these ethical considerations become increasingly critical to responsible AI development. Therefore, this review synthesizes insights from various domains to provide a holistic understanding of the YOLO framework's contributions to the field of object detection. By highlighting both the strengths and limitations of YOLO, this paper offers a foundation for future research directions, particularly in optimizing YOLO for emerging challenges in the ever-evolving landscape of computer vision.

This paper is structured to provide a holistic understanding of the YOLO framework, beginning with an overview of its fundamental architecture and evolutionary advancements. It then delves into a benchmark-based evaluation of various YOLO versions, offering critical insights into their performance across diverse datasets and scenarios. The discussion extends to YOLO's extensive applications across multiple domains, such as healthcare, autonomous systems, agriculture, and industrial automation, highlighting its transformative impact in real-world settings. Finally, the paper addresses ethical considerations associated with YOLO's deployment, including privacy concerns, societal implications, and the need for responsible use. This comprehensive approach aims to provide readers with a well-rounded perspective on the technical capabilities, practical applications, and ethical dimensions of YOLO, serving as a valuable resource for researchers, practitioners, and policymakers alike.

## 2. Literature Search Strategy

Conducting a comprehensive literature review of the YOLO framework requires a systematic and methodologically rigorous approach. This study employed a well-structured strategy to navigate the extensive and multidisciplinary body of literature on YOLO, ensuring a thorough and representative selection of the most relevant and impactful studies.

### 2.1. Search Methodology

To ensure an exhaustive review, we focused on reputable and high-impact sources such as IEEE Xplore, SpringerLink, and key conference proceedings, including CVPR, ICCV, and ECCV. In addition, we utilized search engines like Google Scholar and academic databases, leveraging Boolean search operators to construct detailed queries with phrases such as "object detection", "YOLO", "deep learning", and "neural networks". This approach allowed us to capture the latest research and most significant papers across disciplines related to computer vision and machine learning.

The search covered an array of top-tier publications, including but not limited to the following:

- *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI);*
- *Computer Vision and Image Understanding (CVIU);*
- *Journal of Machine Learning Research (JMLR);*
- *International Journal of Computer Vision (IJCV);*
- *Journal of Artificial Intelligence Research (JAIR).*

The search results yielded an initial pool of 53,200 articles. To manage this large collection, a two-step screening process was applied:

1. Title Screening: The titles were reviewed to eliminate papers not directly related to YOLO or object detection methodologies.
2. Abstract Screening: Abstracts were thoroughly examined to assess the relevance of each article in terms of its focus on YOLO's architectural innovations, applications, or comparative analysis.

## 2.2. Selection Criteria

We applied stringent inclusion and exclusion criteria to refine the literature pool:

- Inclusion Criteria:
  - Studies providing in-depth analysis of YOLO architecture and methodologies.
  - High-impact and widely cited papers.
  - Research papers offering empirical results from YOLO-based applications in various domains.
  - Articles that address both the strengths and limitations of YOLO.
- Exclusion Criteria:
  - Articles that merely mention YOLO without exploring its methodologies or applications.
  - Research papers lacking substantive contributions to the development or application.
  - Duplicate or redundant publications across multiple conferences or journals.

This process refined the literature pool to 135 articles, which were selected for a full-text review. The chosen articles span a wide range of topics, including YOLO's architectural developments, training and optimization strategies, and its application across diverse domains such as medical imaging, autonomous vehicles, and agriculture.

## 2.3. Coding and Classification

The selected articles were further categorized based on specific features of the YOLO framework. Each article was coded according to the following dimensions:

- *Architectural Innovations*: Backbone, neck, and head components, and innovations across YOLO versions (e.g., YOLOv3, YOLOv4, YOLOv5, YOLOv9).
- *Training Strategies*: Data augmentation, transfer learning, and optimization techniques.
- *Performance Metrics*: Evaluation metrics such as mAP (mean Average Precision), FPS (Frames Per Second), and computational cost (FLOPs).
- *Applications*: Medical imaging, autonomous driving, agriculture, industrial applications, and more.

Table 1 provides a detailed breakdown of the different versions of YOLO, their architectural innovations, and methodological approaches. For each version, we analyzed training strategies, loss functions, post-processing techniques, and optimization methods. This systematic classification allows for a nuanced understanding of YOLO's progression and its practical applications.

**Table 1.** YOLO framework version and research methodologies.

YOLO Version	Architectural Innovations	Training Strategies	Optimization Techniques
YOLOv1	Simplified CNN backbone, basic bounding box prediction	Geometric transformations, hue jitter	Stochastic gradient descent, non-maximum suppression (NMS)
YOLOv2	DarkNet-19 backbone, K-means clustering for anchor box refinement	Fine-tuning, pre-trained weights	SGD with momentum, hyperparameter tuning, Adam optimizer

**Table 1.** Cont.

YOLO Version	Architectural Innovations	Training Strategies	Optimization Techniques
YOLOv3	DarkNet-53, multi-scale detection, residual connections	Mix-up, data augmentation, noise	Non-maximum suppression, thresholding, multi-scale object detection
YOLOv4	CSPDarkNet-53, PANet, mosaic data augmentation	Transfer learning, knowledge distillation	Generalized IoU, focal loss, dynamic quantization
YOLOv5	CSPNet, dynamic anchor refinement, lightweight architecture	Mosaic, CutMix, early stopping	Post-training quantization, filter pruning, low-rank approximation
YOLOv6	PANet, CSPDarkNet53	Adversarial training, domain-specific augmentation	IoU loss, confidence thresholding, multi-scale fusion
YOLOv7	EfficientRep backbone, dynamic label assignment	Fine-tuning, adversarial patch detection	NAS, quantization, gradient clipping
YOLOv8	Path Aggregation Network, Dynamic Kernel Attention	Adversarial training, data augmentation	Momentum and Adam optimizer, post-training quantization
YOLOv9	Multi-level auxiliary feature extraction	Fine-tuning, domain-specific augmentation	GELAN module, deep supervision for resource-constrained systems
YOLOv10	Lightweight classification head and separate spatial and channel transformations	NMS free training, dual label assignment	Distinct spatial and channel transformations to increase overall efficiency during down sampling stage
YOLOv11	Introduced C3k2 block in backbone and used C2PSA to enhance spatial attention	Fine-tuning, mix-up, augmentation, adaptive gradient clipping	Quantization, stochastic gradient descent (SGD)

### 3. Single-Stage Object Detectors

#### 3.1. Understanding Single-Stage Detectors in Object Detection: Concepts, Architecture, and Applications

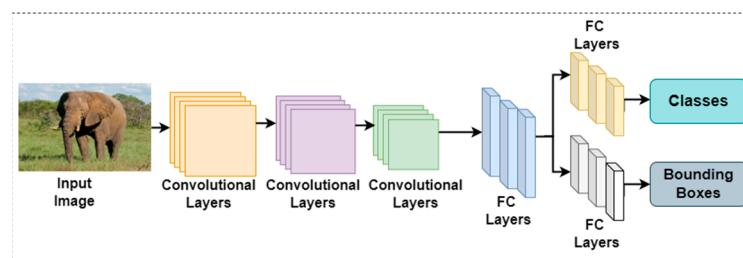
Single-stage object detectors represent a class of models designed to detect objects in an image through a single forward pass of the neural network [15]. Unlike two-stage detectors, which involve separate steps for region proposal and object classification, single-stage detectors perform both tasks simultaneously, streamlining the detection process [16]. This approach has gained popularity due to its simplicity, computational efficiency, and real-time processing capabilities, making it particularly well suited for applications that demand quick inference, such as autonomous vehicles and surveillance systems [17].

A typical single-stage detector directly predicts object class probabilities and bounding boxes without the need for a region proposal network (RPN) [18]. Several key concepts define single-stage object detectors:

1. Unified Architecture: These detectors employ a unified neural network architecture that predicts bounding boxes and class probabilities simultaneously, eliminating the need for a separate region proposal phase [19].
2. Anchor Boxes or Default Boxes: To accommodate varying object scales and aspect ratios, single-stage detectors use anchor boxes (also referred to as default boxes) [20]. These predefined boxes allow the network to make adjustments to better fit objects of different shapes and sizes.
3. Regression and Classification Head: Single-stage detectors consist of two main components: a regression head for predicting bounding box coordinates and a classification head for determining object classes. Both heads operate on the feature maps extracted from the input image [15].

4. Loss Function: The model's training objective involves minimizing a combination of three losses: localization loss (for accurate bounding box predictions), confidence loss (for object presence or absence), and classification loss (for class label accuracy) [21].
5. Non-Maximum Suppression (NMS): After predicting multiple bounding boxes, non-maximum suppression is applied to filter out low-confidence and overlapping predictions. This ensures that only the most confident and non-redundant bounding boxes are retained [22,23].
6. Efficiency and Real-time Processing: One of the primary advantages of single-stage detectors is their computational efficiency, making them suitable for real-time processing. The absence of a separate region proposal step reduces computational overhead, allowing for rapid detection [24].
7. Applications: Single-stage detectors find applications in various domains, including autonomous vehicles, surveillance, robotics, and object recognition in images and videos. Their speed and simplicity make them particularly well-suited for scenarios where real-time detection is crucial. These applications are supported by numerous studies that highlight the efficiency and effectiveness of single-stage detectors in dynamic environments, especially in autonomous driving and pedestrian detection scenarios [25–28]. The effectiveness of single-stage detectors in surveillance systems enables continuous monitoring and quick response, which is essential for security applications [29]. In robotics, these detectors assist in real-time object recognition, facilitating navigation and interaction with the environment [30]. Therefore, the versatility and performance of single-stage detectors have made them a critical component in various modern technology applications.

These characteristics make single-stage detectors a critical tool in object detection, providing a balance between speed and accuracy while supporting a wide range of real-world applications. The flowchart of a single-stage detector's operation is depicted in Figure 1 in which there are convolutional layers of different scales followed by FC layers.



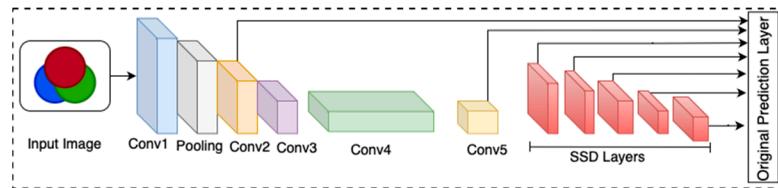
**Figure 1.** Overview of single-stage detectors.

### 3.2. Typical Single-Stage Object Detectors

Several single-stage detectors have been developed over the years, each with unique innovations and optimizations. Below is an overview of key single-stage detectors, their architectures, and contributions to the field of object detection.

#### 3.2.1. SSD (Single Shot Detectors)

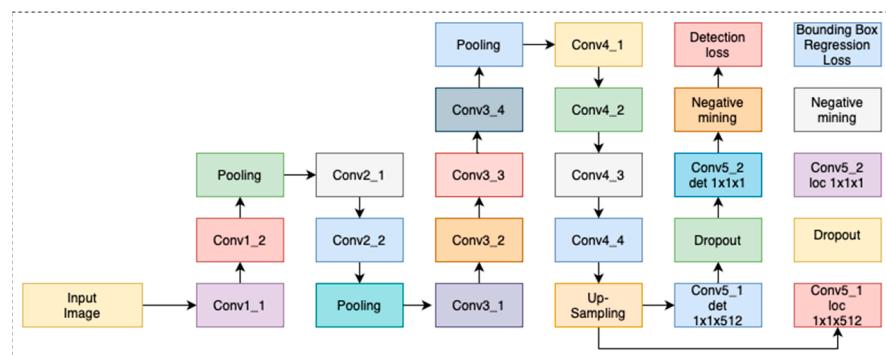
Introduced by Liu et al. in 2016 [31], the Single Shot MultiBox Detector (SSD) leverages a Convolutional Neural Network (CNN) as its backbone for feature extraction. SSD utilizes multiple layers from the base network to generate multi-scale feature maps, allowing it to detect objects at various scales. For each feature map scale, the SSD assigns anchor boxes with different aspect ratios and sizes, simultaneously predicting object class scores and bounding box offsets. After predictions are made, non-maximum suppression is applied to remove duplicate bounding boxes and retain the most confident detections. The overall architecture of an SSD is illustrated in Figure 2. The architecture of an SSD is a combination of convolutional layers that form the backbone and then the SSD head, which performs detection, and the head is composed of a few convolutional layers.



**Figure 2.** Flow diagram of SSD architecture.

### 3.2.2. DenseBox

DenseBox [32], developed by Huang et al., integrates object localization and classification within a single framework. Unlike other models that rely on sparse anchor boxes, DenseBox densely predicts bounding boxes across the entire image, improving localization accuracy. The model employs a deep CNN for feature extraction, followed by non-maximum suppression to refine object detections. DenseBox's dense prediction mechanism allows for enhanced performance in detecting small and closely packed objects. Figure 3 shows the detailed architecture of DenseBox. The whole architecture consists of 16 convolutional layers, the initial 12 layers of which are used VGG19 for initialization.



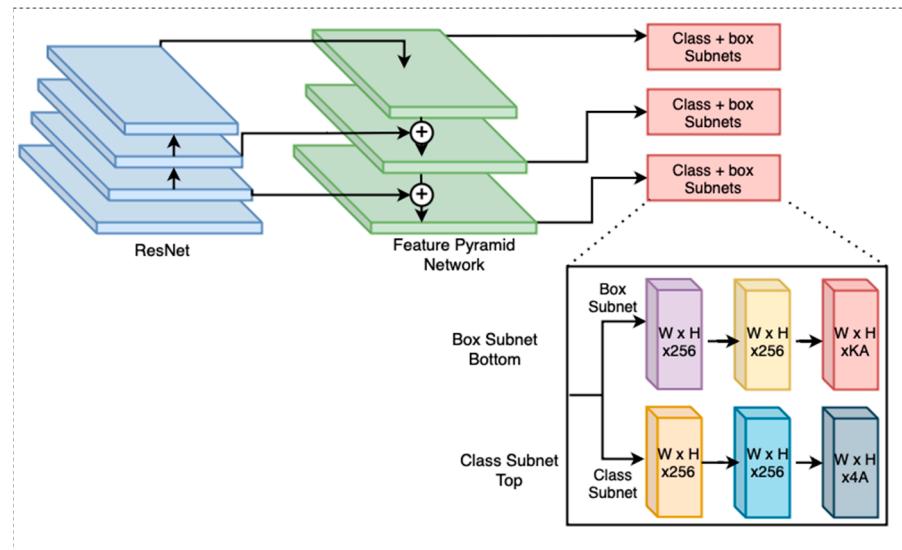
**Figure 3.** Detailed architecture of DenseBox.

### 3.2.3. RetinaNet

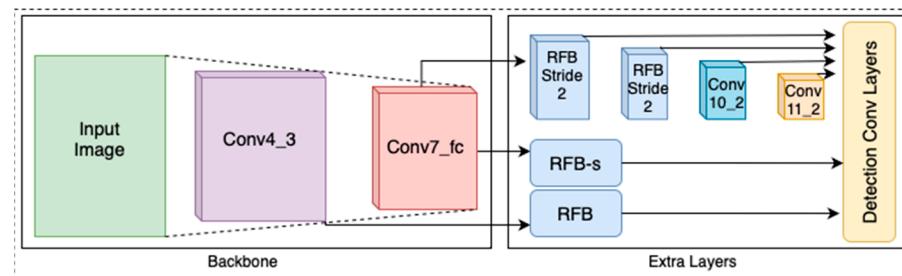
RetinaNet [33], introduced by Lin et al., addresses the issue of class imbalance commonly found in object detection tasks. The architecture incorporates a Feature Pyramid Network (FPN) for multi-scale feature extraction and employs a novel focal loss function to give higher priority to harder-to-detect objects during training. This loss function helps mitigate the imbalance between foreground and background classes, making RetinaNet particularly effective in scenarios with sparse object occurrences. The workflow of RetinaNet is shown in Figure 4. From the figure below, it can be seen that RetinaNet uses both top-down and bottom-up approaches to extract features at different scales and this technique helps in obtaining enriched feature space.

### 3.2.4. RFB Net

The RFB Net (RefineNet with Anchor Boxes) [34], developed by Niu and Zhang, builds upon the basic RefineNet architecture. RFB Net introduces anchor boxes of varying sizes and aspect ratios to improve detection performance, particularly for small objects. This model applies a series of refinement stages that iteratively enhance both localization accuracy and classification confidence. The architecture of RFB Net is shown in Figure 5. A significant difference between the RFB net and SSD is that RFB net uses VGG as a backbone; otherwise, it is architecturally close to SSD.



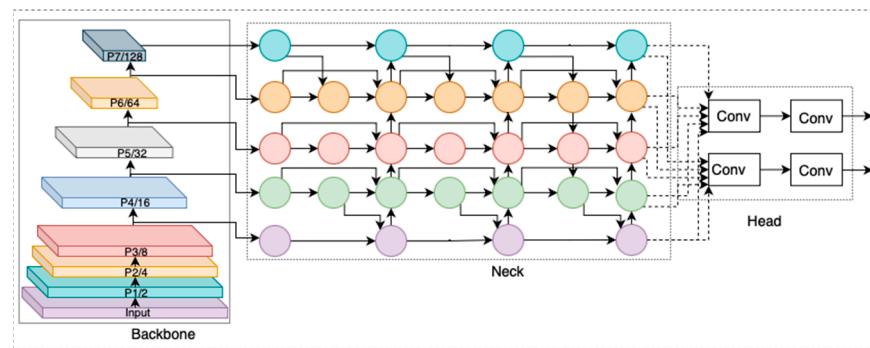
**Figure 4.** Detailed workflow of RetinaNet.



**Figure 5.** The workflow diagram of RFB Net.

### 3.2.5. Efficient Det

EfficientDet [35], proposed by Tan and Le, introduces a compound scaling approach to simultaneously increase network depth, width, and resolution while maintaining computational efficiency. EfficientDet uses a BiFPN (Bidirectional Feature Pyramid Network) for efficient multi-scale feature fusion, balancing model size and performance. This design achieves state-of-the-art object detection results with reduced computational complexity, making it ideal for resource-constrained applications. The architecture of EfficientDet is shown in Figure 6. EfficientDet uses EfficientNet as its backbone architecture and uses repeated layers of Bidirectional Feature Pyramid Network (BiFPN) in the neck part.



**Figure 6.** The architectural overview of EfficientDet.

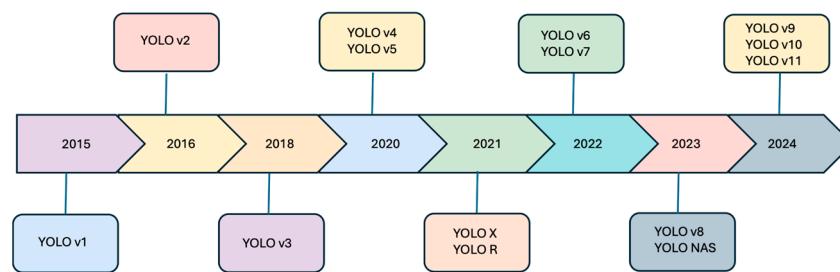
### 3.2.6. YOLO

The YOLO framework, pioneered by Joseph Redmon, revolutionized real-time object detection by introducing a grid-based approach to predict bounding boxes and class

probabilities simultaneously [36]. This innovative design facilitates a highly efficient detection process, making YOLO particularly suitable for applications needing real-time performance.

Since its inception, the YOLO architecture has undergone continuous evolution, with each version from YOLOv1 through to the latest YOLOv9 introducing enhancements in accuracy, speed, and efficiency [37]. Significant advancements include the introduction of anchor boxes, which improve the model's ability to detect objects of varying shapes and sizes, as well as new loss functions aimed at optimizing performance. Additionally, each YOLO iteration features optimized backbones that contribute to faster processing times and better detection capabilities.

The evolution of the YOLO framework from 2015 to 2023 showcases the iterative enhancements made in response to emerging challenges in object detection [38]. Each revision has made strides in minimizing latency while maximizing detection accuracy central goals in the continuing development of real-time object detection technologies. Figure 7. illustrates the timeline of YOLO's evolution from 2015 to 2023.



**Figure 7.** Timeline of YOLO models starting from 2015 to 2023.

To quickly and accurately identify objects, YOLO divides an image into a grid and predicts both bounding boxes and class probabilities simultaneously. Bounding box coordinates and class probabilities are generated by convolutional layers, following feature extraction by a deep Convolutional Neural Network (CNN). YOLO improves the detection of objects at varying sizes using anchor boxes at multiple scales. The final detections are refined using Non-Maximum Suppression (NMS), which filters out redundant and low-confidence predictions, making YOLO a highly efficient and reliable method for object detection.

Each YOLO variant introduces distinct innovations aimed at optimizing both speed and accuracy. For example, YOLOv4 and YOLOv5 integrated advanced backbones and loss functions such as Complete Intersection over Union (CIoU) to improve object localization.

The basic loss function for the YOLO model introduced by Redmon et al. [39] is presented in equation below:

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

where  $1_i^{obj}$  is used to check if there is an object in cell  $i$  while  $1_{ij}^{obj}$  is used to tell that the  $j$ th bounding box in cell  $i$  is responsible for making that prediction. In this basic YOLO loss

function, we can see that only the classification error and bounding box coordinate error are penalized. Table 2 below provides an overview of the loss functions used across YOLO variants, highlighting their contributions to classification and bounding box regression.

**Table 2.** Overview of loss function in YOLO variants.

Model	Bounding Box Regression	Classification
YOLOv1	Mean Squared Error (MSE)	Binary Cross Entropy (BCE)
YOLOv2	Sum Squared Error	BCE
YOLOv3	GIoU/DIoU	Cross Entropy (CE)
YOLOv4	CIoU	BCE/Focal Loss
YOLOv5	CIoU	Focal Loss
YOLOv6	CIoU/DFL	VariFocal Loss
YOLOv7	CIoU	BCE
YOLOv8	CIoU/DFL	CE
YOLOv9	L1 Loss	BCE
YOLOv10	Coordinate loss and confidence loss	Cross Entropy (CE)
YOLOv11	IoU based loss	BCE/Focal Loss + CloU

Note: Mean Squared Error (MSE), Generalized Intersection over Union (GIoU), Distance based Intersection over Union (DIoU), Binary Cross Entropy (BCE), Cross Entropy (CE), Complete Intersection over Union (CIoU), Distributed Focal Loss (DFL).

### 3.3. The YOLO Architecture: Backbone, Neck, and Head

The three primary components of the YOLO architecture backbone, neck, and head undergo significant modifications across its variants to enhance performance:

- *Backbone*: Responsible for extracting features from input data [40], the backbone is typically a CNN pre-trained on large datasets such as ImageNet. Common backbones in YOLO variants include ResNet50, ResNet101, and CSPDarkNet53.
- *Neck*: The neck further processes and refines the feature maps generated by the backbone. It often employs techniques like Feature Pyramid Networks (FPN) and Spatial Attention Modules (SAM) [41] to improve feature representation.
- *Head*: The head processes fused features from the neck to predict bounding boxes and class probabilities. YOLO's head typically uses multi-scale anchor boxes, ensuring effective detection of objects at different scales [42].

Table 3 provides a comparison of the strengths and weaknesses of various YOLO variants, showcasing the tradeoffs between speed, accuracy, and complexity.

**Table 3.** Strengths and weakness comparisons of YOLO variants.

Model	Strengths	Weaknesses
YOLOv1	Fast performance, real-time detection	Lower accuracy compared to two-stage detectors
YOLOv2	Better performance with anchor boxes	Still suffers from accuracy issues
YOLOv3	Improved accuracy with Darknet-53 backbone	Increased complexity, slower than earlier versions
YOLOv4	Adaptable with various heads and backbones	Less user-friendly
YOLOv5	Faster training and more accurate than YOLOv3	Less adaptable than YOLOv4
YOLOv6	Fewer computations and parameters	Less research and resource availability
YOLOv7	Accuracy and speed improvements over YOLOv6	Higher complexity, risk of overfitting
YOLOv8	Lighter and faster compared to YOLOv7	Case-specific optimizations
YOLO-NAS	Good balance of accuracy and speed	Less widely adopted
YOLOv9	Reduced model size, ideal for real-time applications	Focuses too much on specific objects, ignoring the rest

#### 4. Investigation, Evaluation with Benchmark

YOLO's real-time detection capabilities and versatility have revolutionized object detection across numerous domains. In cultural heritage, YOLO is employed to detect and classify artifacts in archaeological studies, aiding in the preservation of cultural history and restoration of structural defects in heritage sites [43,44]. In environmental monitoring, YOLO assists in tracking endangered species and identifying deforestation patterns through satellite imagery [45]. In healthcare, YOLO excels in automating critical tasks, such as detecting tumors in medical imaging and monitoring surgical tools during procedures [46]. In autonomous agriculture, YOLO plays a pivotal role in precision farming by identifying pests, diseases, and nutrient deficiencies, thereby enhancing crop yields and optimizing resource efficiency [47]. In industrial automation, YOLO improves defect detection on production lines, ensuring quality control in manufacturing processes [48]. Additionally, YOLO is a powerful tool in surveillance and security, powering facial recognition systems to enable real-time tracking of individuals in sensitive environments such as airports and banks [49]. The application of YOLO is widespread, making it challenging to enumerate all its use cases. Therefore, we have selected a few representative applications with publicly available datasets to conduct an in-depth investigation into the performance of YOLO models. This analysis thoroughly examines the architectures implemented by various researchers and evaluates their corresponding results. YOLO models are particularly popular in real-time applications due to their single-shot detection framework, which enables both speed and efficiency. In our comprehensive comparison study, we examined how YOLO models perform in key domains such as medical imaging, autonomous vehicles, and agriculture. Additionally, we provided an overview of the suitability of these models based on evaluation metrics such as accuracy, speed, and resource efficiency, highlighting their effectiveness across a diverse range of fields.

##### 4.1. Medicine

The YOLO framework has revolutionized medical image analysis with its ability to efficiently and accurately detect, classify, and segment medical images across a range of applications [50]. The core principles behind YOLO's success in medical image analysis can be divided into three main categories: object detection and classification, segmentation and localization, and compression and enhancement. Each of these frameworks offers unique capabilities to address specific challenges in medical imaging.

###### 4.1.1. Object Detection and Classification Frameworks

YOLO's object detection and classification frameworks focus on the accurate identification and categorization of objects within medical images. These applications streamline the detection of key features, abnormalities, and pathologies, reducing the time required for manual analysis and minimizing the risk of human error. By automating object classification, YOLO enables quicker and more precise medical diagnoses.

**Breast Cancer Detection in Mammograms:** YOLOv3 has been successfully implemented to detect breast abnormalities in mammograms, using fusion models to enhance the accuracy of classification. This model provides reliable detection of early-stage breast cancer, significantly aiding radiologists in making timely and accurate diagnoses [51].

**Face Mask Detection in Medical Settings:** YOLOv2 has been adapted to detect face masks in hospital and clinical environments, achieving an mAP of 81% [52]. This application was particularly important during the COVID-19 pandemic, ensuring that healthcare facilities maintained proper protective measures [53].

**Gallstone Detection and Classification:** YOLOv3 has demonstrated high accuracy (92.7%) in detecting and classifying gallstones in CT scans, making it a valuable tool in radiology for diagnosing gallbladder diseases quickly and accurately [54].

**Red Lesion Detection in Retinal Images:** YOLOv3 has been used to detect red lesions in RGB fundus images, achieving an average precision of 83.3% [55]. This early detection is

critical for preventing vision loss in patients suffering from diabetic retinopathy or other retinal diseases [56,57].

#### 4.1.2. Segmentation and Localization Frameworks

Segmentation and localization frameworks extend YOLO's capabilities beyond object detection to identifying specific regions of interest, such as tumors, and delineating their boundaries. This is particularly useful for medical image analysis, where precise localization of pathologies is critical for treatment planning [58].

**Ovarian Cancer Segmentation for Real-Time Use:** YOLOv5 has been applied for the segmentation of ovarian cancer, providing accurate identification of tumor boundaries [59]. The model's real-time capabilities allow it to be used during surgical procedures, giving doctors valuable information to guide their interventions [60].

**Multi-Modality Medical Image Segmentation:** YOLOv8, when combined with the Segment Anything Model (SAM), has shown strong results across multiple modalities, including X-ray, ultrasound, and CT scan images [61]. This integration enables more precise segmentation and classification, enhancing the diagnostic capabilities of radiologists and surgeons.

**Brain Tumor Localization Using Data Augmentation:** YOLOv3, paired with data augmentation techniques like 180° and 90° rotations, has proven effective in brain tumor detection and segmentation. The use of data augmentation strengthens the model's ability to identify tumors in complex orientations, leading to better treatment planning [62,63].

#### 4.1.3. Compression, Enhancement, and Reconstruction Frameworks

YOLO's application in medical image compression and enhancement helps address challenges in transmitting, storing, and reconstructing high-quality medical data. In telemedicine and secure medical data sharing, preserving image fidelity is critical for ensuring accurate diagnoses. YOLO's capabilities in this area provide efficient solutions for compressing medical images without compromising their quality.

**Medical Image Compression and Encryption:** YOLOv7 has been utilized to compress medical images while maintaining high-quality reconstructions during transmission [64]. In one notable application, liver tumor 3D scans were encrypted and reconstructed with a PSNR of 30.42 and SSIM of 0.94, ensuring that sensitive medical information remains secure and accurate during remote consultations or telemedicine applications [65].

**Secure Medical Data Transmission and Image Reconstruction:** YOLOv7 has also been applied to enhance the transmission and reconstruction of encrypted medical data, which is especially useful for maintaining data integrity in telemedicine. The chaos and encoding methods used in YOLOv7 ensure that high-quality images are transmitted securely across networks.

Table 4 summarizes various studies in medical research, showcasing the capabilities of different YOLO architectures.

**Table 4.** Applications of YOLO in medical image analysis.

YOLO Version	Performance Metrics	Observations/Results	Image Domain
YOLO v7 [66]	mAP	Proposed a unique 2D to 3D bounding box adaptation method using NMS for 3D image analysis. Achieved mAP of 82.10%.	Ultrasound Videos
YOLO v3 [67]	F1-Score, Accuracy, Precision, AUC, Recall	Fusion model accurately identified and categorized breast abnormalities in mammograms.	Breast Cancer
YOLO v2 [68]	mAP	Achieved 81% mAP for medical face mask detection.	Face Mask
YOLO v3 [69]	Recall, Precision, Accuracy	Detected and classified gallstones with an average accuracy of 92.7%.	Radiology

**Table 4.** Cont.

YOLO Version	Performance Metrics	Observations/Results	Image Domain
YOLO v8 [70]	Precision, recall, mAP50	Performed comparative analysis of various YOLOv8 variants to perform detection and classification using ultrasound images	Ovarian Cancer
YOLO v3 [71]	mAP	Best performance in medical imaging with data augmentation techniques, 180° and 90° rotations.	Brain Tumor
YOLO v3 [72]	Accuracy, Specificity, Sensitivity	Introduced a label-free method for cellular analysis using 2D light scattering and neural networks.	Oncology
YOLO v8 [73]	Dice Score, Precision, Recall, F1-Score	YOLO v8 combined with SAM achieved high scores on X-ray, ultrasound, and CT scan data.	X-Ray, Ultrasound, CT-Scan
YOLOv3 [55]	Average Precision	In the proposed work, they used YOLOv3 to detect red lesion in images and predicts bounding boxes and does classification using logistic regression. They achieved an average precision of 83.3%.	RGB Fundus Images
YOLOv7 [74]	Peak-signal-to-noise-ratio (PSNR), structural similarity index (SSIM)	They did image compression using YOLOv7 while preserving information in the images. They used chaos and encoding to encrypt the input images and performed reconstruction at the receiver. They achieved PSNR of 30.42 and SSIM of 0.94	Liver Tumor 3D scans

#### 4.2. Autonomous Vehicles

The application of YOLO models in the field of autonomous vehicles has proven transformative, particularly for enhancing real-time object detection capabilities [75]. As self-driving technology advances and autonomous vehicles become more prevalent, there is an increasing demand for robust computer vision systems capable of accurately detecting and classifying objects in diverse and often challenging environmental conditions [76]. This capability is essential to ensure the safety and reliability of autonomous driving systems.

##### 4.2.1. Challenges in Real-Time Object Detection for Autonomous Vehicles

One of the primary challenges faced by autonomous vehicles is maintaining accurate object detection under various weather conditions, such as fog, snow, and rain. These adverse conditions can distort images and interfere with navigation, making it difficult for the vehicle to accurately discern critical environmental factors like road signs, pedestrians, and other vehicles. The ultimate objective of integrating YOLO models into autonomous vehicle systems is to develop a reliable detection framework that ensures safe and efficient operation, regardless of external conditions.

YOLO's single-shot detection framework, known for its rapid inference time, makes it particularly well-suited for real-time applications. This is critical for autonomous vehicles, which must make quick decisions in dynamic environments. Moreover, advancements in YOLO's architecture, such as improvements in attention mechanisms, backbone networks, and feature fusion techniques, have further increased its accuracy and speed, allowing it to detect objects of varying sizes even smaller objects like pedestrians and distant vehicles [77].

##### 4.2.2. YOLO's Advancements for Harsh Weather Conditions

Real-time object detection in harsh weather is one of the most critical areas of research in autonomous vehicle systems. Weather conditions like snow, fog, and heavy rain can obscure visibility, making object detection challenging [78]. YOLO's rapid processing capabilities have been enhanced with architectural improvements to handle these conditions effectively. For instance, a study using YOLOv8 applied transfer learning to datasets captured under adverse weather conditions, including fog and snow. The model achieved promising results, with mAP scores of 0.672 and 0.704, demonstrating YOLO's adaptability

to real-world challenges and making it a strong candidate for deployment in autonomous driving systems.

#### 4.2.3. Small Object Detection in Autonomous Driving

Detecting small objects at various distances is another significant challenge for autonomous vehicle systems [79]. Small objects, such as pedestrians, cyclists, or road debris, can often be difficult to detect, but they are critical for safe driving. Enhanced versions of YOLO models, particularly YOLOv5, have introduced techniques like structural re-parameterization and the addition of small object detection layers to improve performance. These improvements have been shown to significantly increase the mAP for detecting smaller objects, especially in urban environments where small, fast-moving objects are common.

For example, WOG-YOLO, a variant of YOLOv5, exhibited significant improvements in detecting pedestrians and bikers, increasing the mAP for each by 2.6% and 2.3%, respectively. This focus on small object detection makes YOLO models particularly suitable for urban driving scenarios, where vehicles must detect various obstacles quickly and accurately to ensure safe navigation.

#### 4.2.4. Efficiency and Speed Optimizations for Autonomous Vehicles

In addition to improved accuracy, YOLO models have been optimized for speed and efficiency, making them ideal for edge computing applications in autonomous vehicles, where computational resources are often limited. Lightweight versions of YOLO, such as YOLOv7 and YOLOv5, have incorporated various architectural optimizations such as lightweight backbones, neural architecture search (NAS), and attention mechanisms to improve inference time without sacrificing accuracy.

For instance, YOLOv7 has been enhanced with a hybrid attention mechanism (ACmix) and the Res3Unit backbone, significantly improving its performance, achieving an AP score of 90.8% on road traffic data [80]. These optimizations ensure that YOLO models can deliver real-time performance in dynamic environments, a key requirement for autonomous vehicles operating in various conditions.

#### 4.2.5. YOLO's Performance Across Different Lighting Conditions

Another critical challenge in autonomous vehicle systems is detecting objects under varying lighting conditions, such as nighttime driving. YOLO models, particularly YOLOv8x, have demonstrated strong performance in low-light environments, utilizing advanced feature extraction and segmentation techniques to improve object detection accuracy.

In one study, YOLOv8x outperformed other YOLOv8 variants, achieving precision, recall, and F-score metrics of 0.90, 0.83, and 0.87, respectively, on video data captured during both day and night. This capability ensures that autonomous vehicles can operate safely across different lighting conditions, a vital feature for real-world deployment [81].

Table 5 provides a summary of key studies that highlight YOLO's effectiveness in autonomous vehicle applications. Various YOLO versions, including YOLOv5, YOLOv7, and YOLOv8, have been employed to enhance object detection performance across different datasets and conditions, showcasing YOLO's flexibility and adaptability to the unique challenges of autonomous driving.

**Table 5.** Application of YOLO in the field of autonomous vehicles.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv7 [80]	AP	Proposed modifications including ACmix for hybrid attention, RFLA for feature fusion, and Res3Unit backbone, achieving 89.2% AP.	Road traffic data
YOLOv5 [82]	mAP	Added structural re-parameterization and small object detection layers, achieving 96.1% mAP.	KITTI Dataset (8 classes)

**Table 5.** Cont.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv2, YOLOv3 [83]	mAP, Precision, Recall	Complex YOLO outperformed Tiny-YOLO with a mean Average Precision of 0.217.	KITTI Dataset
YOLOv5 [84]	mAP	WOG-YOLO showed improvements in detecting small objects, increasing mAP by 2.6% for pedestrians and 2.3% for bikers.	Self-built dataset
YOLOv5, YOLOv7 [85]	mAP	Evaluated on different activation functions. YOLOv7 with SiLU achieved an mAP of 94%, while YOLOv5 with LeakyReLU reached 88%.	Sandy weather data
YOLOv3 [86]	Accuracy, Recall	The RES-YOLO network improved vehicle detection accuracy and reduced background noise.	Vehicle dataset
YOLOv8 [87]	mAP	Transfer learning applied to hard weather datasets improved object detection, achieving mAP scores of 0.672 and 0.704.	Fog, snow, and hard weather datasets
YOLOv8 [88]	Precision, Recall, F-score	YOLOv8x outperformed other variants, with precision, recall, and F-score of 0.90, 0.83, and 0.87, respectively.	Video road data for day and night
YOLOv8 [89]	Accuracy	A five-stage model using YOLOv8 for detection and Deep Belief Network for classification achieved 95.6% accuracy.	Aerial imagery dataset
YOLOv8, YOLOv7, YOLOv5, YOLOv4 [90]	mAP	YOLOv7 performed the best with mAP50 scores when deployed on edge devices for real-time inference.	Road dataset
YOLOX [91]	mAP	Combined ShuffDet backbone and attention mechanism to achieve 92.20% mAP while reducing parameters by 34%.	KITTI dataset for road traffic
YOLOv4 [92]	mAP, F1-score, Recall	YOLOv4 was trained on a custom dataset for object detection, achieving an mAP of 74.6%, F1-score of 0.70, and recall of 0.81.	Custom vehicle dataset

#### 4.3. Agriculture

YOLO has transformed agricultural practices by enabling fast, accurate detection of crops, pests, and environmental factors affecting crop health. One of its most significant applications is crop monitoring, where YOLO's real-time object detection helps identify issues such as pests, diseases, and nutrient deficiencies early, enabling timely interventions that improve yields. In precision farming, YOLO helps distinguish crops from weeds, allowing selective herbicide application. This reduces chemical usage, lowers costs, and minimizes environmental impact, supporting sustainable agriculture. YOLO's integration with UAVs further enhances its utility, providing large-scale monitoring and detailed insights that would be difficult and time-consuming to gather manually. Beyond crop monitoring, YOLO is applied to tasks such as fire detection, livestock management, and environmental monitoring. By automating these processes, YOLO contributes to efficient farm management and improved resource use.

Table 6 summarizes notable applications of YOLO in agriculture.

**Table 6.** Application of YOLO in agriculture.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv5 [93]	Precision, Recall	UAV-based detection system for forest degradation, effective even in snowy conditions.	UAV imagery of trees
YOLOv5 [94]	Precision, F1-Score, mAP, Recall	DeepForest with YOLO, mAP of 82%, outperformed other models.	Apple trees in UAV RGB images
YOLOv5s, YOLOv5x [95]	IoU, Recall, Precision, mAP	Lightweight YOLOv5s ideal for embedded systems.	RGB forest images

**Table 6.** Cont.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv3 [96]	F1-Score	Modified YOLOv3, F1-score of 0.817, used for apple detection.	Apple images from orchards
YOLOv5 [97]	F1-Score, mAP	YOLOv5 ensemble technique for real-time forest fire detection, F1-score of 93%.	FLAME dataset from UAVs
YOLOv3 [98]	IoU	YOLOv3 achieved 91.80% average precision for bird detection.	UAV aerial images
YOLOv4 [99]	mAP	YOLOv4 for cherry ripeness detection, increased mAP by 0.5%.	Cherry ripeness detection
YOLOv8s [100]	mAP	Lightweight YOLOv8s model with mAP of 93.4%, suitable for small devices.	Tomato image data
Ag-YOLO [101]	F1-Score	Ag-YOLO for crop monitoring and spraying, F1-score of 0.9205.	UAV imagery of palm trees

#### 4.3.1. Crop Health Monitoring and UAV Integration

One of the most critical applications of YOLO in agriculture is the continuous monitoring of crop health [102]. With the integration of YOLO models and UAV systems, farmers can now scan large-scale agricultural areas, detecting early signs of pest infestations, diseases, or nutrient deficiencies. This proactive approach helps prevent crop loss and optimize farm productivity.

A noteworthy study employed YOLOv5 integrated into UAVs for detecting forest degradation. The model performed impressively, identifying damaged trees even in challenging conditions such as snow [103]. This advancement has far-reaching implications, allowing farmers and environmental researchers to monitor vast landscapes, particularly in remote or difficult-to-access areas, with unprecedented accuracy and speed [104].

#### 4.3.2. Precision Agriculture and Weed Management

YOLO plays a pivotal role in precision agriculture by facilitating accurate crop and weed differentiation. In precision farming, herbicides can be applied selectively to target weeds without affecting crops [105,106]. This method not only reduces the use of chemicals, lowering the cost of farming, but also minimizes environmental damage, leading to more sustainable agricultural practices.

A modified version of YOLOv3 was deployed to monitor apple orchards, achieving an F1-score of 0.817 [107]. The model utilized DenseNet for improved feature extraction and performed exceptionally well at detecting apples at various developmental stages. This level of precision is invaluable for farmers managing orchards, as it enables better decision-making regarding harvesting and pest control [108].

#### 4.3.3. Real-Time Environmental Monitoring and Fire Detection

YOLO has also been applied to monitor environmental hazards, such as forest fires [109,110]. In a study using an ensemble of YOLOv5 and other detection models like DeepLab and LightYOLO, researchers developed a real-time fire detection system mounted on UAVs. The system outperformed conventional models, achieving an F1-score of 93% and mAP of 85.8%. This real-time capability is crucial in mitigating the spread of fires and protecting valuable forest and agricultural resources [111].

#### 4.3.4. Application in Livestock Management and Other Areas

While YOLO's application in crop monitoring remains a primary focus, it has also been applied to livestock management [112]. UAVs equipped with YOLO models can monitor livestock across large grazing areas, providing real-time updates on animal health

and location. This technology is instrumental in reducing the risk of disease outbreaks and improving overall farm management [113].

Furthermore, YOLO's flexibility extends to various agricultural operations. A notable example is Ag-YOLO, a lightweight version of YOLO developed for Intel Neural Compute Stick 2 (NCS2) hardware, which was designed for crop and spray monitoring. Ag-YOLO achieved an F1-score of 0.9205, proving to be an efficient, cost-effective solution for farmers operating in resource-constrained environments [101].

#### 4.4. Industry

In industrial applications, YOLO (You Only Look Once) has become one of the most widely used real-time object detection models due to its high-speed processing and efficient object identification capabilities. The single-stage architecture of YOLO allows it to detect and classify objects in a single pass through the neural network, making it particularly suitable for environments that require rapid decision-making, such as production lines, automated quality control, and anomaly detection. Its adaptability to a wide range of tasks across different industries, from food processing to construction, has made YOLO a versatile tool.

In manufacturing and production, YOLO is used to improve the accuracy and efficiency of automated systems. Whether it's detecting defects in products or monitoring safety in real-time, YOLO contributes to higher-quality outcomes and reduced operational costs [114]. For instance, in food processing, YOLO can be employed to ensure quality control by detecting defects in packaged goods, while in construction, it can help identify safety issues, such as the use of protective gear like helmets [115].

YOLO's implementation in logistics and warehousing has also streamlined processes such as package tracking, inventory management, and equipment monitoring. Robotic systems using YOLO for object detection and identification can automate repetitive tasks, improve safety, and increase production throughput. Despite certain challenges, such as lower accuracy when detecting. Table 7 summarizes various industrial applications of YOLO, showcasing its versatility and effectiveness across different tasks.

**Table 7.** Applications of YOLO in industry.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv5 [116]	mAP	Combined YOLO with transformers to detect small-scale objects and defects. Proposed a bidirectional feature pyramid network, achieving an mAP of 75.2%.	Grayscale imagery of surface defects
YOLOv5 [117]	mAP	PG-YOLO was developed as a lightweight version for edge devices in IoT networks. Improved inference speed without sacrificing accuracy, achieving an mAP of 93.3%.	RGB images of safety-helmet wearing (SHWD)
YOLOv3 [118]	mAP	For detecting weld defects in vehicle wheels, this model achieved an mAP of 98.25% and 84.36% on AP 75 and AP 50, respectively. Environment-specific model.	RGB images of vehicle wheel welding
YOLOv5 [119]	mAP, GFLOPS	ATT-YOLO, inspired by transformers, performed well on surface detection tasks in electronic manufacturing, achieving an mAP of 49.9% with 21.8 GFLOPS.	Laptop surface images
YOLOv3, YOLOv5 [120]	mAP	YOLO models used for process flow tracking in Industry 4.0. Models were applied to datasets with distortions and achieved final mAP scores of 80% (YOLOv3) and 70% (YOLOv5).	Colorful objects in industry setting
YOLO [121]	Precision, Accuracy, mAP	Introduced a deep learning-based system for real-time packaging defect detection. The system achieved 81.8% precision, 82.5% accuracy, and an mAP of 78.6%.	Images of damaged and intact boxes

**Table 7.** Cont.

YOLO Version	Performance Metrics	Observations	Image Domain
YOLOv5s [122]	Precision, Recall, mAP	An upgraded YOLOv5s model used for real-time identification and localization of production line equipment, including robotic arms and AGV carts. Achieved precision of 93.6%, recall of 85.6%, and mAP of 91.8%.	Real-time data from simulated production line
YOLOv5 [123]	mAP	Enhanced YOLOv5 for detecting workpieces on production lines. Improved mAP on COCO dataset by 2.4% and on custom data by 4.2% using ghost bottleneck lightweight deep convolution.	Images of bolts
YOLOD [124]	Average Precision	YOLOD, a unique model addressing uncertainty in object detection, used Gaussian priors in front of YOLOX detection heads. Improved bounding box regression and achieved an AP of 73.9%.	Power line insulator images

#### 4.4.1. Applications of YOLO in Industrial Manufacturing

One of the most impactful uses of YOLO in industry is in manufacturing, where its real-time detection capabilities are leveraged for automated quality control, defect detection, and production optimization. YOLO's speed and accuracy allow manufacturers to identify issues on the production line quickly, reducing downtime and minimizing faulty outputs [125].

**Surface Defect Detection:** YOLOv5, combined with transformers, was used to detect small defects on surfaces in grayscale imagery, improving detection efficiency. The bidirectional feature pyramid network proposed in this study significantly enhanced the model's ability to identify minor defects, achieving an mAP of 75.2% [126].

**Wheel Welding Defect Detection:** YOLOv3 was applied to the specific task of detecting weld defects in vehicle wheels, achieving impressive results with mAP scores of 98.25% and 84.36% at different thresholds. While this model was highly effective in its target environment, it was noted that it may not generalize well to other real-time detection tasks without further adaptation [127].

**Workpiece Detection and Localization:** Another notable application is workpiece detection on production lines, where YOLOv5 was enhanced with lightweight deep convolution layers to improve detection accuracy. The model showed significant improvements, increasing mAP by 2.4% on the COCO dataset and 4.2% on custom industrial datasets [128].

#### 4.4.2. Applications in Automated Quality Control and Safety

YOLO's role in quality control systems is crucial for maintaining the consistency and safety of products in industries like packaging, construction, and electronics. By automatically identifying defects or inconsistencies in real-time, YOLO allows manufacturers to catch problems early in the production process.

**Real-Time Packaging Defect Detection:** In one study, YOLO was used to develop a deep learning-based system for detecting packaging defects in real-time. The model automatically classified product quality by detecting defects in boxed goods, achieving a precision of 81.8%, accuracy of 82.5%, and mAP of 78.6%. Such systems can be deployed to monitor quality across high-speed production lines, reducing the risk of shipping faulty products to customers [129].

**Safety-Helmet Detection in Construction:** Safety in construction is another critical area where YOLO has proven its worth. A lightweight version of YOLOv5, known as PG-YOLO, was specifically developed for edge devices in IoT networks, improving inference speed while maintaining accuracy. The model achieved an mAP of 93.3% for detecting workers wearing safety helmets in construction sites, helping ensure compliance with safety regulations [130].

#### 4.4.3. Industrial Robotics and Automation

In logistics and warehouse management, YOLO's ability to detect and identify objects in real-time is a key asset for automating routine tasks, improving both safety and efficiency. YOLO is integrated into robotic systems that handle object detection for sorting, transporting, and monitoring inventory.

**Production Line Equipment Monitoring:** YOLOv5s was enhanced with a channel attention module, slim-neck, decoupled head, and GSConv lightweight convolution to improve real-time identification and localization of production line equipment, such as robotic arms and AGV carts. This system achieved precision rates of 93.6% and mAP scores of 91.8%, showcasing its effectiveness in automating production line processes [122].

**Power Line Insulator Detection:** In another study, YOLOD was developed to address uncertainty in object detection by placing Gaussian priors in front of the YOLOX detection heads. This model was applied to power line insulator detection, improving the robustness of object detection by using calculated uncertainty scores to refine bounding box predictions. YOLOD achieved an AP of 73.9% [131].

### 5. Evolution and Benchmark-Based Discussion

#### 5.1. Evolution

The YOLO family has undergone considerable evolution, with each new iteration addressing specific limitations of its predecessors while introducing innovations that improve its performance in real-time object detection. YOLOv6, YOLOv7, and YOLOv8 represent key advancements in the early evolution of this framework, each contributing significantly to the landscape of object detection in terms of speed, accuracy, and computational efficiency.

**YOLOv6: Enhanced Speed and Practicality.** YOLOv6 was designed with a focus on speed and practicality, particularly for real-world applications requiring fast and efficient object detection. It introduced modifications in network structure that enabled faster processing while maintaining a decent level of accuracy. YOLOv6 was particularly impactful for lightweight deployment on edge devices, which have limited computing power, making it an attractive option for applications in fields such as surveillance, robotics, and automated inspection systems. However, while YOLOv6 improved efficiency, it faced challenges in complex scenarios involving small or overlapping objects, leading to the need for more advanced versions.

**YOLOv7: Improving Accuracy and Feature Extraction.** YOLOv7 introduced significant architectural changes that enhanced accuracy and feature extraction capabilities. One of the key advancements in YOLOv7 was the integration of cross-stage partial networks (CSPNet), which improved the model's ability to reuse gradients across different stages, allowing for better feature propagation and reducing the model's overall complexity. This improvement translated into better performance in detecting smaller objects or objects within cluttered environments. YOLOv7 also introduced the concept of extended path aggregation, which helped in merging features from different layers to provide a more detailed and robust representation of the input image. These advancements made YOLOv7 more suitable for applications in industries like medical imaging, autonomous driving, and aerial surveillance, where high accuracy in challenging environments is paramount. However, even with these improvements, YOLOv7 was not immune to the problem of vanishing gradients a common issue in deeper neural networks that leads to poor training outcomes due to the weakening of signal propagation as it moves through multiple layers. This was particularly problematic in cases where high-resolution image data required more sophisticated feature extraction.

**YOLOv8: Streamlining for Resource Efficiency.** YOLOv8 further refined the architecture, focusing on achieving better resource efficiency without sacrificing accuracy. One of the most notable advancements in YOLOv8 was its ability to scale efficiently across different hardware configurations, making it a flexible tool for both low-power devices and high-performance computing environments. YOLOv8 streamlined the training process

and introduced optimizations that improved its ability to generalize across a wide range of object detection tasks. YOLOv8 also enhanced the handling of multi-object detection scenarios, where multiple objects of different sizes and shapes appear in a single frame. Despite these improvements, YOLOv8 faced challenges in deeper architectures, particularly with convergence issues. These issues arose from the model's struggle to balance the computational complexity required for deeper networks with the need for real-time inference. As a result, YOLOv8's performance on complex datasets, especially those involving small, overlapping, or occluded objects, was not always consistent.

**Addressing Limitations: The Road to YOLO-NAS and YOLOv9.** The limitations observed in YOLOv6 through YOLOv8 such as vanishing gradients and convergence problems were the catalysts for the development of more sophisticated models like YOLO-NAS and YOLOv9. These models aimed to not only enhance speed and accuracy but also tackle the deeper challenges inherent to neural network architectures, such as gradient management and efficient feature extraction.

### 5.1.1. YOLO-NAS: A Major Turning Point

Before the introduction of YOLOv9 [132], YOLO-NAS [133] developed by Deci AI marked a significant shift in the evolution of the YOLO framework. As object detection models became more widely deployed in real-world applications, there was a growing need for solutions that could balance accuracy with computational efficiency, especially on edge devices that have limited processing power. YOLO-NAS answered this call by incorporating Post-Training Quantization (PTQ), a technique designed to reduce the size and complexity of the model after training [134]. This allowed YOLO-NAS to maintain high levels of accuracy while reducing its computational footprint, making it ideal for resource-constrained environments like mobile devices, embedded systems, and IoT applications.

PTQ enabled YOLO-NAS to deliver minimal latency, which is a crucial factor for real-time object detection, where every millisecond counts. By optimizing the model post-training, PTQ made YOLO-NAS one of the most efficient object detection models for real-time applications, especially in industries where computational resources are scarce, such as autonomous vehicles, robotics, and smart cameras for security systems. The ability to reduce inference time without sacrificing performance positioned YOLO-NAS as a go-to model for developers looking to deploy sophisticated object detection systems on low-power devices.

YOLO-NAS introduced two significant architectural innovations that set it apart from previous models in the YOLO family:

1. Quantization and Sparsity Aware Split-Attention (QSP): The QSP block was designed to enhance the model's ability to handle quantization while still maintaining high accuracy. Quantization often leads to a degradation of model precision because the model is forced to operate with reduced numerical precision (e.g., moving from floating-point to integer operations). QSP mitigated this accuracy drop by using sparsity-aware mechanisms that allowed the model to be more selective in how it used and stored information across different layers. This helped preserve important features, even in a quantized environment.
2. Quantization and Channel-Wise Interactions (QCI): The QCI block further refined the process of quantization by focusing on channel-wise interactions. It enhanced the way features were extracted and processed in the network, ensuring that key information was not lost during the quantization process. By intelligently adjusting how information is passed between channels, QCI ensured that YOLO-NAS could maintain high precision in its predictions, even when the model was reduced to a lightweight architecture. This made it particularly useful for edge applications that require smaller model sizes but cannot afford to lose accuracy.

These innovations were inspired by frameworks like RepVGG and aimed to address the common challenges associated with post-training quantization, specifically the loss of accuracy that typically accompanies such optimization techniques [135]. The combi-

nation of QSP and QCI allowed YOLO-NAS to achieve a high level of precision while retaining a small model size, making it an efficient tool for real-time object detection on constrained hardware.

Despite the impressive advancements introduced in YOLO-NAS, the model did face challenges in handling high-complexity image detection tasks. In scenarios where objects were occluded or had intricate patterns, YOLO-NAS struggled to maintain the same level of accuracy as it did in simpler object detection tasks. For example, applications in agriculture, where leaves and crops often occlude one another, or in medical imaging, where subtle variations in texture and shape are critical, highlighted the limitations of YOLO-NAS. The model's performance often dipped when faced with these complex visual environments, underscoring the need for further architectural improvements.

While YOLO-NAS was an excellent solution for resource-efficient detection in straightforward real-time applications, it required additional improvements to handle the nuances of more complex datasets. These limitations laid the groundwork for the development of future models like YOLOv9, which aimed to address these issues through more advanced gradient handling, better feature extraction, and the use of more sophisticated network architectures.

### 5.1.2. YOLOv9: Groundbreaking Innovations

In response to the challenges encountered by earlier models, YOLOv9 introduced several groundbreaking techniques designed to improve gradient flow, handle error accumulation, and facilitate better convergence during training. These innovations allowed YOLOv9 to extend its applicability to a broader range of real-world object detection tasks.

The key innovations in YOLOv9 include:

1. Programmable Gradient Information (PGI): PGI was designed to tackle the issue of vanishing gradients by enhancing the flow of gradients throughout the model. YOLOv9's PGI ensured smoother backpropagation across multiple prediction branches, significantly improving convergence and overall detection accuracy. PGI is composed of three key components. (a) Main Branch: Responsible for inference tasks. (b) Auxiliary Branch: Manages gradient flow and updates the network parameters. (c) Multi-level Auxiliary Branch: Handles error accumulation and ensures that the gradients propagate effectively across all layers [132]. By addressing gradient backpropagation across complex prediction branches, PGI allowed YOLOv9 to achieve better performance, particularly in detecting multiple objects in challenging environments.
2. Generalized Efficient Layer Aggregation Network (GELAN): The GELAN module was another major innovation in YOLOv9. By drawing from CSPNet [136] and ELAN [137], GELAN provided flexibility in integrating different computational blocks, such as convolutional layers and attention mechanisms. This adaptability allowed YOLOv9 to be fine-tuned for specific detection tasks, from simple object recognition to complex multi-object detection, making it a versatile tool for a wide array of real-time detection applications.
3. Reversible Functions: To ensure information preservation throughout the network, YOLOv9 utilized reversible functions. The formula used for this is:  $X = v\zeta(r\psi(X))$ , where  $r\psi(X)$  represents the transformation of input data through a reversible function, and  $v\zeta$  applies an inverse transformation to recover the original input. These reversible layers allowed YOLOv9 to reconstruct input data perfectly, minimizing information loss during forward and backward passes. The reversible functions allowed for more precise detection and localization of objects, especially in high-dimensional and complex datasets.

### 5.1.3. Evolution to YOLOv10 and YOLOv11

Following YOLOv9, YOLOv10 and YOLOv11 brought further refinements to the YOLO framework, making significant strides in both speed and accuracy.

YOLOv10 introduced groundbreaking innovations that enhanced performance and efficiency, building on the strengths of its predecessors while pushing the boundaries of real-time object detection. A key advancement in YOLOv10 was the introduction of the C3k2 block, an innovative feature that greatly improved feature aggregation while reducing computational overhead [138]. This allowed YOLOv10 to maintain high accuracy even in resource-constrained environments, making it ideal for deployment on edge devices. Additionally, the model's improved attention mechanisms enabled better detection of small and occluded objects, allowing YOLOv10 to outperform previous versions in tasks such as facemask detection and autonomous vehicle applications. Its ability to balance computational efficiency with detection precision set a new standard, with a final mAP50 of 0.944 in benchmark tests.

YOLOv11 further advanced the framework with the introduction of C2PSA (Cross-Stage Partial with Spatial Attention) blocks, which significantly enhanced spatial awareness by enabling the model to focus more effectively on critical regions within an image [139]. This innovation proved especially beneficial in complex scenarios, such as shellfish monitoring and healthcare applications, where precision and accuracy are crucial. YOLOv11 also featured a restructured backbone with smaller kernel sizes and optimized layers, which improved processing speed without sacrificing performance. The inclusion of Spatial Pyramid Pooling-Fast (SPPF) enabled even faster feature aggregation, solidifying YOLOv11 as the most efficient and accurate YOLO model to date. It achieved a final mAP50 of 0.958 across multiple benchmarks, making YOLOv11 a leading choice for real-time object detection tasks across industries ranging from healthcare to autonomous systems.

## 5.2. Benchmarks

With the introduction of YOLOv9, the benchmarks for performance have shifted. We conducted a comprehensive evaluation of YOLOv9, YOLO-NAS, YOLOv8, YOLOv10, and YOLOv11 using well-established datasets like Roboflow 100 [140], Object365 [141], and COCO [142]. These datasets offer diverse real-world challenges, allowing us to assess the models' strengths and weaknesses under different conditions.

### 5.2.1. Benchmark Findings

Our results consistently showed that YOLOv9 outperformed both YOLO-NAS and YOLOv8, particularly in complex image detection tasks where objects are occluded or exhibit detailed patterns. However, with the introduction of YOLOv10 and YOLOv11, the benchmark shifted even further. YOLOv10 introduced new feature aggregation techniques, and YOLOv11's spatial attention mechanisms greatly improved object detection, especially in challenging datasets.

For instance:

- **Shellfish Monitoring:** In tasks such as shellfish monitoring, which involve complex patterns and occlusions, YOLO-NAS and YOLOv8 struggled to maintain accuracy. While YOLOv9 demonstrated greater adaptability, handling the intricate challenges more effectively, YOLOv10 and YOLOv11 pushed the performance further with their superior spatial attention and feature extraction capabilities. YOLOv11 achieved a final mAP50 of 0.563, the highest among the tested models.
- **Medical Image Analysis:** On medical image datasets, particularly for tasks such as blood cell detection, YOLOv9 outperformed YOLO-NAS and YOLOv8. However, the architectural advancements in YOLOv10 and YOLOv11 resulted in even better performance, with YOLOv11 achieving an mAP50 of 0.958. This demonstrates its superior capability to detect small and detailed features within cluttered or noisy data, which is critical for applications in medical diagnostics.

These results are summarized in Table 8, highlighting the mAP50 (mean average precision at 50% intersection over union) scores across several datasets.

**Table 8.** Performance of YOLOv11, YOLOv10, YOLOv9, YOLO-NAS and YOLOv8 on multiple datasets.

Datasets	YOLOv9	YOLO-NAS	YOLOv8	YOLOv10	YOLOv11
Facemask Detection	0.941	0.9199	0.924	0.950	0.962
Shellfish Monitoring	0.534	0.469	0.466	0.542	0.563
Forest Smoke Detection	0.865	0.7811	0.911	0.925	0.945
Human Detection	0.812	0.7145	0.816	0.829	0.854
Pothole Detection	0.78	0.7265	0.745	0.793	0.815
Blood Cell Detection	0.933	0.9041	0.93	0.944	0.958

### 5.2.2. Performance Insights

**Facemask Detection:** All models performed well on this dataset, with YOLOv11 dominating, achieving an mAP50 of 0.962. The improved spatial attention mechanisms in YOLOv11 allowed it to detect subtle variations in facemask patterns, making it the best-performing model in this domain. YOLOv10 also showed improvements over YOLOv9, reaching an mAP50 of 0.950.

**Shellfish Monitoring:** This dataset presented complex challenges with occluded and overlapping objects. Both YOLO-NAS and YOLOv8 struggled, achieving relatively low mAP50 scores of 0.469 and 0.466, respectively. YOLOv9 demonstrated better adaptability with 0.534, while YOLOv10 and YOLOv11 further improved on these results, reaching 0.542 and 0.563, respectively, thanks to their advanced spatial attention mechanisms.

**Forest Smoke Detection:** YOLOv8 performed particularly well in detecting smoke patterns, achieving an mAP50 of 0.911, while YOLOv9 closely followed with 0.865. However, YOLOv10 and YOLOv11 both improved on this with scores of 0.925 and 0.945, respectively. The improvements in feature extraction and attention mechanisms in YOLOv11 gave it a slight edge over previous versions in detecting subtle smoke patterns.

**Human Detection:** For human detection, YOLOv9 outperformed YOLO-NAS but was only marginally better than YOLOv8. YOLOv10 reached 0.829, and YOLOv11 achieved the highest mAP50 at 0.854, benefiting from its optimized backbone and better handling of occlusions in complex scenes.

**Pothole Detection:** YOLOv9 demonstrated superior performance in detecting potholes on road surfaces, achieving an mAP50 of 0.780. However, YOLOv10 and YOLOv11 demonstrated further advancements, reaching 0.793 and 0.815, respectively, making them more reliable for this specific detection task.

**Blood Cell Detection:** In medical image analysis, YOLOv9 showed great precision with an mAP50 of 0.933. However, YOLOv10 and YOLOv11 set a new standard for this dataset, achieving 0.944 and 0.958, respectively. The enhanced gradient flow and feature extraction capabilities in these models contributed to their superior performance in detecting minute and complex patterns, crucial in medical diagnostics.

### 5.2.3. Training Considerations: Extended Epochs for Model Optimization

In our benchmark study, the YOLO models (YOLOv9, YOLO-NAS, YOLOv8, YOLOv10, and YOLOv11) were trained for 20 epochs, providing a baseline for performance evaluation. However, it is evident that this limited number of training cycles does not fully tap into the models' potential. Extended training, involving more epochs, would likely result in smoother learning curves and allow the models to achieve even higher levels of accuracy, especially in tasks that involve complex datasets and intricate object patterns.

Extended training is essential for performance improvement across several aspects of object detection models:

1. **Refined Feature Extraction:** Each epoch allows the model to update and refine its internal feature representations. Models like YOLO-NAS, which are optimized for resource efficiency, benefit from longer training cycles, as this gives the model more op-

portunities to fine-tune its performance on challenging examples like occluded objects or those with non-standard shapes. YOLOv10, with its enhanced feature aggregation blocks, could further refine its detection capabilities with extended epochs, improving accuracy in both resource-constrained environments and complex scenarios.

2. Improved Convergence: While 20 epochs are sufficient for initial insights into performance, convergence when a model reaches its lowest possible error rate often requires more iterations. YOLOv9, YOLOv10, and YOLOv11, featuring more intricate architectures, benefit from prolonged training to reach optimal convergence. Specifically, YOLOv9's use of Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Networks (GELAN) would see further improvements with more epochs, helping the model stabilize gradient flow and improve detection accuracy in environments with occluded or complex objects. YOLOv11, with its C2PSA blocks, would also benefit from more iterations, as this would allow better spatial awareness and feature extraction for difficult detection tasks.
3. Avoiding Overfitting: One concern with prolonged training is overfitting, where the model becomes too tuned to the training data, losing its generalization ability. However, this risk can be mitigated by using early stopping techniques or cross-validation. For models like YOLO-NAS and YOLOv8, incorporating regularization methods such as dropout layers or weight decay would allow them to benefit from extended training epochs without falling prey to overfitting. YOLOv11, with its spatial attention mechanisms, could particularly benefit from extended training, as more iterations would enhance its ability to focus on the most critical regions of the image.
4. Fine-Tuning for Specialized Tasks: In applications like medical imaging, autonomous driving, or industrial automation, where precision is critical, additional training epochs combined with fine-tuning can make a noticeable difference in model performance. YOLOv10, with its enhanced feature aggregation techniques, could greatly benefit from fine-tuning to optimize its performance in specific tasks such as pothole detection or shellfish monitoring. YOLOv11, with its cutting-edge spatial attention blocks, would further improve in complex, specialized tasks like forest smoke detection or health monitoring, where accurate object localization is paramount.
5. Learning Rate Scheduling: As the number of epochs increases, adjusting the learning rate becomes essential. Models like YOLOv9, YOLOv10, and YOLOv11 would benefit from learning rate schedulers that gradually decrease the learning rate during training, ensuring stable convergence and preventing the model from overshooting its optimal parameter values. Adaptive optimization techniques, such as AdamW or RMSProp, could further enhance training performance in extended epochs, particularly in models with complex architectures like YOLOv10 and YOLOv11.

The Specific Impacts on YOLO-NAS, YOLOv9, YOLOv10, and YOLOv11 are as follows:

- YOLO-NAS: Given that YOLO-NAS uses Post-Training Quantization (PTQ), additional epochs would help solidify its quantization strategy, improving performance on both low-resource and high-complexity tasks. The Quantization and Sparsity Aware Split-Attention (QSP) and Quantization and Channel-Wise Interactions (QCI) blocks would benefit from extended training, leading to better feature selection and processing even with quantization. Longer training would also help the model adapt to complex detection tasks that require precision, like health monitoring or industrial automation.
- YOLOv9: YOLOv9's architectural advancements, such as PGI and GELAN, would see improved performance with more training epochs, especially in scenarios involving multiple prediction branches. Extended training would stabilize its gradient flow and error management, improving performance in complex environments like forest smoke detection or medical imaging. Additionally, YOLOv9's reversible functions could further benefit from extended epochs, ensuring that input reconstruction becomes more robust, ultimately enhancing detection precision in real-time applications.
- YOLOv10: As a major advancement over previous versions, YOLOv10 introduced the C3k2 block for more efficient feature aggregation. Extended training would refine

its ability to balance computational efficiency and detection precision, particularly for resource-constrained tasks on edge devices. Additional epochs would enable YOLOv10 to improve its performance on specialized tasks like pothole detection and autonomous vehicle applications, achieving better convergence and stability.

- YOLOv11: The most recent and advanced YOLO model, YOLOv11, introduced the Cross-Stage Partial with Spatial Attention (C2PSA) blocks, greatly enhancing spatial awareness. Prolonged training would allow the model to fine-tune these blocks for better detection of small or occluded objects, especially in complex tasks such as shellfish monitoring or industrial safety detection. The use of Spatial Pyramid Pooling-Fast (SPPF) for feature aggregation would also improve with more epochs, enabling YOLOv11 to excel in real-time detection tasks with a final mAP50 score that sets it apart from earlier models.

As we look toward further optimizing the YOLO family models, a comprehensive training regimen that includes extended epochs, dynamic learning rates, and fine-tuning techniques will be crucial in extracting the best performance from these models. Expanding the datasets to include even more diverse real-world scenarios will allow the models to generalize better across different applications, ensuring they maintain top-tier performance in both simple and complex detection tasks.

While our 20-epoch benchmark provided valuable performance insights, longer training cycles combined with the right optimization techniques would likely unlock even greater accuracy and generalization potential for YOLOv9, YOLO-NAS, YOLOv8, YOLOv10, and YOLOv11, particularly in challenging real-time object detection tasks.

#### 5.2.4. Performance Analysis of YOLO Variants on Benchmark Datasets

The mAP50 charts in Figure 8 illustrate how YOLOv11, YOLOv10, YOLOv9, YOLO-NAS, and YOLOv8 perform across a range of benchmark datasets. These plots provide an in-depth comparison of the models over 20 training epochs, offering key insights into how each model handles different detection tasks. The inclusion of YOLOv10 and YOLOv11 further emphasizes the advancements made in real-time object detection, particularly in complex datasets that challenge earlier models. In this study, mAP50 is employed as the primary metric for evaluating and comparing the performance across different YOLO versions. The choice of mAP50 simplifies the benchmarking process by focusing on straightforward detection tasks, providing a consistent and accessible measure of model effectiveness. By setting a fixed IoU threshold of 50%, mAP50 highlights the models' capabilities in identifying objects with acceptable localization accuracy, making it particularly suitable for general-purpose evaluations and less complex detection scenarios. This approach ensures a balanced comparison, especially when dealing with diverse datasets and varying levels of object detection difficulty.

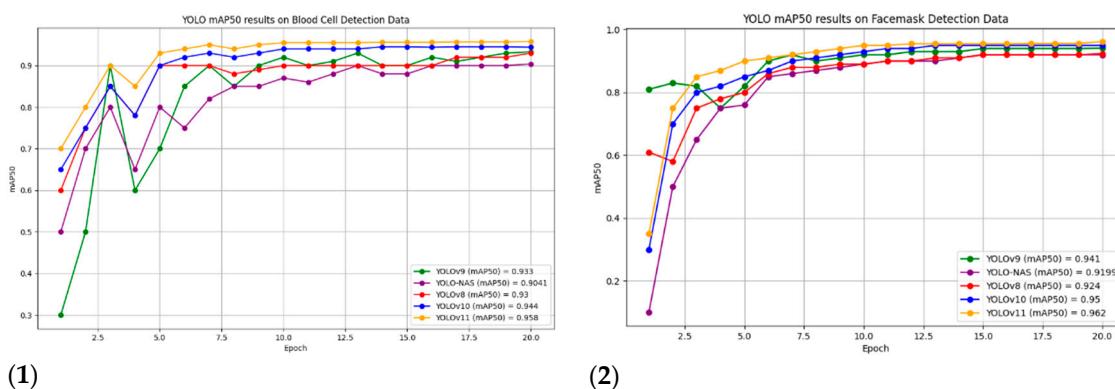
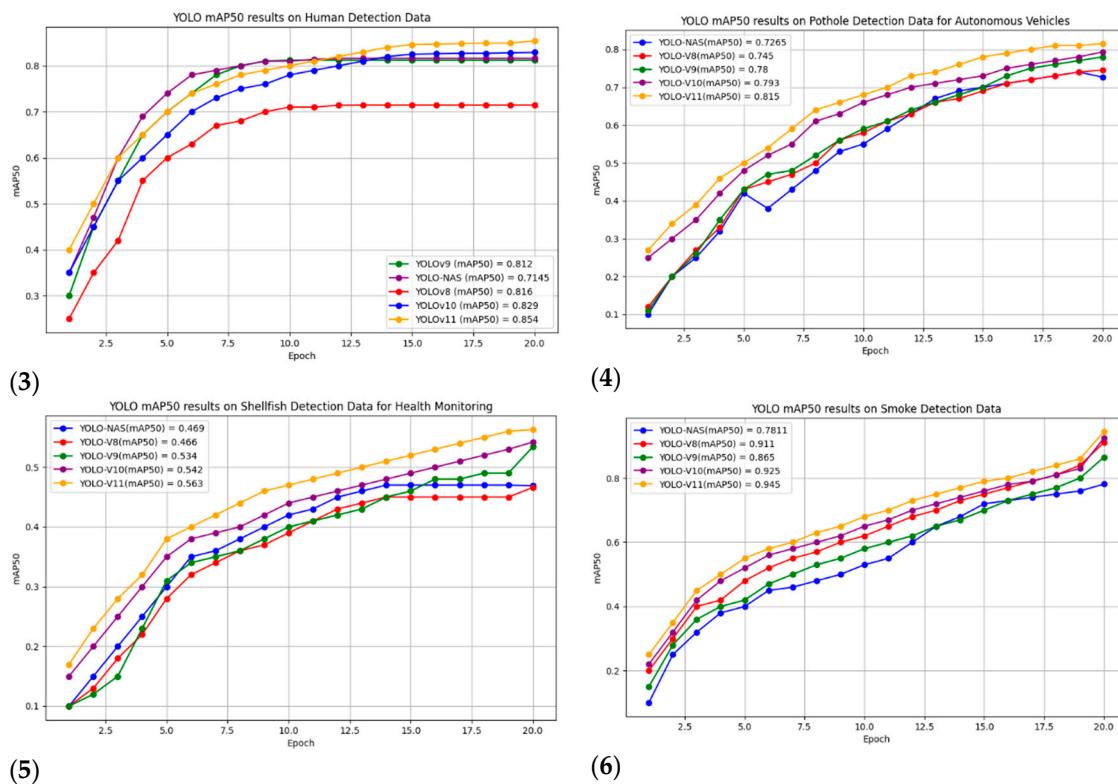


Figure 8. Cont.



**Figure 8.** mAP50 plots of YOLO11, YOLO10, YOLOv9, YOLO-NAS and YOLOv8 on benchmark datasets. (1) shows the performance of various YOLO models on Blood Cell detection task. (2) illustrates the performance of YOLO models on Face mask detection task. (3) represents the results on human detection problem. (4) shows the results on pothole detection task for autonomous vehicles. (5) shows the results for shellfish health monitoring task and (6) shows results for smoke detection task.

- YOLOv11 and YOLOv10 take the lead: Across almost all benchmarks, YOLOv11 consistently outperforms its predecessors, demonstrating the effectiveness of its C2PSA (Cross-Stage Partial with Spatial Attention) blocks, which allow the model to focus more accurately on important regions within the image. YOLOv10 follows closely behind, benefiting from its C3k2 blocks that optimize feature aggregation and balance computational efficiency with accuracy.
  - Blood cell detection: YOLOv11 achieved the highest mAP50 of 0.958, followed by YOLOv10 at 0.944. Both models surpass YOLOv9 (0.933), YOLO-NAS (0.9041), and YOLOv8 (0.93). The C2PSA and C3k2 blocks in YOLOv11 and YOLOv10 allowed these models to detect intricate patterns, such as those found in medical imaging, more effectively than previous iterations.
  - Facemask detection: YOLOv11 also dominated the facemask detection dataset, achieving a mAP50 of 0.962, compared to YOLOv10's 0.950. This was higher than YOLOv9 (0.941), YOLOv8 (0.924), and YOLO-NAS (0.9199), indicating that the newer models are better suited for precise feature extraction in tasks with clear object boundaries, such as facemask detection.
  - Human Detection: In human detection, YOLOv11 scored a mAP50 of 0.854, with YOLOv10 at 0.829. Although YOLOv9 closely followed at 0.812, YOLOv11 and YOLOv10 demonstrated more stability across the 20 training epochs, thanks to their ability to handle occlusions and dynamic backgrounds.
- Real-Time Applications and Edge Device Suitability
 

While YOLO-NAS is known for its efficiency in real-time applications, it was outperformed by YOLOv11 and YOLOv10 in tasks requiring more intricate feature extraction.

However, YOLO-NAS's architecture still proves effective for simpler tasks, such as face-mask detection, where it recorded a mAP50 of 0.9199. YOLOv10 and YOLOv11 offer a strong balance between accuracy and computational efficiency, making them suitable for deployment on edge devices that require fast, resource-efficient models.

- Performance on Autonomous Vehicle and Industrial Tasks

In tasks like pothole detection, which are critical for autonomous vehicles, YOLOv11 again led the models with a mAP50 of 0.815, followed by YOLOv10 at 0.793 and YOLOv9 at 0.78. The newer YOLO models consistently improved over time, showing that their architectures are more adaptable to dynamic, real-time environments. This is crucial in applications such as road safety, where detecting subtle features like potholes in various lighting and weather conditions is essential.

- Handling Complex Data: Shellfish and Smoke Detection

Complex datasets, such as shellfish monitoring and smoke detection, present unique challenges due to the overlapping and occluded objects within the images. YOLOv11 stood out in these tasks, with a mAP50 of 0.563 in shellfish monitoring and 0.945 in smoke detection. YOLOv10 also performed well, with scores of 0.542 and 0.925, respectively. In contrast, YOLOv9, YOLO-NAS, and YOLOv8 struggled more with these datasets, particularly in shellfish monitoring, where YOLOv9 only achieved a mAP50 of 0.534, and YOLO-NAS fell behind at 0.469.

In smoke detection, YOLOv8 initially performed better than YOLOv9, with a mAP50 of 0.911 compared to YOLOv9's 0.865. However, YOLOv11 surpassed both models, making it the top performer by the end of the training epochs. YOLO-NAS, while still competitive, recorded a mAP50 of 0.7811, highlighting its limitations in handling more dynamic tasks that require detailed motion analysis and fine-grained object detection.

- Training Epochs: The Importance of Extended Training

As illustrated in the mAP50 charts, extending the training epochs beyond 20 could further enhance the performance of all models, particularly YOLOv10 and YOLOv11. Both models showed a strong upward trajectory throughout the 20 epochs, suggesting that additional training could further improve their ability to handle complex object detection tasks. YOLO-NAS, while designed for efficiency, exhibited earlier flattening in its performance, indicating that its architectural limitations may prevent it from reaching the same level of accuracy in high-complexity tasks.

Therefore, the addition of YOLOv10 and YOLOv11 has shifted the performance standards in real-time object detection. While YOLO-NAS remains a strong candidate for resource-constrained environments, and YOLOv8 continues to deliver adaptable performance, YOLOv11 has emerged as the most versatile and accurate model across a range of challenging datasets. Whether in medical imaging, autonomous driving, or industrial monitoring, YOLOv11's ability to handle complex patterns, occlusions, and dynamic environments makes it the preferred choice for tasks requiring high precision and model stability.

In Table 9, we present the results of testing YOLO variants on a facemask dataset. The objective was to evaluate the performance of older YOLO versions (YOLOv5, YOLOv6, and YOLOv7) to understand their capabilities in handling relatively simple image classification tasks like facemask detection.

**Table 9.** Results of different YOLO variants on facemask data.

Model Type	mAP50
YOLOv7	0.927
YOLOv6	0.6771
YOLOv5	0.791

YOLOv7 achieved the highest mAP50 (mean average precision at 50% Intersection over Union (IoU) threshold) score of 0.927, outperforming both YOLOv6 and YOLOv5. This higher score suggests that YOLOv7 is particularly well-suited for simple object detection tasks involving clear and distinguishable objects. YOLOv7's efficiency and architectural simplicity enabled it to outperform even newer models like YOLO-NAS in tasks with fewer complexities, where lighter architectures excel.

- YOLOv7's performance highlights its capability to handle real-time detection while balancing precision, making it suitable for applications such as facemask detection.
- On the other hand, YOLOv6 and YOLOv5 underperformed relative to YOLOv7, with mAP50 scores of 0.6771 and 0.791, respectively. YOLOv6 and YOLOv5, while capable of decent performance in general object detection, require further optimization, including additional epochs of training and fine-tuning of model parameters to improve their accuracy in specialized tasks like facemask detection.

The results show that older models like YOLOv5 and YOLOv6 could still be relevant with appropriate adjustments, but for tasks that prioritize high speed and accuracy on relatively simple datasets, YOLOv7 stands out as the optimal choice.

These results underscore the importance of selecting models based on the complexity of the dataset and task. While newer models may offer advanced features for handling complex image detection, older, simpler architectures can still perform optimally when the task at hand involves more straightforward detection.

### 5.3. Discussions

#### 5.3.1. Limitations, Challenges, and Integration of YOLO with Emerging Technologies

Despite its broad applicability and success, YOLO faces several significant limitations that constrain its performance in certain scenarios. A major challenge is its difficulty in detecting small objects, particularly in cluttered or complex environments. Small objects often lack sufficient feature representation in YOLO's grid-based framework, as the resolution of feature maps tends to diminish with increasing model depth [143]. This limitation becomes critical in applications such as traffic monitoring, where distinguishing between distant vehicles and pedestrians is essential, or in medical imaging, where small anomalies like microcalcifications in mammograms can go undetected. Overlapping or occluded objects exacerbate this issue, as YOLO relies on anchor boxes and non-maximum suppression for object localization, which can lead to false negatives and overlooked detections.

Another limitation is YOLO's computational demands, which hinder its deployment on resource-constrained devices like IoT sensors, mobile platforms, and drones [144]. While lightweight variants such as YOLOv5s address this issue to some extent, they often sacrifice accuracy for efficiency, making them unsuitable for tasks that require high precision. The high computational cost also limits YOLO's usability in real-time applications where latency is a critical factor, such as autonomous vehicles and emergency response systems. Additionally, YOLO's fixed input size requirement may necessitate resizing images, potentially leading to loss of important details or distortions in object shapes, further degrading performance in scenarios where fine-grained detail is crucial.

Domain adaptability is another pressing challenge for YOLO. The model's performance tends to degrade significantly in domains with irregular or noisy data, such as low-light environments, underwater imaging, or thermal imagery [145]. These domains often present unique challenges that YOLO's conventional architecture is not optimized to address, resulting in reduced detection accuracy and robustness. For example, in underwater exploration, YOLO may struggle to distinguish objects due to variable lighting conditions, reflections, and distortions caused by water. Similarly, in medical imaging, variations in resolution, contrast, and texture can negatively impact YOLO's ability to generalize across datasets, necessitating extensive domain-specific tuning and pre-processing.

A further challenge is YOLO's vulnerability to adversarial attacks, which can manipulate the model by introducing imperceptible perturbations to input images [146]. These attacks can cause YOLO to misclassify objects, leading to potentially disastrous conse-

quences in high-stakes applications such as security, defense, or healthcare. For instance, an adversarial attack on a surveillance system powered by YOLO might result in failing to detect a weapon or misidentifying a harmless object as a threat. This vulnerability underscores the need for robust defenses, such as adversarial training, input filtering, and enhanced feature extraction mechanisms. Additionally, ethical concerns related to privacy and dataset biases present another layer of challenges. YOLO's deployment in surveillance and facial recognition systems often raises privacy issues, while biases in training datasets can result in unfair or inaccurate detections, especially in diverse populations or underrepresented environments. These challenges highlight the importance of not only technical advancements but also ethical considerations in YOLO's design and application.

### 5.3.2. Advancing YOLO Through Integration with AI Tools and Emerging Technologies

To address the challenges faced by YOLO, researchers are exploring innovative trends to enhance its adaptability and performance. Few-shot learning has emerged as a promising approach, enabling YOLO to operate effectively even with limited labeled data by leveraging transfer learning and meta-learning techniques [147]. This reduces the dependency on large-scale datasets, making YOLO more accessible for specialized applications where data scarcity is common. Another advancement is the development of dynamic detection models, which allow YOLO to adapt to evolving criteria in real-time scenarios [148]. Applications such as robotics, where conditions change rapidly, or personalized retail systems, which require tailored recommendations, greatly benefit from this adaptability. Additionally, extending YOLO to handle 3D object detection has become a critical focus area [149,150]. By incorporating technologies like LiDAR, depth maps, and volumetric analysis, YOLO is now applied in advanced fields such as autonomous vehicles and AR/VR systems [151], where spatial precision and contextual understanding are paramount [152]. These innovations not only address existing limitations but also open new opportunities for YOLO in emerging technologies and complex real-world environments.

The integration of YOLO with advanced AI tools, particularly Edge AI, has transformed the landscape of real-time object detection by significantly enhancing efficiency and functionality [153]. Leveraging hardware accelerators such as FPGAs and TPUs enables YOLO to achieve faster inference with reduced power consumption, making it ideal for deployment on resource-constrained devices such as IoT systems, drones, and mobile platforms. Lightweight YOLO models handle initial detection on edge devices, while more complex computations are offloaded to cloud infrastructure, creating a hybrid system that optimizes resource usage and minimizes latency. This architecture is particularly valuable for large-scale, real-time applications, including smart homes, wearable technologies, and industrial monitoring systems, where speed and efficiency are critical.

Integration with contextual AI frameworks, such as Vision Transformers, further enhances YOLO's ability to operate in dynamic environments [154]. This combination allows YOLO to incorporate spatial and temporal context into its predictions, improving accuracy and reliability in scenarios such as crowded public spaces or rapidly changing industrial sites. Such advancements streamline real-time analytics, making it possible to process large volumes of data for applications like traffic management, crowd monitoring, and safety systems in smart cities. By providing actionable insights derived from real-time data, this integration empowers informed decision-making and enhances operational efficiency across various sectors.

These synergistic advancements have unlocked transformational applications across numerous domains. In autonomous vehicles, YOLO integrated with Edge AI ensures reliable, real-time object detection even under resource constraints, improving safety and navigation. In agriculture, UAVs equipped with YOLO and Edge AI enable large-scale crop monitoring, identifying pests, nutrient deficiencies, and other yield-impacting factors, reducing manual interventions and optimizing efficiency. In industrial automation, these systems perform real-time defect detection on production lines, maintaining consistent quality control. Additionally, YOLO's integration with multimodal AI, such as combining visual

and auditory data, has expanded its utility to innovative areas like assistive technologies and advanced surveillance systems. Together, these advancements redefine the boundaries of intelligent detection and decision-making across diverse, real-world applications.

## 6. Ethical Considerations in the Deployment of YOLO: A Deeper Examination

The YOLO (You Only Look Once) framework, with its transformative real-time object detection capabilities, has significantly reshaped a broad range of industries. Its efficiency and speed have enabled breakthroughs in fields such as healthcare, autonomous driving, agriculture, and industrial automation. However, as with any disruptive technology, the widespread use of YOLO raises profound ethical questions. These issues extend beyond basic concerns of privacy or fairness and tap into deeper societal, philosophical, and environmental considerations. To responsibly harness YOLO's potential, we must delve into these ethical challenges with a nuanced understanding.

### 6.1. The Erosion of Privacy and the Threat of Surveillance Dystopias

As YOLO continues to enhance surveillance capabilities, we face the risk of creating a society where individuals are constantly monitored without their consent. Beyond the simple argument of privacy violations, YOLO-enabled systems threaten to embed a culture of surveillance into the fabric of daily life. The rapid pace of technological innovation is outpacing legislative and ethical frameworks, leading to a world where the omnipresent eye of cameras, drones, and smart devices can track and analyze human behavior in granular detail.

The ethical challenge here is not just about balancing security and privacy; it is about safeguarding the very concept of autonomy and freedom. In an environment where every action is logged and analyzed by real-time object detection systems, individuals may begin to self-censor or alter their behavior to avoid suspicion. This normalization of constant surveillance can lead to a dystopian future where free expression and individuality are compromised, and dissenting voices are stifled.

### 6.2. Bias and the Reinforcement of Social Inequalities

YOLO models, like many AI technologies, are only as fair as the data on which they are trained. However, the bias problem runs deeper than simple data misrepresentation. The use of YOLO in critical systems such as predictive policing, healthcare diagnostics, or hiring processes can reinforce systemic inequalities. By automating decision-making processes, we risk entrenching the biases present in historical data and perpetuating discriminatory practices.

In healthcare, for example, biased YOLO models could lead to unequal diagnostic outcomes across different racial or socioeconomic groups, where misdiagnosis or delayed detection could be life-threatening. In law enforcement, biased models might disproportionately target certain communities, leading to over-policing and further marginalization. The ethical dilemma lies in how we reconcile the tension between technological advancement and social justice. It is not enough to train YOLO on more diverse datasets; we must rigorously interrogate how the algorithms themselves make decisions, ensuring that they do not replicate or amplify existing biases.

### 6.3. Accountability in an Age of Automated Decision-Making

One of the most complex ethical issues surrounding YOLO's deployment is accountability. As these systems become increasingly autonomous, the line between human and machine responsibility blurs. When a YOLO-powered system makes a wrong decision whether in diagnosing a disease, misidentifying a pedestrian, or flagging an innocent person as a security threat who is accountable for the consequences?

The issue of accountability goes beyond merely ensuring transparency in the development of YOLO models. It extends into legal and moral territory, where developers, deployers, and users must navigate an intricate web of responsibility. If a YOLO-powered

autonomous vehicle is involved in an accident, who should be held responsible—the developer who designed the model, the organization that deployed it, or the user who trusted it? As we integrate YOLO into critical decision-making systems, we must establish clear ethical and legal frameworks for accountability, ensuring that responsibility is distributed appropriately across all stakeholders.

#### 6.4. The Ethical Dilemmas in Medical and Life-Critical Applications

YOLO's application in healthcare particularly in diagnostics, surgery, and patient monitoring holds immense promise, but it also raises high-stakes ethical questions. Errors in object detection in these domains can have life-or-death consequences. A misdiagnosis caused by an incorrect detection of a tumor or a missed abnormality in a critical scan can lead to delayed treatments or improper medical interventions.

This challenges the trust between healthcare professionals and AI systems. While YOLO can assist in increasing the accuracy of medical assessments, it should not undermine the authority and expertise of healthcare professionals. Ensuring that YOLO remains a tool that complements human judgment, rather than replaces it, is vital. The ethical debate extends to questions about the humanization of care how much reliance on automated systems is acceptable before the personal touch of medical practitioners is lost?

#### 6.5. Environmental Sustainability and the Hidden Cost of Automation

One of the overlooked ethical issues with YOLO is its environmental impact. The growing demand for AI and machine learning models has led to significant increases in energy consumption, particularly in training and deploying large-scale YOLO models. While the technology itself is seen as cutting-edge, the infrastructure required to support its deployment is resource-intensive, contributing to the carbon footprint of AI technologies.

This ethical concern goes beyond efficiency in training and into the realm of sustainability. As YOLO models are integrated into a broader range of industries, the need for high-performance computing resources grows. Data centers, GPUs, and cloud computing infrastructures essential for training large-scale models consume vast amounts of energy. Therefore, it is essential that we focus on developing greener AI technologies and optimizing YOLO variants for energy efficiency. This includes prioritizing lightweight models that maintain performance while minimizing environmental harm, as well as exploring renewable energy sources for data centers.

#### 6.6. The Social Impact of YOLO and the Displacement of Human Labor

The automation potential of YOLO extends beyond its technical capabilities and into societal concerns. As industries adopt YOLO for object detection and automation, particularly in manufacturing, agriculture, and logistics, the displacement of human labor becomes a significant ethical concern. The technology that drives efficiency in industrial settings often comes at the cost of jobs, particularly in sectors reliant on manual labor for tasks such as quality control and monitoring.

The ethical challenge here is twofold. First, there is a need to address the potential economic disruption caused by widespread automation. Policymakers, businesses, and technology developers must collaborate to create strategies for retraining and upskilling displaced workers. Second, there is the broader philosophical question about the value of human labor in an automated society. As machines take over more tasks, how do we ensure that humans are not rendered obsolete? A society that over-automates without regard for the social consequences risks creating deep divides between those who benefit from automation and those who are left behind.

#### 6.7. Ethical Frameworks for Responsible YOLO Deployment

The rapid pace of YOLO's evolution demands the parallel development of ethical frameworks that guide its responsible use. These frameworks should prioritize human dignity, fairness, privacy, and sustainability. At the core of this endeavor is the commitment

to responsible AI development, where transparency, accountability, and inclusivity are foundational principles.

Developers and organizations must adopt stringent ethical guidelines for training, deploying, and monitoring YOLO-powered systems. These guidelines should include considerations of data fairness, model interpretability, privacy protections, and energy efficiency. By embedding ethical principles into the development lifecycle, we can ensure that YOLO is deployed in ways that benefit society without compromising fundamental rights or causing harm.

## 7. Conclusions

This paper has examined the remarkable evolution of the YOLO (You Only Look Once) family of object detection models, from earlier versions like YOLOv6, YOLOv7, and YOLOv8, through to groundbreaking innovations such as YOLO-NAS, YOLOv9, and the most recent releases, YOLOv10 and YOLOv11. Each iteration has brought advancements in speed, accuracy, and computational efficiency, solidifying YOLO's role as a dominant framework in real-time object detection. YOLO's single-stage detection architecture has enabled rapid, efficient object identification across diverse and time-sensitive fields, such as healthcare, autonomous driving, agriculture, and industrial automation.

YOLO-NAS introduced a key innovation with Post-Training Quantization (PTQ), optimizing the model for resource-constrained environments without compromising accuracy. YOLOv9 further enhanced performance by introducing features like Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Networks (GELAN), allowing the model to tackle more complex detection tasks, including those with occlusions and intricate patterns.

The recent developments in YOLOv10 and YOLOv11 have further pushed the boundaries of performance. YOLOv10's C3k2 blocks and YOLOv11's Cross-Stage Partial with Spatial Attention (C2PSA) blocks significantly improved the models' ability to detect small and occluded objects while maintaining computational efficiency. In particular, YOLOv11 emerged as the most accurate and efficient model in benchmark tests, outperforming previous versions in tasks such as facemask detection, blood cell analysis, and autonomous vehicle applications.

Comprehensive benchmarks across datasets like Roboflow 100, Object365, and COCO have demonstrated the distinct advantages of YOLOv9, YOLOv10, and YOLOv11, particularly in complex object detection scenarios. Among these, YOLOv11 consistently achieved the highest performance, confirming its status as the current gold standard for real-time detection tasks in applications that require high precision, such as healthcare, environmental monitoring, and autonomous systems.

However, with these technological advancements come important ethical considerations. YOLO's ability to enable real-time object tracking and identification raises concerns about privacy and the potential for intrusive surveillance, which could threaten individual freedoms and civil liberties. Additionally, the automation of tasks previously performed by humans, especially in industries such as manufacturing and agriculture, could lead to job displacement. Therefore, the need for stringent ethical frameworks and guidelines becomes crucial to ensure that YOLO's powerful capabilities are used responsibly and for the benefit of society.

Therefore, YOLO has demonstrated itself as a highly adaptable and powerful tool for real-time object detection across multiple domains. As research continues to refine its performance, particularly through the development of lightweight and energy-efficient models, YOLO will undoubtedly remain at the forefront of object detection technology. However, ensuring its responsible and ethical deployment will be essential in maximizing its potential while safeguarding societal values.

This research provides a comprehensive review of the YOLO framework, emphasizing its evolution and transformative applications across diverse domains. By analyzing performance benchmarks for YOLO versions from YOLOv8 to YOLOv11, the study sheds

light on key technical advancements, such as improved feature aggregation and attention mechanisms, and their implications for real-time object detection. It offers domain-specific insights into YOLO's effectiveness in areas such as precision farming, healthcare diagnostics, and autonomous systems, demonstrating its adaptability and utility. Furthermore, the discussion of ethical considerations addresses critical issues like privacy, bias, and societal impact, offering a holistic perspective on the responsible deployment of YOLO models. This work serves as a resource for researchers and practitioners by synthesizing innovations and applications in the YOLO framework while pointing toward future research directions.

Despite its breadth, this review has certain limitations. The performance benchmarks rely heavily on open-source datasets, limiting validation with real-world samples. The analysis focuses primarily on YOLO's internal evolution rather than comparing it extensively with other object detection frameworks. Additionally, models were evaluated using a fixed number of training epochs, which might not fully capture their optimal performance potential. Finally, dataset biases inherent to publicly available data may influence the reported outcomes and generalizability of the results. These limitations highlight areas for further research, including extended training and validation with practical, real-life scenarios.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [[CrossRef](#)] [[PubMed](#)]
2. Vijayakumar, A.; Vairavasundaram, S. YOLO-based Object Detection Models: A Review and its Applications. *Multimed. Tools Appl.* **2024**, *83*, 83535–83574. [[CrossRef](#)]
3. Wang, X.; Li, H.; Yue, X.; Meng, L. A comprehensive survey on object detection YOLO. In Proceedings of the 5th International Symposium on Advanced Technologies and Applications in the Inter of Things (ATAIT), Kusatsu, Japan, 28–29 August 2023.
4. Soviany, P.; Ionescu, R.T. Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. *arXiv* **2018**, arXiv:1803.08707.
5. Chandana, R.; Ramachandra, A. Real time object detection system with YOLO and CNN models: A review. *arXiv* **2022**, arXiv:2208.00773.
6. Kaur, R.; Singh, S. A comprehensive review of object detection with deep learning. *Digit. Signal Process.* **2023**, *132*, 103812. [[CrossRef](#)]
7. Zhao, X.; Ni, Y.; Jia, H. Modified object detection method based on YOLO. In Proceedings of the CCF Chinese Conference on Computer Vision, Singapore, 11–14 October 2017; pp. 233–244.
8. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
9. Ragab, M.G.; Abdulkader, S.J.; Muneer, A.; Alqushaibi, A.; Sumiea, E.H.; Qureshi, R.; Al-Selwi, S.M.; Alhussian, H. A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *IEEE Access* **2024**, *12*, 57815–57836. [[CrossRef](#)]
10. Shetty, A.K.; Saha, I.; Sanghvi, R.M.; Save, S.A.; Patel, Y.J. A review: Object detection models. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–8.
11. Du, J. Understanding of object detection based on CNN family and YOLO. *J. Phys. Conf. Ser.* **2018**, *1004*, 012029. [[CrossRef](#)]
12. Zou, X. A review of object detection techniques. In Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 10–11 August 2019; pp. 251–254.
13. Sanchez, S.; Romero, H.; Morales, A. A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *844*, 012024. [[CrossRef](#)]
14. Ashraf, I.; Hur, S.; Kim, G.; Park, Y. Analyzing performance of YOLOx for detecting vehicles in bad weather conditions. *Sensors* **2024**, *24*, 522. [[CrossRef](#)]
15. Wu, S.; Li, X.; Wang, X. IoU-aware single-stage object detector for accurate localization. *Image Vis. Comput.* **2020**, *97*, 103911. [[CrossRef](#)]
16. Carranza-García, M.; Torres-Mateo, J.; Lara-Benítez, P.; García-Gutiérrez, J. On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sens.* **2020**, *13*, 89. [[CrossRef](#)]
17. Yasmine, G.; Maha, G.; Hicham, M. Overview of single-stage object detection models: From YOLOv1 to Yolov7. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 1579–1584.
18. 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*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]

19. Deng, J.; Xuan, X.; Wang, W.; Li, Z.; Yao, H.; Wang, Z. A review of research on object detection based on deep learning. *J. Phys. Conf. Ser.* **2020**, *1684*, 012028. [[CrossRef](#)]
20. Zhong, Y.; Wang, J.; Peng, J.; Zhang, L. Anchor box optimization for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 1286–1294.
21. Luo, S.; Dai, H.; Shao, L.; Ding, Y. M3dssd: Monocular 3D single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6145–6154.
22. Devi, S.; Thopalli, K.; Malarvezhi, P.; Thiagarajan, J.J. Improving single-stage object detectors for nighttime pedestrian detection. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2250034. [[CrossRef](#)]
23. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4507–4515.
24. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3DSSD: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
25. Wang, L.; Hua, S.; Zhang, C.; Yang, G.; Ren, J.; Li, J. YOLOdrive: A Lightweight Autonomous Driving Single-Stage Target Detection Approach. *IEEE Internet Things J.* **2024**, *11*, 36099–36113. [[CrossRef](#)]
26. Dai, X. HybridNet: A fast vehicle detection system for autonomous driving. *Signal Process. Image Commun.* **2019**, *70*, 79–88. [[CrossRef](#)]
27. Ortiz Castello, V.; del Tejo Catalá, O.; Salvador Igual, I.; Perez-Cortes, J.-C. Real-time on-board pedestrian detection using generic single-stage algorithms and on-road databases. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420929175. [[CrossRef](#)]
28. Ren, J.; Chen, X.; Liu, J.; Sun, W.; Pang, J.; Yan, Q.; Tai, Y.-W.; Xu, L. Accurate single stage detector using recurrent rolling convolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5420–5428.
29. Zhang, H.; Li, P.; Du, Z.; Dou, W. Risk entropy modeling of surveillance camera for public security application. *IEEE Access* **2020**, *8*, 45343–45355. [[CrossRef](#)]
30. Lu, Z.; Rathod, V.; Votet, R.; Huang, J. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14668–14678.
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I **14**; pp. 21–37.
32. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. Densebox: Unifying landmark localization with end to end object detection. *arXiv* **2015**, arXiv:1509.04874.
33. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *arXiv* **2017**, arXiv:1708.02002.
34. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
35. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
36. Jiang, Z.; Zhao, L.; Li, S.; Jia, Y. Real-time object detection method based on improved YOLOv4-tiny. *arXiv* **2020**, arXiv:2011.04244.
37. Kumar, C.; Punitha, R. Yolov3 and yolov4: Multiple object detection for surveillance applications. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 1316–1321.
38. Das, A.; Nandi, A.; Deb, I. Recent Advances in Object Detection Based on YOLO-V4 and Faster RCNN: A Review. In *Mathematical Modeling for Computer Applications*; Wiley Online Library: Hoboken, NJ, USA, 2024; pp. 405–417.
39. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
40. Mahasin, M.; Dewi, I.A. Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 backbones on YOLO v4 as object detector. *Int. J. Eng. Sci. Inf. Technol.* **2022**, *2*, 64–72. [[CrossRef](#)]
41. Chen, H.; Chen, Z.; Yu, H. Enhanced YOLOv5: An Efficient Road Object Detection Method. *Sensors* **2023**, *23*, 8355. [[CrossRef](#)] [[PubMed](#)]
42. Li, R.; Zeng, X.; Yang, S.; Li, Q.; Yan, A.; Li, D. ABYOLOv4: Improved YOLOv4 human object detection based on enhanced multi-scale feature fusion. *EURASIP J. Adv. Signal Process.* **2024**, *2024*, *6*. [[CrossRef](#)]
43. Gao, C.; Zhang, Q.; Tan, Z.; Zhao, G.; Gao, S.; Kim, E.; Shen, T. Applying optimized YOLOv8 for heritage conservation: Enhanced object detection in Jiangnan traditional private gardens. *Herit. Sci.* **2024**, *12*, 31. [[CrossRef](#)]
44. Fu, X.; Angkawisittpan, N. Detecting surface defects of heritage buildings based on deep learning. *J. Intell. Syst.* **2024**, *33*, 20230048. [[CrossRef](#)]
45. Su, X.; Zhang, J.; Ma, Z.; Dong, Y.; Zi, J.; Xu, N.; Zhang, H.; Xu, F.; Chen, F. Identification of Rare Wildlife in the Field Environment Based on the Improved YOLOv5 Model. *Remote Sens.* **2024**, *16*, 1535. [[CrossRef](#)]
46. Byeon, H. YOLO v10-Based Brain Tumor Detection: An Innovative Approach in CT Imaging. *Nanotechnol. Percept.* **2024**, *20*, 113–125.
47. Li, T.; Zhang, L.; Lin, J. Precision agriculture with YOLO-Leaf: Advanced methods for detecting apple leaf diseases. *Front. Plant Sci.* **2024**, *15*, 1452502. [[CrossRef](#)]

48. Xiao, G.; Hou, S.; Zhou, H. PCB defect detection algorithm based on CDI-YOLO. *Sci. Rep.* **2024**, *14*, 7351. [[CrossRef](#)]
49. Nguyen, H.H.; Ta, T.N.; Nguyen, N.C.; Pham, H.M.; Nguyen, D.M. Yolo based real-time human detection for smart video surveillance at the edge. In Proceedings of the 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE), Phu Quoc Island, Vietnam, 13–15 January 2021; pp. 439–444.
50. Kaur, A.; Singh, Y.; Neeru, N.; Kaur, L.; Singh, A. A Survey on Deep Learning Approaches to Medical Images and a Systematic Look up into Real-Time Object Detection. *Arch. Comput. Methods Eng.* **2022**, *29*, 2071–2111. [[CrossRef](#)]
51. Coşkun, D.; Karaboga, D.; Baştürk, A.; Akay, B.; Nalbantoglu, Ö.U.; Doğan, S.; Paçal, İ.; Karagöz, M.A. A comparative study of YOLO models and a transformer-based YOLOv5 model for mass detection in mammograms. *Turk. J. Electr. Eng. Comput. Sci.* **2023**, *31*, 1294–1313. [[CrossRef](#)]
52. Han, Z.; Huang, H.; Fan, Q.; Li, Y.; Li, Y.; Chen, X. SMD-YOLO: An efficient and lightweight detection method for mask wearing status during the COVID-19 pandemic. *Comput. Methods Programs Biomed.* **2022**, *221*, 106888. [[CrossRef](#)]
53. Vibhuti; Jindal, N.; Singh, H.; Rana, P.S. Face mask detection in COVID-19: A strategic review. *Multimed. Tools Appl.* **2022**, *81*, 40013–40042. [[CrossRef](#)] [[PubMed](#)]
54. Yu, C.J.; Yeh, H.J.; Chang, C.C.; Tang, J.H.; Kao, W.Y.; Chen, W.C.; Huang, Y.J.; Li, C.H.; Chang, W.H.; Lin, Y.T.; et al. Lightweight deep neural networks for cholelithiasis and cholecystitis detection by point-of-care ultrasound. *Comput. Methods Programs Biomed.* **2021**, *211*, 106382. [[CrossRef](#)] [[PubMed](#)]
55. Monemian, M.; Rabbani, H. Detecting red-lesions from retinal fundus images using unique morphological features. *Sci. Rep.* **2023**, *13*, 3487. [[CrossRef](#)] [[PubMed](#)]
56. Alyoubi, W.L.; Abulkhair, M.F.; Shalash, W.M. Diabetic Retinopathy Fundus Image Classification and Lesions Localization System Using Deep Learning. *Sensors* **2021**, *21*, 3704. [[CrossRef](#)] [[PubMed](#)]
57. Farahat, Z.; Zrira, N.; Souissi, N.; Benamar, S.; Belmekki, M.; Ngote, M.N.; Megdiche, K. Application of Deep Learning Methods in a Moroccan Ophthalmic Center: Analysis and Discussion. *Diagnostics* **2023**, *13*, 1694. [[CrossRef](#)] [[PubMed](#)]
58. Sobek, J.; Medina Inojosa, J.R.; Medina Inojosa, B.J.; Rassoulinejad-Mousavi, S.M.; Conte, G.M.; Lopez-Jimenez, F.; Erickson, B.J. MedYOLO: A Medical Image Object Detection Framework. *J. Imaging Inform. Med.* **2024**, *37*, 3208–3216. [[CrossRef](#)] [[PubMed](#)]
59. Vasker, N.; Sowrov, A.R.A.; Hasan, M.; Ali, M.S.; Rashid, M.R.A.; Islam, M.M. Unmasking Ovary Tumors: Real-Time Detection with YOLOv5. In Proceedings of the 2023 4th International Conference on Big Data Analytics and Practices (IBDAP), Bangkok, Thailand, 25–27 August 2023; pp. 1–6.
60. Sadeghi, M.H.; Sina, S.; Omidi, H.; Farshchitabrizi, A.H.; Alavi, M. Deep learning in ovarian cancer diagnosis: A comprehensive review of various imaging modalities. *Pol. J. Radiol.* **2024**, *89*, e30–e48. [[CrossRef](#)] [[PubMed](#)]
61. Zhang, S.; Wang, K.; Zhang, H.; Wang, T.; Gao, X.; Song, Y.; Wang, F. An improved YOLOv8 for fiber bundle segmentation in X-ray computed tomography images of 2.5D composites to build the finite element model. *Compos. Part A Appl. Sci. Manuf.* **2024**, *185*, 108337. [[CrossRef](#)]
62. Hossain, A.; Islam, M.T.; Islam, M.S.; Chowdhury, M.E.H.; Almutairi, A.F.; Razouqi, Q.A.; Misran, N. A YOLOv3 Deep Neural Network Model to Detect Brain Tumor in Portable Electromagnetic Imaging System. *IEEE Access* **2021**, *9*, 82647–82660. [[CrossRef](#)]
63. Chen, A.; Lin, D.; Gao, Q. Enhancing brain tumor detection in MRI images using YOLO-NeuroBoost model. *Front. Neurol.* **2024**, *15*, 1445882. [[CrossRef](#)] [[PubMed](#)]
64. Sara, U.; Akter, M.; Uddin, M.S. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *J. Comput. Commun.* **2019**, *7*, 8–18. [[CrossRef](#)]
65. Amin, J.; Anjum, M.A.; Sharif, M.; Kadry, S.; Nadeem, A.; Ahmad, S.F. Liver Tumor Localization Based on YOLOv3 and 3D-Semantic Segmentation Using Deep Neural Networks. *Diagnostics* **2022**, *12*, 823. [[CrossRef](#)]
66. Iriani Sapitri, A.; Nurmaini, S.; Naufal Rachmatullah, M.; Tutuko, B.; Darmawahyuni, A.; Firdaus, F.; Rini, D.P.; Islami, A. Deep learning-based real time detection for cardiac objects with fetal ultrasound video. *Inform. Med. Unlocked* **2023**, *36*, 101150. [[CrossRef](#)]
67. Baccouche, A.; Garcia-Zapirain, B.; Zheng, Y.; Elmaghraby, A.S. Early detection and classification of abnormality in prior mammograms using image-to-image translation and YOLO techniques. *Comput. Methods Programs Biomed.* **2022**, *221*, 106884. [[CrossRef](#)]
68. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustain. Cities Soc.* **2021**, *65*, 102600. [[CrossRef](#)]
69. Pang, S.; Ding, T.; Qiao, S.; Meng, F.; Wang, S.; Li, P.; Wang, X. A novel YOLOv3-arch model for identifying cholelithiasis and classifying gallstones on CT images. *PLoS ONE* **2019**, *14*, e0217647. [[CrossRef](#)] [[PubMed](#)]
70. Pham, T.-L.; Le, V.-H. Ovarian Tumors Detection and Classification from Ultrasound Images Based on YOLOv8. *J. Adv. Inf. Technol.* **2024**, *15*, 264–275. [[CrossRef](#)]
71. Safdar, M.F.; Alkobaisi, S.S.; Zahra, F.T. A Comparative Analysis of Data Augmentation Approaches for Magnetic Resonance Imaging (MRI) Scan Images of Brain Tumor. *Acta Inf. Med.* **2020**, *28*, 29–36. [[CrossRef](#)]
72. Li, S.; Li, Y.; Yao, J.; Chen, B.; Song, J.; Xue, Q.; Yang, X. Label-free classification of dead and live colonic adenocarcinoma cells based on 2D light scattering and deep learning analysis. *Cytom. Part A* **2021**, *99*, 1134–1142. [[CrossRef](#)]
73. Pandey, S.; Chen, K.-F.; Dam, E.B. Comprehensive Multimodal Segmentation in Medical Imaging: Combining YOLOv8 with SAM and HQ-SAM Models. *arXiv* **2023**, arXiv:2310.12995.

74. Priyanka; Baranwal, N.; Singh, K.N.; Singh, A.K. YOLO-based ROI selection for joint encryption and compression of medical images with reconstruction through super-resolution network. *Future Gener. Comput. Syst.* **2024**, *150*, 1–9. [CrossRef]
75. Haimer, Z.; Mateur, K.; Farhan, Y.; Madi, A.A. YOLO Algorithms Performance Comparison for Object Detection in Adverse Weather Conditions. In Proceedings of the 2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Casablanca, Morocco, 18–19 May 2023; pp. 1–7.
76. Ding, Q.; Li, P.; Yan, X.; Shi, D.; Liang, L.; Wang, W.; Xie, H.; Li, J.; Wei, M. Cf-yolo: Cross fusion yolo for object detection in adverse weather with a high-quality real snow dataset. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10749–10759. [CrossRef]
77. Lv, Z.; Wang, R.; Wang, Y.; Zhou, F.; Guo, N. Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO. *IEEE Access* **2023**, *11*, 121190–121201. [CrossRef]
78. Rothmeier, T.; Wachtel, D.; von Dem Bussche-Hünnefeld, T.; Huber, W. I had a bad day: Challenges of object detection in bad visibility conditions. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–6.
79. Guo, X. A novel Multi to Single Module for small object detection. *arXiv* **2023**, arXiv:2303.14977.
80. Zhang, Y.; Sun, Y.; Wang, Z.; Jiang, Y. YOLOv7-RAR for Urban Vehicle Detection. *Sensors* **2023**, *23*, 1801. [CrossRef] [PubMed]
81. Elesawy, A.; Mohammed Abdelkader, E.; Osman, H. A Detailed Comparative Analysis of You Only Look Once-Based Architectures for the Detection of Personal Protective Equipment on Construction Sites. *Eng* **2024**, *5*, 347–366. [CrossRef]
82. Jia, X.; Tong, Y.; Qiao, H.; Li, M.; Tong, J.; Liang, B. Fast and accurate object detector for autonomous driving based on improved YOLOv5. *Sci. Rep.* **2023**, *13*, 9711. [CrossRef] [PubMed]
83. Dazlee, N.M.A.A.; Khalil, S.A.; Abdul-Rahman, S.; Mutalib, S. Object Detection for Autonomous Vehicles with Sensor-based Technology Using YOLO. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 129–134. [CrossRef]
84. Xu, L.; Yan, W.; Ji, J. The research of a novel WOG-YOLO algorithm for autonomous driving object detection. *Sci. Rep.* **2023**, *13*, 3699. [CrossRef] [PubMed]
85. Aloufi, N.; Alnori, A.; Thayananthan, V.; Basuhail, A. Object Detection Performance Evaluation for Autonomous Vehicles in Sandy Weather Environments. *Appl. Sci.* **2023**, *13*, 10249. [CrossRef]
86. Li, Y.; Wang, J.; Huang, J.; Li, Y. Research on Deep Learning Automatic Vehicle Recognition Algorithm Based on RES-YOLO Model. *Sensors* **2022**, *22*, 3783. [CrossRef]
87. Kumar, D.; Muhammad, N. Object Detection in Adverse Weather for Autonomous Driving through Data Merging and YOLOv8. *Sensors* **2023**, *23*, 8471. [CrossRef] [PubMed]
88. Afdhal, A.; Saddam, K.; Sugiarto, S.; Fuadi, Z.; Nasaruddin, N. Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars in a Mixed Traffic Environment. In Proceedings of the 2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), Banda Aceh, Indonesia, 2–3 August 2023; pp. 260–265.
89. Al Mudawi, N.; Qureshi, A.M.; Abdelhaq, M.; Alshahrani, A.; Alazez, A.; Alonazi, M.; Algarni, A. Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences. *Sustainability* **2023**, *15*, 14597. [CrossRef]
90. Azevedo, P.; Santos, V. Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices. *Robot. Auton. Syst.* **2024**, *171*, 104558. [CrossRef]
91. He, Q.; Xu, A.; Ye, Z.; Zhou, W.; Cai, T. Object Detection Based on Lightweight YOLOX for Autonomous Driving. *Sensors* **2023**, *23*, 7596. [CrossRef] [PubMed]
92. Sarda, A.; Dixit, S.; Bhan, A. Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 1370–1374.
93. Puliti, S.; Astrup, R. Automatic detection of snow breakage at single tree level using YOLOv5 applied to UAV imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102946. [CrossRef]
94. Jemaa, H.; Bouachir, W.; Leblon, B.; Bouguila, N. Computer vision system for detecting orchard trees from UAV images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *XLIII-B4-2022*, 661–668. [CrossRef]
95. Idrissi, M.; Hussain, A.; Barua, B.; Osman, A.; Abozariba, R.; Aneiba, A.; Asyhari, T. Evaluating the Forest Ecosystem through a Semi-Autonomous Quadruped Robot and a Hexacopter UAV. *Sensors* **2022**, *22*, 5497. [CrossRef] [PubMed]
96. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
97. Bahhar, C.; Ksibi, A.; Ayadi, M.; Jamjoom, M.M.; Ullah, Z.; Soufiene, B.O.; Sakli, H. Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics* **2023**, *12*, 228. [CrossRef]
98. Hong, S.-J.; Han, Y.; Kim, S.-Y.; Lee, A.-Y.; Kim, G. Application of Deep-Learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery. *Sensors* **2019**, *19*, 1651. [CrossRef] [PubMed]
99. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2021**, *35*, 13895–13906. [CrossRef]
100. Yang, G.; Wang, J.; Nie, Z.; Yang, H.; Yu, S. A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy* **2023**, *13*, 1824. [CrossRef]
101. Qin, Z.; Wang, W.; Dammer, K.-H.; Guo, L.; Cao, Z. Ag-YOLO: A Real-Time Low-Cost Detector for Precise Spraying with Case Study of Palms. *Front. Plant Sci.* **2021**, *12*, 753603. [CrossRef] [PubMed]

102. Zhang, W.; Huang, H.; Sun, Y.; Wu, X. AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning. *Front. Plant Sci.* **2022**, *13*, 1079384. [[CrossRef](#)] [[PubMed](#)]
103. Liu, D.; Lv, F.; Guo, J.; Zhang, H.; Zhu, L. Detection of Forestry Pests Based on Improved YOLOv5 and Transfer Learning. *Forests* **2023**, *14*, 1484. [[CrossRef](#)]
104. Jayagopal, P.; Purushothaman Janaki, K.; Mohan, P.; Kondapaneni, U.B.; Periyasamy, J.; Mathivanan, S.K.; Dalu, G.T. A modified generative adversarial networks with Yolov5 for automated forest health diagnosis from aerial imagery and Tabu search algorithm. *Sci. Rep.* **2024**, *14*, 4814. [[CrossRef](#)]
105. Dandekar, Y.; Shinde, K.; Gangan, J.; Firdausi, S.; Bharne, S. Weed Plant Detection from Agricultural Field Images using YOLOv3 Algorithm. In Proceedings of the 2022 6th International Conference On Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 26–27 August 2022; pp. 1–4.
106. Dang, F.; Chen, D.; Lu, Y.; Li, Z. YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. *Comput. Electron. Agric.* **2023**, *205*, 107655. [[CrossRef](#)]
107. MacEachern, C.B.; Esau, T.J.; Schumann, A.W.; Hennessy, P.J.; Zaman, Q.U. Detection of fruit maturity stage and yield estimation in wild blueberry using deep learning convolutional neural networks. *Smart Agric. Technol.* **2023**, *3*, 100099. [[CrossRef](#)]
108. Rai, N.; Zhang, Y.; Ram, B.G.; Schumacher, L.; Yellavajjala, R.K.; Bajwa, S.; Sun, X. Applications of deep learning in precision weed management: A review. *Comput. Electron. Agric.* **2023**, *206*, 107698. [[CrossRef](#)]
109. Han, Y.; Duan, B.; Guan, R.; Yang, G.; Zhen, Z. LUFD-YOLO: A Lightweight Model for UAV Remote Sensing Forest Fire Detection Based on Attention Mechanism and Multi-Level Feature Fusion. *Remote Sens.* **2024**, *16*, 2177. [[CrossRef](#)]
110. Wu, Z.; Xue, R.; Li, H. Real-Time Video Fire Detection via Modified YOLOv5 Network Model. *Fire Technol.* **2022**, *58*, 2377–2403. [[CrossRef](#)]
111. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Comput. Electron. Agric.* **2024**, *223*, 109090. [[CrossRef](#)]
112. Qiao, Y.; Guo, Y.; He, D. Cattle body detection based on YOLOv5-ASFF for precision livestock farming. *Comput. Electron. Agric.* **2023**, *204*, 107579. [[CrossRef](#)]
113. Kaiyu, T.; Lei, N.; Bozedan, G.; Congsheng, J.; Yuanmeng, F.; Liang, T.; Zhongling, H.; Xiang, J.; Kun, J. Application of Yolo Algorithm in Livestock Counting and Identification System. In Proceedings of the 2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 15–17 December 2023; pp. 1–6.
114. Zhu, L.; Zhang, J.; Jia, C. An Improved YOLOv5-based Method for Surface Defect Detection of Steel Plate. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022; pp. 2233–2238.
115. Wang, B.; Wang, M.; Yang, J.; Luo, H. YOLOv5-CD: Strip steel surface defect detection method based on coordinate attention and a decoupled head. *Meas. Sens.* **2023**, *30*, 100909. [[CrossRef](#)]
116. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [[CrossRef](#)] [[PubMed](#)]
117. Dong, C.; Pang, C.; Li, Z.; Zeng, X.; Hu, X. PG-YOLO: A Novel Lightweight Object Detection Method for Edge Devices in Industrial Internet of Things. *IEEE Access* **2022**, *10*, 123736–123745. [[CrossRef](#)]
118. Tianjiao, L.; Hong, B. A optimized YOLO method for object detection. In Proceedings of the 2020 16th International Conference on Computational Intelligence and Security (CIS), Nanning, China, 27–30 November 2020; pp. 30–34.
119. Wang, J.; Dai, H.; Chen, T.; Liu, H.; Zhang, X.; Zhong, Q.; Lu, R. Toward surface defect detection in electronics manufacturing by an accurate and lightweight YOLO-style object detector. *Sci. Rep.* **2023**, *13*, 7062. [[CrossRef](#)]
120. Schneidereit, S.; Yarahmadi, A.M.; Schneidereit, T.; Breuß, M.; Gebauer, M. YOLO-Based Object Detection in Industry 4.0 Fischertechnik Model Environment. In *Intelligent Systems and Applications*; Arai, C.K., Ed.; Springer: Cham, Switzerland, 2024; pp. 1–20.
121. Vu, T.-T.-H.; Pham, D.-L.; Chang, T.-W. A YOLO-based Real-time Packaging Defect Detection System. *Procedia Comput. Sci.* **2023**, *217*, 886–894. [[CrossRef](#)]
122. Yu, M.; Wan, Q.; Tian, S.; Hou, Y.; Wang, Y.; Zhao, J. Equipment Identification and Localization Method Based on Improved YOLOv5s Model for Production Line. *Sensors* **2022**, *22*, 10011. [[CrossRef](#)] [[PubMed](#)]
123. Liu, Z.; Lv, H. YOLO\_Bolt: A lightweight network model for bolt detection. *Sci. Rep.* **2024**, *14*, 656. [[CrossRef](#)] [[PubMed](#)]
124. Dai, Z. Uncertainty-aware accurate insulator fault detection based on an improved YOLOX model. *Energy Rep.* **2022**, *8*, 12809–12821. [[CrossRef](#)]
125. Zendehdel, N.; Chen, H.; Leu, M.C. Real-time tool detection in smart manufacturing using You-Only-Look-Once (YOLO)v5. *Manuf. Lett.* **2023**, *35*, 1052–1059. [[CrossRef](#)]
126. Le, H.F.; Zhang, L.J.; Liu, Y.X. Surface Defect Detection of Industrial Parts Based on YOLOv5. *IEEE Access* **2022**, *10*, 130784–130794. [[CrossRef](#)]
127. Cong, P.; Lv, K.; Feng, H.; Zhou, J. Improved YOLOv3 Model for Workpiece Stud Leakage Detection. *Electronics* **2022**, *11*, 3430. [[CrossRef](#)]
128. Xu, X.; Zhang, X.; Zhang, T. Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images. *Remote Sens.* **2022**, *14*, 1018. [[CrossRef](#)]

129. Nishant, B.; Gill, S.S.; Raj, B. Improving Quality Assurance: Automated Defect Detection in Soap Bar Packaging Using YOLO-V5. In Proceedings of the 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM), Roorkee, India, 26–27 August 2023; pp. 1–6.
130. Tran, V.T.; To, T.S.; Nguyen, T.-N.; Tran, T.D. Safety Helmet Detection at Construction Sites Using YOLOv5 and YOLOR. In *Intelligence of Things: Technologies and Applications*; Nguyen, N.-T., Dao, N.-N., Pham, Q.-D., Le, H.A., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 339–347.
131. Chen, Y.; Liu, H.; Chen, J.; Hu, J.; Zheng, E. Insu-YOLO: An Insulator Defect Detection Algorithm Based on Multiscale Feature Fusion. *Electronics* **2023**, *12*, 3210. [[CrossRef](#)]
132. Wang, C.-Y.; Yeh, I.H.; Hong, Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
133. Ultralytics. YOLO-NAS. 2023. Available online: <https://docs.ultralytics.com/models/yolo-nas/#which-tasks-and-modes-are-supported-by-yolo-nas-models> (accessed on 15 October 2024).
134. Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; Gao, W. Post-training quantization for vision transformer. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28092–28103.
135. Chu, X.; Li, L.; Zhang, B. Make RepVGG Greater Again: A Quantization-aware Approach. *arXiv* **2022**, arXiv:2212.01593. [[CrossRef](#)]
136. Wang, C.-Y.; Hong, Y.; Yeh, I.H.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *arXiv* **2019**, arXiv:1911.11929.
137. Zhang, X.; Zeng, H.; Guo, S.; Zhang, L. Efficient Long-Range Attention Network for Image Super-resolution. *arXiv* **2022**, arXiv:2203.06697.
138. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
139. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
140. Ciaglia, F.; Zuppichini, F.S.; Guerrie, P.; McQuade, M.; Solawetz, J. Roboflow 100: A rich, multi-domain object detection benchmark. *arXiv* **2022**, arXiv:2211.13523.
141. Shao, S.; Li, Z.; Zhang, T.; Peng, C.; Yu, G.; Zhang, X.; Li, J.; Sun, J. Objects365: A large-scale, high-quality dataset for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8430–8439.
142. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V *13*. pp. 740–755.
143. Mittal, P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artif. Intell. Rev.* **2024**, *57*, 242. [[CrossRef](#)]
144. Hussain, M. Yolov1 to v8: Unveiling each variant—a comprehensive review of yolo. *IEEE Access* **2024**, *12*, 42816–42833. [[CrossRef](#)]
145. Wang, J.; Yang, P.; Liu, Y.; Shang, D.; Hui, X.; Song, J.; Chen, X. Research on improved yolov5 for low-light environment object detection. *Electronics* **2023**, *12*, 3089. [[CrossRef](#)]
146. Kazmi, S.M.K.A.; Aafaq, N.; Khan, M.A.; Khalil, M.; Saleem, A. From Pixel to Peril: Investigating Adversarial Attacks on Aerial Imagery Through Comprehensive Review and Prospective Trajectories. *IEEE Access* **2023**, *11*, 81256–81278. [[CrossRef](#)]
147. Wen, Y.; Wang, L. Yolo-sd: Simulated feature fusion for few-shot industrial defect detection based on YOLOv8 and stable diffusion. *Int. J. Mach. Learn. Cybern.* **2024**, *15*, 4589–4601. [[CrossRef](#)]
148. Chen, J.; Er, M.J. Dynamic YOLO for small underwater object detection. *Artif. Intell. Rev.* **2024**, *57*, 165. [[CrossRef](#)]
149. Liu, L.; Song, X.; Song, H.; Sun, S.; Han, X.-F.; Akhtar, N.; Mian, A. Yolo-3DMM for Simultaneous Multiple Object Detection and Tracking in Traffic Scenarios. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9467–9481. [[CrossRef](#)]
150. El Ghazouali, S.; Mhirit, Y.; Oukhrad, A.; Michelucci, U.; Nouira, H. FusionVision: A comprehensive approach of 3D object reconstruction and segmentation from RGB-D cameras using YOLO and fast segment anything. *Sensors* **2024**, *24*, 2889. [[CrossRef](#)] [[PubMed](#)]
151. Ali, M.L.; Zhang, Z. Natural Human-Computer Interface Based on Gesture Recognition with YOLO to Enhance Virtual Lab Users' Immersive Feeling. In Proceedings of the 2024 ASEE Annual Conference & Exposition, Portland, OR, USA, 23–26 June 2024.
152. Talha, S.A.; Manasreh, D.; Nazzal, M.D. The Use of Lidar and Artificial Intelligence Algorithms for Detection and Size Estimation of Potholes. *Buildings* **2024**, *14*, 1078. [[CrossRef](#)]
153. Al Amin, R.; Hasan, M.; Wiese, V.; Obermaisser, R. FPGA-based Real-Time Object Detection and Classification System using YOLO for Edge Computing. *IEEE Access* **2024**, *12*, 73268–73278. [[CrossRef](#)]
154. Xue, C.; Xia, Y.; Wu, M.; Chen, Z.; Cheng, F.; Yun, L. EL-YOLO: An efficient and lightweight low-altitude aerial objects detector for onboard applications. *Expert Syst. Appl.* **2024**, *256*, 124848. [[CrossRef](#)]