Proceedings of the 4th International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/87/2025.20335

The Evolution of YOLO: From YOLOv1 to YOLOv11 with a Focus on YOLOv7's Innovations in Object Detection

Yuzhao Luo[1,a,*]

[1]*Courant Institute of Mathematical Sciences, New York University, NY, USA*
a. yl10756@nyu.edu
**corresponding author*

Abstract: Throughout the evolution of You Only Look Once (YOLO) series, staring from base YOLO to latest YOLOv11, each version takes advantages of different techniques and mechanism, incorporating innovations that enhance object detection capabilities by improving both speed and accuracy. From introduction of anchor boxes in YOLOv2 to multi-scale predictions in YOLOv3 and Cross-Stage Partial Networks in YOLOv4, each iteration has brought unique improvements. In YOLOv7, two major advancements, Extended Efficient Layer Aggregation Network and Planned Re-parameterized Convolution, were introduced to address challenges in feature aggregation and parameter utilization, while maintaining optimal gradient flow. Additionally, advanced label assignment strategies, such as lead head guided label assigner and coarse-to-fine label assigner, further improve learning efficiency. These innovations enable YOLOv7 to set new standards in object detection, especially for applications in autonomous driving, video surveillance, medical imaging, and beyond.

Keywords: Object Detection, YOLO series, YOLOv7.

1. Introduction

In fact, object detection has been evolving over the past years, closely related to the deep learning. During the traditional object detection era, the Viola-Jones Detector and HOG Detector were the pioneering work that started the traditional object detection methods [https://viso.ai/deep-learning/object-detection/]. With the emergence and popularity of deep learning, many different methods have emerged such as R-CNN, Fast R-CNN, YOLO, and SSD (Single Shot Multibox Detector).

Object detection capabilities have been significantly enhanced through advancements in Convolutional Neural Networks (CNN) and Vision Transformers (ViT). In the past survey of object detection, researchers proposed many methods based of CNN and ViT. For example, Ren et al. [1] utilized Faster R-CNN for object detection back in 2015. Carion et al. [2] also proposed end-to-end object detection with transformers in 2020. Traditionally, object detection methods could be categorized into two-stage, one-stage detectors, and transformer-based methods. Two-stage detectors generates a set of possible object proposals from the input image, referred as the first stage. During the second stage, these region proposals are classified into different object types, and their bounding boxes are further refined to improve localization accuracy. One-stage detectors perform both region proposal generation and classification in one step. One-stage detectors predict class probabilities and bounding boxes directly from the image without a separate proposal stage, predicting the bounding

DOI: 10.54254/2753-8818/87/2025.20335

boxes and class probabilities directly from the image. Unlike one-stage or two-stage detection methods, transformer-based methods abandon the traditional region proposal approach and does not rely on anchor boxes or sliding windows. Instead, these methods employ encoder-decoder architecture to output a fixed-size set of bounding boxes and class labels.

Though one-stage object detectors are struggle with identifying irregularly shaped objects or a group of tiny objects, one-stage object detectors skip the region proposal phase and run detection directly over densely sampled locations, prioritizing inference speed and have greater structural simplicity and efficiency at the cost the performance compared to multi-stage and transformer-based detectors. The most widely used one-stage object detection models include YOLO, SSD, and RetinaNet.

Among these typical models, the family of YOLO models has been evolving since it firstly introduced by Joseph Redmon et al. [3]. YOLO is well-known for its simplicity, speed, and high performance in real-time object detection. Furthermore, YOLO models could also complete tiny object detection task, one of the difficult tasks in computer vision due to the small object size and limited information available.

The following sections in this paper would discuss the bae YOLO, YOLOv7 - one variant of YOLO that introduced computing scaling (scaling width, depth, and resolution) to optimize the model for different applications, and several applications of YOLO based models.

2. YOLO Family

2.1. Overview of Base YOLO

Before the emergence of the first generation YOLO model, object detection work rely on classifiers to perform detection. YOLO treats object detection as a regression problem, predicting the bounding boxes coordinates and the associated class probabilities for the detected objects.

Back to 2016, most methods like R-CNN focus on identifying possible object regions within an image, followed by running a classifier on each proposed region. After classification a post-processing step refines the bounding boxes, removes duplicate detections, and adjusts the scores based on interactions with other objects in the scene [4]. As a novel object detection method at that time, YOLO transcends other detection methods like DPM and R-CNN for several reasons. YOLO is fast because there is no need for complex pipeline if regarding detection as a regression problem. YOLO reasons globally about the image when making predictions. YOLO effectively captures contextual details about objects during both training and testing. This helps YOLO avoid false positives on background regions, reducing errors in situations where other detectors might misidentify background regions as objects. It also learns representations that can generalize well, reducing the chance of failure when used on new domains or unexpected inputs.

The architecture of YOLO is unified, predicting multiple bounding boxes and class probabilities in a single forward pass. Generally, the network decides the input image into an S x S grid. Each grid cell predicts bounding boxes, confidence scores, and class probabilities, based on the assumption whether it contains an object. The confidence scores reflect both the likelihood of the grid cell containing an object and how well the bounding box fits it.

DOI: 10.54254/2753-8818/87/2025.20335

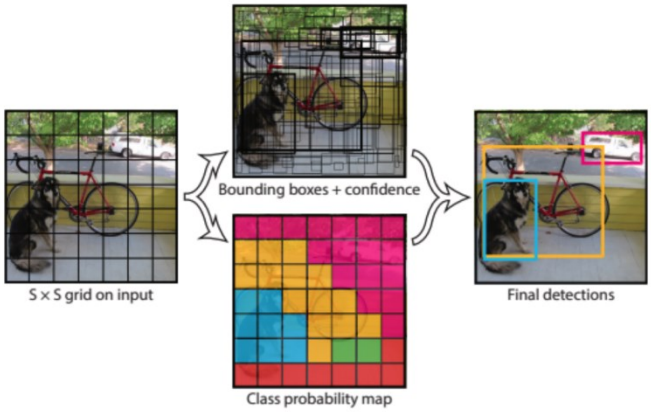

Figure 1: Process of Base YOLO[3]

YOLO utilizes a convolutional neural network design influenced by GoogLeNet [5], featuring 24 convolutional layers followed by 2 fully connected layers. The base YOLO model is pre-trained on ImageNet for classification and then fine-tuned for object detection using the PASCAL VOC dataset [6]. YOLO specifically splits the image into a 7x7 grid, with each grid cell tasked with detecting objects whose center falls within the cell. For each grid cell, it predicts two bounding boxes and a set of class probabilities. The full network of base YOLO is shown in Figure 2.

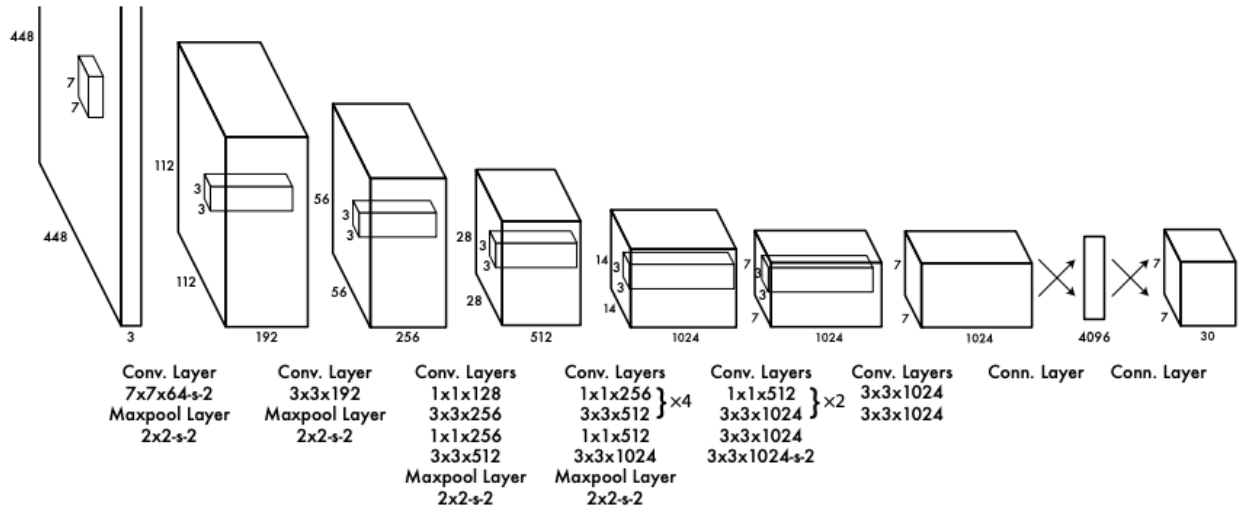

Figure 2: Base YOLO Architecture [3]

The main advantage of the base YOLO model is its speed, processing images at 45 frames per second (FPS). Fast YOLO, which has fewer convolutional layers and filters, can process 155 FPS, outperforming slower region-based methods like Faster R-CNN[7], which can process fewer than 10 FPS. Taking the performance of Fast YOLO on the PASCAL VOC dataset as an example, it achieves 52.7% mAP while processing at 155 FPS. While the full achieves a higher 63.4% mAP.

Although YOLO can quickly detect objects, it lags behind other methods like Faster R-CNN, especially in precise localization, particularly for small objects that are close together. YOLO's coarse grid structure limits its ability to detect multiple small, closely packed objects within a single grid cell, making it harder to manage varying object sizes and shapes. Since each grid is limited to predicting two bounding boxes, resulting in YOLO struggles with detection multiple objects within

DOI: 10.54254/2753-8818/87/2025.20335

a single grid cell. Also, YOLO's architecture imposes significant spatial constraints on the bounding boxes, limiting its ability to handle objects of different shapes and aspect ratios.

2.2. History of YOLO Family

Over time, YOLO has undergone several iterations, each improving upon performance, speed, and scalability. YOLOv1 (introduced by Joseph Redmon et al. in 2016) revolutionized object detection by addressing it as a single neural network problem. YOLOv2, also known as YOLO9000, introduced several innovations, including batch normalization for more stable training and anchor boxes inspired by Faster R-CNN to handle varying object sizes and shapes. [8]. YOLOv2 maintained real-time detection speeds while enhancing accuracy and extending to 9000 classes, with multi-scale predictions in YOLOv3 allowing it to detect objects at small, medium, and large scales [9]. YOLOv3 balanced accuracy and speed, making it popular for many real-time applications like surveillance [10]. YOLOv4 introduced a variety of techniques to enhance detection performance and make improvements in the architecture, such as Weighted Residual Connections (WRC), Cross-Stage Partial Connections (CSP), and others. [11]. YOLOv4 reached up to 43.5% mAP on the COCO dataset, running at over 100 FPS on GPUs. YOLOv5 was developed by the Ultralytics team, it became notable for its ease of implementation, with out-of-the-box support for PyTorch and introduction of a flexible model architecture [12]. YOLOv6, introduced by Meituan, focused on improving industrial applications. It introduced better optimization for edge devices and provided more robust models, especially for tasks requiring faster detection [13]. YOLOv7 employed compound scaling (adjusting width, depth, and resolution) to optimize the model for different tasks, from edge devices to high-end GPUs, using the Extended Efficient Layer Aggregation Network (E-ELAN) for improved multi-scale detection and feature extraction [14]. YOLOv8 was developed by Ultralyitics. It continued building on YOLOv5's foundation, introducing significant architectural and methodological innovations. YOLOv8 achieves a mAP of 37.3% on the COCO dataset and a speed of 0.99ms on A100 TensorRT [15]. YOLOv9 introduced a new object detection framework, focusing on addressing information bottlenecks and improving gradient flow during training in deep neural networks [16]. This is achieved through the introduction of Programmable Gradient Information (PGI) and the creation of a lightweight network architecture called Generalized Efficient Layer Aggregation Network (GELAN). However, the challenges of non-maximum suppression (NMS) in previous YOLO versions introduced computation redundancy and inference latency. YOLOv10 eliminated redundancy and reduced inference latency by using an NMS-free training strategy, improving supervision during training with consistent dual assignments. [17]. YOLOv11 is the latest version in the Ultralytics YOLO series, building on the advancements made by previous YOLO models.

Juan Terven and Diana Cordova-Esparze tracked the evolution of the YOLO family, examining the innovations and advancements introduced in each version, from the original YOLO to the latest iterations, including YOLOv8, YOLO-NAS, and YOLO models enhanced with transformer architectures on February, 2024 [18]. Since the emergences of YOLOv9 and YOLOv10,Chien-Yao Wang and Hong-Yuan Mark Liao also published a review of YOLO's development from YOLOv1 to YOLOv10 in August 2024.

DOI: 10.54254/2753-8818/87/2025.20335

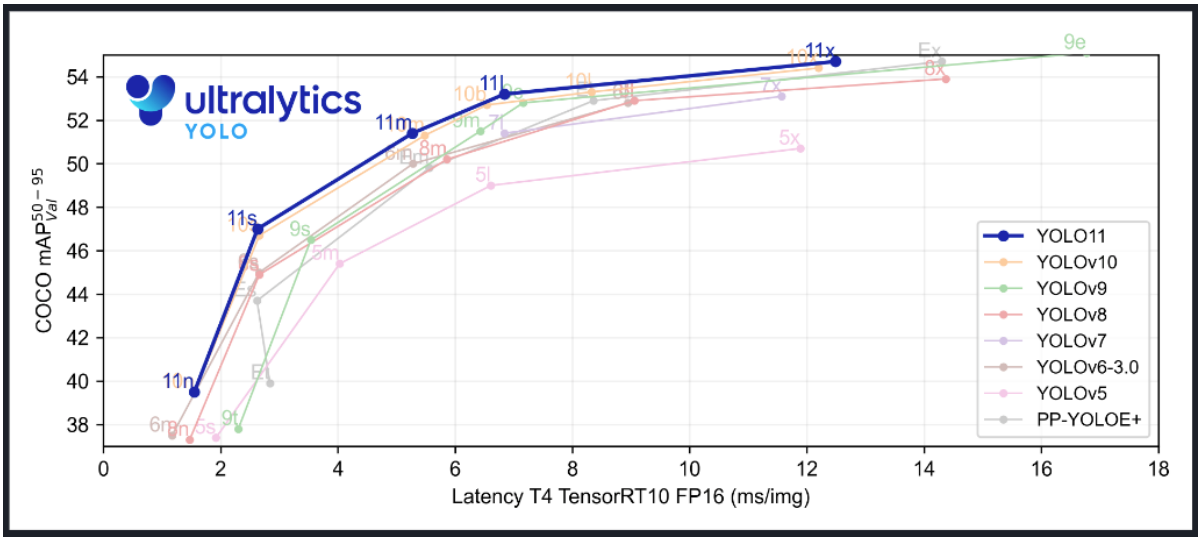

Figure 3: Comparison between various YOLO models in terms of mAP and latency [19]

2.3. YOLOv7

YOLOv7 is crucial in the YOLO family because it introduced cutting-edge techniques to address common challenges in feature extraction and computational efficiency. Though later versions like YOLOv8 perform better, discussing YOLOv7 in detail helps highlight how modern object detection models are pushing the boundaries of performance and adaptability.

YOLOv7 introduced two key architectural advancements that significantly improve object detection performance: E-ELAN and Planned Re-parameterized Convolution (RepConv). Extended Efficient Layer Aggregation Network (ELAN) [20] was not firstly introduced in YOLOv7 but improved in YOLOv7 to implement E-ELAN. Specifically, assuming there is a feature map of 2c channels passed from the previous module for a ELAN structure. In the ELAN structure, the 2c channels will be used directly or passed to convolution layers for merging finally. ELAN architecture provides several benefits that enhance the overall performance of the network, including better feature extraction, improved gradient flow, and more efficient parameter utilization. E-ELAN enhances the learning capabilities of the network by improving parameter utilization. The key idea behind E-ELAN is to maintain the gradient path while expanding the network's feature extraction capabilities through strategic use of group convolutions.

For ELAN structure, if more computational blocks are stacked unlimitedly, the stability could be compromised, leading to the decrease of parameter utilization rate. E-ELAN addresses this by using expand, shuffle, and merge cardinality techniques to enable continuous learning improvement without altering the original gradient path. E-ELAN introduced group convolutions to increases the cardinality. After applying group convolutions, the feature maps are shuffled into different groups and then merged. This shuffling and merging mechanism allows the network to capture more diverse features across different layers, enhancing learning without changing the overall gradient transmission path.

The structures of ELAN and E-ELAN are shown in Figure 4. Essentially, E-ELAN is equivalent to two parallel ELANs and adding the outputs.

DOI: 10.54254/2753-8818/87/2025.20335

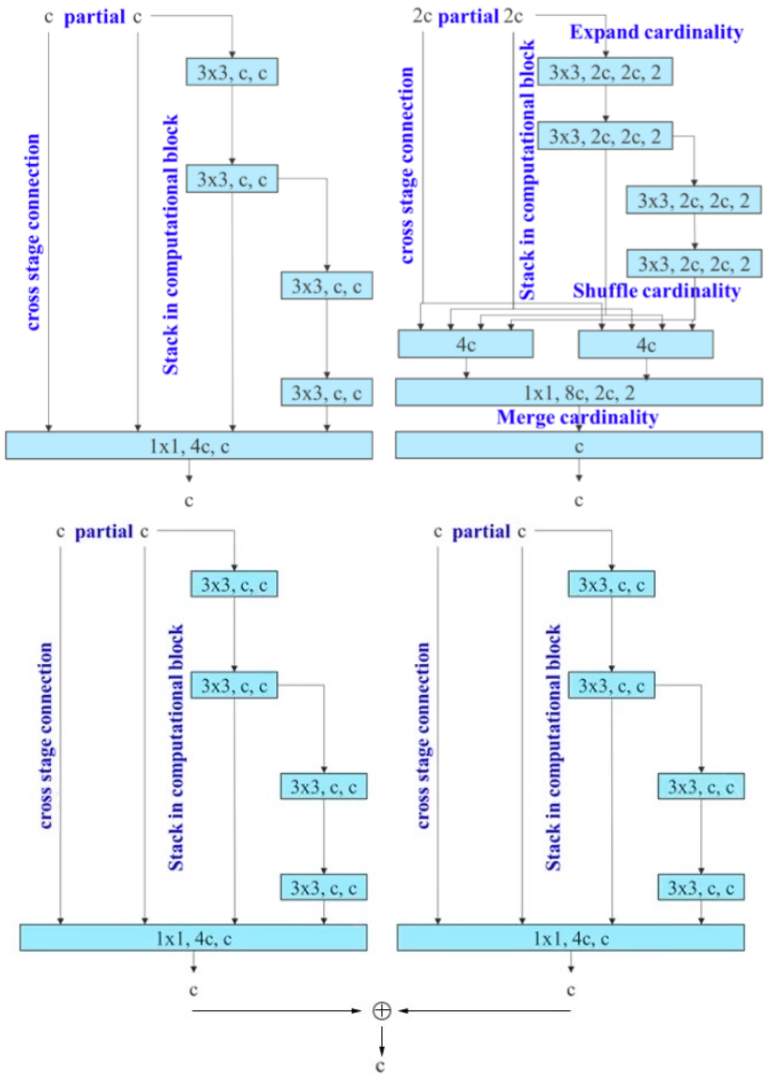

Figure 4: ELAN vs. E-ELAN [14]

YOLOv7 uses model scaling to adjust certain model attributes to create different versions that meet various inference speed requirements. Since the model of YOLOv7 is quite special, the splicing operation is used in its basic module ELAN. Special processing is required when scaling the model for this splicing-based model, so YOLOv7 uses a composite model scaling method while changing the width of the model. The enlarged model is more complex but more accurate, allowing it to be used in different usage scenarios. Among all models of YOLOv7, YOLOv7x is a scaling of YOLOv7. Taking the E-ELAN module in the two models as an example, the E-ELAN of YOLOv7x adds input to the convolutional layer and the splicing layer, which leads to the expansion of the depth and width of the model.

The planned Re-parameterized Convolution is applied to residual or concatenation-based modules, simplifying complex structures during training and improving the model's performance without affecting inference speed. [21]. The author of YOLOv7 found that although re-parameterized convolution achieved good results on VGG, when directly applied to structures with residual modules such as ResNet [22] and DenseNet [23], the accuracy was greatly reduced. . Therefore, YOLOv7 uses the RepConvN structure, which removes the identity connection based on RepConv. YOLOv7 employed planned RepConvN to optimize the network's performance during both training and

DOI: 10.54254/2753-8818/87/2025.20335

inference by merging multiple computational modules into one during the inference stage, achieving a better balance between training efficiency and inference performance.

YOLOv7 introduced lead head guided label assigner and coarse-to-fine head guided label assigner to optimize the label assignment process during training. These methods aim to enhance how labels are assigned to predicted bounding boxes, significantly impacting how well the model learns to detect and classify objects. The lead head guided label assigner improve the assignment of ground truth labels to the model's predictions by focusing on the lead head during training. In object detection networks with multiple heads, the lead head is typically the one responsible for handling a specific scale or level of detail. The coarse-to-fine head guided label assigner is introduced to refine the label assignment process progressively during training. It focuses on moving from coarse to fine predictions, ensuring that the network hones in on the most accurate label assignments over time.

2.4. Applications of YOLOv7

YOLOv7's real-time performance, efficiency, and accuracy make it a highly versatile model for a wide range of applications. Its ability to handle multiple scales, work in edge computing environments, and provide high high accuracy in challenging conditions has made it a good choice for industries like autonomous driving, medical imaging, surveillance, agriculture and tiny object detection, etc.

Figure 5: Tiny Bird Detection using YOLOv7

3. Discussion

Object detection, as an important task in computer vision, has made significant progress but still faces many challenges, including: (1) Small target detection. Small target objects in the image have fewer pixels and less distinct features, making them difficult to detect. (2) Occlusion and partially visible targets: When a target is partially occluded, existing object detection algorithms may not be able to recognize it correctly. (3) Background interference: Complex information in the background may interfere with the judgment of the object detection model. (4) Category imbalance: Some categories

DOI: 10.54254/2753-8818/87/2025.20335

in the dataset have a much larger number of samples than others, resulting in the trained model being biased towards identifying those categories with larger sample sizes. (5) Real time requirements: In certain application scenarios, such as autonomous driving, object detection requires real-time processing, which places high demands on the speed of the algorithm. (6) Multi scale object detection: There may be objects of different sizes in the image, and the model needs to have good multi-scale detection capabilities.

In the future, object detection technology will further develop towards real-time performance, robustness, multimodal fusion, interpretability, and other aspects to meet the needs of more complex application scenarios. For example, improving the detection accuracy of small targets by enhancing feature representation capabilities and introducing multi-scale feature fusion techniques; Processing object detection in occlusion situations through local feature matching or contextual information assistance. Meanwhile, with the improvement of hardware computing ability and the arrival of the big data era, object detection algorithms will become more efficient and intelligent. To this end, the promising directions also includes: explore how to train object detection in conjunction with other visual tasks such as segmentation, recognition, etc. to improve overall performance; Combining various modalities of data such as images, videos, and sounds to improve the robustness and accuracy of detection; Optimizing algorithm architecture and hardware acceleration schemes to improve the detection speed.

4. Conclusion

The evolution of the YOLO family from its inception to YOLOv11 represents a continuous pursuit of optimizing real-time object detection, focusing on improving both speed and accuracy. Each version has introduced significant architectural changes, making YOLO models one of the most widely-used object detection algorithms today. YOLOv7 stands out as a pivotal advancement in the evolution of real-time object detection models, pushing the boundaries of performance, speed, and efficiency. By integrating key innovations such as E-ELAN and RepConvN, YOLOv7 addresses challenges in feature aggregation, parameter utilization, and gradient flow, making it one of the most efficient models in terms of accuracy and computational demand.

References

[1] *Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.*

[2] *Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In European conference on computer vision (pp. 213-229). Cham: Springer International Publishing.*

[3] *Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.*

[4] *Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).*

[5] *Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).*

[6] *Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111, 98-136.*

[7] *Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.*

[8] *Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).*

[9] *Farhadi, A., & Redmon, J. (2018, June). Yolov3: An incremental improvement. In Computer vision and pattern recognition (Vol. 1804, pp. 1-6). Berlin/Heidelberg, Germany: Springer.*

DOI: 10.54254/2753-8818/87/2025.20335

[10] *Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. Journal of Big data, 8(1), 66.*

[11] *Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.*

[12] *Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.*

[13] *Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). 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).*

[14] *Yaseen, M. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. arXiv preprint arXiv:2408.15857.*

[15] *Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616.*

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

[17] *Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction, 5(4), 1680-1716.*

[18] *Wang, C. Y., & Liao, H. Y. M. (2024). YOLOv1 to YOLOv10: The fastest and most accurate real-time object detection systems. arXiv preprint arXiv:2408.09332.*

[19] *Wang, C. Y., Liao, H. Y. M., & Yeh, I. H. (2022). Designing network design strategies through gradient path analysis. arXiv preprint arXiv:2211.04800.*

[20] *Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13733-13742).*

[21] *He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).*

[22] *Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).*

[23] *Shigematsu, K. (2023). Small Bird Detection using YOLOv7 with Test-Time Augmentation. arXiv preprint arXiv:2401.01018.*