Evolution of YOLO: A Literature Review

Authors

Audric Sudirman, **Padayatchy Sandragassen**, **Amisha Sherekar**, **Nikhil Rasmalla**, **Nikhil Donthataboina**, **Bhavishya Maturi**, **Vishnupriya Dachuri**

Abstract

Recent years have seen a rise in the popularity of You Only Look Once (YOLO) object detection algorithms due to their speed and high accuracy. Our paper gives an in-depth review of YOLO variants, including YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, and YOLOv8. Over the years, YOLO has undergone successive versions, each refining its architecture and addressing limitations. It started out as a framework to predict bounding boxes and class probabilities in a single step. The continuous evolution of YOLO reflects the dynamic nature of the field, where innovation and adaptation propel object detection capabilities to new heights, catering to the diverse needs of applications spanning surveillance, robotics, and beyond. We provided an analysis of the YOLO revolution in this paper. This study presents the improvements made to each YOLO version. Additionally, YOLO-related metrics and applications are also provided. We begin by giving a general overview of object detection and YOLO, then briefly analysing each YOLO variant and its offshoot versions. Every variation's enhancement is also examined. Next, metrics used to evaluate YOLO are discussed with a table comparing each YOLO variant with those metrics. Furthermore, the applications of each YOLO variant are mentioned.

Keywords

YOLO (You Only Look Once), Object detection, Convolutional Neural Networks (CNNs), Mean Average Precision (mAP), Anchor boxes, One-stage Detector

Introduction

Object detection is a computer vision task that involves identifying and locating objects within an image or video. Many applications exist for object detection, such as optical character recognition, self-driving cars, tracking objects, and face detection. Object Detection models are split into two types: single-stage and multi-stage [1].

YOLO (You Only Look Once) is an exceptional real-time, single-stage detector. It is an individual neural network that can predict several bounding boxes and class probabilities simultaneously. Beginning with "You Only Look Once: Unified, Real-Time Object Detection" by Redmon et al., the YOLO algorithm has undergone several iterations in recent years, each bringing improvements in speed and accuracy [2].

When compared with other models like Faster R-CNN, SSD, and RetinaNet, YOLO distinguishes itself with its balance of speed and accuracy for real-time object detection. For example, the detection accuracy of YOLO and Fast-RCNN is 63.4 and 70, respectively, but the inference time of YOLO is around 300 times faster [1].

Through a literature survey, our paper compares YOLO variants from YOLOV1 to YOLOV8 with their offshoot versions. The article elaborates on the evolution of each version with performance and its application. The study also discusses each version's new advantages, disadvantages, architectural advancements, and evaluation metrics for object detection. The review concludes by examining the performance of YOLO and its variants.

YOLO

YOLOv1 [3] is a game-changing neural network for object detection that predicts bounding boxes and class probabilities straight from complete photos in a single pass. The working principle consists of splitting an image into a grid. Each grid cell forecasts bounding boxes and confidence scores which reflect the model's assurance of containing an object on an SxS grid. Bounding boxes are rectangles drawn around objects in an image, specifying an object's location and size within that cell. The network design is inspired by GoogLeNet and consists of 24 convolutional layers and two fully linked layers [3] as seen in the Figure 1. When compared to two-stage models, YOLOv1 processes the full image during training and testing, gathering global contextual information and lowering background errors which achieves a direct prediction of objects and detection of images faster. As a result, the base model achieves real-time processing at 45 frames per second, making it perfect for applications such as self-driving automobiles and video processing [3].

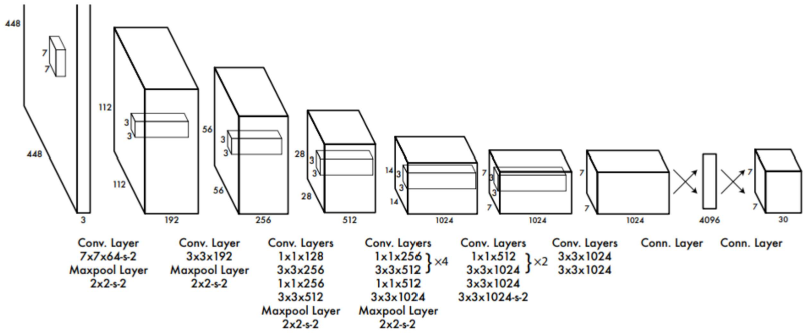

Figure 1: Detailed architecture of YOLO architecture [3].

YOLOv2

YOLOv2 was released in 2016 by Joseph Redmon and Ali Farhadi, and was published in July 2017 as "YOLO9000: Better, Faster, Stronger". It outperformed algorithms such as SSD and Faster R-CNN [4]. YOLOv2 improved on YOLOv1 by introducing batch normalisation for stability, anchor boxes for precise bounding box predictions, and higher-resolution training. This resulted in improved speed and accuracy and better resolution for object identification compared to other competitive models [4]. Anchor boxes in YOLOv2 are pre-made shapes that help the model better forecast object borders by providing different-sized templates, improving object recognition accuracy and recall [4]. YOLOv2 replaced fully connected layers by k-means clustering for more effective priors which resolved bounding box prediction difficulties [4]. The architecture used Darknet-19, which consists of 19 convolutional layers and 5 max-pooling layers, and achieved better accuracy. The model was strengthened by training on WordNet-labeled data and using a merged ImageNet and COCO datasets with a Word Tree method. On Pascal VOC 2007, YOLOv2 achieved an amazing 76.8 mAP, outperforming YOLOv1 by 13.4% [4]. The real-time detection system, which can recognize over 9000 object categories, demonstrates its speed and accuracy [4].

YOLOv3

YOLOv3 was developed by Joseph Redmon and Ali Farhadi [5] who introduced several design changes to improve accuracy and speed. YOLOv3 gives a multilabel classification because it can be used as a logistic classifier for each class in place of the softmax layer used by YOLOv1 to calculate class probabilities. In YOLOv3, the system predicts object bounding boxes using dimension clusters as anchor boxes and logistic regression for objectness scores, resulting in better real-time object identification accuracy relevant to recommendation systems, process management, and less human input. The network predicts boxes at three different scales and uses a new feature extractor called Darknet-53, which is more powerful than Darknet-19 and more efficient than ResNet-101 or ResNet-152 [5]. During the creation of YOLOv3, testing with prediction methods revealed that anchor box (x, y) offsets impacted stability while linear (x, y) predictions resulted in a modest mAP decline [5]. The improvements also address the previous struggle of YOLO with small objects, as the new multi-scale predictions show relatively high APS performance. YOLOv3 outperforms YOLOv2, increasing average precision for small objects by 13.3. The offshoot, YOLOv3-Tiny, is faster than YOLOv3 in real-time object detection and efficiently identifies objects with improved speed performance for various image dimensions at the cost of some accuracy. [6].

YOLOv4

The YOLO-v4 model, developed by Alexey Bochkovskiy et al. [7], is an effective object detection model made for real-time use on a traditional GPU. It implemented numerous new features in its architecture like optimised anchors which are carefully selected initial bounding box sizes for training that increase detection accuracy in YOLOv4. Dynamic Mini-Batch Size was also added which improves object detection by altering the number of training samples dependent on the network's performance in YOLOv4. The Bag-of-Specials modules improve YOLOv4's object recognition by introducing specialised techniques such as feature aggregation, attention mechanisms, and upgraded convolutional layers for greater accuracy and performance [7]. On a Tesla V100 Graphics Processing Unit (GPU), its real-time speed reaches roughly 65 frames per second (FPS), a significant improvement over YOLOv3.[8] SlimYOLOv4, a YOLOv4 offshoot developed by Peng Ding et al., improves efficiency by replacing YOLOv4's feature extraction network with MobileNetV2, which maintains accuracy while requiring fewer calculations. It uses sophisticated convolutional layers to improve performance and ReLU6 to improve numerical resolution. It outperforms YOLOv2's mAP of 29.2% with a mAP of 70.83%. SlimYOLOv4 is optimised for mobile devices, with a real-time performance of 60.19 FPS. By addressing storage and power limits through a lightweight design, it emphasises the need for mobile-suitable networks. [8].

YOLOv5

YOLOv5 was put on GitHub by ‘’Glenn Jocher,’’ CEO of ‘’Ultralytics,’’ two months after YOLOv4 in 2020 [9]. The YOLOv5 repository is a natural extension of the YOLOv3 PyTorch repository [10].

YOLOv5 is, nevertheless, distinct from prior releases. As shown in Figure 2, it comprises three main parts: the backbone, the neck, and the head. It makes use of CSPDarknet53 as its backbone. This backbone addresses the repeating gradient information in big backbones and integrates gradient change into a feature map, which boosts inference speed, accuracy, and model size by minimizing the parameters [11]. As its neck, YOLOv5 uses a path aggregation network (PANet) to improve the flow of information. PANet employs a novel feature pyramid network architecture to increase low-level feature propagation with an improved bottom-up strategy [12].

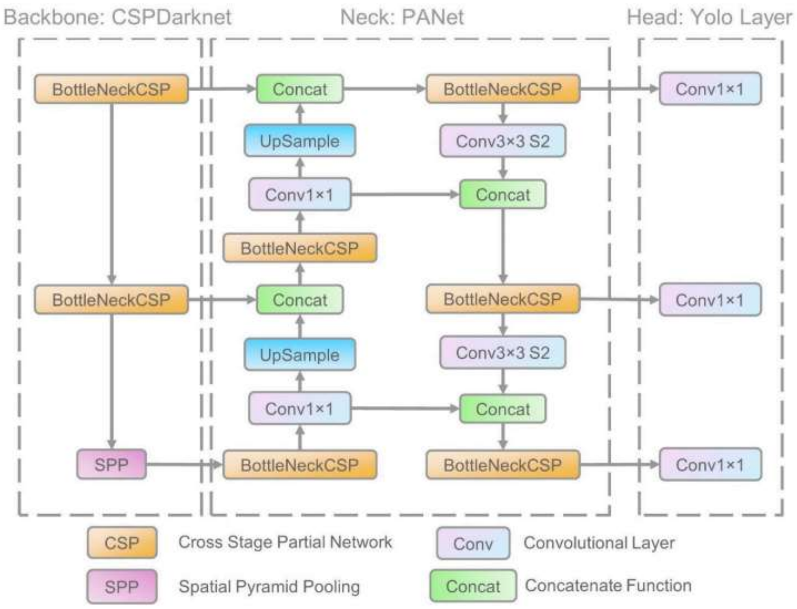

Figure 2: Detailed architecture of YOLOv5 architecture [13].

In addition, YOLOv5 introduces new data augmentation strategies to increase the model’s generalizability and reduce overfitting. This includes ‘’Mosaic Augmentation’’, a strategy for combining four training photos into one that encourages object identification models to handle different object scales and translations better [14].

Additional scaled versions, from small, medium, large, to extra large, are available for devices with different computational constraints. In order, and With ‘’small’’ as the fastest, they trade inference speed for higher accuracy and computational power due to more parameters.

YOLOv6

YOLOv6 is an improvement over its predecessors. It was introduced in 2022 by Chuyi Li et al. [15]. The models in the YOLOv6 framework are designated with the letters N, S, M, and L, which stand for various sizes and capacities. Regarding network design, YOLOv6 features different-sized architectures tailored for industrial applications, with small models having a single-path backbone and large models built on efficient multi-branch blocks.

YOLOv6 uses RepVGG backbones to improve feature representation in small networks while preserving quick inference. YOLOv5, on the other hand, uses EfficientNet-based spines. YOLOv6's neck utilises the PANet architecture from YOLOv4 and YOLOv5, which is further optimised for an effective Rep-PAN using RepBlocks or CSPStackRep Blocks. YOLOv6 reduces the number of middle 3x3 convolutional layers to one by streamlining its decoupled head and using the Efficient Decoupled Head architecture. A new version, YOLOv6 v3.0, was presented by Chuyi Li et al. to improve accuracy and performance while lowering inference latencies [16].

YOLOv7

YOLOv7 was developed and released by Wang, Bochkovskiy, and Liao in 2022, delivering remarkable speed and precision [17]. It is characterized by a significant change in architecture, and a trainable "bag-of-freebies", a term originally conceived in the YOLOv4 paper from the same authors, a set of machine learning methods to improve the accuracy of an object detection model.

Extended-Efficient Layer Aggregation Network builds upon the original ELAN by using group convolution to increase the richness of extra features. It uses a shuffle and merge strategy to mix these enriched features from multiple groups. This strategy improves the learning of many features across different maps while also optimizing parameter consumption and calculations [17].

The ability to find an optimal balance between parameters, called Model Scaling, computational power, speed, and accuracy is necessary to better fit the requirements of different devices. This process typically involves adjusting picture size, layer count, channel number, and feature pyramid levels. Here, they propose a new compound scaling method for concatenation-based models, in which the three factors are considered together, namely the depth factor and output channel of the computational block, and the width factor on the transition layers [17].

RepConv, a performant convolutional approach, was found to perform poorly with different backbones like ResNet and DenseNet due to its identity connections in conjunction with ResNet's residual and DenseNet's concatenation. Thus, RepConvN was proposed without these identity connections.

Deep supervision involves integrating extra heads within networks to bolster performance, even within well-established architectures like ResNet and DenseNet. Label assignment methods have transitioned from directly labelling ground truth to creating softer labels by amalgamating network predictions with ground truth. Effectively allocating these softer labels to both auxiliary and primary heads was traditionally done by handling label assignments separately. By contrast, the proposed approach uses predictions from the primary head to guide the learning process for both heads, including tactics like lead head-guided assignment and a coarse-to-fine strategy, dynamically regulating label significance throughout training to optimize learning while preserving crucial information. These methodologies aim to refine training processes by leveraging enhanced label assignment and deep supervision for superior neural network proficiency [17].

YOLOv8

Released in January 2023 by Ultralytics [18], YOLOv8 appears to outperform earlier iterations, but it is difficult to make strong statements about this particular model due to the lack of peer-reviewed academic papers that evaluate its performance relative to previous canonical YOLO versions. Using the MS COCO dataset, Ultralytics claims that their YOLOv8 model outperforms YOLOv5, YOLOv6-2.0, and YOLOv7 in both accuracy and inference speed [18]. These are attributed to architectural changes, which include an “anchor-free split Ultralytics head”, and “Advanced Backbone and Neck Architectures”, details of which have not been disclosed yet. Github user RangeKing summarised and illustrated the YOLOv8 architecture, with significant changes involving replacing C3 modules with an improved version called C2f [19]. Others have at least corroborated YOLOv8’s state-of-the-art performance, including Solawetz and Frencesco (2023) [20]. But this is not necessarily universal, as seen by Li et al. (2023) and Wang, Bagci, and Adu-Gyamfi (2023), who have observed an increase in accuracy at the cost of equivalent or worse inference speed [21], [22].

Metrics

Object detection models, including YOLO, must be evaluated using a variety of metrics to assess different aspects of its capabilities. Frames Per Second (FPS) evaluates the speed of the various techniques [23].

There is often a tradeoff between precision and recall regarding object detection. Plotting precision against recall and varying the confidence threshold for detection creates a Precision-Recall curve. Average Precision (AP) is the average of precision values at different recall levels and is calculated by computing the area under the Precision-Recall curve. An expansion of average precision is Mean Average Precision, or mAP. While we add individual objects for Average precision, mAP provides the precision for the entire model.

Variant	Input size	Dataset	FPS	AP	mAP
YOLOV1	448*448	Voc 2007 + Voc 2012	45	63.4	63.4
YOLOV2	608*608	COCO	40	-	48.1
YOLOV3	608*608	COCO	20	57.9	57.9
YOLOV4	608*608	COCO	33	65.7	65.7
YOLOV5	608*608	COCO	48	-	50.7
YOLOV6	608*608	COCO	520	-	43.1
YOLOV7	608*608	COCO	160	69.7	56.8
YOLOV8	608*608	COCO	280	53.9	-

Table 1: Comparative Analysis of YOLO variants with metrics [9].

Applications

Applications of YOLO models are incredibly diverse, including autonomous vehicles [24]–[27], facial recognition [28]–[32], aerial photography [33]–[37], and medical imagery [38]. Generic object detection models struggle with several features that are common to imagery for autonomous vehicles and aerial photography: small objects; serious and prolonged occlusion; class imbalance; and different weather and lighting conditions. However, it can be argued that these applications do not necessarily include significant challenges that deviate from generic object detection. For example, Qi et al. (2023) posit that “face detection is just a sub-task of general object detection” because many of these challenges are present in general object detection, and what is intuitively seen as unique properties, like facial expressions and makeup, are analogous to distortion and colour [37]. In addition, model size, parameters, and complexity are important due to constraints on computing resources in these applications. These challenges have led authors to augment existing architectures with mixed results, often creating trade-offs between improved accuracy, inference speed, and model size [26].

Neural architecture optimization is often done to compress the model size and parameters for use in resource-constrained devices. This typically manifests in proposed models as replacements of various modules for more lightweight counterparts. For example, the insertion of C3Ghost and Ghost modules into the YOLOv5s’ neck network by Dong et al. (2022) has shown a small increase in accuracy and a significant reduction in the number of parameters, model size, and GFLOPs [26].

Conclusion

The evolution of YOLO has been a constant effort to introduce innovation to improve speed and accuracy while overcoming the limitations of the previous version. The paper outlines the architecture of each YOLO version and discusses enhancements from the prior version. This article also highlights metrics and a table that compares the performance of all YOLO variants on those metrics, making it easy to glance at the overall evaluation of each YOLO variant. The paper further mentions the standard application of YOLO, which makes this model more relevant to integrate into the real world.

Some directions for future research have been identified in this review.
1. Use advances in the object detection field to iteratively improve the YOLO model by producing new versions through modifications. An example of such a change is replacing the backbone, dependent on the target device and constraints.
2. Deploy the YOLO models in real-world applications, and improve them through augmentations to solve common problems. For example, advances in object detection for aerial photography have found solutions to small object detection, such as transformer-style attention mechanisms.
3. Exhaustive evaluation of Ultralytics YOLOv8 through ablation studies to identify key changes that contribute to its increased performance.

References

[1] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using YOLO: challenges, architectural successors, datasets and applications,” *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Aug. 2022, doi: 10.1007/s11042-022-13644-y.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” arXiv, 1506.02640, May 2016. Available: https://arxiv.org/abs/1506.02640

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” arXiv, 2015. doi: 10.48550/arxiv.1506.02640.

[4] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger.” arXiv, 2016. doi: 10.48550/arxiv.1612.08242.

[5] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement.” arXiv, 2018. doi: 10.48550/arxiv.1804.02767.

[6] P. Adarsh, P. Rathi, and M. Kumar, “YOLO v3-Tiny: Object Detection and Recognition using one stage improved model,” in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, Mar. 2020. doi: 10.1109/icaccs48705.2020.9074315.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection.” arXiv, 2020. doi: 10.48550/arxiv.2004.10934.

[8] P. Ding, H. Qian, and S. Chu, “SlimYOLOv4: lightweight object detector based on YOLOv4,” *J Real-Time Image Proc*, vol. 19, no. 3, pp. 487–498, Feb. 2022, doi: 10.1007/s11554-022-01201-7.

[9] 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,” *Int J Comput Intell Syst*, vol. 16, no. 1, Aug. 2023, doi: 10.1007/s44196-023-00302-w.

[10] J. S. Jun 29 and 2020. 15. M. Read □ □ □ □, “What is YOLOv5? A Guide for Beginners.” Roboflow Blog. Accessed: Dec. 14, 2023. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

[11] 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, Jan. 2022, doi: 10.3390/s22020464.

[12] A. Krenzer *et al.*, “A Real-Time Polyp-Detection System with Clinical Application in Colonoscopy Using Deep Convolutional Neural Networks,” *J. Imaging*, vol. 9, no. 2, p. 26, Jan. 2023, doi: 10.3390/jimaging9020026.

[13] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A Forest Fire Detection System Based on Ensemble Learning,” *Forests*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/f12020217.

[14] Ultralytics, “Architecture Summary.” Accessed: Dec. 14, 2023. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/architecture_description

[15] C. Li *et al.*, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications.” arXiv, 2022. doi: 10.48550/arxiv.2209.02976.

[16] C. Li *et al.*, “YOLOv6 v3.0: A Full-Scale Reloading.” arXiv, 2023. doi: 10.48550/arxiv.2301.05586.

[17] 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, 2022. doi: 10.48550/arxiv.2207.02696.

[18] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics.” Jan. 2023. Accessed: Dec. 14, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[19] “Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics,” GitHub. Accessed: Dec. 14, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189

[20] J. Solawetz, F. Jan 11, and 2023. 12. M. Read □ □ □ □, “What is YOLOv8? The Ultimate Guide.” Roboflow Blog. Accessed: Dec. 14, 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/

[21] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, “A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition,” *Drones*, vol. 7, no. 5, p. 304, May 2023, doi: 10.3390/drones7050304.

[22] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, “Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8.” arXiv, 2023. doi: 10.48550/arxiv.2304.08256.

[23] J. Kaur and W. Singh, “Tools, techniques, datasets and application areas for object detection in an image: a review,” *Multimed Tools Appl*, vol. 81, no. 27, pp. 38297–38351, Apr. 2022, doi: 10.1007/s11042-022-13153-y.

[24] D. Tian *et al.*, “SA-YOLOv3: An Efficient and Accurate Object Detector Using Self-Attention Mechanism for Autonomous Driving,” *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 5, pp. 4099–4110, May 2022, doi: 10.1109/tits.2020.3041278.

[25] R. Wang *et al.*, “A Real-Time Object Detector for Autonomous Vehicles Based on YOLOv4,” *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–11, Dec. 2021, doi: 10.1155/2021/9218137.

[26] X. Dong, S. Yan, and C. Duan, “A lightweight vehicles detection network model based on YOLOv5,” *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104914, Aug. 2022, doi: 10.1016/j.engappai.2022.104914.

[27] Y. Han, F. Wang, W. Wang, X. Li, and J. Zhang, “YOLO-SG: Small traffic signs detection method in complex scene,” *J Supercomput*, Jul. 2023, doi: 10.1007/s11227-023-05547-y.

[28] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, “UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective,” *Sensors*, vol. 20, no. 8, p. 2238, Apr. 2020, doi: 10.3390/s20082238.

[29] Y. Hu, X. Wu, G. Zheng, and X. Liu, “Object Detection of UAV for Anti-UAV Based on Improved YOLO v3,” in *2019 Chinese Control Conference (CCC)*, IEEE, Jul. 2019. doi: 10.23919/chicc.2019.8865525.

[30] K. Boudjit and N. Ramzan, “Human detection based on deep learning YOLO-v2 for real-time UAV applications,” *Journal of Experimental &amp; Theoretical Artificial Intelligence*, vol. 34, no. 3, pp. 527–544, Apr. 2021, doi: 10.1080/0952813x.2021.1907793.

[31] Z. Liu, X. Gao, Y. Wan, J. Wang, and H. Lyu, “An Improved YOLOv5 Method for Small Object Detection in UAV Capture Scenes,” *IEEE Access*, vol. 11, pp. 14365–14374, 2023, doi: 10.1109/access.2023.3241005.

[32] L. Zhao and M. Zhu, “MS-YOLOv7:YOLOv7 Based on Multi-Scale for Object Detection on UAV Aerial Photography,” *Drones*, vol. 7, no. 3, p. 188, Mar. 2023, doi: 10.3390/drones7030188.

[33] S. Ramachandran, J. George, S. Skaria, and V. V.V., “Using YOLO based deep learning network for real time detection and localization of lung nodules from low dose CT scans,” in *Medical Imaging 2018: Computer-Aided Diagnosis*, K. Mori and N. Petrick, Eds., SPIE, Feb. 2018. doi: 10.1117/12.2293699.

[34] C. Li, R. Wang, J. Li, and L. Fei, “Face Detection Based on YOLOv3,” in *Recent Trends in Intelligent Computing, Communication and Devices*, Springer Singapore, 2019, pp. 277–284. doi: 10.1007/978-981-13-9406-5_34.

[35] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, “YOLO-face: a real-time face detector,” *Vis Comput*, vol. 37, no. 4, pp. 805–813, Mar. 2020, doi: 10.1007/s00371-020-01831-7.

[36] S. Khan, A. Akram, and N. Usman, “Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV,” *Wireless Pers Commun*, vol. 113, no. 1, pp. 469–480, Mar. 2020, doi: 10.1007/s11277-020-07224-2.

[37] D. Qi, W. Tan, Q. Yao, and J. Liu, “YOLO5Face: Why Reinventing a Face Detector,” in *Lecture Notes in Computer Science*, Springer Nature Switzerland, 2023, pp. 228–244. doi: 10.1007/978-3-031-25072-9_15.

[38] I. Pacal *et al.*, “An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets,” *Computers in Biology and Medicine*, vol. 141, p. 105031, Feb. 2022, doi: 10.1016/j.compbiomed.2021.105031.