International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I12P103

Volume 73 Issue 12, 24-33, December 2025
© 2025 Seventh Sense Research Group®

Original Article

A Longitudinal Study on the Evolution of YOLO Architectures: From YOLOv1 to YOLOv12

Rajaa Miftah[1], Abdessamad Belangour[2], Mostafa Hanoune[3], Sara Bouraya[4]

[1,2,3,4]*Hassan II University, Faculty of Sciences, Ben M'sik Casablanca, Morocco.*

[4]*Corresponding Author : sarabouraya95@gmail.com*

Received: 07 July 2025 Revised: 08 November 2025 Accepted: 17 November 2025 Published: 19 December 2025

Abstract *- Computer vision is a branch of artificial intelligence that allows machines to read and comprehend visual data in the world, like images and videos. Video tagging is a technique in computer vision that is used to find and label objects, actions, or scenes over multiple successive frames of a video without human input. The ability enables many applications such as surveillance, autonomous driving, content moderation, and massive video analysis. The YOLO (You Only Look Once) family of real-time object detectors is one of the approaches available, and it provides a powerful tradeoff between speed and accuracy that is essential to an effective video annotation. The paper will give a longitudinal analysis of YOLO models since version 1 all the way to version 12 with a view to how the design principles, the backbone architecture, and the feature fusion strategies of the models have changed over the years. The successive iterations had improvements to improve the precision of detection, the rate of computation, and the stability of the systems. The study is based on the history of detection techniques, beginning with simple grid-based schemes, all the way to current anchor-free and reparameterized models. The study explores how architectural innovations can be used to facilitate improved scalability and deployment to a variety of computing environments that go beyond edge devices into cloud services. The paper shows how YOLO was developed by illustrating these major improvements that justify its role in contemporary computer vision systems employed on real-time video tagging.*

Keywords *- Computer vision, Video Tagging, Yolo, One Stage detectors, Object Detection.*

1. Introduction

Over the past few years, computer vision has been one of the fastest-growing fields of artificial intelligence that allows machines to see, perceive, and react to visual objects of reality around them. The computer vision techniques can now be used in a broad range of tasks, such as autonomous driving, surveillance, medical diagnostics, and smart manufacturing, whether it comes to working with still images or with high-resolution video streams. Video tagging is significant in this regard, as it can automatically label what one can see and what one can say as it progresses over time. It is a video frame-to-video frame object, action, or event detection that allows one to develop a structure of understanding of full video sequences. The process of video indexing, scene analysis, anomaly detection, and interactive retrieval of content is just a few of the tasks that require the accomplishment of precise and real-time tagging. The YOLO (You Only Look Once) family of models has been a particularly successful object detector, particularly due to its capacity to trade both speed and accuracy [1-3]. YOLOv1 ushered in the concept of an integrated real-time detector, and the new direction in the field began. Newer architectures, Yolo, One Stage detectors, Object Detection, and finally YOLOv12 have since continued to make significant improvements on the architecture, including better backbone networks, multi-scale features fusion, anchor-free detection methods, and reparameterization of models (Figure 1).

These advancements have made YOLO particularly well-suited for video tagging tasks, where accurate and consistent object detection across frames is essential. With each new version, the models have become more resilient to common challenges in video analysis, such as motion blur, object occlusion, and rapidly changing backgrounds.

In addition, their lightweight design and fast inference speed make them highly effective for real-time deployment on embedded and edge devices. The current study conducts a longitudinal discussion of YOLO architectures from version 1 through version 12, explaining how their design principles, backbone networks, and feature-fusion strategies have evolved up to now.

Then, it explains how successive versions have incrementally improved detection accuracy with efficiency for real-time video-tagging applications. Although there are several related reviews, most of them normally cover only subsets of the YOLO family, such as reviews on state-of-the-art versions up to YOLOv8 or YOLOv9, and often lack detailed documentation on architectural tradeoffs and performance over every iteration. This work addresses that gap by documenting major architectural innovations-trade-offs along each iteration from YOLOv1 to YOLOv12. This longitudinal perspective gives light on each specific novelty and its cumulative impact on advancing real-time object-detection systems.

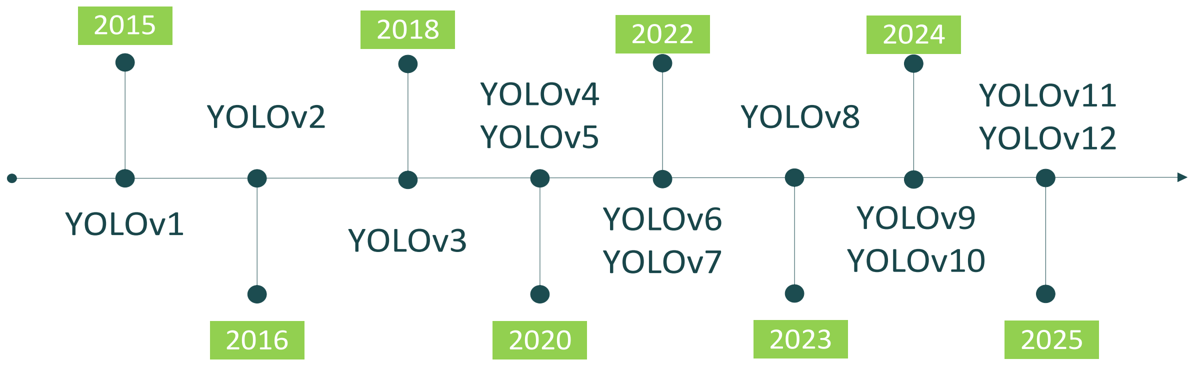

Fig. 1 Yolo models' evolution

2. Challenges in Video Tagging

Video tagging as an activity, in spite of its increasing significance, is a challenging and intricate activity, due in part to the fundamental difficulties of video data:

2.1. Temporal Consistency

Objects that are observed in more than one frame can give inconsistent detection, including loss of tags or flickering, which leads to reduced tagging reliability. Temporal coherence is also necessary in the identification of objects in video sequences since it allows the same object to be labeled consistently across the whole sequence to provide robust video tagging. The consistency can be maintained by using approaches like object tracking algorithms or temporal models like recurrent neural networks and Transformers that can analyze object identities across frames. Flickering can also be minimized and reliability enhanced by means of post-processing methods such as temporal smoothing.

2.2. Motion Blur and Fast Object Movement

Objects and camera movement are prone to quick movement, causing the effects of motion blur on the object and camera, making it difficult to detect and classify the object. Consequently, models might demonstrate a decreased level of confidence with their predictions or may not even identify the objects in the impacted frames. As solutions, there are motion compensation, training models on blurred images in data augmentation, training models on motion-aware architectures, and the use of motion-sensitive architectures. Cameras with a high frame rate are also capable of deblurring images in recorded videos to enhance the accuracy of detection.

2.3. Occlusions and Overlapping Objects

It is not rare in a dynamic situation to have objects partially or entirely covered by others, particularly in busy surroundings. This complicates the process of consistent tagging between frames, especially of small or fast-moving things. Strategies that can be used to address these problems are multi-object tracking, instance segmentation, and the use of temporal context to predict temporarily occluded objects.

2.4. Varying Lighting and Weather Conditions

Videos shot in real-life conditions tend to have a change in lighting, shadows, glare, or environmental characteristics like rain, fog, or low light. Such factors have the potential to lower the detection accuracy and render the reliability of assigned tags ineffective. Methods such as domain adaptation, data augmentation to simulate different lighting or weather conditions, and multi-spectral or infrared imaging can enhance robustness to such challenging situations.

2.5. Scene Complexity and Background Variation

Unlike in still pictures, the background of video scenes is usually dynamic and cluttered. Noise that is caused by frequent changes of scenes or background motion can result in false positives or missed detections. Background subtraction, attention mechanisms in deep learning models, and hierarchical approaches that combine object-level and scene-level features can help models focus on relevant objects and reduce background interference.

2.6. Scalability and Real-Time Constraints

Fast and lightweight models are demanded for processing real-time long video streams. Edge devices or mobile platforms with limited resources are exposed to deployment restrictions due to high computational requirements. High computational requirements can restrict deployment on edge devices or mobile platforms with limited resources. Lightweight architecture, model optimization techniques like pruning or quantization, and adaptive frame selection strategies can reduce computational load and enable real-time deployment. Hardware acceleration using GPUs or NPUs can further enhance performance.

2.7. Class Imbalance and Rare Events

In most video tagging systems, one or more classes of objects are significantly more common than the other classes. Models may be biased to dominant classes, and it would be more difficult to notice uncommon but important events or objects. Weighted loss functions, synthetic data generation of rare classes, and active learning are some of the possible approaches that can balance the datasets and enhance the detection of underrepresented objects.

2.8. Annotation Complexity

Building annotated video datasets is time and cost-intensive, especially when frame-tracked object tracking and labeling are necessary.

This restricts access to large-scale and high-quality training data for video tagging tasks. Weakly supervised and semi-supervised learning, quality-controlled crowdsourcing, and synthetic video generation can minimize the need to use manual labels and also have enough data to train powerful video tagging models.

3. Pros and Cons of YOLO Models

Despite its growing importance, video tagging is a complicated and difficult task to perform, mostly because of the natural challenges posed by video data. The following table summarizes the key strengths and weaknesses of each of the major versions of YOLO. It serves as a practical guide for researchers and practitioners to select the most suitable model based on specific project needs, whether the focus is on real-time performance, small object detection, deployment efficiency, or support for tasks such as segmentation and tracking (Table 1).

4. Architecture Evolution of YOLO Models

The YOLO (You Only Look Once) family of object detection models has transformed the way we approach real-time computer vision tasks. Introduced in 2015, YOLO's philosophy was simple yet revolutionary: detecting all objects in a single forward pass of the network. Over the years, this idea has evolved through twelve iterations, each bringing significant advancements in architecture, accuracy, and efficiency.

Table 1. Pros and Cons of YOLOv models

Model	Pros	Cons
YOLOv1	Fast, simple, end-to-end single network	Poor localization, fails on small/overlapping objects
YOLOv2	Better accuracy, multi-scale training, and 9K classes support	Still struggles with very small objects
YOLOv3	Improved small object detection, multi-scale prediction	More complex and heavier than v2
YOLOv4	High accuracy and real-time speed, many training tricks	High GPU requirements, complex architecture
YOLOv5	User-friendly, easy to train, multi-scale versions	Unofficial naming, not from original authors
YOLOv6	Efficient for deployment, low latency	Less community support, limited research documentation
YOLOv7	State-of-the-art real-time detection, efficient training tricks	Complex training pipeline
YOLOv8	Anchor-free, supports segmentation and tracking	Multiple variants can be confusing
YOLOv9	Programmable gradient head, high accuracy	Higher complexity and FLOPs
YOLOv10	NMS-free, efficient architecture	The training pipeline is more complex
YOLOv11	Unified multi-task head, good accuracy-speed tradeoff	Slightly more complex than previous versions
YOLOv12	Latest improvements, best for advanced multi-task learning	Diminishing returns vs v11, heavier compute

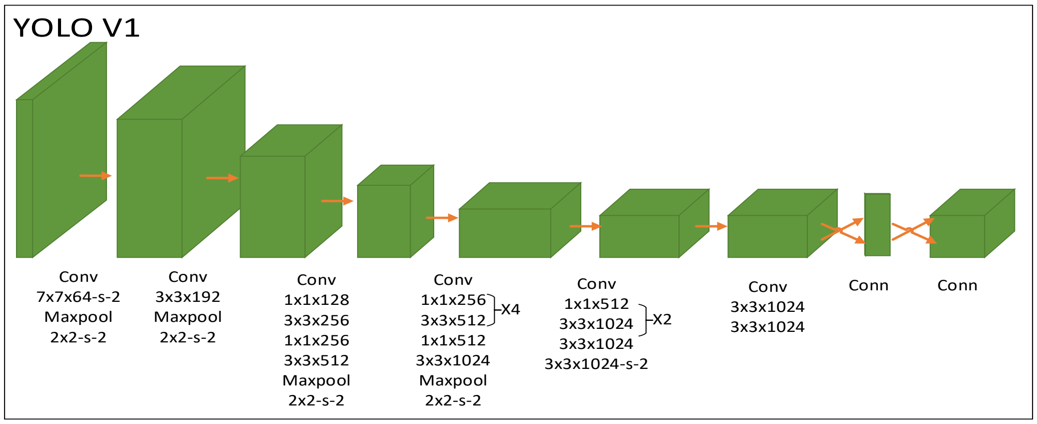
YOLO V1
Conv 7x7x64-s-2 Maxpool 2x2-s-2
Conv 3x3x192 Maxpool 2x2-s-2
Conv 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool 2x2-s-2
Conv 1x1x256 3x3x512 X4 1x1x512 3x3x1024 Maxpool 2x2-s-2
Conv 1x1x512 3x3x1024 X2 3x3x1024 3x3x1024-s-2
Conv 3x3x1024 3x3x1024
Conn
Conn

Fig. 2 YOLOV1 architecture

The first framework proposed in 2015, called YOLOv1 [4], made the radical move of framing object detection as one regression problem. It employed a tailored convolutional neural network to break the image down into a grid, and concurrently forecast the presence of a bounding box and the probability of a specific class. This gave very rapid performance, but had its own weaknesses, particularly in the detection of small or overlapping objects, during which the model performed poorly.

(Figure 2) Along with the launch of YOLOv2 in 2016 [5], there have been a number of significant changes. The model used a more robust backbone, Darknet-19, and added anchor boxes in order to improve the prediction of bounding boxes.

Further methods like batch normalization, input images of a higher resolution, enabled YOLOv2 to obtain a far richer tradeoff between speed and accuracy, which enhanced the detection of a broad range of object categories (Figure 3).

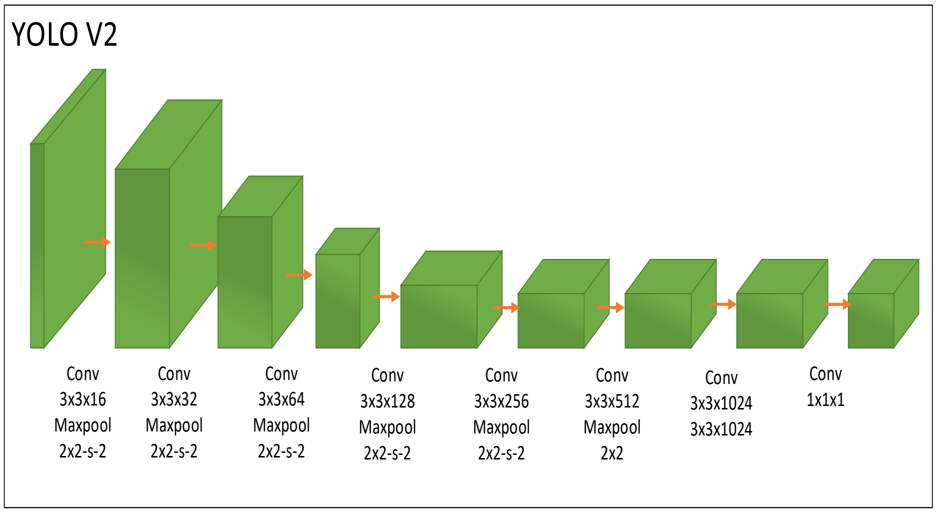
YOLO V2
Conv 3x3x16 Maxpool 2x2-s-2
Conv 3x3x32 Maxpool 2x2-s-2
Conv 3x3x64 Maxpool 2x2-s-2
Conv 3x3x128 Maxpool 2x2-s-2
Conv 3x3x256 Maxpool 2x2-s-2
Conv 3x3x512 Maxpool 2x2
Conv 3x3x1024 3x3x1024
Conv 1x1x1

Fig. 3 YOLOV2 model architecture

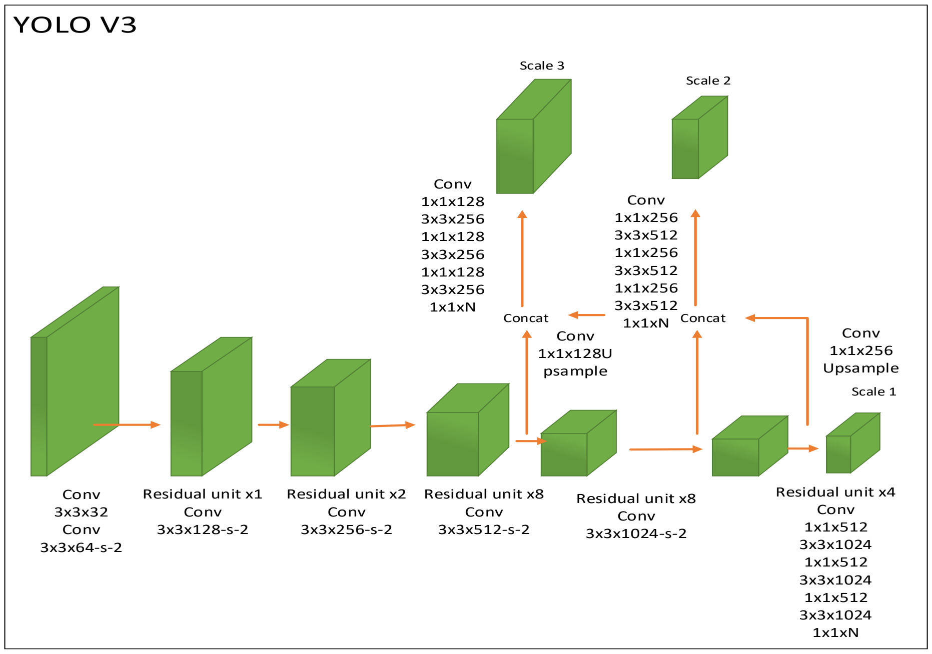
YOLO V3
Scale 3
Scale 2
Conv 1x1x128 3x3x256 1x1x128 3x3x256 1x1x128 3x3x256 1x1xN
Conv 1x1x256 3x3x512 1x1x256 3x3x512 1x1x256 3x3x512 1x1xN
Concat
Concat
Conv 1x1x128U psample
Conv 1x1x256 Upsample
Scale 1
Conv 3x3x32 Conv 3x3x64-s-2
Residual unit x1 Conv 3x3x128-s-2
Residual unit x2 Conv 3x3x256-s-2
Residual unit x8 Conv 3x3x512-s-2
Residual unit x8 Conv 3x3x1024-s-2
Residual unit x4 Conv 1x1x512 3x3x1024 1x1x512 3x3x1024 1x1x512 3x3x1024 1x1xN

Fig. 4 YOLOV3 model architecture

YOLOv3 was later released in 2018 [6], and it made the model deeper and more powerful. It had a Darknet-53 backbone, the performance of which was higher in feature extraction and also offered multi-scale prediction through a Feature Pyramid Network-like method. These improvements enabled YOLOv3 to achieve much higher accuracy on small object detection at image-vital real-time, and became popular as a version until a few years ago (Figure 4).

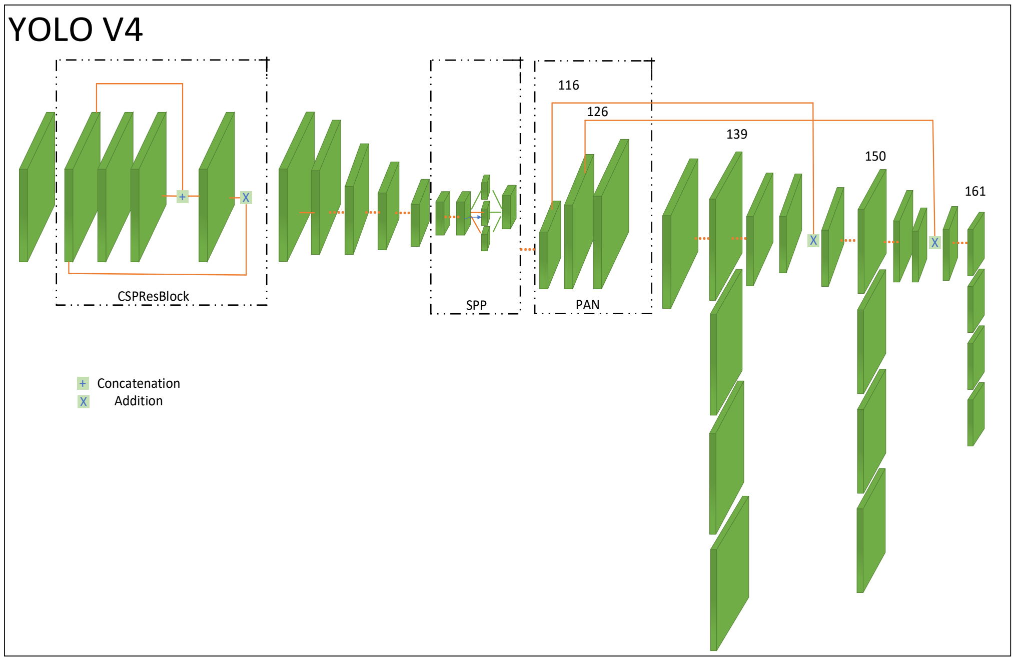

Fig. 5 YOLOV4 model archtecture

YOLOv4 was a significant innovation in 2020 [7]. It incorporated the latest methods, such as Cross-Stage Partial connections (CSPDarknet53 backbone), PANet neck, Spatial Pyramid Pooling (SPP), and other methods of data augmentation, including Mosaic and DropBlock. YOLOv4 was created with a fair tradeoff of performance and practicality, providing the finest quality performance without the need to use high-performance hardware. (Figure 5)

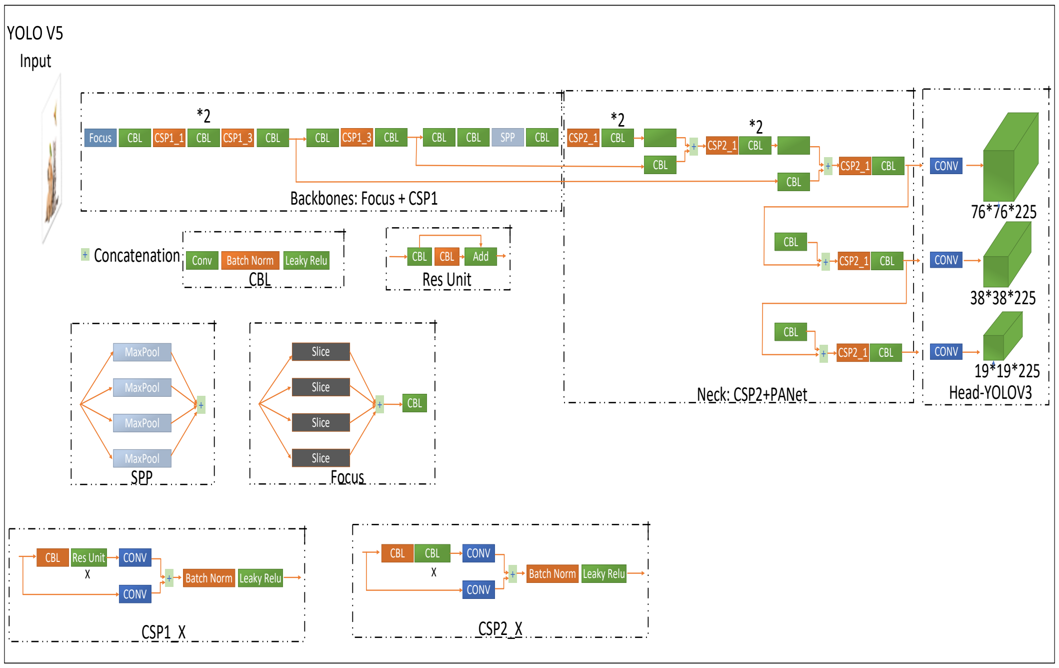

Fig. 6 YOLOV5 model architecture

About the same period, YOLOv5 was published, but not by the original authors, and it developed rapidly in popularity because of its PyTorch implementation and modular design that is easily user-friendly. Having a CSP-based backbone and decoupled head, YOLOv5 provided several model sizes (n, s, m, l, x) based on the hardware limitations. It was popularized as flexible, easily trainable, and with the help of the Ultralytics community (Figure 6).

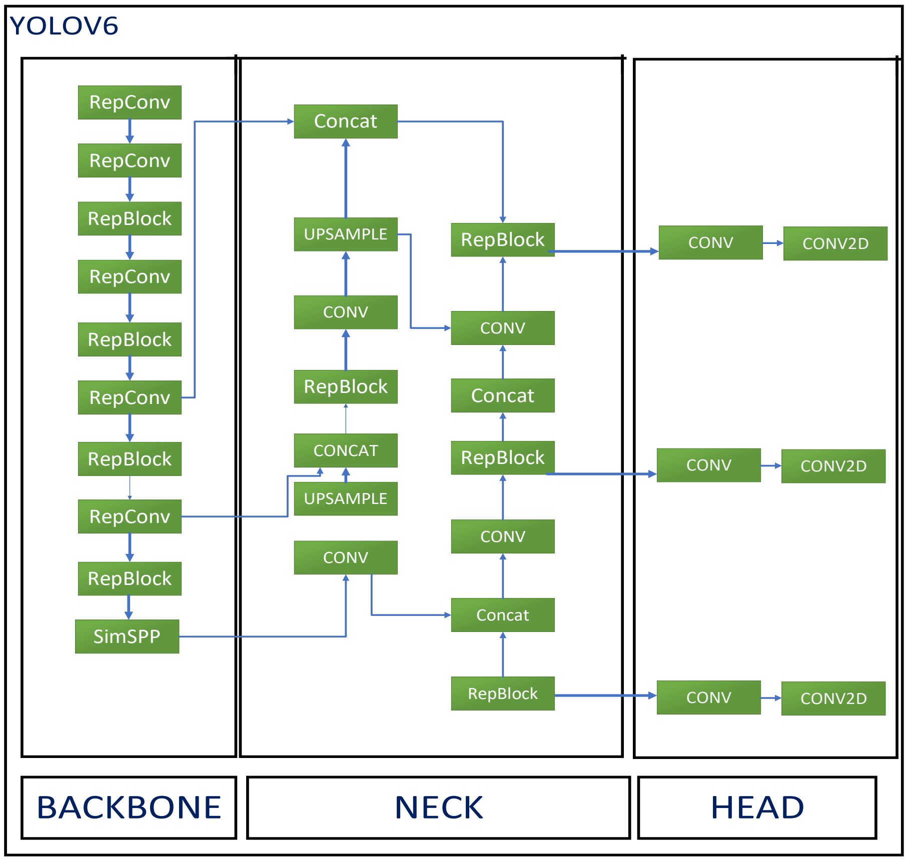

Fig. 7 YOLOV6 model architecture

YOLOv6 was published in 2022, designed to be at the industrial and production scale of object detection. It had a more effective CSP backbone and an optimized PANet neck. YOLOv6 was also redesigned to be lighter and faster, enabling YOLOv6 to be used in edge devices and scale to real-world use, particularly where low latency is of critical importance (Figure 7). In 2022, YOLOv7 was also released [8], introducing such strong architectural changes as E-ELAN (Extended Efficient Layer Aggregation Network) and multi-branch reparameterization. It was designed keeping the multi-task learning in mind, which allows detection, segmentation, and more in one unified building. Having effective inference and competitive accuracy, YOLOv7 was a state-of-the-art real-time detector (Figure 8).

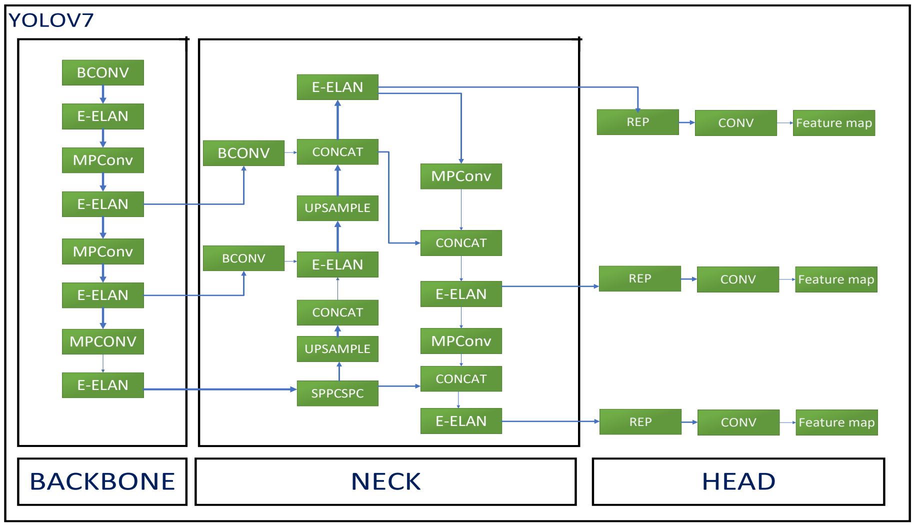

Fig. 8 YOLOV7 model architecture

In 2023, a significant paradigm shift was made by YOLOv8. It transitioned to an anchor-free structure, which made the process of detection simpler and enhanced generalization. YOLOv8 was created by Ultralytics by design, prioritizing modularity and versatility, which enabled it to detect objects along with classification and segmentation with relative simplicity (Figure 9). In 2024, the YOLOv9 [9] emerged and brought in definitions of Generalized ELAN and hybrid design philosophies. It was intended to combine anchor-based and anchor-free systems to achieve improved generalization between tasks and data sets. It is particularly noteworthy that it introduced Diverse Branch Blocks (DBB) and more rigorous regularization, thereby inspiring a performance limit without sacrificing simplicity.

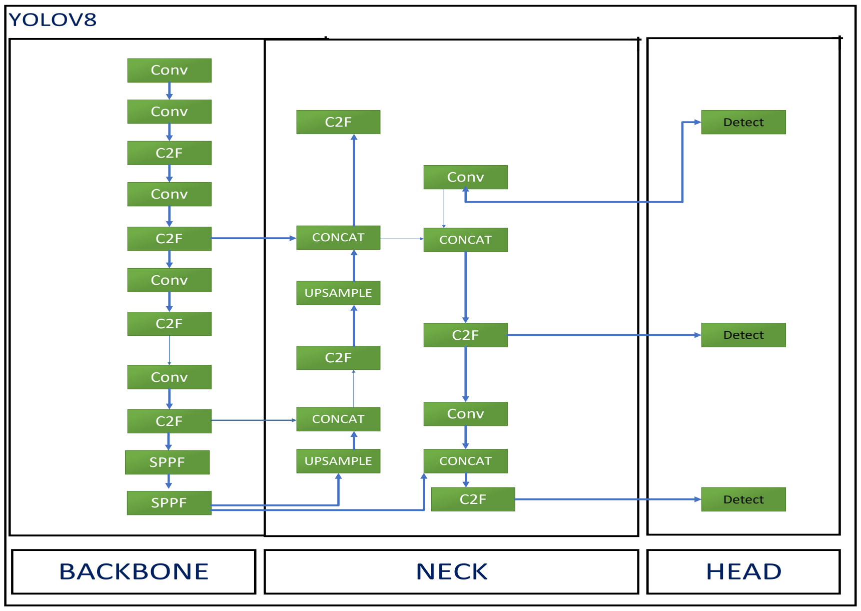

Fig. 9 YOLOV8 model architecture

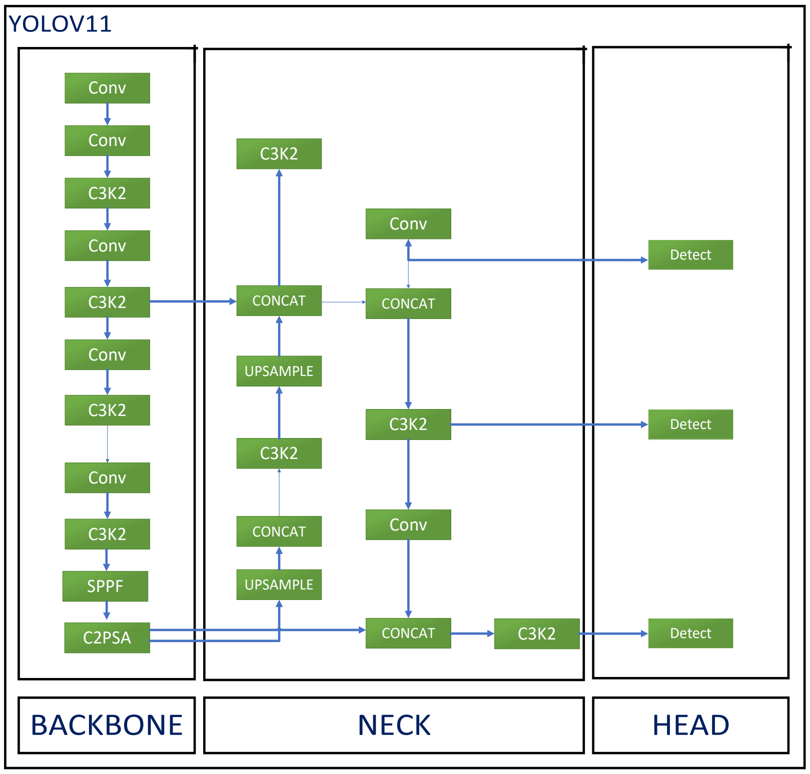

Fig. 10 YOLOV11 model architecture

Shortly after, YOLOv10 emerged and is heavily targeted at running on embedded systems [10]. This version was the most suitable in low-resource settings with the arrival of Efficient Model Optimization (EMO). It maintained high detection accuracy while being highly optimized for CPUs and mobile devices. By 2025, YOLOv11 pushed the boundary further by introducing a new backbone architecture called GELAN (Generalized Efficient Layer Aggregation Network) (Figure 10). Finally, YOLOv12, the latest evolution, combined the best of both CNNs and transformers. It introduced adaptive attention pooling and global reasoning through transformer-inspired modules. YOLOv12 blurs the line between traditional object detectors and vision transformers, achieving outstanding performance in complex and cluttered scenes.

5. YOLO Models Comparison

Since its launch in 2015, the YOLO (You Only Look Once) family of models has been evolving in terms of architecture. Every new version presents backbone network refinements, neck structure, head design, and model efficiency, which lead to better precision, speed, and capability to handle more complex tasks, which include object detection, segmentation, and tracking. The table below gives a summarized representation of YOLO v1 to v12 and identifies:

- Version and release year
- Backbone (feature extraction architecture)
- Neck (multi-scale feature aggregation)
- Head (detection and prediction mechanism)
- Model size and computational complexity (parameters and FLOPs)
- Detection performance (mAP on the COCO dataset)

This comparison (Table 2) helps researchers and engineers choose the most suitable version based on performance requirements, computational resources, and application needs, ranging from real-time edge deployment to high-accuracy cloud-based systems.

Table 2. YOLO model comparison

Version	**Year**	**Backbone**	**Neck**	**Head**	**Params / FLOPs***	**mAP (COCO)**
YOLOv1	2015	Custom CNN	None	Fully connected	~60M / ~30B	~63% @ IoU=0.5
YOLOv2	2016	Darknet-19	None	Anchor-based	~50M / ~70B	~76% @ IoU=0.5
YOLOv3	2018	Darknet-53	FPN-like	3-scale YOLO head	~62M / ~140B	~57.9% AP50:95
YOLOv4	2020	CSPDarknet53	SPP + PANet	YOLOv3 head	~64M / ~147B	~55.5% AP50:95
YOLOv5	2020	CSP-based (n/s/m/l/x)	PANet	Decoupled head	7M-90M	~65-50% (varies by model size)
YOLOv6	2022	Efficient industrial CSP	Optimized PAN	Efficient YOLO head	10-30M	Unknown
YOLOv7	2022	Enhanced CSP-Darknet	PAN + BoF	YOLO head	~36M	~56.8% AP50:95
YOLOv8	2023	Anchor-free CSP	CSPPAN	YOLO head	11-155M	~60-50%
YOLOv9	2024	CSP + PGA	CSPPAN	Programmable-gradient	2-57M / 7-189G	up to 60.6%
YOLOv10	2024	Large-kernel anchor-free	Spatial-channel decoupled	NMS-free head	-	SOTA at release
YOLOv11	2025	Optimized CSP	CSPPAN v2	Unified multi-task	-	Very high (unspecified)
YOLOv12	2025	Latest Ultralytics	Advanced modules	Multi-task head	-	Marginal gain over v11

6. Discussion

The development of the YOLO (You Only Look Once) family over the past ten years has become an example of the amazing breakthrough in the area of real-time object recognition. The different versions of YOLO have solved certain drawbacks of its predecessor, coupled with the introduction of architectural innovations. This longitudinal study identifies some of the common trends and design philosophies that have been used in the advancement of these models.

First, the evolution exhibits a change in the simplicity and monolithic nature of the structures, such as the ones in YOLOv1, to the more modular and hierarchical design of the subsequent ones. The advent of methods like multi-scale detection, anchor boxes, and subsequently anchor-free methods is indicative of a wider trend of more flexible and fine-grained recognition of objects. The innovative combinations, such as PANet, SPP, CSP connection, and others, have greatly added to the feature aggregation and spatial representation, which helped in improving the detection performance, particularly in cluttered or complex environments.

Second, the backbone networks have become increasingly more efficient and deeper with every version. Beginning with the custom CNN in YOLOv1, the network architectures were developed to encompass Darknet versions, CSPDarknet, and later GELAN and transformer-inspired networks. These advances have enabled YOLO models to preserve or increase accuracy and decrease latency, a highly important aspect of a real-time system. In addition, the architecture of the heads that are used to make the prediction also changes, in which fully connected layers in YOLOv1 have been replaced by decoupled, programmable, and NMS-free heads in subsequent versions. These developments are in line with the increased significance of efficient decoding and task-specific flexibility, especially on tasks that extend beyond object detection, to include instance segmentation, multi-object tracking, and action recognition.

The other theme that is essential is the increasing popularity of the edge and embedded deployment. Architectural optimization and quantization-friendly models such as YOLOv6 and YOLOv10 were developed to operate on low-resource devices. This scalability guarantees that YOLO can be used in data centers as well as mobile, IoT, and industrial settings. Lastly, the new versions (YOLOv11 and YOLOv12) highlight multi-task learning and unified heads, which enable one model to execute several tasks (e.g., detection, segmentation, tracking) at the same time. This is indicative of the growing demand for general-purpose, all-in-one vision systems in real-life applications.

The superior performance of recent YOLO versions arises from specific architectural innovations rather than dataset changes alone. The introduction of Cross Stage Partial (CSP) connections in YOLOv4 reduced gradient duplication and enhanced feature reuse. Later, Efficient Layer Aggregation Networks (ELAN) and PANet modules improved spatial information flow between layers. YOLOv12 further integrates attention-based refinement, enabling precise object boundary detection and reduced false positives. These cumulative enhancements explain YOLO's higher mAP and FPS relative to two-stage detectors such as Faster R-CNN, while maintaining robustness in complex real-time scenarios. Overall, this longitudinal study shows that YOLO's evolution is not just about improving detection metrics, but also about expanding applicability, enhancing robustness, and simplifying deployment across a range of use cases.

7. Conclusion

In this paper, a longitudinal analysis of the YOLO object detection models from version 1 to version 12 is provided, as well as the architectural changes and technical modifications that have influenced the family over the years. Since its inception as a single regression-based detector, the YOLO has evolved into a flexible, effective, and high-performing system that can process a variety of tasks in real-time vision.

This discussion shows a steady push up to increased accuracy, swift inference, and increased compatibility with the application. Every YOLO version presents a specific improvement in backbones, necks, heads, and optimization plans, which, together, expand the limits of the applications of real-time detection models. The gradual adoption of methods such as multi-scale prediction, CSP modules, transformer-inspired blocks, and multi-task heads is indicative of the response of the community to real-world issues in video tagging, deployment at the edge, and embedded AI. The influence of YOLO on computer vision, particularly time-sensitive programs like autonomous driving, surveillance, and intelligent cities, is indisputable. In the ever-evolving field, more versions of YOLO are expected to adopt hybrid architectures, self-supervised learning, and adaptive intelligence to be on the frontline of real-time object detection.

This longitudinal view is not only useful in helping researchers and practitioners to comprehend the tradeoffs of the design of every version of YOLO, but also to assist them in choosing the most suitable model to run certain tasks and limitations. The story of the implementation of innovations in the face of the YOLO system v1-v12 is a testimony to the fact that the iterative method of innovation can result in the development of strong, scalable, and popular AI applications.

7.1. Future Research Directions

Though the evolution of YOLO models is rather rapid, there are still a number of open research opportunities. Future research might work on ultra-lightweight model compression and quantization to be deployed on microcontrollers and low-power IoT devices. Cross-modal integration, i.e., bringing together YOLO and natural-language understanding or multimodal transformers to enhance contextual awareness in challenging scenes, is another avenue to pursue. The topic of federated and privacy-preserving learning should be considered to enable the YOLO models to learn on distributed datasets without disclosing sensitive data. Also, explainability and interpretability should be of more importance, especially in fields where safety is a concern, like healthcare and autonomous driving, to enable the predictions of the YOLO to be audited and trusted. Addressing these challenges will guide the next phase of innovation in real-time object detection.

References

[1] Bhaumik Vaidya, and Chirag Paunwala, *Deep Learning Architectures for Object Detection and Classification*, Smart Techniques for a Smarter Planet, Springer, Cham, pp. 53-79, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Licheng Jiao et al., "A Survey of Deep Learning-Based Object Detection," *IEEE Access*, vol. 7, pp. 128837-128868, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Iffat Zafar et al., *Hands-On Convolutional Neural Networks with Tensorflow: Solve Computer Vision Problems with Modeling in Tensorflow and Python*, Packt Publishing Ltd, 2018. [Google Scholar] [Publisher Link]

[4] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection" *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779-788, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Joseph Redmon, and Ali Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 6517-6525, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Joseph Redmon, and Ali Farhadi, "YOLOv3: An Incremental Improvement," *arXiv Preprint*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv Preprint*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Taiwan, pp. 7464-7475, 2023. [Google Scholar] [Publisher Link]

[9] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *Computer Vision - ECCV 2024*, Springer, Cham, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Hui Chen et al., "YOLOv10: Real-Time End-to-End Object Detection," *Neural Information Processing Systems Foundation, Inc. (NeurIPS)*, Vancouver, Canada, pp. 107984-108011, 2024. [CrossRef] [Google Scholar] [Publisher Link]