YOLOv12 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series

Ranjan Sapkota
Cornell University
45 PUBLICATIONS 236 CITATIONS

Rizwan Qureshi
University of Texas MD Anderson Cancer Center
133 PUBLICATIONS 1,895 CITATIONS

Marco Flores-Calero
56 PUBLICATIONS 355 CITATIONS

Chetan Badgujar
University of Tennessee at Knoxville
30 PUBLICATIONS 274 CITATIONS

YOLOv12 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series

Ranjan Sapkota*
Biological & Environmental Engineering
Cornell University
Ithaca, New York, USA
rs2672@cornell.edu

Rizwan Qureshi
Center for Research in Computer Vision
The University of Central Florida
Orlando, FL, USA

Marco Flores-Calero
Department of Electrical, Electronics and Telecommunications
Universidad de las Fuerzas Armadas
Sangolquí, Ecuador

Chetan Badgujar
Biosystems Engineering and Soil Sciences
The University of Tennessee
Knoxville, TN, USA

Upesh Nepal
Cooper Machine Company, Inc.
Wadley, GA, USA

Alwin Poulose
School of Data Science
Indian Institute of Science Education and Research Thiruvananthapuram (IISER TVM)
Thiruvananthapuram, Kerala, India

Peter Zeno
ZenoRobotics, LLC
Billings, MT, USA

Uday Bhanu Prakash Vaddevolu
Biological and Agricultural Engineering
Texas A&M University
College Station, TX, USA

Sheheryar Khan
School of Professional Education and Executive Development
The Hong Kong Polytechnic University
Hong Kong, SAR China

Maged Shoman
University of Tennessee
Knoxville, TN, USA

Hong Yan
Department of Electrical Engineering
City University of Hong Kong
Center for Intelligent Multidimensional Data Analysis (CIMDA)
Kowloon, Hong Kong, SAR China

Manoj Karkee*
Biological & Environmental Engineering
Cornell University
Ithaca, New York, USA
mk2684@cornell.edu

***Abstract*—This review systematically examines the progression of the You Only Look Once (YOLO) object detection algorithms from YOLOv1 to the recently unveiled YOLOv12. Employing a reverse chronological analysis, this study examines the advancements introduced by YOLO algorithms, beginning with YOLOv12 and progressing through YOLO11 (or YOLOv11), YOLOv10, YOLOv9, YOLOv8, and subsequent versions to explore each version's contributions to enhancing speed, detection accuracy, and computational efficiency in real-time object detection. Additionally, this study reviews the alternative versions derived from YOLO architectural advancements of YOLO-NAS, YOLO-X, YOLO-R, DAMO-YOLO, and Gold-YOLO. By detailing the incremental technological advancements in subsequent YOLO versions, this review chronicles the evolution of YOLO, and discusses the challenges and limitations in each of the earlier versions. The evolution signifies a path towards integrating YOLO with multimodal, context-aware, and Artificial General Intelligence (AGI) systems for the next YOLO decade, promising significant implications for future developments in AI-driven applications.**

***Index Terms*—You Only Look Once, YOLO, YOLOv1 to YOLOv12, YOLOv11, YOLOv10, YOLO configurations, CNN, Deep learning, Real-time object detection, Artificial intelligence, Computer vision, Healthcare and Medical Imaging, Autonomous vehicles, Traffic safety, Industrial manufacturing, Surveillance, Agriculture**

I. Introduction

Object detection is a critical component of computer vision systems, which enables automated systems to identify and locate objects of interest within images or video frames [1]–[5]. Real-time object detection has become integral to numerous applications requiring real- and near-real-time analysis, monitoring and interaction with dynamic environments such as agriculture, transportation, education, and health-care [6]–[11]. For instance, real-time object detection is the foundational technology for the success of autonomous vehicles and robotic systems [12]–[14], allowing the system to quickly recognize and track different objects of interests such as vehicles, pedestrians, bicycles, and other obstacles, enhancing navigational safety and efficiency [15], [16]. The utility of object recognition extends beyond vehicular applications, and is also pivotal in action recognition within video sequences, useful in digital surveillance, monitoring, sports analysis, cityscapes [17] and human-machine interaction [6], [18], [19]. These areas benefit from the capability to analyze and respond to situational dynamics in real-time, illustrating its broad applicability, acceptance, and impact. However, the problem of object detection involves several challenges:

- Complexity of Real-World Environments: Real-world environments/scenes are highly variable and unpredictable. Objects can appear in various orientations, scales, distances and lighting conditions, making it difficult for a detection algorithm to generalize and maintain accuracy in real time [20].
- Illumination Factors: Illumination plays a crucial role in object detection, as factors like lighting intensity, direction, shadows, and glare can significantly affect performance [21], [22]. Non-uniform or low light, color temperature changes, and dynamic lighting variations can obscure object features or cause false detections. Solutions include controlled lighting setups, preprocessing techniques like normalization and color correction, and training models with diverse, augmented datasets to enhance robustness [23].
- Occlusions and Clutter: Objects may be partially or fully obscured by other objects, creating cluttered scenes that result in incomplete information, which requires careful interpretation for accurate analysis [24], [25].
- Speed and Efficiency: Many applications necessitate rapid processing of visual data to enable timely decision-making. This requires detection algorithms to achieve a balance between high accuracy and low latency, ensuring that the systems can deliver efficient and reliable results in real- or near-real-time scenarios, such as autonomous vehicles and traffic safety, healthcare and medical imaging, industrial manufacturing, security and surveillance and agricultural automation [26].

Addressing these challenges required innovative techniques, which initially relied on hand-crafted features and classical machine learning methods. Later, the focus shifted towards automated feature learning, and end-to-end deep learning methods.

A. *Traditional Object Detection Approaches*

Before the advent of deep learning, object detection relied on a combination of hand-crafted features and machine learning classifiers [27]. Some of the notable traditional methods include:

- **Correlation Filters:** Used to detect objects by correlating a filter with the image, such as matching templates [28]. These approaches struggle with variations in the appearance of objects and lighting conditions [29].
- **Sliding Window Approach**: This method involves moving a fixed-size window across the image and applying a classifier to each window to determine whether it contains an object [30]. However, it struggles with varying object sizes and aspect ratios, which can lead to inaccurate detections and a high computational cost due to the exhaustive search involved.
- **Viola-Jones Detector**: The Viola-Jones detector, introduced in 2001, uses Haar-like features [31] and a cascade of AdaBoost trained classifiers [32] to detect objects in images efficiently [33].

Supporting these methods are various hand-crafted feature extraction techniques, including:

- **Gabor Features:** Extracted texture features using Gabor filters, which are effective for texture representation but computationally intensive [34].
- **Histogram of Oriented Gradients (HOG):** Captures edge or gradient structures that characterize the shape of objects, typically combined with Support Vector Machines (SVM) for classification [35].
- **Local Binary Patterns (LBP):** Utilizes pixel intensity comparisons to form a binary pattern, used in texture classification and face recognition [36].
- **Haar-like Features**: These features consider adjacent rectangular regions in a detection window, sum up the pixel intensities in each region, and calculate the difference between these sums. This difference is then used to categorize subsections of an image [37].
- **Deformable Part Models (DPM):** DPM [38] represents objects as a collection of deformable parts arranged in a spatial structure. It proved particularly effective for detecting objects under occlusions, pose variations, and cluttered backgrounds.
- **Scale-Invariant Feature Transform (SIFT):** SIFT is a robust method for detecting and describing local features in images [39]. Beyond feature extraction, it has been effectively used for object detection by matching keypoints between input images and reference templates, leveraging its invariance to scale, rotation, and illumination changes.
- **Speeded-Up Robust Features (SURF):** A faster alternative to SIFT, SURF detects and describes features using an efficient Hessian matrix approximation [40], making it suitable for real-time object detection tasks [41] .

1) Classification Methods: Some of the most commonly employed classification methods for these detectors include Support Vector Machine (SVM), statistical classifiers (e.g., Bayesian Classifier) and ensemble methods (e.g. Adaboost, Random Forest) and Multilayer Perceptrons (MLP) Neural Networks [42], [43]. These traditional methods in early computer vision, reliant on hand-crafted features and classical classifiers, offered moderate success under controlled conditions but struggled with robustness and generalization in diverse real-world scenarios, lacking the accuracy achieved by modern deep learning techniques [44]. Figure 1 shows the historical development of computer vision systems emphasizing how object detection algorithms evolved.

B. Emergence of Convolutional Neural Networks

After 2010, the performance of handcrafted features plateaued, leading to saturation in object detection research. However, in 2012, the world witnessed the birth of convolutional neural networks (CNNs), marking a significant turning point in the field [45]–[48]. As a deep convolutional network, CNNs are able to learn robust and high-level feature representations of an image, and are particularly effective because:

- **Hierarchical Feature Learning:** CNNs learn to extract low-level features (e.g., edges, textures) in early layers and high-level features (e.g., object parts, shapes) in deeper layers, facilitating robust object representation [49].
- **Spatial Invariance:** Convolutional layers enable CNNs to recognize objects regardless of their position within the image, enhancing detection robustness [50].
- **Scalability and Generalizability**: CNNs can be scaled to handle larger datasets and more complex models, improving performance and robustness on a wide range of tasks and application environments [51].

However, CNNs can not be directly applied to the object detection task, due to varying number of objects, varying sizes, aspect ratios, and orientation. CNNs were primarily designed for image classification, meaning they output a single label for the entire image [5]. Whereas object detection tasks require not only classifying the object but also localizing it in the image, i.e., identifying the position of the object through bounding boxes.

C. Timeline of Object Detection Paradigms and the Evolution of YOLO Models

One of the earliest deep learning-based object detectors was R-CNN, introduced in 2014 by Girshick et al. [54]. It marked a pivotal milestone in the development of detection models, breaking the stagnation in object detection by introducing Regions with CNN features (R-CNN). This groundbreaking approach revolutionized the field, sparking rapid advancements and accelerating the evolution of object detection at an unprecedented pace.

The idea behind R-CNN is simple, it uses the selective search algorithm to generate about 2000 region proposals, which are then processed by a CNN to extract features [54]. Finally, linear SVM classifiers are utilized to detect objects within each region and identify their respective categories. While R-CNN achieved significant progress, it has notable drawbacks: the redundant feature computations across a large number of overlapping proposals (over 2,000 boxes per image) result in extremely slow detection speeds, taking 14 seconds per image even with GPU acceleration [46].

After that, Fast R-CNN, introduced in 2015, improved object detection by addressing the redundant feature computations across numerous overlapping proposals [55]. It integrated region proposal feature extraction and classification into a single pass, significantly enhancing efficiency and speed compared to previous methods like R-CNN [56]. Building on this, Faster R-CNN advanced the approach further by introducing Region Proposal Networks (RPNs), enabling end-to-end training. This innovation eliminated the reliance on selective search, reducing computational complexity and streamlining the pipeline, thus allowing for faster and more accurate object detection without the need for external proposal generation [8], [25].

Later, the Single Shot Multibox Detector (SSD) [57], introduced in 2016, discretizes bounding boxes into predefined default boxes of various scales and aspect ratios. It predicts object scores and adjusts box shapes accordingly. By leveraging multi-scale feature maps, SSD handles objects of different sizes effectively. Unlike earlier methods, it eliminates the need for a separate proposal generation step, simplifying the training process and improving performance, particularly in detecting small objects.

These breakthroughs laid the foundation for numerous subsequent advancements in the field. Over the years, several other detectors have been developed, and by 2024, the YOLO series has become a dominant force in the domain, continuously evolving to improve speed, accuracy, and efficiency in real-time object detection tasks.

Figure 1 presents a chronological overview of deep learning-based detectors, categorized into YOLO and others. Traditional detectors, shown on the right side, as well as Transformer-based detectors, are included for visual comparison.

D. You Only Look Once Approach

The ”You Only Look Once” (YOLO) object detection algorithm was first introduced by Joseph Redmon et al., [58] in 2015, revolutionized real-time object detection by combining region proposal and classification into a single neural network, significantly reducing computation time. YOLO's unified architecture divides the image into a grid, predicting bounding boxes and class probabilities directly for each cell, enabling end-to-end learning [58]. YOLOv1 utilized a simplified CNN backbone, setting the stage with basic bounding box predictions. Subsequent versions like YOLOv2 incorporated the DarkNet-19 backbone and refined anchor boxes through K-means clustering. YOLOv3 expanded this further with a DarkNet-53 architecture, integrating multi-scale detection and residual connections. The series continued to

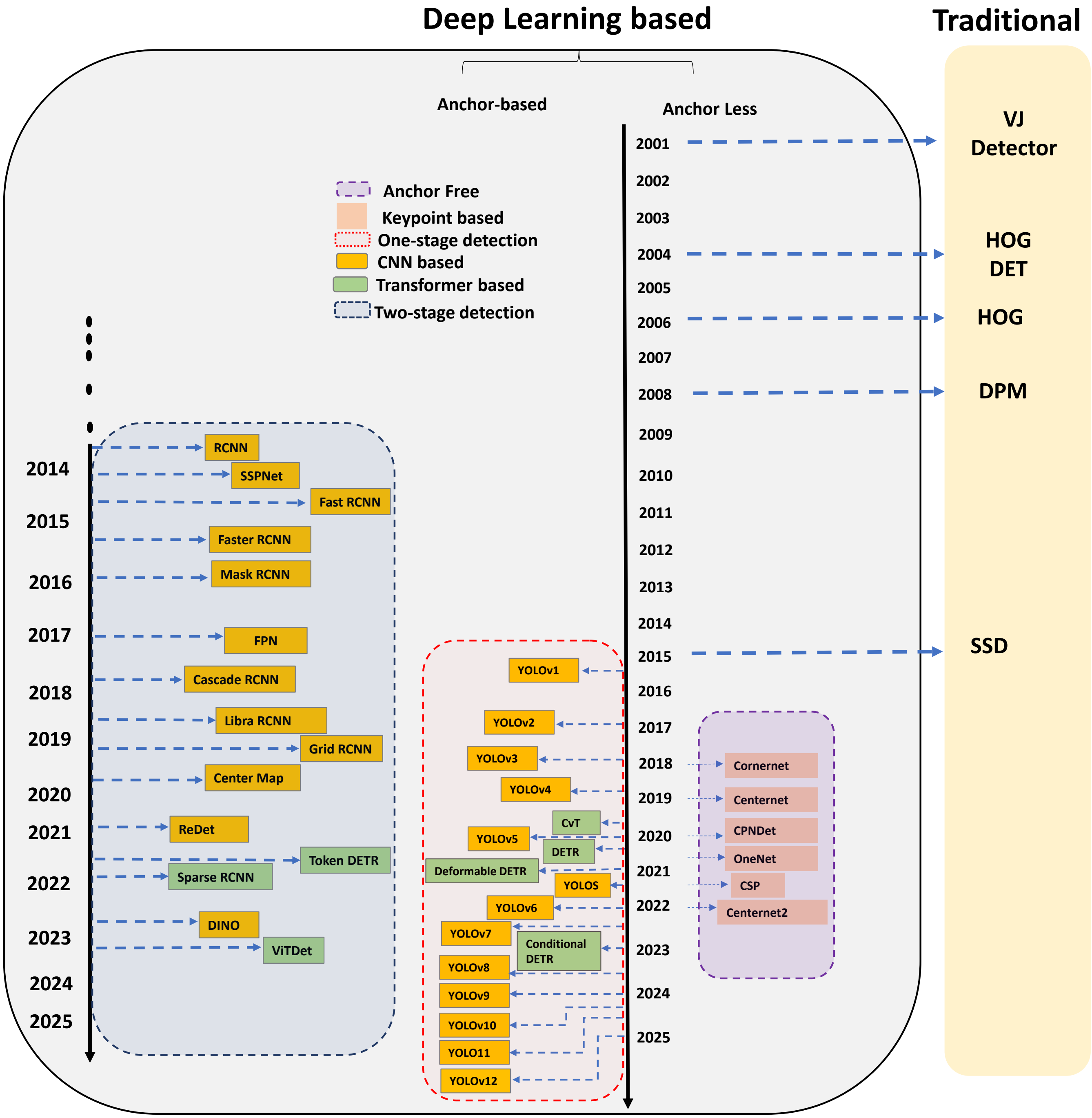


Fig. 1: **Timeline of object detection paradigms and evolution of YOLO models. The figure shows the progression from traditional methods like VJ Detector and HOG to deep learning-based approaches, including R-CNN, Fast R-CNN, Faster R-CNN, and YOLO series. Recent advancements also highlight transformer-based models, such as DETR (Detection Transformer) [52] and ViTDet (Vision Transformer for Detection) [53], which have demonstrated significant progress in object detection tasks.**

innovate with YOLOv4 implementing CSPDarkNet-53 and PANet alongside mosaic data augmentation. YOLOv5 and YOLOv6 introduced CSPNet with dynamic anchor refinement and further enhancements in PANet, respectively. YOLOv7 featured an EfficientRep backbone with dynamic label assignment, while YOLOv8 introduced a Path Aggregation Network with Dynamic Kernel Attention. YOLOv9 developed multi-level auxiliary feature extraction, and YOLOv10 optimized the system with a lightweight classification head and distinct spatial and channel transformations. YOLOv11 introduced the C3k2 block in its backbone and utilized C2PSA for improved spatial attention. The latest, YOLOv12, marks a significant shift towards an attention-centric design, introducing the Area Attention (A^2) module for efficient large receptive field processing, Residual Efficient Layer Aggregation Networks (R-ELAN) for enhanced feature aggregation, and architectural optimizations including FlashAttention and adjusted MLP ratios. This attention-based approach allows YOLOv12 to achieve state-of-the-art performance in both accuracy and efficiency, surpassing previous CNN-based models while maintaining real-time detection capabilities [59]. autonomous vehicles and traffic safety [4], [16], [60], [61], healthcare [62]–[64], industrial applications [60], [65], [66], surveillance and security [67], [68] and agriculture [69]–[79], where accuracy and speed are crucial.

E. Motivation and Organization of the Study

Since ”You Only Look Once” has been widely adopted in the field of computer vision, a search for this keyword in Google Scholar yields approximately 5,550,000 results as of June 9, 2024. The acronym ”YOLO” further emphasizes its popularity, generating around 210,000 search results at the same time instant. Thousands of researchers have cited YOLO papers, highlighting its significant influence. This study aims to review and critically summarize the YOLO's decadal progress and its advancements over time, as visually summarized in the mind-map, shown in Figure 2.

This comprehensive analysis starts with Section 2: **YOLO Trajectory**, tracing the evolution from YOLOv1 to YOLOv11. In Section 3: **Context and Distinctions of Prior YOLO Literature** offers insights into the background and unique aspects of earlier studies. Section 4: **Review of YOLO Versions** details the key features and improvements of each version. In Section 5: **Applications** various use cases across different domains are highlighted, demonstrating the versatility of YOLO models. Following this, section 6 **Challenges, Limitations and Future Directions** addresses current issues and potential advancements. Finally, the **Conclusion** section summarizes the findings of this comprehensive review. Each section is further divided into various sub-subsections to present and discuss specific topic areas relevant to the corresponding sections.

II. The Evolution of YOLO: Trajectory and Variants

Figure 1 illustrates the development timeline of the YOLO models, beginning with the release of YOLOv1 and progressing through to the latest version, YOLOv11. This timeline highlights the key advancements and iterations in the YOLO series.

YOLOv1 [58] was introduced in 2016 as a novel approach to object detection, offering good accuracy and computational speed by processing images using a single stage network architecture. The first YOLO version laid the foundation for real-time applications of machine vision systems, setting a new standard for subsequent developments.

YOLOv2, or YOLO9000 [80], [81], expanded on the foundation of YOLOv1 by improving the resolution at which the model operated and by expanding the capability to detect over 9000 object categories, thus enhancing its versatility and accuracy. YOLOv2 introduced two primary variants: a smaller version optimized for speed and a larger version focused on higher accuracy.

YOLOv3 further advanced these capabilities by implementing multi-scale predictions and a deeper network architecture, which allowed better detection of smaller objects [82]. YOLOv3 introduced three primary variants, each designed to balance model size and performance: YOLOv3-spp (Small), Standard, and YOLOv3-tiny(Tiny), catering to different trade-offs between speed and accuracy.

The series continued to evolve with YOLOv4 and YOLOv5, each introducing more refined techniques and optimizations to improve detection performance (i.e., accuracy and speed) even further [83]–[85]. YOLOv4 introduced four main variants: the standard version, YOLOv4-CSP, which incorporates Cross-Stage Partial (CSP) networks to enhance performance and reduce computational cost; YOLOv4x-mish, which utilizes the Mish activation function to improve accuracy while maintaining efficiency; and YOLOv4-tiny, a lightweight version optimized for real-time applications and edge devices, sacrificing some accuracy for speed. YOLOv5, developed by Ultralytics [86] [1], brought significant improvements in terms of ease of use and performance, establishing itself as a popular choice in the computer vision community. YOLOv5 introduced five primary variants to meet various performance needs: YOLOv5s (small), optimized for speed and efficiency in resource-constrained environments; YOLOv5m (medium), offering a balanced trade-off between speed and accuracy; YOLOv5l (large), designed for higher accuracy at the expense of resources; YOLOv5x (extra-large), focused on top-tier accuracy for powerful hardware; and YOLOv5n (nano), a lightweight version tailored for rapid inference and low computational demands, ideal for real-time applications and edge devices.

Subsequent versions, YOLOv6 through YOLO11, have continued to build on this success, focusing on enhancing model scalability, reducing computational demands, and improving real-time performance metrics.

Li et al. [87] introduced YOLOv6 in 2022. Developed by a team from Meituan, a Chinese e-commerce platform, YOLOv6 features a novel backbone and neck architecture. It also incorporates advanced training techniques such as Anchor-Aided Training (AAT) and Self-Distillation to enhance performance and efficiency. YOLOv6 introduces three main variants: the standard version, balancing accuracy and speed for general detection tasks; YOLOv6-Nano, optimized for real-time applications with a focus on speed and performance on

[1] Ultralytics specializes in AI and deep learning, known for YOLOv5 and YOLOv8 models used in object detection, segmentation, and classification for computer vision tasks.

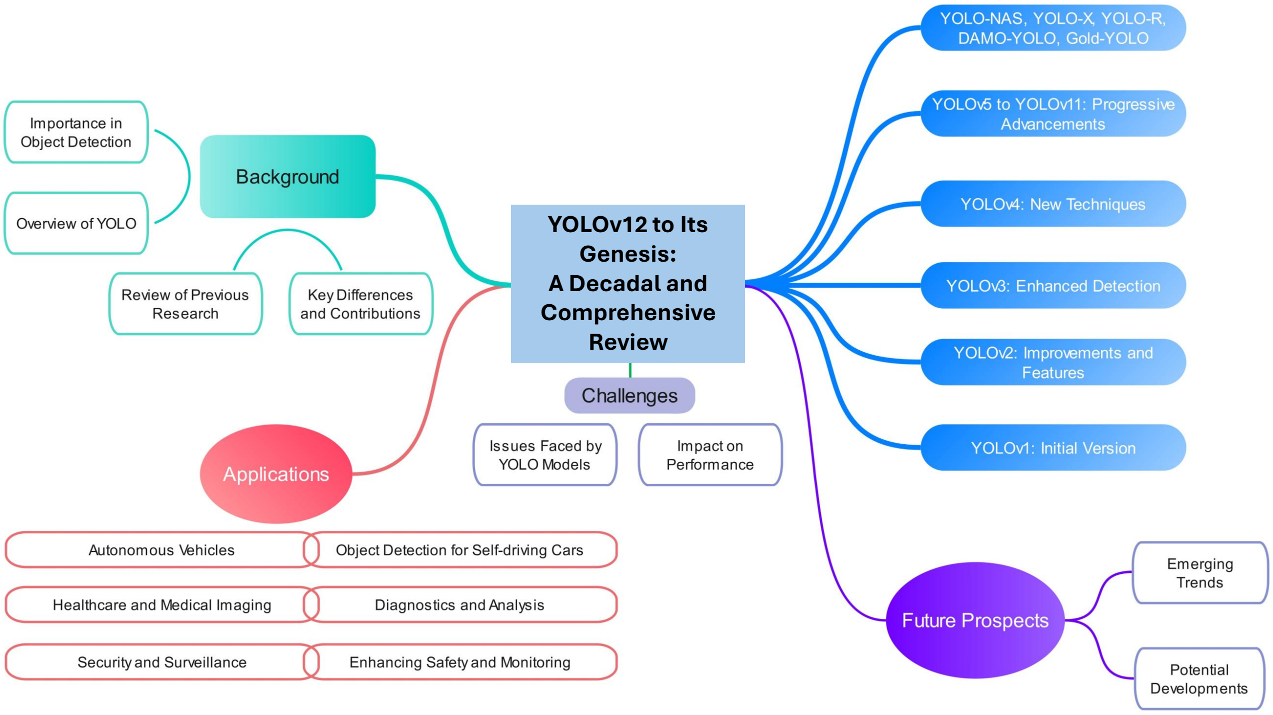

YOLOv12 to Its Genesis: A Decadal and Comprehensive Review
Background
Importance in Object Detection
Overview of YOLO
Review of Previous Research
Key Differences and Contributions
Applications
Autonomous Vehicles
Object Detection for Self-driving Cars
Healthcare and Medical Imaging
Diagnostics and Analysis
Security and Surveillance
Enhancing Safety and Monitoring
Challenges
Issues Faced by YOLO Models
Impact on Performance
YOLO-NAS, YOLO-X, YOLO-R, DAMO-YOLO, Gold-YOLO
YOLOv5 to YOLOv11: Progressive Advancements
YOLOv4: New Techniques
YOLOv3: Enhanced Detection
YOLOv2: Improvements and Features
YOLOv1: Initial Version
Future Prospects
Emerging Trends
Potential Developments

Fig. 2: **A schematic of the manuscript. We discuss all versions of YOLO, with comparative analysis. Applications in key areas; such as, autonomous vehicles, and security and surveillance are also presented. Challenges in each YOLO version, with performance enhancements, are also highlighted, we also provide visionary thoughts on the future impact of YOLO on industry and society.**

edge devices; and YOLOv6-Tiny, designed for even faster inference on low-resource hardware, trading off some accuracy.

YOLOv7 [88], [89] introduces advanced techniques like trainable bag-of-freebies (optimizations that improve accuracy without increasing inference cost) and dynamic label assignment. It introduces three variants: the standard version, balancing speed and accuracy; YOLOv7-X, a more powerful variant optimized for performance but requiring more computational resources; and YOLOv7-Tiny, a lightweight version designed for real-time applications on edge devices, prioritizing speed over accuracy.

YOLOv8 released in 2023 by Ultralytics [90]. It features a more efficient architecture, enhanced training techniques, and support for larger datasets. Its user-friendly implementation in PyTorch makes it accessible for both research and production. YOLOv8 introduces four variants: YOLOv8-S, optimized for fast inference on edge devices with some accuracy trade-offs; YOLOv8-M, balancing accuracy and speed for general tasks; YOLOv8-L, prioritizing accuracy at the cost of computational demand; and YOLOv8-Tiny, a lightweight version for real-time applications.

YOLOv9 [91] proposed the concept of programmable gradient information (PGI) to cope with the various changes required by deep networks to achieve multiple objectives. PGI can provide complete input information for the target task to calculate objective function, so that reliable gradient information can be obtained to update network weights. In addition, a new lightweight network architecture – Generalized Efficient Layer Aggregation Network (GELAN), based on gradient path planning is designed. GELAN's architecture confirms that PGI has gained superior results on lightweight models. We verified the proposed GELAN and PGI on MS COCO dataset based object detection. Its variants are YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, YOLOv9e [92].

YOLOv10 [93], developed by researchers at Tsinghua University, introduces a novel approach to real-time object detection, addressing the limitations of both post-processing and model architecture in previous YOLO versions. By eliminating non-maximum suppression (NMS) and optimizing key components of the model, YOLOv10 offers significant improvements in efficiency and performance. As the penultimate version in the YOLO series, it introduces six distinct variants as YOLOv10-N, YOLOv10-S, YOLOv10-M, YOLOv10-B, YOLOv10-L, and YOLOv10-X [94] Notably, YOLOv10-N and YOLOv10-S exhibit the lowest latencies at 1.84 ms and 2.49 ms, respectively, making them highly suitable for applications requiring low latency. These models outperform their predecessors, with YOLOv10-X achieving the highest mAP of 54.4% and a latency of 10.70 ms, reflecting a well-balanced enhancement in both accuracy and inference speed.

YOLOv11 [95] represents the latest advancement in the YOLO series of algorithms, marking a significant step forward in object detection technology. This version introduces a highly sophisticated backbone and neck architecture designed to extract features with exceptional accuracy. Additionally, it incorporates optimized designs and streamlined training pipelines, resulting in notable improvements in speed, efficiency, and overall performance. One of the standout features of YOLOv11 is its ability to balance precision and computational efficiency, making it suitable for a wide range of applications, from embedded systems to large-scale deployments. By addressing the limitations of its predecessors, YOLOv11 not only enhances detection accuracy but also ensures real-time processing capabilities under diverse and challenging conditions. Additionally, the YOLOv11 model can be classified into five variants—YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11L, and YOLOv11x—based on the depth of the network.

Each iteration of the YOLO series has set new benchmarks for object detection capabilities and significantly impacted various application areas, from autonomous vehicles and traffic safety to healthcare, industrial automation and smart farming.

A. Significance of Latency and mAP Scores in YOLO

Inference Time (T_{inf}) and mean Average Precision (mAP) are critical metrics to assess the performance of object detection models such as YOLO [47], [96]. Inference Time specifically measures the duration required for the model to process an image and generate predictions, focusing solely on the computational phase and is typically measured in milliseconds (ms) [96]. This metric excludes any delays from image preprocessing or post-processing, providing a clear measure of the model's computational efficiency. Lower inference times are crucial for real-time applications such as autonomous driving, surveillance, and robotics, where rapid and accurate detections are essential [97]. High inference times can lead to delays that compromise safety and effectiveness in these dynamic settings [98].

Frames Per Second (FPS) is another essential metric that indicates how many images the model can evaluate each second, complementing inference time by illustrating the model's ability to handle streaming video or rapid image sequences. Both inference time and FPS provide a detailed view of the real-time operational performance of a model.

It is also important to note that these performance metrics are highly dependent on the hardware platform used for testing. Differences in computational power can significantly influence results, which makes it essential to standardize hardware during benchmark tests to ensure fair comparisons. Likewise, mAP is a comprehensive metric used to evaluate the accuracy of object detection models [99]. It considers both precision and recall (Table I), and it is calculated by taking the average precision (AP) across all classes and then averaging these AP scores [99], [100]. It provides a balanced view of how well the model performs across different object categories and varying conditions within the dataset. Other metrics used for comprehensive evaluation of YOLO models [101], [102] are detailed in Table I.

Here, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are the key performance evaluators. TP is instance where the model correctly identifies an object as present. TN occurs when the model correctly predicts the absence of an object. FP arises when the model incorrectly identifies an object as present, and FN happens when the model fails to detect an object that is actually present. These metrics are crucial for assessing the accuracy and reliability of the YOLO object detection [99], [100], [102].

B. Single-stage detection with YOLO

The Single Shot MultiBox Detector (SSD) [57] introduced in 2016 revolutionized object detection by streamlining the process through a single-stage approach, significantly inspiring subsequent developments in YOLO models [57], [103], [104]. Unlike two-stage models like R-CNN, which rely on a region proposal step before actual object detection, SSD and by extension, YOLO variants, perform detection and classification in a single sweep across the image. This paradigm shift enhances the detection process by eliminating intermediate steps, thus facilitating faster and more efficient object detection suitable for real-time applications. The architecture of SSD, which YOLO models have adapted, utilizes multiple feature maps at different resolutions to detect objects of various sizes, employing a diverse array of anchor boxes at each feature map location to improve localization accuracy [105], [106].

Figure 3 illustrates a YOLO model that incorporates SSD's architectural principles to enhance real-time detection capabilities through improved feature extraction using Multi-Headed Attention layers. This adoption from SSD methodology significantly boosts the processing speed and detection accuracy of models such as YOLOv8, YOLOv9, and YOLOv10, making them ideal for rapid and reliable object detection in resource-constrained environments [108], [109]. The efficient single-shot mechanism, which directly classifies and localizes objects, highlights the ongoing evolution of the YOLO series to meet the accuracy and speed requirements of diverse real-world scenarios [104].

For the comprehensive review articles, it is advantageous to pinpoint a specific gap that the proposed review will address. For instance, a common oversight in the existing literature is the omission of the latest YOLO iterations, particularly YOLOv9, YOLOv10, and YOLOv11 or neglecting to cover the application domains of interest. Given the YOLO algorithm's ten-year milestone, there is a pressing need to systematically document and critically evaluate these newer models. Our review aims to fill this void by providing updated, in-depth insights and comparative analysis of YOLOv9 and YOLOv11, extending across various applications to serve the wider research and technical community. This state-of-the-art review intends to highlight the continued advancements and capabilities of these models within the dynamic field of object detection technology.

In this review paper, we adopt a unique reverse-chronological approach to analyze the progression of YOLO, beginning with the most recent versions and moving back-

TABLE I: **Summary of Performance Metrics Used in Model Evaluation**

No.	Performance Metric	Symbol	Equation	Description
1	Precision	P	$P = \frac{TP}{TP+FP}$	Ratio of true positive detections to the total predicted positives.
2	Recall	R	$R = \frac{TP}{TP+FN}$	Ratio of true positive detections to the total actual positives.
3	F1 Score	$F1$	$F1 = 2 \cdot \frac{P \cdot R}{P+R}$	Harmonic mean of precision and recall, balancing both metrics to provide a single performance measure for the model
4	Intersection over Union	IoU	$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$	Measures the overlap between the predicted and actual bounding boxes.
5	Frames Per Second	FPS	$\text{FPS} = \frac{1}{L}$	Number of images the model processes per second, inversely related to latency.
6	Non-Maximum Suppression	NMS	-	NMS is a post-processing step in YOLO to remove redundant bounding-boxes.

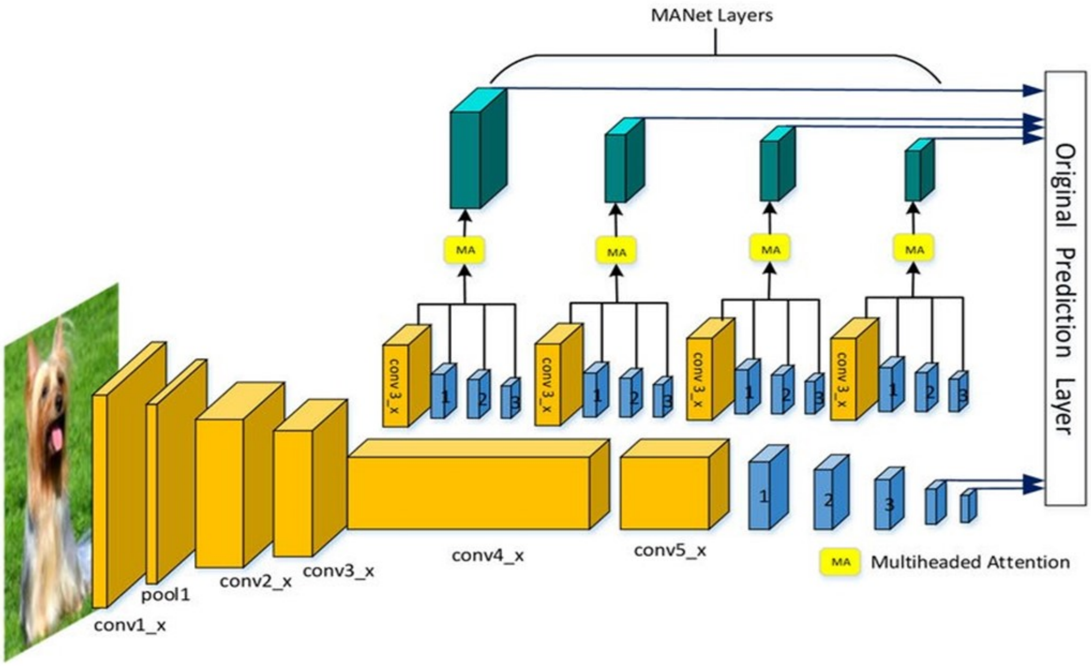


Fig. 3: **Enhanced YOLO model architecture incorporating SSD's single-stage detection approach with Multi-Headed Attention (MA) layers for superior real-time object detection performance [107].**

wards. The analysis is divided into six distinct subsections. The first subsection covers the latest iterations, YOLO11, The second subsection examines YOLOv10, YOLOv9, and YOLOv8, where we delve into the architecture and advancements that define the forefront of object detection technology. This approach not only shows the most cutting-edge developments but also sets the stage for understanding the incremental improvements that have been realized over time. The third subsection reviews YOLOv7, YOLOv6, and YOLOv5, tracing further back in the series to highlight the evolutionary steps contributing to the enhancements observed in the later versions. We analyze each model's technical and scientific aspects to provide a comprehensive view of the progress within these iterations. The fourth subsection addresses the earlier YOLO versions, offering a complete historical perspective that enriches the reader's understanding of the foundational technologies and the methodologies, refined through successive updates. The fifth subsection presents alternative versions derived from YOLO.

To close this section, we discuss the application of the YOLO models in reverse order across five critical real-world domains: autonomous vehicles and traffic safety, healthcare and medical image analysis, surveillance and security, industrial manufacturing and agriculture. For each application, we present a detailed examination and corresponding tabular data in reverse chronological order, showcasing how YOLO technologies have been adapted and implemented to meet specific industry needs and challenges. This reverse review

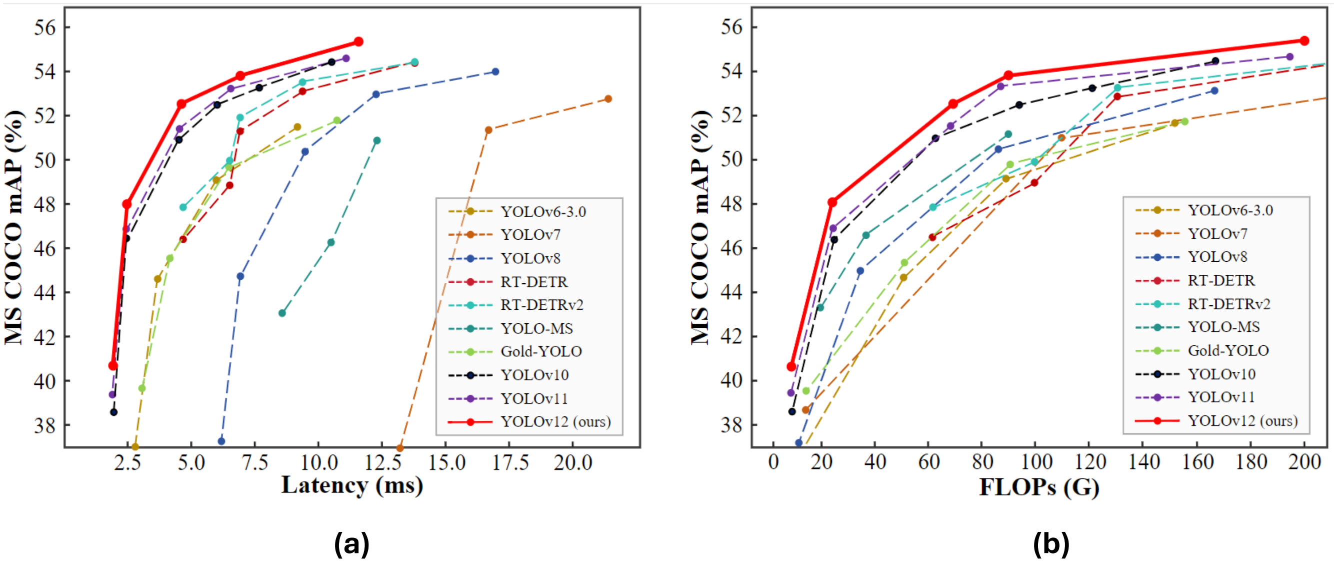


Fig. 4: **(a) Latency comparison on the MS COCO benchmark reveals significantly faster inference achieved with YOLOv12 compared to the same achieved with previous YOLO versions. (b) GFLOPs analysis also shows enhanced computational efficiency. [59]**

strategy not only emphasizes the state-of-the-art but also provides a narrative of technological evolution, illustrating how each iteration builds upon the last to push the boundaries of what's possible in object detection. By understanding where YOLO technology stands today and how it got there, readers gain a comprehensive view of its capabilities and potential future directions. This methodical unpacking of the YOLO series not only highlights technological advancements but also offers insights into the broader implications and utility of these models in practical scenarios, setting the groundwork for anticipating future innovations in object detection technology.

III. Review of YOLO Versions

This section reviews YOLO series models, starting from the advanced and latest version, YOLOv12, and progressively tracing back to the foundational YOLOv1. By first highlighting the most recent technological advancements, this approach enables immediate insights into the state-of-the-art capabilities of object detection. Subsequently, the narrative explores how earlier models laid the groundwork for these innovations.

A. YOLOv12 and YOLO11

YOLOv12 [59] is the most recent YOLO version introduced in February 2025, which marks a substantial advancement in real-time object detection by integrating attention mechanisms into the YOLO framework while maintaining competitive inference speeds. This attention-centric framework not only surpasses popular real-time detectors (e.g., xxx) in accuracy but also achieves state-of-the-art performance through a combination of innovative attention methods, Residual Efficient Layer Aggregation Networks (R-ELAN), and several architectural optimizations.

Figure 4a demonstrates latency comparisons on the MS COCO benchmark dataset, highlighting YOLOv12's significantly lower inference latency compared to YOLOv11, YOLOv10, YOLOv9, and YOLOv8. The curve reveals a substantial reduction in latency, enabling faster processing speeds while maintaining high detection accuracy (reference). Complementing this, Figure 4b presents comparison in terms of GFLOPs, which shows YOLOv12 achieves higher computational efficiency, reflecting its ability to handle complex computations effectively. This balance between speed and computational power demonstrates YOLOv12's robust performance.

On the COCO dataset, YOLOv12 sets a new state-of-the-art standard, with the lightweight YOLOv12-N achieving 40.6% mAP and the larger YOLOv12-X reaching 55.2% mAP. These results, combined with the latency and GFLOPs improvements, establish YOLOv12 as a new benchmark in real-time object detection. The model framework is available in five scales: YOLOv12-N, S, M, L, and X, each optimized for specific applications. For instance, YOLOv12-N achieves 40.6% mAP at 1.64 ms on a T4 GPU, outperforming YOLOv10-N and YOLOv11-N by 2.1% and 1.2% mAP, respectively. Similarly, YOLOv12-S attains 48.0% mAP at 2.61 ms/image, surpassing YOLOv8-S, YOLOv9-S, YOLOv10-S, and YOLOv11-S by margins of 3.0%, 1.2%, 1.7%, and 1.1% mAP. The larger models in the YOLOv12 family continue to show improvements in efficiency and performance. Notably, YOLOv12-M achieves 52.5% mAP at 4.86 ms/image. In terms of computational effieciency, YOLOv12-L demonstrates a significant reduction in FLOPs, decreasing by 31.4G compared to its predecessor, YOLOv10-L. At the highest end of the scale, YOLOv12-X showcases superior performance, outperforming both YOLOv10-X and YOLOv11-X in detection accuracy [59].

YOLOv12 also surpasses end-to-end detectors like RT-DETR and RT-DETRv2. For example, YOLOv12-S runs 42% faster than RT-DETR-R18 and RT-DETRv2-R18, using only 36% of the computation and 45% of the parameters. Residual connections show minimal impact on convergence in smaller models (YOLOv12-N) but are critical for stable training in larger models (YOLOv12-L/X), with YOLOv12-X requiring a scaling factor of 0.01. The area attention module reduces inference time by 0.7 ms on an RTX 3080 with FP32 precision, while FlashAttention further accelerates inference by 0.3–0.4 ms.

Visualization analyses confirm that YOLOv12 produces clearer object contours and more precise foreground activations than its predecessors. A convolution-based attention implementation proves to be faster than linear alternatives. Additionally, a hierarchical design, extended training (approximately 600 epochs), an optimized convolution kernel size (7 × 7), absence of positional embedding, and an MLP ratio of 1.2 collectively enhance the framework's performance and efficiency.

As discussed before, the YOLOv12 architecture (Figure 5a), demonstrates an advanced integration of A^2 (Area Attention) modules, R-ELAN (Residual Efficient Layer Aggregation Networks) blocks, and a streamlined detection head. This design optimizes the model's visual information processing while maintaining high accuracy. The major innovations on the YOLOv12 architecture are listed below.

1) YOLOv12 Architectural Innovation: label=•

- **Area Attention (A^2) Module:** This module implements segmented feature processing with Flash Attention integration, reducing computational complexity by 50% through spatial reshaping while maintaining large receptive fields. AA enables real-time detection at fixed $n = 640$ resolution through optimized memory access patterns, as illustrated in Figure 5a.
- **Residual ELAN (R-ELAN) Hierarchy:** R-ELAN combines residual shortcuts (0.01 scaling) with dual-branch processing to mitigate the gradient vanishing problem. The model also features a streamlined final aggregation stage that reduces parameters by 18% and FLOPs by 24% compared to baseline architectures, as shown in Figure 5b.
- **Efficient Architectural Revisions:** YOLOv12 replaces positional encoding with 7×7 depth-wise convolution for implicit spatial awareness. It also implements adaptive MLP ratio (1.2×) and shallow block stacking to balance the computational load, achieving 4.1 ms inference latency on V100 hardware.
- **Optimized Training Framework:** The model was trained over 600 epochs using SGD with cosine scheduling (initial lr=0.01). The model also incorporates Mosaic-9 and Mixup augmentations with 12.8% mAP gain on COCO dataset, maintaining real-time performance through selective kernel convolution integration.

Figure 5b presents an architectural comparison of popular attention modules: CSPNet, ELAN, C3K2 (a case of GELAN), and the proposed R-ELAN. Brief summary of these modules is blow.

- **CSPNet (Cross Stage Partial Network):** CSPNet enhances gradient flow by splitting feature maps into two paths, one for learning and one for propagation, reducing computational bottlenecks and improving inference speed. This model is visually depicted in Figure 5b (leftmost module).
- **ELAN (Efficient Layer Aggregation Network):** ELAN improves feature integration by aggregating multi-scale features efficiently, enhancing the model's ability to detect objects at various scales. However, as shown in Figure 5b (second module), ELAN can introduce instability due to gradient blocking and lacks of residual connections, particularly in large-scale models.
- **C3K2 (Compact GELAN):** This module is a compact version of GELAN (Generalized Efficient Layer Aggregation Network) that offers a balance between computational efficiency and feature expressiveness, suitable for resource-constrained environments. The module is also illustrated in Figure 5b (third module).
- **R-ELAN (Residual ELAN):** R-ELAN introduces residual connections and redesigns feature aggregation to address optimization challenges in attention-based models, combining the benefits of residual learning with efficient feature aggregation. As shown in Figure 5b (rightmost module), R-ELAN applies a residual shortcut with a scaling factor (default 0.01) and processes the input through a transition layer, followed by a bottleneck structure for improved stability and performance.

The R-ELAN design, as depicted in Figure 5b, addresses the limitations of ELAN by introducing residual connections and a revised aggregation approach. Unlike ELAN, which splits the input into two parts and processes them separately, R-ELAN applies a transition layer to adjust channel dimensions and processes the feature map through subsequent blocks before concatenation. This design mitigates gradient blocking and ensures stable convergence, particularly in large-scale models like YOLOv12-L and YOLOv12-X. The integration of residual connections and attention mechanisms in R-ELAN, as shown in Figure 5b, highlights YOLOv12's architectural advancements in balancing efficiency and accuracy.

YOLO11: YOLO11 developed by Ultralytics represents the most recent version building upon the foundations established by its predecessors in the YOLO family. This latest iteration introduces several architectural innovations that enhance its performance across a wide spectrum of tasks as depicted in Figure 6. The model incorporates the C3k2 (Cross Stage Partial with kernel size 2) block, which replaces the C2f block used in previous versions, offering improved computational efficiency [112]. Additionally, YOLOv11 retains the SPPF (Spatial Pyramid Pooling - Fast) component and introduces the C2PSA (Convolutional block with Parallel Spatial Attention)

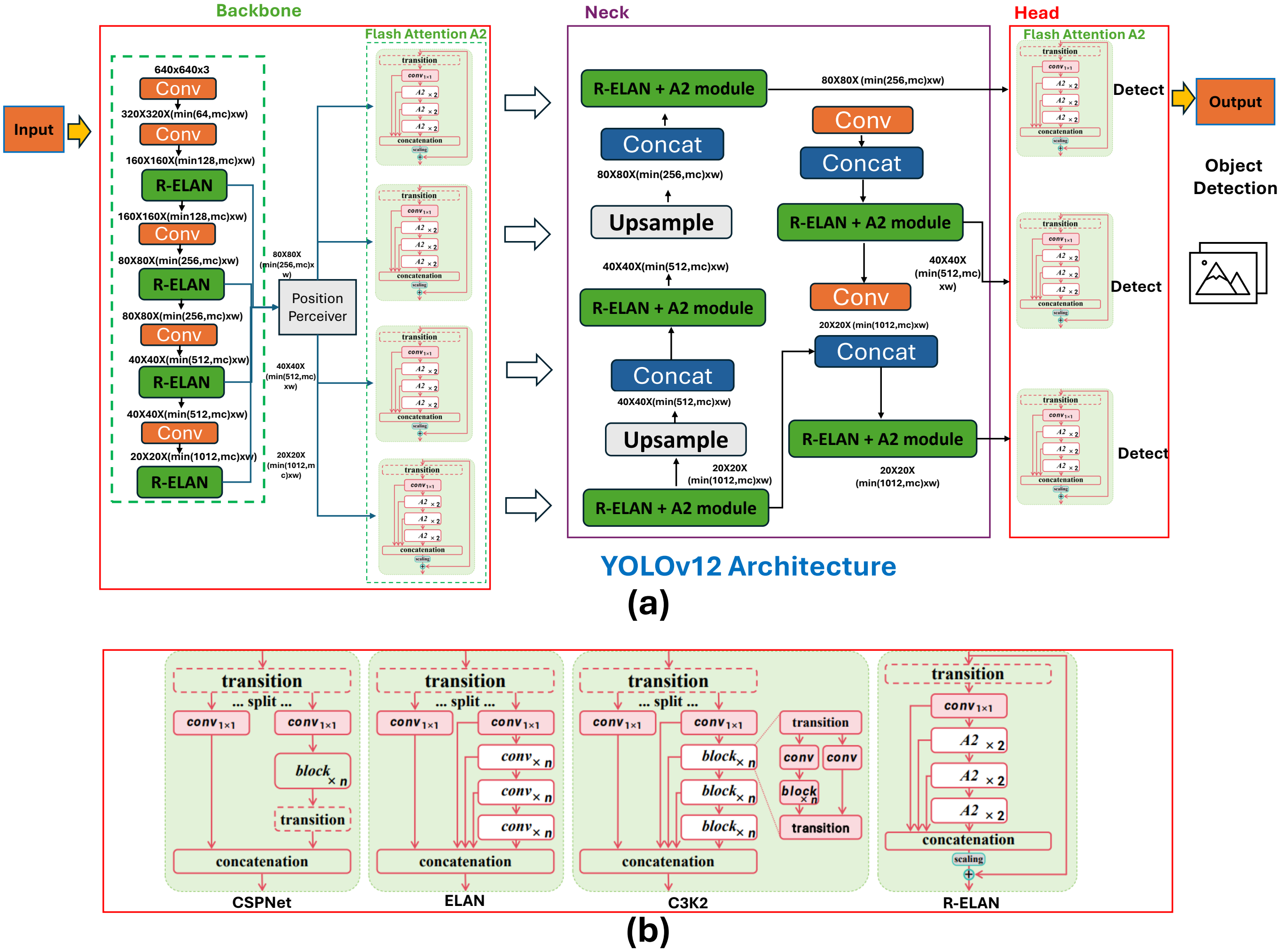


Fig. 5: **(a) Architecture of the YOLOv12 object detection model that integrated Area Attention (A²) modules, R-ELAN blocks, and a streamlined detection head; (b) Comparison of "Attention Module" architectures: CSPNet [110], ELAN [111], C3K2 (used in YOLOv9) [91], and the novel R-ELAN [59] introduced with YOLOv12, which improved residual connections and enhanced feature aggregation, demonstrating superior performance.**

module, collectively enhancing feature extraction capabilities [79]. These architectural enhancements enable YOLOv11 to capture intricate image details with greater precision, particularly in challenging scenarios involving small or occluded objects. The model's versatility is evident in its support for a broad range of computer vision tasks, including object detection, instance segmentation, pose estimation, image classification, and oriented bounding box (OBB) detection (https://docs.ultralytics.com/models/yolo11/).

Empirical evaluations of YOLOv achieves a higher mean Average Precision (mAP) score on the COCO dataset while utilizing 22% fewer parameters compared to its YOLOv8m counterpart [113]. This reduction in parameter count contributes to faster model performance without significantly impacting overall accuracy. Furthermore, YOLOv11 exhibits inference times approximately 2% faster than YOLOv10, making it particularly well-suited for real-time applications. The model's efficiency extends across various deployment environments, including edge devices, cloud platforms, and systems supporting NVIDIA GPUs. YOLOv11 is available in multiple variants, ranging from nano to extra-large, catering to diverse computational requirements and use cases. These advancements position YOLOv11 as a state-of-the-art solution for industries requiring rapid and accurate image analysis, such as autonomous driving, surveillance, and industrial automation.

YOLOv10, incorporates advanced techniques like automated architecture search and more refined loss functions to enhance detection accuracy and speed, tailored for both edge and cloud computing environments. This version and its predecessors, YOLOv9 and YOLOv8, introduce substantial improvements in network architecture, such as the integration of cross-stage partial networks (CSPNets) and the use of transformer-based backbones for better feature extraction across different scales. YOLOv7 and YOLOv6 continued to build on these improvements by optimizing computational effi-

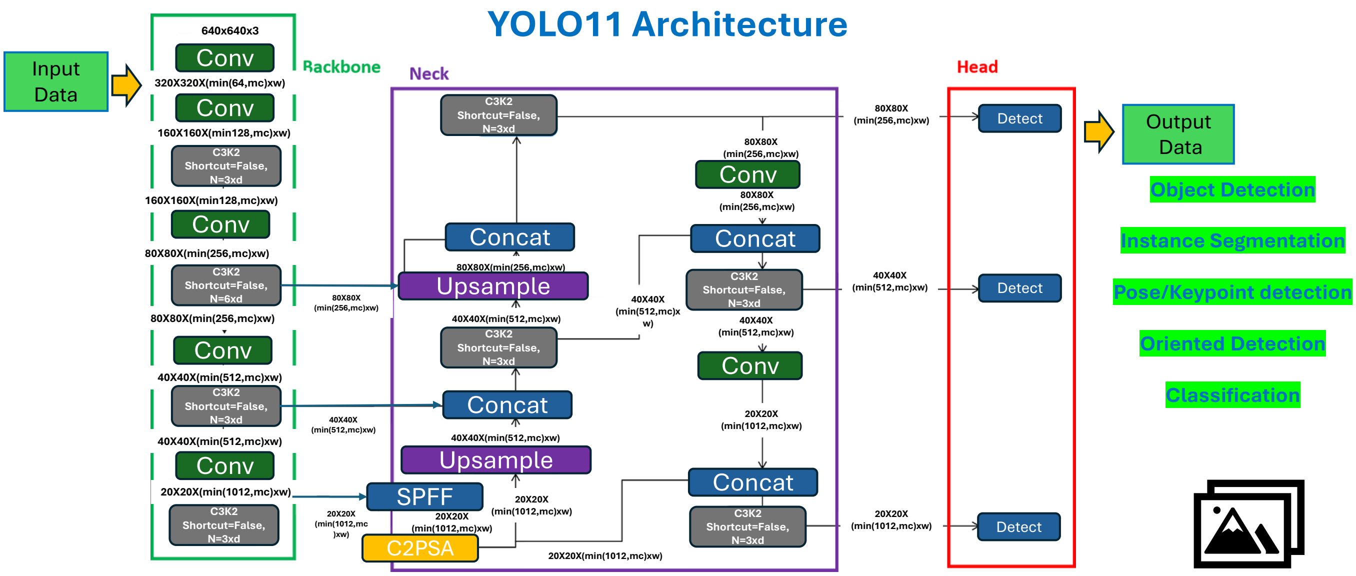


Fig. 6: **YOLOv11 architecture diagram: Enhanced backbone with C3k2 blocks replacing C2f, SPPF for multi-scale feature extraction, C2PSA for attention mechanism, and optimized neck. Efficient feature processing through multiple scales, culminating in a Detect head for multi-class object detection and localization.**

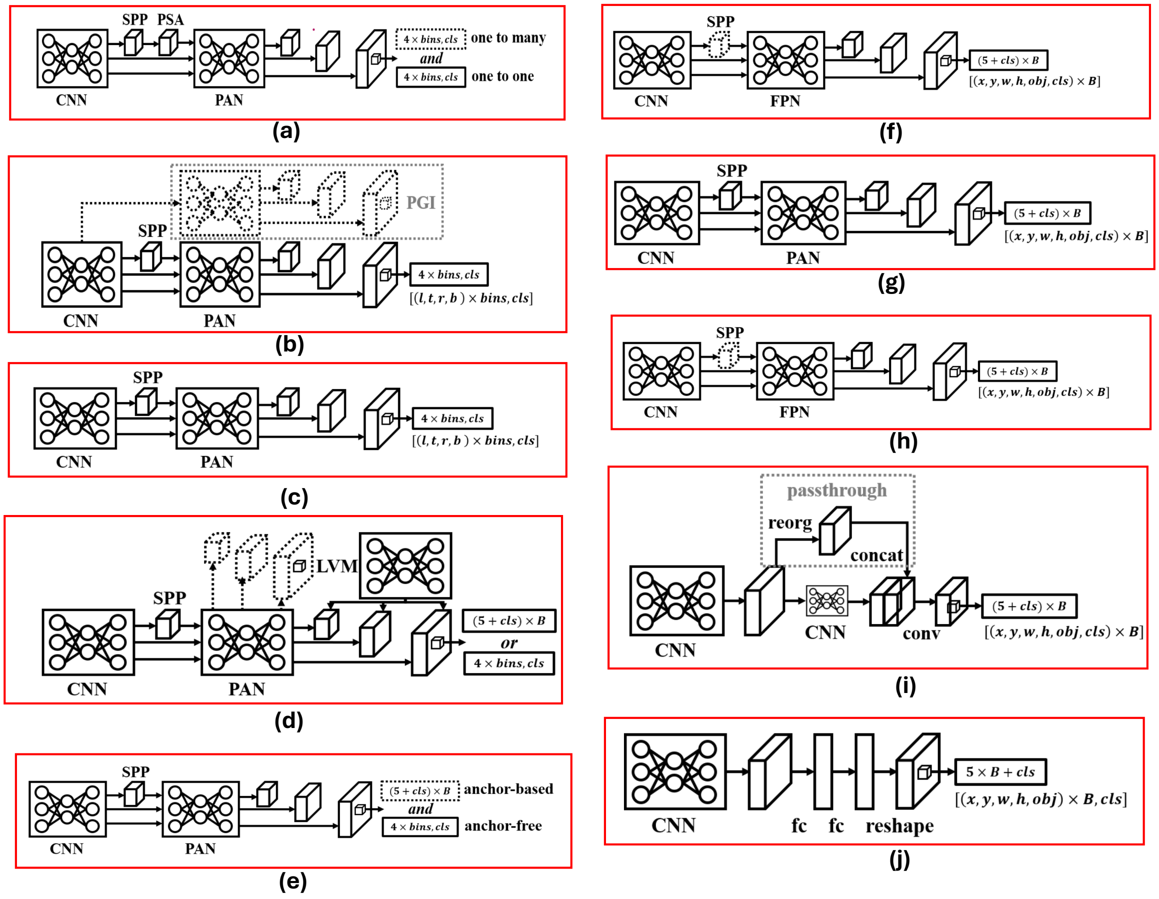


Fig. 7: **Simplified Architecture diagrams for : a)YOLOv10 ;b) YOLOv9 ; c) YOLOv8 ; d) YOLOv7 ; e) YOLOv6 ; f) YOLOv5 ; g) YOLOv4 ; h) YOLOv3 ; i)YOLOv2 ; j) YOLOv1. The diagrams are detailed in [114]**

ciency and expanding model scalability. Meanwhile, YOLOv5 introduced PyTorch support, which significantly enhanced the model's accessibility and adaptability, thus broadening its application in industry and academia. YOLOv4, on the other hand, marked a pivotal point in YOLO history by integrating features like Mish activation and Cross-Stage Partial connections, setting new standards for speed and accuracy in real-time applications. The mid-generations, starting with YOLOv3, were notable for introducing multi-scale predictions and bounding box predictions across different layers, which greatly improved the model's ability to detect small objects—a longstanding challenge in earlier versions. YOLOv3 was also one of the first YOLO models to leverage deeper feature extractors like Darknet-53, which significantly boosted its performance over YOLOv2. YOLOv2 itself had introduced important features such as batch normalization and high-resolution classifiers, which enhanced the overall accuracy without compromising the speed. The original YOLO model, YOLOv1, was revolutionary, proposing a single-stage detection framework that unified the object detection process into a single neural network model.

Figure 7 a illustrates a sophisticated transformer-based model that simplifies the detection process by integrating dual label assignments and eliminating the need for non-max suppression (NMS), achieving a streamlined, end-to-end object detection. YOLOv9, as shown in Figure 7 b, introduces the Programmable Gradient Information (PGI) system to enhance model interpretability and robustness, significantly improving generalization across various tasks. Moving to YOLOv8 and YOLOv7, Figures 7 c and 7 d respectively depict their architectures which incorporate elements like ELAN and CSPNet to boost performance and flexibility across computing devices. YOLOv6, highlighted in Figure 7 e, focuses on industry applications with enhancements in model quantization and real-time performance.

YOLOv5, represented in Figure 7 f, marks a pivotal development with its adoption of PyTorch and improvements in training methods that enhance model accessibility and efficiency. In contrast, YOLOv4, shown in Figure 7 g, integrates technologies like CSPNet and Path Aggregation Network (PAN) [115] to optimize real-time detection. YOLOv3, visualized in Figure 7 h, introduces significant architectural changes with Darknet-53 and multi-scale predictions, which substantially enhance the detection of small objects. YOLOv2, depicted in Figure 7 i, advances the architecture with dimension clusters and fine-grained features, improving the model's efficiency and adaptability. Finally, YOLOv1, as outlined in Figure 7 j, revolutionizes object detection by integrating a single-stage detector that performs grid-based predictions in real-time, significantly reducing model complexity and enhancing speed.

B. YOLOv10, YOLOv9 and YOLOv8

YOLOv10 [94], developed at Tsinghua University, China, represents a breakthrough in the YOLO series for real-time object detection, achieving unprecedented performance. This version eliminates the need for non-maximum suppression (NMS) [116], a traditional bottleneck in earlier models, thereby drastically reducing latency. YOLOv10 introduces a dual assignment strategy in its training protocol, which optimizes detection accuracy without sacrificing speed with the help of one-to-many and one-to-one label assignments, ensuring robust detection with lower latency [117], [118]. The architecture of YOLOv10 includes several innovative components that enhance both computational efficiency and detection performance. Among these are lightweight classification heads [119] that reduce computational demands, spatial-channel decoupled downsampling to minimize information loss during feature reduction [120], and rank-guided block design that optimizes parameter use [121]. These architectural advancements ensure that YOLOv10 operates synergistically across various scales—from YOLOv10-N (Nano) to YOLOv10-X (Extra Large), making it adaptable to diverse computational constraints and operational requirements [94]. According to wang et al. [94], performance evaluations on benchmark datasets like MS-COCO [122] demonstrate that YOLOv10 not only surpasses its predecessors—YOLOv9 and YOLOv8—in both accuracy and efficiency but also sets new industry standards. For instance, YOLOv10-S substantially outperforms comparable models (e.g., YOLOv9 BASE, yoloV9 Gelan, YOLOv8, YOLOv7) with an improved mAP and lower latency. This version also incorporates holistic efficiency-accuracy driven design, large-kernel convolutions, and partial self-attention modules, collectively improving the trade-off between computational cost and detection capability. The architecture diagrams of YOLOv10, YOLOv9, and YOLOv8 are summarized in Figures 8, 9, and 10, respectively.

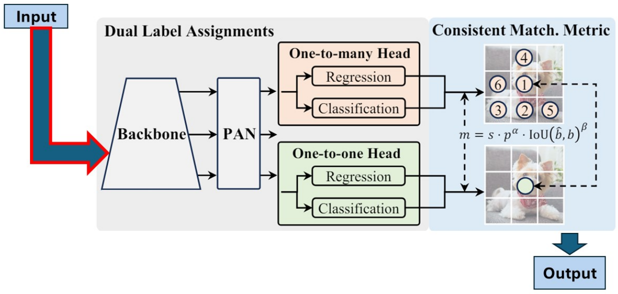


Fig. 8: **YOLOv10 architecture, which employs a dual label assignment strategy to improve detection accuracy. A backbone processes the input image, while PAN (Path Aggregation Network) enhances feature representation. Employed heads are (1) one-to-many head for regression and classification tasks, and (2) one-to-one head for precise localization [94]**

The YOLOv10 model offers various configurations, each tailored to specific performance needs within real-time object detection frameworks. Starting with YOLOv10-N (Nano), it demonstrates a rapid detection capability with a mAP of

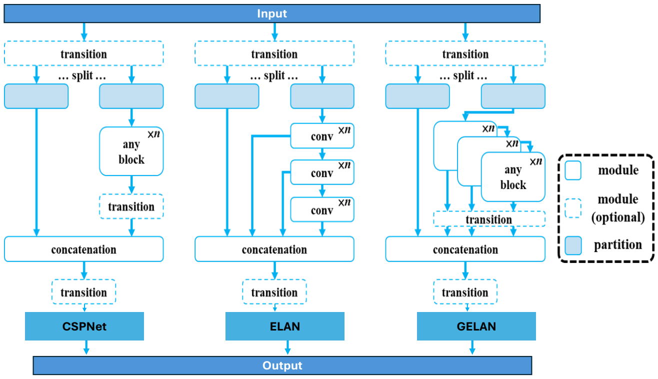


Fig. 9: **YOLOv9 architecture [91] with CSPNet, ELAN, and GELAN modules. CSPNet enhances gradient flow and reduces computational load through feature map partitioning. ELAN focuses on the linear aggregation of features for improved learning efficiency, while GELAN generalizes this approach to combine features from multiple depths and pathways, providing greater flexibility and accuracy in feature extraction.**

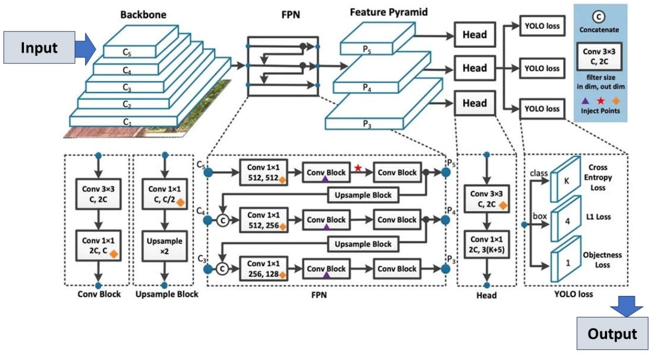


Fig. 10: **YOLOv8 architecture [123]: showcasing the key components and their connections. The backbone network processes the input image through multiple convolutional layers (C1 to C5), extracting hierarchical features. These features are then passed through the Feature Pyramid Network (FPN) to create a feature pyramid (P3, P4, P5), which enhances detection at different scales. The network heads perform final predictions, incorporating convolutional blocks and upsample blocks to refine features.**

38.5% at an exceptionally reduced latency to 1.84 ms, making it highly suitable for scenarios demanding quick responses. Progressing through the series, YOLOv10-S (Small) and YOLOv10-M (Medium) offer progressively higher mAP values of 46.3% and 51.1% at latencies of 2.49 ms and 4.74 ms, respectively, providing a balanced performance for versatile applications. The larger variants, YOLOv10-B (Balanced) and YOLOv10-L (Large), cater to environments requiring detailed detections, with mAPs of 52.5% and 53.2% and latencies of 5.74 ms and 7.28 ms respectively. The largest model, YOLOv10-X (Extra Large), excels with the highest mAP of 54.4% at a latency of 10.70 ms, designed for complex detection tasks where precision is paramount. These configurations underscore YOLOv10's adaptability across a spectrum of operational requirements.

Reflecting on YOLO's evolution, starting from YOLOv1, which set the benchmark with an mAP of 63.4% and a latency of 45 ms, to the latest YOLOv10, significant technological strides have been evident. YOLOv10's predecessors, YOLOv9 and YOLOv8, display comparable mAP scores to YOLOv10 but with marginally higher latency, indicating the incremental enhancements YOLOv10 brings to the table. Specifically, YOLOv9 and YOLOv8 models, such as YOLOv9-N and YOLOv8-N, showcase mAPs of 39.5% and 37.3%, respectively, at latency indicative of their generational improvements. Meanwhile, the higher end of these series, YOLOv9-X, and YOLOv8-X, achieve mAPs of 54.4% and 53.9%, respectively, with YOLOv10 outperforming them in efficiency. The YOLO series, from YOLOv1 through YOLOv8, YOLOv9, and now YOLOv10, has continually advanced the frontier of real-time object detection, enhancing both the speed and accuracy of detections, and thus broadening the scope for practical applications in sectors like autonomous driving, surveillance, and real-time video analytics.

YOLOv9 [91] marks a significant advancement in real-time object detection by addressing the efficiency and accuracy challenges associated with earlier versions, particularly by mitigating information loss in deep neural processing. It introduces the innovative Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN) architecture. These enhancements focus on preserving crucial information across the network, ensuring robust and reliable gradients that prevent data degradation, which is common in deep neural networks [124]. Compared to its successor, YOLOv10, YOLOv9 sets a foundational stage by addressing the information bottleneck problem that typically hinders deep learning models. While YOLOv9's PGI strategically maintains data integrity throughout the processing layers, YOLOv10 builds upon this foundation by eliminating the need for NMS and further optimizing model architecture for reduced latency and enhanced computational efficiency. YOLOv10 also introduces dual assignment strategies for NMS-free training, significantly enhancing the system's response time without compromising accuracy, which reflects a direct evolution from the groundwork laid by YOLOv9's innovations [125]. Furthermore, YOLOv9's GELAN architecture represents a pivotal improvement in network design, offering a flexible and efficient structure that effectively integrates multi-scale features. While GELAN contributes significantly to YOLOv9's performance, YOLOv10 extends these architectural improvements to achieve even greater efficiency and adaptability [126]. It reduces computational overhead and increases the model's applicability to various real-time scenarios, showcasing an advanced level of refinement that leverages and enhances the capabilities introduced by YOLOv9.

YOLOv8 was released in January 2023 by Ultralytics, marking a significant progression in the YOLO series with an introduction of multiple scaled versions designed to cater to a wide range of applications [127], [128]. These versions included YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large), each optimized for specific performance and computational needs. This flexibility made YOLOv8 highly versatile, supporting many vision tasks such as object detection, segmentation, pose estimation, tracking, and classification, significantly broadening its application scope in real-world scenarios [128]. The architecture of YOLOv8 underwent substantial refinements to enhance its detection capabilities. It retained a similar backbone to YOLOv5 but introduced modifications in the CSP Layer, now evolved into the C2f module—a cross-stage partial bottleneck with dual convolutions that effectively combine high-level features with contextual information to bolster detection accuracy. YOLOv8 transitioned to an anchor-free model with a decoupled head, allowing independent processing of object detection, classification, and regression tasks, which, in turn, improved overall model accuracy [129]. The output layer employed a sigmoid activation function for objectness scores and softmax for class probabilities, enhancing the precision of bounding box predictions. YOLOv8 also integrated advanced loss functions like CIoU [130] and Distribution Focal Loss (DFL) [131] for bounding-box optimization and binary cross-entropy for classification, which proved particularly effective in enhancing detection performance for smaller objects. YOLOv8's architecture, demonstrated in detailed diagrams, features the modified CSPDarknet53 backbone with the innovative C2f module, augmented by a spatial pyramid pooling fast (SPPF) layer that accelerates computation by pooling features into a fixed-size map. This model also introduced a semantic segmentation variant, YOLOv8-Seg, which utilized the backbone and C2f module, followed by two segmentation heads designed to predict semantic segmentation masks efficiently. This segmentation model achieved state-of-the-art results on various benchmarks while maintaining high speed and accuracy, evident in its performance on the MS COCO dataset where YOLOv8x reached an AP of 53.9% at 640 pixels image size—surpassing the 50.7% AP of YOLOv5—with a remarkable speed of 280 FPS on an NVIDIA A100 using TensorRT. As we progress backwards through the YOLO series, from YOLOv10 to YOLOv8 and soon to YOLOv7, these architectural and functional advancements highlight the series' evolutionary trajectory in optimizing real-time object detection networks.

YOLOv8 was released in January 2023 by Ultralytics, marking a significant progression in the YOLO series with an introduction of multiple scaled versions designed to cater to a wide range of applications [127], [128]. These versions included YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large), each optimized for specific performance and computational needs. This flexibility made YOLOv8 highly versatile, supporting many vision tasks such as object detection, segmentation, pose estimation, tracking, and classification, significantly broadening its application scope in real-world scenarios [128]. YOLOv8's architecture underwent significant upgrades to boost its detection performance, maintaining a backbone similar to YOLOv5 but enhancing it with the evolved C2f module, a cross-stage partial bottleneck with dual convolutions. This module integrates high-level features with contextual information, improving accuracy. The model transitioned to an anchor-free system with a decoupled head for independent objectness, classification, and regression tasks, enhancing accuracy [129]. The output layer now uses sigmoid for objectness and softmax for class probabilities, refining bounding box precision. Additionally, YOLOv8 employs CIoU [130], Distribution Focal Loss (DFL) [131] for bounding-box optimization, and binary cross-entropy for classification, significantly boosting performance, particularly for smaller objects.

This model also introduced a semantic segmentation variant, YOLOv8-Seg [132], which utilized the backbone and C2f module, followed by two segmentation heads designed to predict semantic segmentation masks efficiently. This segmentation model achieved state-of-the-art results on various benchmarks while maintaining high speed and accuracy, evident in its performance on the MS COCO dataset where YOLOv8x reached an AP of 53.9% at 640 pixels image size—surpassing the 50.7% AP of YOLOv5—with a remarkable speed of 280 FPS on an NVIDIA A100 using TensorRT. As we progress backwards through the YOLO series, from YOLOv10 to YOLOv8 and soon to YOLOv7, these architectural and functional advancements highlight the series' evolutionary trajectory in optimizing real-time object detection networks.

C. YOLOv7, YOLOv6 and YOLOv5

The YOLOv7 model introduces enhancements in object detection tailored for drone-captured scenarios, particularly through the Transformer Prediction Head (TPH-YOLOv5) variant [133], which emphasizes improvements in handling scale variations and densely packed objects [89]. By incorporating TPH and the Convolutional Block Attention Module (CBAM) [134], YOLOv7 substantially boosts its capacity to focus on relevant regions in cluttered environments. These features particularly enhance the model's ability to detect objects across varied scales, an essential trait for drone applications where altitude changes affect object size perception drastically. The model integrates sophisticated strategies like multi-scale testing [17] and a self-trained classifier, which refines its performance on challenging categories by specifically addressing common issues in drone imagery, such as motion blur and occlusion. These adaptations have shown notable improvements, with YOLOv7 achieving competitive results in drone-specific datasets and challenges [135]. The model's adaptability and robustness in such specialized conditions demonstrate its potential beyond conventional settings, catering effectively to next-generation applications like urban surveillance and wildlife monitoring.

YOLOv6 emerges as a robust solution in industrial applications by delivering a finely balanced trade-off between speed and accuracy, crucial for deployment across various hardware platforms [87]. It iterates on previous versions by

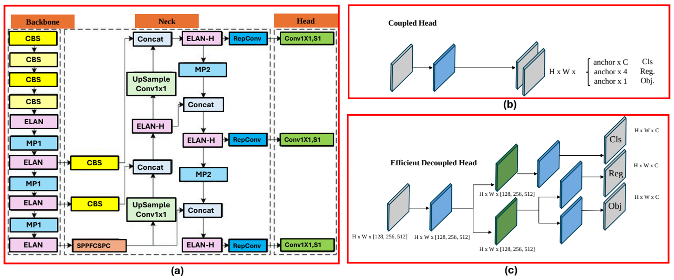


Fig. 11: **Architecture diagrams for a) YOLOv7 ; b) YOLOv6 ; and c) YOLOv5**

incorporating cutting-edge network designs, training strategies, and quantization techniques to enhance its efficiency and performance significantly. This model has been optimized for diverse operational requirements with its scalable architecture, ranging from YOLOv6-N to YOLOv6-X, each offering different performance levels to suit specific computational budgets [136]. Significant innovations in YOLOv6 include advanced label assignment techniques and loss functions that refine the model's predictive accuracy and operational efficiency. By leveraging state-of-the-art advancements in machine learning, YOLOv6 not only excels in traditional object detection metrics but also sets new standards in throughput and latency, making it exceptionally suitable for real-time applications in industrial and commercial domains.

YOLOv6 and YOLOv7 each introduced innovative features that build on the foundation set by YOLOv5. YOLOv6, released in October 2021, introduced lightweight nano models optimized for mobile and CPU environments alongside a more effective backbone for improved small object detection. YOLOv7 further advanced this development by incorporating a new backbone network, PANet [137], enhancing feature aggregation and representation, and introducing the CIOU loss function for better object scaling and aspect ratio handling. YOLO-v6 significantly shifts the architecture to an anchor-free design, incorporating a self-attention mechanism to better capture long-range dependencies and employing adaptive training techniques to optimize performance during training [138]. These versions collectively push the boundaries of object detection performance, emphasizing speed, accuracy, and adaptability across various deployment scenarios.

YOLOv5 has significantly contributed to the YOLO series evolution, focusing on user-friendliness and performance enhancements [139], [140]. Its introduction by Ultralytics brought a streamlined, accessible framework that lowered the barriers to implementing high-speed object detection across various platforms. YOLOv5's architecture incorporates a series of optimizations including improved backbone, neck, and head designs which collectively enhance its detection capabilities. The model supports multiple size variants, facilitating a broad range of applications from mobile devices to cloud-based systems [139]. YOLOv5's adaptability is further evidenced by its continuous updates and community-driven enhancements, which ensure it remains at the forefront of object detection technologies. This version stands out for its balance of speed, accuracy, and utility, making it a preferred choice for developers and researchers looking to deploy state-of-the-art detection systems efficiently.

YOLOv5 marks a significant evolution in the YOLO series, focusing on production-ready deployments with streamlined architecture for real-world applications. This version emphasizes reducing the model's complexity by refining its layers and components, enhancing its inference speed without sacrificing detection accuracy. The backbone and feature extraction layers were optimized to accelerate processing, and the network's architecture was simplified to facilitate faster data throughput. Importantly, YOLO v5 enhances its deployment flexibility, catering to edge devices with limited computational resources through model modularity and efficient activations. These architectural refinements ensure YOLO v5 operates effectively in diverse environments, from high-resource servers to mobile devices, making it a versatile tool in the arsenal of object detection technologies.

D. *YOLOv4, YOLOv3, YOLOv2 and YOLOv1*

The introduction of YOLOv4 [141] in 2020 marked the latest developments, employing CSPDarknet-53 [142]as its backbone. This modified version of Darknet-53 uses Cross-Stage Partial connections to reduce computational demands while enhancing learning capacity. YOLOv4 incorporates innovative features such as Mish activation [144], replacing traditional ReLU to maintain smooth gradients, and utilizes new data augmentation techniques such as Mosaic and CutMix [145]. Additionally, it introduces advanced regularization methods, including DropBlock regularization [146] and Class Label Smoothing to prevent overfitting [147], alongside optimization

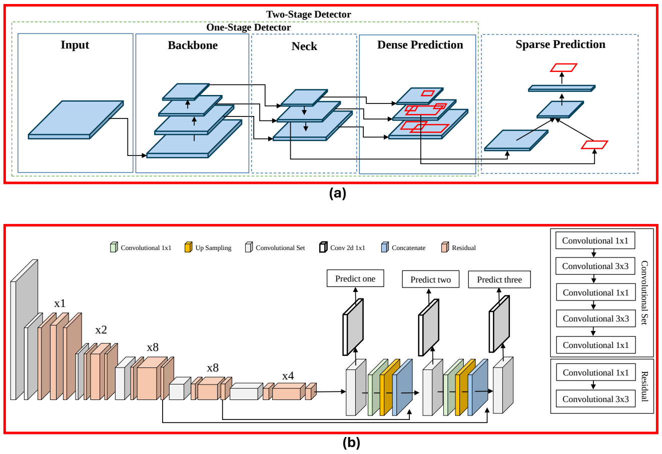
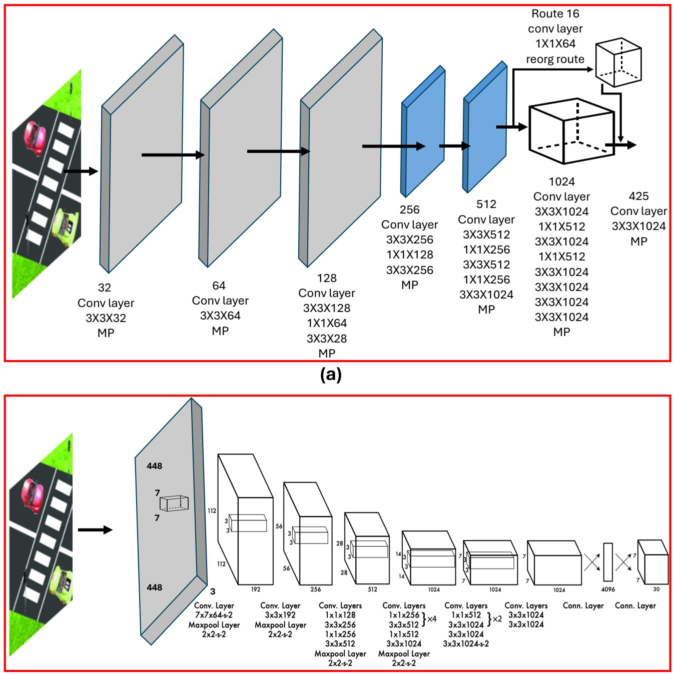


Fig. 12: **Comparison of YOLOv4 [141] and YOLOv3 [143] architectures. (a) YOLOv4 architecture shows a two-stage detector with a backbone, neck, dense prediction, and sparse prediction modules. (b) YOLOv3 architecture features convolutional and upsampling layers that lead to multi-scale predictions. This highlights the structural advancements in object detection between the two versions**

strategies termed BoF (Bag of Freebies) [148] and BoS (Bag of Specials) that enhance training and inference efficiency. ”YOLOv3, introduced in 2018 before the release of YOLOv4, employed the Darknet-53 architecture, incorporating principles of residual learning. Initially trained on ImageNet, this version excelled in detecting objects of various sizes due to its multi-scale detection capabilities within the architecture. The subsequent development of YOLOv4 built upon the success of YOLOv3, further enhancing the framework’s robustness and accuracy.



(a)

(b)

Fig. 13: **(a) YOLOv2 architecture [149], illustrating improvements such as the use of batch normalization, higher resolution input, and anchor boxes ; (b) YOLOv1 architecture [150], showing the sequence of convolutional layers, max-pooling layers, and fully connected layers used for object detection. This model performs feature extraction and prediction in a single unified step, aiming for real-time performance.**

YOLOv3 [143] improved detection accuracy, especially for small objects, by using three different scales for detection, thereby capturing essential features at various resolutions. Earlier, YOLOv2 and the original YOLO (YOLOv1) laid the groundwork for these advancements [150].

Earlier, YOLOv2 and the original YOLO (YOLOv1) laid the groundwork for these advancements. Released in 2016, YOLOv2 introduced a new 30-layer architecture with anchor boxes from Faster R-CNN and batch normalization to speed up convergence and enhance model performance. YOLOv1, debuting in 2015 by Joseph Redmon, revolutionized object detection with its single-shot mechanism that predicted bounding boxes and class probabilities in one network pass, utilizing a simpler Darknet-19 architecture. This initial approach significantly accelerated the detection process, establishing the foundational techniques that would be refined in later versions of the YOLO series. YOLOv4 and YOLOv3, showcasing their advanced architectures and features, are illustrated in Figure 12a and b, respectively, while YOLOv2 and YOLOv1 are depicted in Figure 13a and b, showcasing the foundational developments in the series.

E. Alternative Versions Derived from YOLO

Several alternative YOLO models have been developed from different versions, with the five primary ones being YOLO-NAS, YOLO-X, YOLO-R, DAMO-YOLO, and Gold-YOLO.

1) YOLO-NAS: YOLO-NAS, developed by Deci AI, represents a significant advancement in object detection technology [151]. This model leverages Neural Architecture Search (NAS) [152] to address limitations of previous YOLO iterations [153]. YOLO-NAS introduces a quantization-friendly basic block, enhancing performance with minimal precision loss post-quantization. The architecture employs quantization-aware blocks and selective quantization, resulting in superior object detection capabilities. Notably, when converted to INT8, the model experiences only a slight precision drop, outperforming its predecessors. YOLO-NAS utilizes sophisticated training schemes and post-training quantization techniques, further improving its efficiency [151]. The model is pre-trained on datasets such as COCO, Objects365, and Roboflow 100, making it suitable for various downstream object detection tasks. YOLO-NAS is available in three variants: Small (s), Medium (m), and Large (l), each optimized for different computational requirements. These variants offer a balance between Mean Average Precision (mAP) and latency, with the INT-8 versions demonstrating impressive performance metrics. The architecture of YOLO-NAS (Figure 14a) supports inference, validation, and export modes, though it does not support training. YOLO-NAS’s innovative design and superior performance position it as a critical tool for developers and researchers in the field of computer vision.

2) YOLO-X: YOLOX, developed by Megvii Technology, represents a significant advancement in the YOLO series of object detectors. This model introduces several key improvements to enhance performance and efficiency. YOLOX adopts an anchor-free approach, departing from the anchor-based methods of its predecessors [154]. It incorporates a decoupled head, separating classification and regression tasks to address the known conflict between these objectives in object detection [155]. The model also implements SimOTA, an advanced label assignment strategy, further improving its detection capabilities [156]. Architecturally (Figure 14b), YOLOX-DarkNet53 builds upon the YOLOv3-SPP baseline, incorporating enhancements such as EMA weights updating, cosine learning rate scheduling, IoU loss, and an IoU-aware branch. The decoupled head consists of a 1x1 convolution layer for channel dimension reduction, followed by two parallel branches with two 3x3 convolution layers each [157]. This design significantly improves convergence speed and is crucial for end-to-end detection performance. YOLOX demon-

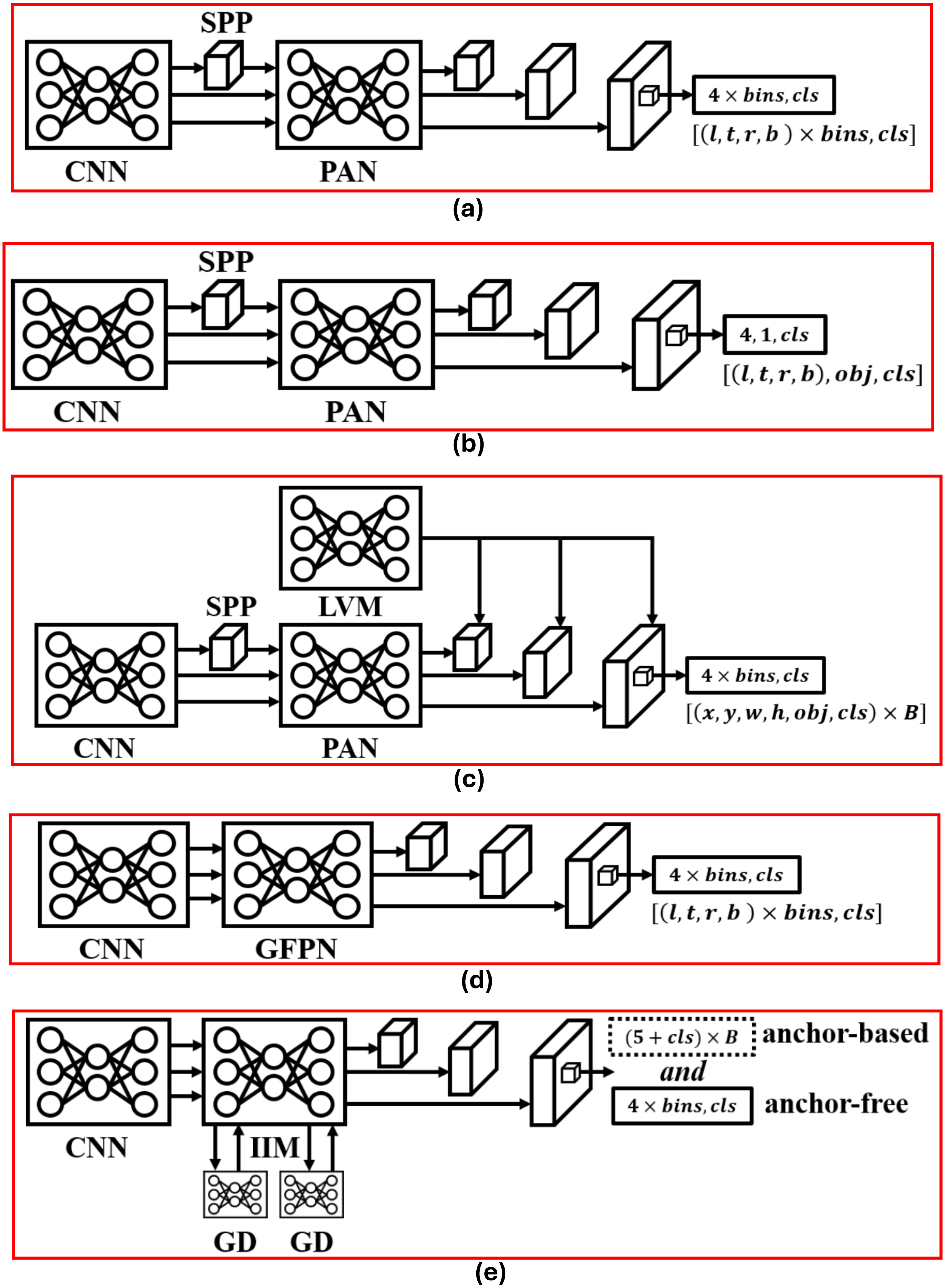


Fig. 14: **Architecture diagram of a) YOLONAS; b)YOLOX ; c) YOLOR ; d) DAMO YOLO ; e) GOLD YOLO**

strates superior performance across various model sizes . The YOLOX-L variant achieves 50.0% AP on COCO at 68.9 FPS on Tesla V100, surpassing YOLOv5-L by 1.8% AP. Even the lightweight YOLOX-Nano, with only 0.91M parameters and 1.08 GFLOPs, attains 25.3% AP on COCO, outperforming NanoDet by 1.8% AP. These advancements position YOLOX as a state-of-the-art object detector, balancing accuracy and efficiency across a wide range of model scales [154].

3) YOLO-R: YOLOR (You Only Learn One Representation) is a state-of-the-art object detection algorithm developed by Chien-Yao Wang et al. [158] . Unlike previous YOLO versions, YOLOR introduces a novel approach to multi-task learning by unifying implicit and explicit knowledge representation [159]. The algorithm's core concept is inspired by human cognition, aiming to process multiple tasks simultaneously given a single input. YOLOR's architecture (Figure 14c) incorporates three key components: kernel space alignment, prediction refinement, and a CNN with multi-task learning capabilities. This unified network encodes both implicit knowledge (learned subconsciously from deep layers) and explicit knowledge (obtained from shallow layers and clear metadata), resulting in a more refined and generalized representation [158], [160]. Compared to other YOLO algorithms, YOLOR significantly improves both speed and accuracy. It achieves comparable object detection accuracy to Scaled YOLOv4 while increasing inference speed by 88%, making it one of the fastest object detection algorithms in modern computer vision. On the MS COCO dataset, YOLOR outperforms PP-YOLOv2 by 3.8% in mean average precision at the same inference speed [158].

4) DAMO-YOLO: DAMO-YOLO is developed by Alibaba's DAMO Academy, significantly enhances performance by integrating novel technologies like Neural Architecture Search (NAS), a reparameterized Generalized-FPN (RepGFPN), and lightweight head architectures (Figure 14d) with AlignedOTA label assignment and distillation enhancement [161]. Leveraging MAE-NAS, the model employs a heuristic, training-free approach to architect detection backbones under strict latency and performance constraints, generating efficient structures akin to ResNet and CSPNet [151]. The neck's design emphasizes a robust ”large neck, small head” architecture, optimizing the fusion of high-level semantic and low-level spatial features through an enhanced FPN. This approach effectively balances computational efficiency and detection accuracy, particularly notable in its deployment across various model scales, from lightweight versions for edge devices to more robust configurations for general industry applications. DAMO-YOLO's architectural prowess is showcased through impressive performance metrics, achieving mAP scores ranging from 43.6 to 51.9 on COCO datasets with relatively low latency on T4 GPUs. Moreover, the model's lightweight variants demonstrate substantial efficacy on edge devices, underscoring its adaptability and broad application potential. Such capabilities are further augmented by strategic enhancements in label assignment and knowledge distillation, addressing common challenges in object detection like label misalignment and model generalization.

5) Gold-YOLO: Gold-YOLO was developed by the team at Huawei Noah's Ark Lab to significantly enhance multi-scale feature fusion through an innovative Gather-and-Distribute (GD) mechanism [162]. This mechanism, which utilized convolution and self-attention operations, was implemented to optimize the exchange and integration of information across different levels of the feature pyramid. This approach facilitated a more effective balance between latency and detection accuracy [114]. Furthermore, an MAE-style unsupervised pretraining was incorporated into the YOLO-series for the first time, which was reported to enhance learning efficiency and overall model performance. Gold-YOLO's architecture (Figure 14e) aimed to address the limitations inherent in traditional Feature Pyramid Networks (FPNs) by preventing recursive information loss and enabling more direct and efficient feature fusion. This was achieved by a method where features from all levels were gathered to a central processing node, enhanced, and then redistributed, ensuring enriched feature maps that retained critical information across scales. The effectiveness of this novel design was demonstrated through impressive performance metrics; Gold-YOLO achieved a 39.9% AP on the COCO dataset with high throughput speeds on a T4 GPU, surpassing previous state-of-the-art models like YOLOv6-3.0-N. The contributions made by this paper were significant, as they not only enhanced the YOLO model's capabilities to handle various object sizes and complexities but also established a new benchmark for the integration of advanced neural network techniques with traditional convolutional architectures for real-time applications.

IV. Challenges and Limitations

YOLOv11:

As the most recent addition to the YOLO series, this version must address and overcome

- Challenge in Detecting Small and Rotated Objects: Despite advancements, YOLOv11 struggles with small, low-resolution objects and those with varied orientations. This limitation is due to its architectural constraints, which may not fully capture the complexities of such objects, leading to potential inaccuracies in detection.
- Susceptibility to Overfitting: YOLOv11 is prone to overfitting, particularly when trained on limited or homogeneous datasets. This overfitting can adversely affect the model's performance on new or varied datasets, indicating a need for improved generalization capabilities or training approaches.
- Computational Efficiency vs. Accuracy Trade-off: While YOLOv11 has improved computational efficiency, there remains a trade-off with accuracy, particularly in complex detection environments. This trade-off highlights the ongoing challenge of balancing speed and accuracy to fulfill the requirements of real-time applications.

YOLOv10:

- YOLOv10 has not yet seen widespread adoption in published research. Its release promises cutting-edge improvements in object detection capabilities, but the lack of extensive testing and real-world application data makes it difficult to ascertain its full potential and limitations.

- Preliminary evaluations suggest that while YOLOv10 might offer advancements in speed and accuracy, integrating it into existing systems could present challenges due to compatibility and computational demands. Potential users may hesitate to adopt this version until more comprehensive studies and benchmarks are available, which articulate its advantages over previous models.
- The expectation with YOLOv10, much like its predecessors, is that it will drive further research in object detection technologies. Its eventual widespread implementation could pave the way for addressing complex detection scenarios with higher accuracy, particularly in dynamic environments. However, as with any new technology, the adaptation phase will be crucial in understanding its practical limitations and operational challenges.

YOLOv9:

- Despite YOLOv9's enhancements in detection capabilities, it has only been featured in a handful of studies, which limits a comprehensive understanding of its performance across diverse applications. This lack of extensive validation may deter organizations from adopting it until more empirical evidence and comparative analyses establish its efficacy and efficiency over earlier versions.
- While YOLOv9 improves upon the speed and accuracy of its predecessors, it may still struggle with detecting small or overlapping objects in cluttered scenes. This is a recurring challenge in high-density environments like crowded urban areas or complex natural scenes in transportation and agriculture, where precise detection is critical for applications such as autonomous driving, wildlife monitoring and robotic fruit picking.
- Future developments for YOLOv9 could focus on enhancing its robustness in adverse conditions, such as varying weather, lighting, or occlusions. Integrating more adaptive and context-aware mechanisms could help in mitigating false positives and improving the reliability of the system under different operational conditions. The implementation of advanced training techniques such as federated learning could also be explored to enhance its adaptability and learning efficiency from decentralized data sources.

YOLOv8:

- YOLOv8 has shown significant improvements in object detection tasks, particularly in real-time applications. However, it continues to face challenges in terms of computational efficiency and resource consumption when deployed on lower-end hardware [163]. This can limit its applicability in resource-constrained environments where deploying advanced hardware solutions is not feasible [164].
- The future direction for YOLOv8 could involve optimizing its architectural design to reduce computational load without compromising detection accuracy. Enhancing its scalability to efficiently process images of varying resolutions and conditions can broaden its application scope. Moreover, incorporating adaptive scaling and context-aware training methods could potentially address the detection challenges in complex scenes, making it more robust against diverse operational challenges.

YOLOv7:

- Although YOLOv7 introduces significant improvements in detection accuracy and speed, its adoption across varied real-world applications reveals a persistent challenge in handling highly dynamic scenes. For instance, in environments with rapid motion or in scenarios involving occlusions, YOLOv7 can still experience drops in performance. The algorithm's ability to generalize across different types of blur and motion artifacts remains an area for further research and enhancement.
- The complexity of YOLOv7's architecture, while beneficial for accuracy, imposes a substantial computational burden. This makes it less ideal for deployment on edge devices or platforms with limited processing capabilities, where maintaining a balance between speed and power efficiency is crucial [165], [166]. Efforts to streamline the model for such applications without significant loss of performance are necessary.
- Looking forward, there is significant potential in expanding YOLOv7's capabilities through the integration of semi-supervised or unsupervised learning paradigms. This would enable the model to leverage unlabeled data effectively, a common challenge in the real-world where annotated datasets are often scarce or expensive to produce. Additionally, enhancing the model's resilience to adversarial attacks and variability in data quality could further solidify its utility in security-sensitive applications like surveillance and fraud detection.

YOLOv6:

- One of the notable challenges with YOLOv6 is its handling of scale variability within images, which can affect its efficacy in environments where objects appear at diverse distances from the camera. While YOLOv6 shows improved accuracy and speed over its predecessors, it sometimes struggles with small or partially occluded objects, which are common in crowded scenes or complex industrial environments [167], [168]. This limitation can be critical in applications such as automated surveillance or advanced manufacturing monitoring.
- YOLOv6, while efficient, still requires considerable computational resources when compared to other models optimized for edge devices. Its deployment in resource-constrained environments such as mobile or embedded systems often requires a trade-off between detection performance and operational efficiency. Further optimizations and model pruning are necessary to achieve the best of both worlds—real-time performance with reduced computational demands.

- Future enhancements for YOLOv6 could focus on incorporating more advanced feature extraction techniques that improve its robustness to variations in object appearance and environmental conditions. Additionally, integrating more adaptive and context-aware learning mechanisms could help overcome some of the challenges related to background clutter and similar adversities. Enhancing the model's capacity to learn from a limited number of training samples, through techniques such as few-shot learning or transfer learning, could address the scarcity of labeled training data in specialized applications.

YOLOv5:

- YOLOv5 has made significant strides in improving detection speed and accuracy, but it faces challenges in consistently detecting small objects due to its spatial resolution constraints. This is particularly evident in fields such as medical imaging or satellite image analysis, where precision is crucial for identifying fine details. Techniques such as spatial pyramid pooling or enhanced up-sampling may be needed to increase the receptive field and improve the detection of smaller objects without compromising the model's efficiency [169]–[171].
- While YOLOv5 offers faster training and inference times compared to previous versions, its deployment on edge devices is limited by high memory and processing requirements [172], [173]. Although optimized models like YOLOv5s provide a solution, they sometimes do so at the cost of detection accuracy. Optimizing network architecture through neural architecture search (NAS) could potentially offer a more balanced solution, enhancing both performance and efficiency for real-time object detection applications.
- The adaptability of YOLOv5 to varied environmental conditions and different types of data distribution remains an area for development. Future research could focus on enhancing the robustness of YOLOv5 through advanced data augmentation techniques and domain adaptation strategies. This would enable the model to maintain high accuracy levels across diverse application settings, from urban surveillance to complex natural environments, effectively handling variations in lighting, weather, and seasonal changes.

YOLOv4, YOLOv3, YOLOv2 and YOLOv1:

- While YOLOv4 introduced notable enhancements in speed and accuracy, it still exhibits performance inconsistencies across different datasets, particularly with class imbalance and the detection of rare objects. The model's high computational demand also restricts its deployment on low-power devices. Continued efforts to improve model compression and increase adaptability to varying environmental conditions are essential to extend its practical utility in diverse real-world applications.
- YOLOv3 improved upon the balance of speed and accuracy, yet it struggles with small object detection due to its grid limitation. Its computational efficiency poses challenges for deployment in resource-constrained environments, prompting research towards optimization techniques to improve efficiency without sacrificing performance. Additionally, enhancing the model's robustness to environmental variations could improve its reliability for applications like autonomous driving and urban surveillance.
- Despite the incremental improvements introduced in YOLOv2, it faces challenges in detecting small objects, balancing speed with accuracy, and maintaining relevance with the advent of more capable successors. This version's reliance on a fixed grid system hampers its ability to perform in high-precision detection tasks. Future developments may shift towards adapting YOLOv2's core strengths in new architectures that enhance its spatial resolution and dynamic scaling capabilities.

For the versions of YOLO under YOLOv5, their use may decrease and discontinue in the future as newer versions are replacing the older YOLO versions in overall performance and efficiency.

- The potential for YOLOv4, YOLOv3, and YOLOv2 in future research involves exploring adaptive mechanisms that can tailor learning rates and augment data to better handle diverse operational scenarios. Integrating these models with newer technologies like model pruning and feature fusion may address existing inefficiencies and extend their applicability to a wider range of applications.
- YOLOv1 was revolutionary for its time, introducing real-time object detection by processing the entire image at once as a single regression problem. However, it faces significant challenges in dealing with small objects due to each grid cell predicting only two boxes and the probabilities for the classes. This structure often leads to poor performance on groups of small objects that are close together, such as flocks of birds or traffic scenes with multiple vehicles at a distance. Improvements in subsequent models focus on increasing the number of predictions per grid and incorporating finer-grained feature maps to enhance small object detection.
- Another limitation of YOLOv1 is the spatial constraints of its bounding boxes. Since each cell in the grid can only predict two boxes and has limited context about its neighboring cells, the precision in localizing objects, especially those with complex or irregular shapes, is often compromised. This challenge is particularly evident in medical imaging and satellite image analysis, where the exact contours of the objects are crucial. Advances in convolutional neural network designs and cross-layer feature integration in later versions seek to address these drawbacks.
- Although YOLOv1 laid the groundwork for real-time object detection, its direct usage has significantly diminished, with advancements in the field largely driven by more recent iterations such as YOLOv4 and beyond. These newer models have not only retained the core

principles of YOLOv1 but have also introduced improved mechanisms for handling diverse object sizes and aspect ratios. Current and future research is less likely to concentrate on YOLOv1 and earlier versions like YOLOv3, but rather on advancing these later iterations or developing hybrid models that might incorporate elements of YOLOv1's architecture to benefit applications where high speed and low latency are paramount, despite potential trade-offs in detection precision and detail.

- Future iterations could focus on dynamic grid systems, lighter network architectures, and advanced scaling features to tackle the challenges of small object detection and computational limitations. These improvements could enhance their deployment in emerging areas such as edge computing, where real-time processing and low power consumption are crucial.
- As newer models like YOLOv8 and YOLOv9 continue to evolve, the foundational aspects of YOLOv4, YOLOv3, and YOLOv2 can still offer valuable insights for developing hybrid models or specialized applications. Research may increasingly focus on leveraging these older versions for their speed attributes while compensating for their detection limitations through composite and hybrid modeling approaches.

A. *Challenges in Statistical Metrics for Evaluation*

Threat: Evaluating YOLO detection systems requires a unique approach, as each version, from the original YOLO to the latest YOLOv12, targets different aspects of detection capability, such as speed, accuracy, or computational efficiency. For a comprehensive evaluation, it is essential to employ a diverse array of metrics, including precision, recall, GFLOPs, and model size. This approach allows for a more complete comparison and understanding of each model's strengths and weaknesses in various real-world applications. This multi-metric evaluation is crucial to assessing the practical utility and technological advancement of the YOLO series. Future YOLO versions are expected to introduce novel evaluation metrics that capture emerging capabilities in edge computing, multi-modal fusion, and adaptive architecture optimization, necessitating an even more sophisticated evaluation framework to accurately assess their performance across diverse deployment scenarios.

Mitigation: Despite this limitation, our main premise is that the selected metrics enable us to compare various YOLO systems and adequately assess their overall effectiveness. Recognizing the inherent limitations of statistical summaries is crucial when conducting a comprehensive evaluation of detection systems across different applications. Therefore, we aim to improve the clarity and reliability of our review by openly acknowledging these potential threats to construct validity. This approach provides a more nuanced understanding of the limitations associated with various aspects of YOLO techniques for object detection in diverse domains.

Spectral versus RGB Images: Beyond traditional RGB imaging, spectral features encompass a broader spectrum, including infrared, ultraviolet, and even multispectral and hyperspectral imaging. These advanced spectral techniques can significantly enhance YOLO's object detection capabilities by providing additional information not visible in the RGB spectrum. For example, hyperspectral imaging can detect subtle variations in plant health for agricultural applications or distinguish between materials based on their spectral signatures in industrial settings. This expansion into wider spectral data not only improves detection accuracy but also opens up new avenues for application-specific optimizations, reinforcing YOLO's versatility and potential across various fields.

Over the past decade, the series of YOLO models have significantly impacted various sectors, demonstrating the powerful capabilities of deep learning in real-world applications. As a pioneering object detection algorithm, YOLO has facilitated rapid advancements across diverse fields by offering high-speed, real-time detection with commendable accuracy. One of the most notable applications has been in public safety and surveillance, where YOLO models have improved the efficacy of monitoring systems, enhancing the detection of suspicious activities and ensuring public safety more efficiently. In the realm of automotive technology, YOLO has been integral in developing advanced driver-assistance systems (ADAS) [13], contributing to object detection that supports collision avoidance systems and pedestrian safety. Furthermore, YOLO has transformed the healthcare sector by accelerating medical image analysis, enabling quicker and more accurate detection of pathologies which is critical for diagnostics and treatment planning. In industrial settings, YOLO has optimized quality control processes by identifying defects in manufacturing lines in real-time, thereby reducing waste and increasing production efficiency. Additionally, in the retail sector, YOLO has supported inventory management through automated checkouts and stock monitoring, enhancing customer experience and operational efficiency, whereas in agriculture, YOLO has played a key role to enhance timely crop stress detection, pest localization and precision crop management while improving worker health and safety.

V. Future Directions in Object Detection with YOLO

A. *YOLO deployment on Edge and IoT Devices*

The deployment of YOLO on edge devices unlocks several promising avenues for future research and development. One potential direction involves enhancing the algorithm's efficiency and accuracy for even more constrained environments, such as ultra-low-power Microcontrollers and embedded systems. This can be achieved through further optimization techniques, including model pruning, quantization, and the development of specialized hardware accelerators. Additionally, integrating YOLO with advanced communication protocols, edge computing frameworks and IoT devices could facilitate more seamless collaboration between edge devices and centralized cloud services, enhancing the overall system performance and scalability. Exploring the integration of YOLO with other

AI-driven functionalities, such as anomaly detection and predictive analytics, may unlock new applications in areas like healthcare, smart cities, and industrial automation. As edge computing continues to evolve, the adaptation of YOLO to support federated learning paradigms could ensure the data privacy while enabling continuous learning and improvement of object detection models. These future directions will not only expand the capabilities of YOLO but also contribute significantly to the advancement of intelligent edge computing systems [174]–[177].

B. *YOLO and Embodied Artificial Intelligence*

Embodied Artificial Intelligence (EAI) refers to AI systems integrated with physical entities or bodies, enabling them to interact with the real world in a natural and human-like manner [178]. Incorporating YOLO into these systems significantly enhances their sensory capabilities, allowing for more efficient and accurate interaction with the physical environment. Applications of YOLO in EAI include autonomous vehicles, drones, robots [179], human-robot interaction [180]. Additionally, it plays a significant role in healthcare, particularly with robotic surgical assistants [181], among other innovative uses [182].

VI. Expanding YOLO Object Detection into broader AI Domains

A. *YOLO and Artificial General Intelligence*

Artificial General Intelligence (AGI) refers to an intelligent agent with human-level or higher intelligence, capable of solving a variety of complex problems in diverse domains [183], [184]. In this context, an AGI system would need to integrate object detection capabilities, similar to those provided by YOLO, with other essential cognitive functions, such as advanced natural language understanding, reasoning, and decision-making. This fusion will enable the system to handle a broad spectrum of tasks in real-time, adapting to dynamic environments and complex scenarios effectively, thus advancing the current AI systems towards achieving a true AGI.

B. *YOLO Integration with Large Language Models*

One effective way to advance AGI capabilities is to integrate YOLO with Large Language Models (LLMs) by seamlessly merging advanced visual data interpretation with sophisticated natural language understanding, reasoning, and contextual awareness. This fusion would allow the AGI to not only recognize and analyze objects in real-time but also engage in meaningful interactions with stakeholders (or end users), making more informed decisions and adapting to complex tasks with greater autonomy and precision. This synergy would enable AGI systems to operate in complex environments handling multi-modal inputs simultaneously, such as navigating through an environment using visual cues identified by YOLO while interpreting and acting on spoken commands through capabilities provided by LLMs [112], [185], [186]. Such integration is expected to lead to a highly versatile and intelligent system, capable of performing real-time, multifaceted operations across diverse application domains. By combining advanced visual recognition with robust language processing, it brings us a step closer to realizing true AGI, where the system can autonomously adapt, learn, and perform complex tasks with human-like flexibility and reasoning.

VII. YOLO and Environmental Impact

Training and retraining YOLO is extremely energy-intensive, leading to substantial energy and water consumption, as well as significant carbon dioxide emissions. This environmental impact underscores concerns about the sustainability of AI development, emphasizing the urgent need for more efficient practices to reduce the ecological footprint of large-scale model training [187], [188].

VIII. Conclusion

In this comprehensive review, we explored the evolution of the YOLO models from the most recent YOLOv12 to the inaugural YOLOv1, including alternative versions of YOLO as YOLO-NAS, YOLO-X, YOLO-R, DAMO-YOLO, and Gold-YOLO. This retrospective analysis covered a decade of advancements, highlighting theapplied use of each version and their respective impacts across five critical application areas: autonomous vehicles and traffic safety, healthcare and medical imaging, security and surveillance, manufacturing, and agriculture. Our review outlined the significant enhancements in detection speed, accuracy, and computational efficiency that each iteration brought, while also addressing the specific challenges and limitations faced by earlier versions. Furthermore, we identified gaps in the current capabilities of YOLO models and proposed potential directions for future research, such as trade-off between detection speed versus accuracy, handling small and overlapping Objects, and generalization across diverse datasets and domains. Predicting the trajectory of YOLO's development, we anticipate a shift towards multimodal data processing, leveraging advancements in large language models and natural language processing to enhance object detection systems. This fusion is expected to broaden the utility of YOLO models, enabling more sophisticated, context-aware applications that could revolutionize the interaction between AI systems and their environments using Generative AI and multi-modal LLMs. Thus, this review not only serves as a detailed chronicle of YOLO's evolution but also sets a prospective blueprint for its integration into the next generation of technological innovations.

IX. Authors Contribution

Ranjan Sapkota: principal conceptualizer, research design, formal analysis, original draft preparation, manuscript writing, and editing. **Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, Hong Yan:** methodology refinement, critical revisions, manuscript review, and editing. **Manoj Karkee**: Funding and supervision, methodology refinement, critical

revisions, manuscript review, and editing. **Ranjan Sapkota and Manoj Karkee:** Corresponding authors.

X. Acknowledgement

This work was supported by the National Science Foundation and United States Department of Agriculture, National Institute of Food and Agriculture through the "Artificial Intelligence (AI) Institute for Agriculture" Program under Award AWD003473. Additionally, this work is partially supported by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA) and by EcuTS2025 from UFA-ESPE.

Our more research on CV and YOLO [79], [189], [112], [190], [191], [192], [193], [194], [195], [196]

References

[1] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261–318, 2020.

[2] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review," *Computers and Electronics in Agriculture*, vol. 223, p. 109090, Aug. 2024.

[3] H. M. Ahmad and A. Rahimi, "Deep learning methods for object detection in smart manufacturing: A survey," *Journal of Manufacturing Systems*, vol. 64, pp. 181–196, 2022.

[4] C. Gheorghe, M. Duguleana, R. G. Boboc, and C. C. Postelnicu, "Analyzing real-time object detection with yolo algorithm in automotive applications: A review," *CMES - Computer Modeling in Engineering and Sciences*, vol. 141, no. 3, p. 1939 – 1981, 2024. Cited by: 0.

[5] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul, "A survey: object detection methods from cnn to transformer," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21353–21383, 2023.

[6] R. A. S. Fernandez, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy, "Natural user interfaces for human-drone multi-modal interaction," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1013–1022, IEEE, 2016.

[7] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *Advances in neural information processing systems*, vol. 31, 2018.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[9] J. Tang, C. Ye, X. Zhou, and L. Xu, "Yolo-fusion and internet of things: Advancing object detection in smart transportation," *Alexandria Engineering Journal*, vol. 107, pp. 1–12, 2024.

[10] H. Chen and J. Guan, "Teacher–student behavior recognition in classroom teaching based on improved yolo-v4 and internet of things technology," *Electronics*, vol. 11, no. 23, p. 3998, 2022.

[11] M. G. Ragab, S. J. Abdulkader, A. Muneer, A. Alqushaibi, E. H. Sumiea, R. Qureshi, S. M. Al-Selwi, and H. Alhussian, "A comprehensive systematic review of yolo for medical object detection (2018 to 2023)," *IEEE Access*, 2024.

[12] D. Flippo, S. Gunturu, C. Baldwin, and C. Badgujar, "Tree Trunk Detection of Eastern Red Cedar in Rangeland Environment with Deep Learning Technique," *Croatian journal of forest engineering*, vol. 44, no. 2, pp. 357–368, 2023.

[13] V. Malligere Shivanna and J.-I. Guo, "Object detection, recognition, and tracking algorithms for adass—a study on recent trends," *Sensors*, vol. 24, no. 1, 2024.

[14] M. Flores-Calero, C. A. Astudillo, D. Guevara, J. Maza, B. S. Lita, B. Defaz, J. S. Ante, D. Zabala-Blanco, and J. M. Armingol Moreno, "Traffic sign detection and recognition using yolo object detection algorithm: A systematic review," *Mathematics*, vol. 12, no. 2, 2024. Cited by: 18; All Open Access, Gold Open Access.

[15] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[16] M. Shoman, D. Wang, A. Aboah, and M. Abdel-Aty, "Enhancing traffic safety with parallel dense video captioning for end-to-end event analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 7125–7133, June 2024.

[17] M. Hnewa and H. Radha, "Integrated multiscale domain adaptive yolo," *IEEE Transactions on Image Processing*, vol. 32, pp. 1857–1867, 2023.

[18] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2018.

[19] M. Shoman, G. Lanzaro, T. Sayed, and S. Gargoum, "Autonomous vehicle–pedestrian interaction modeling platform: A case study in four major cities," *Journal of Transportation Engineering, Part A: Systems*, vol. 150, no. 9, p. 04024045, 2024.

[20] M. Kaushal, B. S. Khehra, and A. Sharma, "Soft computing based object detection and tracking approaches: State-of-the-art survey," *Applied Soft Computing*, vol. 70, pp. 423–464, 2018.

[21] J. Xiang, H. Fan, H. Liao, J. Xu, W. Sun, and S. Yu, "Moving object detection and shadow removing under changing illumination condition," *Mathematical problems in Engineering*, vol. 2014, no. 1, p. 827461, 2014.

[22] Y. Xiao, A. Jiang, J. Ye, and M.-W. Wang, "Making of night vision: Object detection under low-illumination," *IEEE Access*, vol. 8, pp. 123075–123086, 2020.

[23] S. Seoni, A. Shahini, K. M. Meiburger, F. Marzola, G. Rotunno, U. R. Acharya, F. Molinari, and M. Salvi, "All you need is data preparation: A systematic review of image harmonization techniques in multi-center/device studies for medical support systems," *Computer Methods and Programs in Biomedicine*, p. 108200, 2024.

[24] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 3, pp. 505–519, 2008.

[25] T. Mostafa, S. J. Chowdhury, M. K. Rhaman, and M. G. R. Alam, "Occluded object detection for autonomous vehicles employing yolov5, yolox and faster r-cnn," in *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0405–0410, IEEE, 2022.

[26] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, 2021.

[27] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[28] K. Park, T. Patten, J. Prankl, and M. Vincze, "Multi-task template matching for object detection, segmentation and pose estimation using depth images," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7207–7213, IEEE, 2019.

[29] S. Liu, D. Liu, G. Srivastava, D. Połap, and M. Woźniak, "Overview and methods of correlation filter algorithms in object tracking," *Complex & Intelligent Systems*, vol. 7, pp. 1895–1917, 2021.

[30] M. Teutsch and W. Kruger, "Robust and fast detection of moving vehicles in aerial videos using sliding windows," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 26–34, 2015.

[31] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings. international conference on image processing*, vol. 1, pp. I–I, IEEE, 2002.

[32] G. Jun-Feng and L. Yu-Pin, "A comprehensive study for asymmetric adaboost and its application in object detection," *Acta Automatica Sinica*, vol. 35, no. 11, pp. 1403–1409, 2009.

[33] Q. Li, U. Niaz, and B. Merialdo, "An improved algorithm on viola-jones object detector," in *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, IEEE, 2012.

[34] X.-d. Hu, X.-q. Wang, F.-j. Meng, X. Hua, Y.-j. Yan, Y.-y. Li, J. Huang, and X.-l. Jiang, "Gabor-CNN for object detection based on small samples," *Defence Technology*, vol. 16, no. 6, pp. 1116–1129, 2020.

[35] T. Surasak, I. Takahiro, C.-h. Cheng, C.-e. Wang, and P.-y. Sheng, "Histogram of oriented gradients for human detection in video," in *2018 5th International conference on business and industrial research (ICBIR)*, pp. 172–176, IEEE, 2018.

[36] M. S. Karis, N. R. A. Razif, N. M. Ali, M. A. Rosli, M. S. M. Aras, and M. M. Ghazaly, "Local binary pattern (lbp) with application to variant object detection: A survey and method," in *2016 IEEE*

12th international colloquium on signal processing & its applications (CSPA), pp. 221–226, IEEE, 2016.

[37] T. Mita, T. Kaneko, and O. Hori, "Joint haar-like features for face detection," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, pp. 1619–1626, IEEE, 2005.

[38] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2497–2504, 2014.

[39] P. Piccinini, A. Prati, and R. Cucchiara, "Real-time object detection and localization with sift-based clustering," *Image and Vision Computing*, vol. 30, no. 8, pp. 573–587, 2012.

[40] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, and L. Shapiro, "Principal curvature-based region detector for object recognition," in *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2007.

[41] J. Li and Y. Zhang, "Learning surf cascade for fast and accurate object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3468–3475, 2013.

[42] H.-J. Chiu, T.-H. S. Li, and P.-H. Kuo, "Breast cancer–detection system using pca, multilayer perceptron, transfer learning, and support vector machine," *IEEE Access*, vol. 8, pp. 204309–204324, 2020.

[43] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, p. 99129 – 99149, 2022. Cited by: 170; All Open Access, Gold Open Access.

[44] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE winter conference on applications of computer vision*, pp. 75–82, IEEE, 2014.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[46] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented r-cnn for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3520–3529, 2021.

[47] S. Tang and Y. Yuan, "Object detection based on convolutional neural network," in *International Conference-IEEE–2016*, 2015.

[48] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," in *2017 36th Chinese control conference (CCC)*, pp. 11104–11109, IEEE, 2017.

[49] X. Li, D. Song, and Y. Dong, "Hierarchical feature fusion network for salient object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 9165–9175, 2020.

[50] E. Crawford and J. Pineau, "Spatially invariant unsupervised object detection with convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3412–3420, 2019.

[51] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[52] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[53] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*, pp. 280–296, Springer, 2022.

[54] R. Girshick, J. Donahue, T. Darrell, J. Malik, and E. Mercan, "R-cnn for object detection," in *IEEE Conference*, 2014.

[55] S. Bhat, K. A. Shenoy, M. R. Jain, and K. Manasvi, "Detecting crops and weeds in fields using YOLOv6 and Faster R-CNN object detection models," in *2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS)*, pp. 43–48, IEEE, 2023.

[56] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[57] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016.

[58] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[59] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[60] C. Wang, Q. Sun, X. Dong, and J. Chen, "Automotive adhesive defect detection based on improved YOLOv8," *Signal, Image and Video Processing*, pp. 1–13, 2024.

[61] M. Shoman, T. Ghoul, G. Lanzaro, T. Alsharif, S. Gargoum, and T. Sayed, "Enforcing traffic safety: A deep learning approach for detecting motorcyclists' helmet violations using yolov8 and deep convolutional generative adversarial network-generated images," *Algorithms*, vol. 17, no. 5, 2024.

[62] G. S. Patel, A. A. Desai, Y. Y. Kamble, G. V. Pujari, P. A. Chougule, and V. A. Jujare, "Identification and separation of medicine through robot using YOLO and CNN algorithms for healthcare," in *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI)*, vol. 1, pp. 1–5, IEEE, 2023.

[63] Y. Luo, Y. Zhang, X. Sun, H. Dai, X. Chen, *et al.*, "Intelligent solutions in chest abnormality detection based on yolov5 and resnet50," *Journal of healthcare engineering*, vol. 2021, 2021.

[64] A. Salinas-Medina and A. Neme, "Enhancing hospital efficiency through web-deployed object detection: A yolov8-based approach for automating healthcare operations," in *2023 Mexican International Conference on Computer Science (ENC)*, pp. 1–6, IEEE, 2023.

[65] D.-L. Pham, T.-W. Chang, *et al.*, "A yolo-based real-time packaging defect detection system," *Procedia Computer Science*, vol. 217, pp. 886–894, 2023.

[66] J. Klarák, R. Andok, P. Malík, I. Kuric, M. Ritomskỳ, I. Klačková, and H.-Y. Tsai, "From anomaly detection to defect classification," *Sensors*, vol. 24, no. 2, p. 429, 2024.

[67] M. A. Arroyo, M. T. I. Ziad, H. Kobayashi, J. Yang, and S. Sethumadhavan, "YOLO: frequently resetting cyber-physical systems for security," in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*, vol. 11009, pp. 166–183, SPIE, 2019.

[68] N. Bordoloi, A. K. Talukdar, and K. K. Sarma, "Suspicious activity detection from videos using yolov3," in *2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1–5, IEEE, 2020.

[69] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with you look only once (YOLO) algorithm: A bibliometric and systematic literature review," *arXiv preprint arXiv:2401.10379*, 2024.

[70] J. Li, Y. Qiao, S. Liu, J. Zhang, Z. Yang, and M. Wang, "An improved YOLOv5-based vegetable disease detection method," *Computers and Electronics in Agriculture*, vol. 202, p. 107345, 2022.

[71] L. Fu, Y. Feng, J. Wu, Z. Liu, F. Gao, Y. Majeed, A. Al-Mallahi, Q. Zhang, R. Li, and Y. Cui, "Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model," *Precision Agriculture*, vol. 22, pp. 754–776, 2021.

[72] Y. Zhong, J. Gao, Q. Lei, and Y. Zhou, "A vision-based counting and recognition system for flying insects in intelligent agriculture," *Sensors*, vol. 18, no. 5, p. 1489, 2018.

[73] Y. Wang, L. Yang, H. Chen, A. Hussain, C. Ma, and M. Al-gabri, "Mushroom-yolo: A deep learning algorithm for mushroom growth recognition based on improved yolov5 in agriculture 4.0," in *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*, pp. 239–244, IEEE, 2022.

[74] K. Jiang, T. Xie, R. Yan, X. Wen, D. Li, H. Jiang, N. Jiang, L. Feng, X. Duan, and J. Wang, "An attention mechanism-improved yolov7 object detection algorithm for hemp duck count estimation," *Agriculture*, vol. 12, no. 10, p. 1659, 2022.

[75] G. Chen, Y. Hou, T. Cui, H. Li, F. Shangguan, and L. Cao, "YOLOv8-CML: A lightweight target detection method for color-changing melon ripening in intelligent agriculture," *ResearchSquare*, 2023.

[76] X. Yu, D. Yin, H. Xu, F. Pinto Espinosa, U. Schmidhalter, C. Nie, Y. Bai, S. Sankaran, B. Ming, N. Cui, *et al.*, "Maize tassel number and tasseling stage monitoring based on near-ground and uav rgb images by improved yolov8," *Precision Agriculture*, pp. 1–39, 2024.

[77] L. Jia, T. Wang, Y. Chen, Y. Zang, X. Li, H. Shi, and L. Gao, "Mobilenet-ca-yolo: An improved yolov7 based on the mobilenetv3 and attention mechanism for rice pests and diseases detection," *Agriculture*, vol. 13, no. 7, p. 1285, 2023.

[78] M. Umar, S. Altaf, S. Ahmad, H. Mahmoud, A. S. N. Mohamed, and R. Ayub, "Precision agriculture through deep learning: Tomato plant multiple diseases recognition with cnn and improved yolov7," *IEEE Access*, 2024.

[79] R. Sapkota, D. Ahmed, and M. Karkee, "Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments," *Artificial Intelligence in Agriculture*, 2024.

[80] R. Li and J. Yang, "Improved YOLOv2 object detection model," in *2018 6th international conference on multimedia computing and systems (ICMCS)*, pp. 1–6, IEEE, 2018.

[81] H. Nakahara, H. Yonekawa, T. Fujii, and S. Sato, "A lightweight YOLOv2: A binarized CNN with a parallel support vector regression for an FPGA," in *Proceedings of the 2018 ACM/SIGDA International Symposium on field-programmable gate arrays*, pp. 31–40, 2018.

[82] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection," in *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pp. 1–6, IEEE, 2018.

[83] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, 2022.

[84] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms," *Agronomy*, vol. 12, no. 2, p. 319, 2022.

[85] N. Mohod, P. Agrawal, and V. Madaan, "YOLOv4 vs YOLOv5: Object detection on surveillance videos," in *International Conference on Advanced Network Technologies and Intelligent Computing*, pp. 654–665, Springer, 2022.

[86] "Ultralytics," 2020. Accessed on 2024-12-31.

[87] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[88] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022. Accessed: 2024-06-05.

[89] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[90] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics, yolov 8," 2023. Accessed on 2024-12-31.

[91] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, 2024.

[92] "Ultralytics, yolov 9," 2023. Accessed on 2024-12-31.

[93] "Ultralytics, yolov10: Real-time end-to-end object detection," 2023. Accessed on 2024-12-31.

[94] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[95] "Ultralytics, yolo11 new," 2024. Accessed on 2024-12-31.

[96] H. Mao, X. Yang, and W. J. Dally, "A delay metric for video object detection: What average precision fails to tell," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 573–582, 2019.

[97] B. Chen, G. Ghiasi, H. Liu, T.-Y. Lin, D. Kalenichenko, H. Adam, and Q. V. Le, "Mnasfpn: Learning latency-aware pyramid architecture for object detection on mobile devices," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13607–13616, 2020.

[98] D. Pestana, P. R. Miranda, J. D. Lopes, R. P. Duarte, M. P. Véstias, H. C. Neto, and J. T. De Sousa, "A full featured configurable accelerator for object detection with yolo," *IEEE Access*, vol. 9, pp. 75864–75877, 2021.

[99] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 528–537, 2018.

[100] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, "Probabilistic object detection: Definition and evaluation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1031–1040, 2020.

[101] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*, pp. 345–359, Springer, 2005.

[102] Z. Liang, Z. Zhang, M. Zhang, X. Zhao, and S. Pu, "Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7140–7149, 2021.

[103] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[104] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4203–4212, 2018.

[105] L. Cui, R. Ma, P. Lv, X. Jiang, Z. Gao, B. Zhou, and M. Xu, "Mdssd: multi-scale deconvolutional single shot detector for small objects," *arXiv preprint arXiv:1805.07009*, 2018.

[106] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 988–996, 2017.

[107] J. Jiang, H. Xu, S. Zhang, and Y. Fang, "Object detection algorithm based on multiheaded attention," *Applied Sciences*, vol. 9, no. 9, p. 1829, 2019.

[108] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 797–813, 2018.

[109] Z. Li, L. Yang, and F. Zhou, "Fssd: feature fusion single shot multibox detector," *arXiv preprint arXiv:1712.00960*, 2017.

[110] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.

[111] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *arXiv preprint arXiv:2211.04800*, 2022.

[112] R. Sapkota, Z. Meng, and M. Karkee, "Synthetic meets authentic: Leveraging llm generated datasets for yolo11 and yolov10-based apple detection through machine vision sensors," *Smart Agricultural Technology*, vol. 9, p. 100614, 2024.

[113] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024.

[114] C.-Y. Wang, H.-Y. M. Liao, *et al.*, "Yolov1 to yolov10: The fastest and most accurate real-time object detection systems," *APSIPA Transactions on Signal and Information Processing*, vol. 13, no. 1, 2024.

[115] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.

[116] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part I 12*, pp. 290–306, Springer, 2015.

[117] S. Li, M. Li, R. Li, C. He, and L. Zhang, "One-to-few label assignment for end-to-end dense detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7350–7359, 2023.

[118] Y. Tian, N. Deng, J. Xu, and Z. Wen, "A fine-grained dataset for sewage outfalls objective detection in natural environments," *Scientific Data*, vol. 11, no. 1, p. 724, 2024.

[119] S. Bhagat, M. Kokare, V. Haswani, P. Hambarde, and R. Kamble, "Wheatnet-lite: A novel light weight network for wheat head detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1332–1341, 2021.

[120] Y. Hu, W. Tan, F. Meng, and Y. Liang, "A decoupled spatial-channel inverted bottleneck for image compression," in *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 1740–1744, IEEE, 2023.

[121] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention," *Agronomy*, vol. 13, no. 7, p. 1824, 2023.

[122] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[123] G. Jocher *et al.*, "Yolov8: A comprehensive improvement of the yolo object detection series." https://docs.ultralytics.com/yolov8/, 2022. Accessed: 2024-06-05.

[124] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in *2015 ieee information theory workshop (itw)*, pp. 1–5, IEEE, 2015.

[125] B. Zhang, J. Li, Y. Bai, Q. Jiang, B. Yan, and Z. Wang, “An improved microaneurysm detection model based on swinir and yolov8,” *Bioengineering*, vol. 10, no. 12, p. 1405, 2023.

[126] C.-T. Chien, R.-Y. Ju, K.-Y. Chou, and J.-S. Chiang, “Yolov9 for fracture detection in pediatric wrist trauma x-ray images,” *arXiv preprint arXiv:2403.11249*, 2024.

[127] Ultralytics, “Home — docs.ultralytics.com.” https://docs.ultralytics.com/. [Accessed 28-05-2024].

[128] Ultralytics, “YOLOv8 Object Detection Model: What is, How to Use — roboflow.com.” https://roboflow.com/model/yolov8. [Accessed 28-05-2024].

[129] Ultralytics, “Ultralytics YOLOv8 Solutions: Quick Walkthrough — ultralytics.medium.com.” https://ultralytics.medium.com/ultralytics-yolov8-solutions-quick-walkthrough-b802fd6da5d7. [Accessed 28-05-2024].

[130] S. Du, B. Zhang, P. Zhang, and P. Xiang, “An improved bounding box regression loss function based on ciou loss for multi-scale object detection,” in *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, pp. 92–98, IEEE, 2021.

[131] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, *et al.*, “Pp-yoloe: An evolved version of yolo,” *arXiv preprint arXiv:2203.16250*, 2022.

[132] X. Yue, K. Qi, X. Na, Y. Zhang, Y. Liu, and C. Liu, “Improved yolov8-seg network for instance segmentation of healthy and diseased tomato plants in the growth stage,” *Agriculture*, vol. 13, no. 8, p. 1643, 2023.

[133] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2778–2788, 2021.

[134] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[135] Z. Bai, X. Pei, Z. Qiao, G. Wu, and Y. Bai, “Improved yolov7 target detection algorithm based on uav aerial photography,” *Drones*, vol. 8, no. 3, p. 104, 2024.

[136] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, “Statistical analysis of design aspects of various yolo-based deep learning models for object detection,” *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126, 2023.

[137] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “Panet: Few-shot image semantic segmentation with prototype alignment,” in *proceedings of the IEEE/CVF international conference on computer vision*, pp. 9197–9206, 2019.

[138] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, “Vit-yolo: Transformer-based yolo for object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2799–2808, 2021.

[139] Ultralytics, “Comprehensive Guide to Ultralytics YOLOv5 — docs.ultralytics.com.” https://docs.ultralytics.com/yolov5/. [Accessed 28-05-2024].

[140] Ultralytics, “GitHub - ultralytics/yolov5: YOLOv5 in PyTorch ¿ ONNX ¿ CoreML ¿ TFLite — github.com.” https://github.com/ultralytics/yolov5. [Accessed 28-05-2024].

[141] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[142] M. Mahasin and I. A. Dewi, “Comparison of cspdarknet53, cspresnext-50, and efficientnet-b0 backbones on yolo v4 as object detector,” *International Journal of Engineering, Science and Information Technology*, vol. 2, no. 3, pp. 64–72, 2022.

[143] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[144] D. Misra, “Mish: A self regularized non-monotonic activation function,” *arXiv preprint arXiv:1908.08681*, 2019.

[145] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

[146] G. Ghiasi, T.-Y. Lin, and Q. V. Le, “Dropblock: A regularization method for convolutional networks,” *Advances in neural information processing systems*, vol. 31, 2018.

[147] R. Müller, S. Kornblith, and G. E. Hinton, “When does label smoothing help?,” *Advances in neural information processing systems*, vol. 32, 2019.

[148] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of freebies for training object detection neural networks,” *arXiv preprint arXiv:1902.04103*, 2019.

[149] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[150] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv preprint arXiv:1506.02640*, 2016. Accessed: 2024-06-05.

[151] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[152] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, “A comprehensive survey of neural architecture search: Challenges and solutions,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–34, 2021.

[153] M. Mithun and S. J. Jawhar, “Detection and classification on mri images of brain tumor using yolo nas deep learning model,” *Journal of Radiation Research and Applied Sciences*, vol. 17, no. 4, p. 101113, 2024.

[154] Z. Ge, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.

[155] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, “Complete and accurate holly fruits counting using yolox object detection,” *Computers and Electronics in Agriculture*, vol. 198, p. 107062, 2022.

[156] J. Liu and W. Sun, “Yolox-based ship target detection for shore-based monitoring,” in *Proceedings of the 2022 5th International Conference on Signal Processing and Machine Learning*, pp. 234–241, 2022.

[157] I. Ashraf, S. Hur, G. Kim, and Y. Park, “Analyzing performance of yolox for detecting vehicles in bad weather conditions,” *Sensors*, vol. 24, no. 2, p. 522, 2024.

[158] H.-S. Chang, C.-Y. Wang, R. R. Wang, G. Chou, and H.-Y. M. Liao, “Yolor-based multi-task learning,” *arXiv preprint arXiv:2309.16921*, 2023.

[159] T. Andrei-Alexandru, C. Cosmin, S. Ioan, T. Adrian-Alexandru, and D. E. Henrietta, “Novel ceramic plate defect detection using yolo-r,” in *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6, IEEE, 2022.

[160] H. Sun, D. Lu, X. Li, J. Tan, J. Zhao, and D. Hou, “Research on multi-apparent defects detection of concrete bridges based on yolor,” in *Structures*, vol. 65, p. 106735, Elsevier, 2024.

[161] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, and X. Sun, “Damo-yolo: A report on real-time object detection design,” *arXiv preprint arXiv:2211.15444*, 2022.

[162] C. Wang, W. He, Y. Nie, J. Guo, C. Liu, Y. Wang, and K. Han, “Gold-yolo: Efficient object detector via gather-and-distribute mechanism,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[163] R. Ye, Q. Gao, Y. Qian, J. Sun, and T. Li, “Improved yolov8 and sahi model for the collaborative detection of small targets at the micro scale: A case study of pest detection in tea,” *Agronomy*, vol. 14, no. 5, p. 1034, 2024.

[164] E. Soylu and T. Soylu, “A performance comparison of yolov8 models for traffic sign detection in the robotaxi-full scale autonomous vehicle competition,” *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 25005–25035, 2024.

[165] O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, “A comparative study of yolov5 and yolov7 object detection algorithms,” *Journal of Computing and Social Informatics*, vol. 2, no. 1, pp. 1–12, 2023.

[166] N. AFFES, J. KTARI, N. BEN AMOR, T. FRIKHA, and H. HAMAM, “Comparison of YOLOV5, YOLOV6, YOLOV7 and YOLOV8 for intelligent video surveillance.,” *Journal of Information Assurance & Security*, vol. 18, no. 5, 2023.

[167] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y.-I. Cho, “A yolov6-based improved fire detection approach for smart city environments,” *Sensors*, vol. 23, no. 6, p. 3161, 2023.

[168] N. Li, M. Wang, G. Yang, B. Li, B. Yuan, and S. Xu, “Dens-yolov6: A small object detection model for garbage detection on water surface,” *Multimedia Tools and Applications*, pp. 1–21, 2023.

[169] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "Yolo-z: Improving small object detection in yolov5 for autonomous vehicles," *arXiv preprint arXiv:2112.11798*, 2021.

[170] H.-K. Jung and G.-S. Choi, "Improved yolov5: Efficient object detection using drone images under various conditions," *Applied Sciences*, vol. 12, no. 14, p. 7255, 2022.

[171] A. Wang, T. Peng, H. Cao, Y. Xu, X. Wei, and B. Cui, "Tia-yolov5: An improved yolov5 network for real-time detection of crop and weed in the field," *Frontiers in Plant Science*, vol. 13, p. 1091655, 2022.

[172] T.-H. Wu, T.-W. Wang, and Y.-Q. Liu, "Real-time vehicle and distance detection based on improved yolo v5 network," in *2021 3rd World Symposium on Artificial Intelligence (WSAI)*, pp. 24–28, IEEE, 2021.

[173] X. Jia, Y. Tong, H. Qiao, M. Li, J. Tong, and B. Liang, "Fast and accurate object detector for autonomous driving based on improved yolov5," *Scientific reports*, vol. 13, no. 1, p. 9711, 2023.

[174] H. F. Yang, Y. Ling, C. Kopca, S. Ricord, and Y. Wang, "Cooperative traffic signal assistance system for non-motorized users and disabilities empowered by computer vision and edge artificial intelligence," *Transportation research part C: emerging technologies*, vol. 145, p. 103896, 2022.

[175] P. Ghaziamin, K. Bajaj, N. Bouguila, and Z. Patterson, "A privacy-preserving edge computing solution for real-time passenger counting at bus stops using overhead fisheye camera," in *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 25–32, IEEE, 2024.

[176] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and T. Alsboui, "Domain feature mapping with yolov7 for automated edge-based pallet racking inspections," *Sensors*, vol. 22, no. 18, p. 6927, 2022.

[177] D. Zhang, "A yolo-based approach for fire and smoke detection in iot surveillance systems.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, 2024.

[178] R. Pfeifer and F. Iida, "Embodied artificial intelligence: Trends and challenges," *Lecture notes in computer science*, pp. 1–26, 2004.

[179] N. J. Sanket, *Active vision based embodied-ai design for nano-uav autonomy*. PhD thesis, University of Maryland, College Park, 2021.

[180] T. Wang, P. Zheng, S. Li, and L. Wang, "Multimodal human–robot interaction for human-centric smart manufacturing: A survey," *Advanced Intelligent Systems*, vol. 6, no. 3, p. 2300359, 2024.

[181] A. Lakshmipathy, M. Vardhineedi, V. R. P. Sekharamahanthi, D. D. Patel, S. Saini, and S. Mohammed, "Medicaption: Integrating yolo-driven computer vision and nlp for advanced pharmaceutical package recognition and annotation," *Authorea Preprints*, 2024.

[182] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, *et al.*, "Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation," *arXiv preprint arXiv:2403.09227*, 2024.

[183] A. K. Pande, P. Brantley, M. H. Tanveer, and R. C. Voicu, "From ai to agi-the evolution of real-time systems with gpt integration," in *SoutheastCon 2024*, pp. 699–707, IEEE, 2024.

[184] Y. Qu, C. Wei, P. Du, W. Che, C. Zhang, W. Ouyang, Y. Bian, F. Xu, B. Hu, K. Du, H. Wu, J. Liu, and Q. Liu, "Integration of cognitive tasks into artificial general intelligence test for large models," *iScience*, vol. 27, 2024.

[185] A. Rouhi, D. Patiño, and D. K. Han, "Enhancing object detection by leveraging large language models for contextual knowledge," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 15317 LNCS, p. 299 – 314, 2025. Cited by: 0.

[186] R. Sapkota, A. Paudel, and M. Karkee, "Zero-shot automatic annotation and instance segmentation using llm-generated datasets: Eliminating field imaging and manual annotation for deep learning model development," *arXiv preprint arXiv:2411.11285*, 2024.

[187] R. Xu, K. Ji, Z. Yuan, C. Wang, and Y. Xia, "Exploring the evolution trend of china's digital carbon footprint: A simulation based on system dynamics approach," *Sustainability (Switzerland)*, vol. 16, no. 10, 2024. All Open Access, Gold Open Access.

[188] P. Dhar, "The carbon impact of artificial intelligence," *Nature Machine Intelligence*, vol. 2, no. 10, 2020.

[189] R. Sapkota, D. Ahmed, M. Churuvija, and M. Karkee, "Immature green apple detection and sizing in commercial orchards using yolov8 and shape fitting techniques," *IEEE Access*, vol. 12, pp. 43436–43452, 2024.

[190] R. Sapkota and M. Karkee, "Integrating yolo11 and convolution block attention module for multi-season segmentation of tree trunks and branches in commercial apple orchards," *arXiv preprint arXiv:2412.05728*, 2024.

[191] R. Sapkota and M. Karkee, "Yolo11 and vision transformers based 3d pose estimation of immature green fruits in commercial apple orchards for robotic thinning," *arXiv preprint arXiv:2410.19846*, 2024.

[192] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive performance evaluation of yolo11, yolov10, yolov9 and yolov8 on detecting and counting fruitlet in complex orchard environments," *arXiv preprint arXiv:2407.12040*, 2024.

[193] Z. Meng, X. Du, R. Sapkota, Z. Ma, and H. Cheng, "Yolov10-pose and yolov9-pose: Real-time strawberry stalk pose detection models," *Computers in Industry*, vol. 165, p. 104231, 2025.

[194] R. Sapkota and M. Karkee, "Comparing yolov11 and yolov8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment," *arXiv preprint arXiv:2410.19869*, 2024.

[195] R. Sapkota, S. Raza, M. Shoman, A. Paudel, and M. Karkee, "Image, text, and speech data augmentation using multimodal llms for deep learning: A survey," *arXiv preprint arXiv:2501.18648*, 2025.

[196] R. Sapkota, S. Raza, and M. Karkee, "Comprehensive analysis of transparency and accessibility of chatgpt, deepseek, and other sota large language models," *Preprints. org DOI:10.20944/preprints202502.1608.v1*, 2025.