Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD38757

YOLO-Based Object Detection: Architectural Evolution and Applications in Autonomous Driving and Medical Imaging

Yang Liu

Beijing Concord College of Sino-Canada, Beijing, China
Andyliuyang8@gmail.com

Abstract. Object detection is a fundamental task in computer vision, in which both localization and classification of objects are required. In the traditional two-stage method, for example Faster R-CNN, high accuracy is achieved but real-time inference speed is limited, restricting its use in real-time detection applications. With the introduction of the You Only Look Once (YOLO) model, this limitation is addressed by reformulating image detection as a regression problem, thereby achieving faster inference. This paper reviews the structural developments of YOLO models, from the grid-based prediction and anchor box mechanism in early versions, to the recently introduced NMS-free detection method in YOLO26. Based on published papers and research studies from the original YOLO model to the most recent benchmarking works, this paper analyzes the technical mechanism behind each major improvement, and with representative applications in autonomous driving and medical imaging. The result shows that each version of YOLO keeps improving the balance between detection speed and accuracy, and the latest version also achieves better efficiency during real-time applications. In conclusion, the continuous development of YOLO makes it an important model for real-time applications, while achieving accurate small object detection and maintaining high performance under adverse conditions remain challenging.

Keywords: Object detection, YOLO model, Real-time detection, Two-stage detector

1. Introduction

Object detection is a longstanding research focus in computer vision, with broad application demand in areas such as facial identification, autonomous driving, traffic monitoring, and robotic vision [1]. The task generally involves detecting instances of different categories, such as pedestrians and vehicles, using an image or video frame. Traditional object detection relies heavily on carefully designed hand-crafted feature extraction methods, such as the Histogram of Oriented Gradients (HOG) [2]. In 2012, the deep convolutional neural network AlexNet achieved a significant advantage in the ImageNet image recognition competition [3], after which deep learning began to attract widespread attention [4], and object detection gradually entered the deep learning era.

Deep learning-based object detection methods are generally divided into two categories according to their detection pipelines [5]: two-stage detection and one-stage detection. The former follows a coarse-to-fine process, whereas the latter completes detection in a single stage. Two-stage detectors usually achieve higher localization and classification accuracy, whereas one-stage

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD38757

detectors are generally faster [5]. One-stage detection attempts to directly classify each region of interest as either background or a target object, producing class probabilities and bounding box coordinates through a single pass [6]. You Only Look Once (YOLO) is a representative one-stage detection framework [7].

Introduced in 2015, YOLOv1 divides an image into a fixed grid and simultaneously predicts bounding boxes and class probabilities for each grid cell, substantially improving detection speed compared with contemporary two-stage detectors, although its localization accuracy, particularly for small objects, remained comparatively limited [7]. Since then, the YOLO framework has undergone continuous architectural evolution, extending from its original design to versions incorporating anchor-free prediction and, most recently, to inference pipelines that eliminate the dependence on Non-Maximum Suppression (NMS) post-processing.

Although several existing reviews summarize the general development history of the YOLO series, most focus primarily on cataloging version updates, with comparatively limited attention paid to the specific mechanisms by which each architectural change affects the trade-off between detection speed and accuracy, and few connect these technical developments to concrete real-world deployment scenarios. This paper reviews the architectural evolution of YOLO, from early grid-based prediction and anchor box mechanisms to the recently introduced NMS-free detection paradigm exemplified by YOLO26, and further examines representative applications in autonomous driving and medical imaging. This paper analyzes the evolution of the core technology of YOLO, explores and discusses typical applications, and analyzes future research directions.

2. Background: from two-stage to one-stage detection

2.1. Two-stage detectors: the case of faster R-CNN

Two-stage detectors formulate object detection as a coarse-to-fine process consisting of a region proposal stage followed by a classification and regression stage [5]. Early approaches in this family, including R-CNN and Fast R-CNN, generate candidate object regions through external proposal algorithms before classifying each region with a convolutional neural network [8]. Faster R-CNN further integrates the proposal generation process into the network itself through a Region Proposal Network (RPN), which shares convolutional features with the detection network and substantially reduces computational redundancy [9]. This design allows the entire pipeline to be trained end-to-end while retaining the two-stage structure. Owing to the additional refinement provided by the proposal stage, two-stage detectors such as Faster R-CNN generally achieve higher localization and classification accuracy than one-stage alternatives, however, the sequential nature of the pipeline limits inference speed, which restricts their applicability in scenarios that require real-time detection [5].

2.2. The emergence of one-stage detection: YOLO

To address the speed limitations of two-stage detectors, one-stage detection reformulates object detection as a single regression problem, directly predicting bounding box coordinates and class probabilities from the input image in one forward pass [6]. You Only Look Once (YOLO) is a representative framework of this category [7]. Rather than relying on a separate region proposal stage, YOLOv1 divides the input image into an S × S grid. Each grid cell predicts the bounding boxes and confidence scores of objects whose centers fall within that cell [7]. This end-to-end formulation removes the sliding-window and region-proposal steps used in earlier methods,

DOI: 10.54254/2755-2721/2026.LD38757

resulting in a substantial improvement in inference speed. The elimination of a dedicated proposal stage nonetheless introduces a trade-off: because each grid cell predicts a limited number of bounding boxes, early YOLO versions were less effective at detecting small or densely packed objects compared with two-stage detectors of the same period [7]. Subsequent versions of YOLO retain this fundamental one-stage design while progressively refining the network architecture, prediction mechanism, and post-processing strategy to narrow this accuracy gap, as discussed in Section 3. Figure 1 summarizes this evolutionary trajectory, from the original YOLOv1 architecture to the NMS-free design introduced in YOLO26.

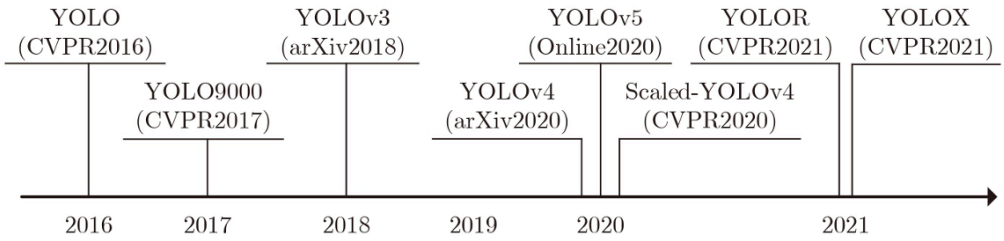

Figure 1. Timeline of YOLO architectural 1

3. Core technical evolution of YOLO

3.1. Grid-based prediction and the anchor box mechanism

Building on the grid-based prediction scheme introduced in YOLOv1 [7], subsequent versions of the framework retained this fundamental design while progressively addressing its main sources of localization error. In YOLOv1, bounding boxes are predicted directly through fully connected layers, a design that discards much of the spatial structure encoded in the convolutional feature map and consequently results in imprecise localization, particularly for small or densely arranged objects. YOLOv2, also referred to as YOLO9000, addresses this limitation by replacing the fully connected prediction layers with anchor boxes, a set of predefined boxes with fixed scales and aspect ratios [10]. Figure 2 illustrates this anchor-based prediction mechanism, showing how the network regresses from a predefined anchor box to the final predicted bounding box.

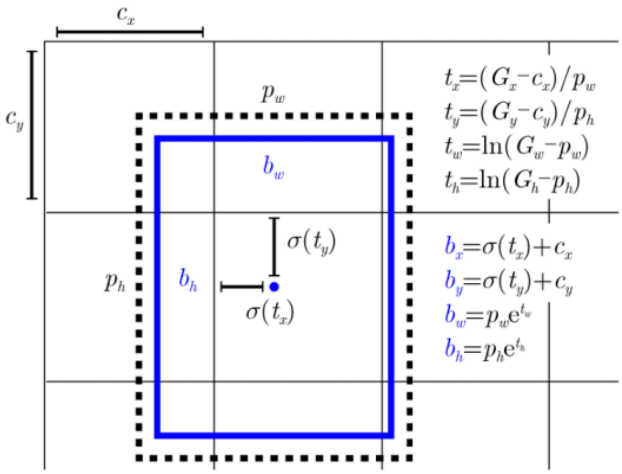

Figure 2. Bounding box prediction with dimension priors

DOI: 10.54254/2755-2721/2026.LD38757

Rather than predicting absolute bounding box coordinates, the network predicts offsets relative to these anchor priors. This reformulation improves training convergence and allows the model to accommodate a wider range of object shapes. YOLOv2 further adopts a new backbone network, Darknet-19, and resizes the input resolution to 416×416, producing an odd-numbered 13×13 output grid with a single center cell, which simplifies the prediction of large, centrally located objects [10].

YOLOv3 extends this mechanism through Darknet-53, a substantially deeper backbone that incorporates residual connections adapted from ResNet, enabling greater network depth while mitigating the vanishing-gradient problems that otherwise hinder convergence [11]. In addition to a deeper backbone, YOLOv3 introduces multi-scale prediction, generating output feature maps at three distinct spatial resolutions, each paired with three anchor boxes of different scales. This multi-scale design substantially improves detection performance across object sizes, and in particular narrows the accuracy gap for small targets that had remained a persistent weakness of earlier YOLO versions [12].

3.2. From NMS-dependent to NMS-free detection: the case of YOLO26

Grid-based, anchor-based prediction typically produces multiple overlapping bounding boxes for the same object, since neighboring grid cells and multiple anchors within a cell can independently generate high-confidence candidate detections for a single target. To resolve this redundancy, YOLO models have historically relied on Non-Maximum Suppression (NMS) as a post-processing step, in which candidate boxes are ranked by confidence score and any box that substantially overlaps, as measured by Intersection over Union, with a higher-confidence box is suppressed, leaving only the most confident detection for each object [13]. Although effective, this hand-crafted procedure introduces several practical drawbacks: it depends on a manually tuned IoU threshold, is non-differentiable and therefore cannot be optimized jointly with the rest of the network, and adds inference latency that becomes difficult to bound in dense scenes, a limitation of particular concern for deployment on power- and compute-constrained edge devices.

YOLOv10 was the first version to directly address this dependency, introducing a consistent dual-assignment training strategy in which the network is supervised simultaneously with one-to-many and one-to-one label assignments, enabling NMS-free inference while retaining accuracy competitive with NMS-dependent counterparts [14]. Building on this direction, YOLO26, released in 2026, treats NMS-free inference not as an optional training refinement but as a structural property of the detection head itself, producing final, non-redundant bounding box predictions directly, without hand-tuned post-processing thresholds. The architecture additionally removes the Distribution Focal Loss (DFL) module used in prior versions, further simplifying the overall inference pipeline. Benchmarking results indicate that these architectural changes yield substantially reduced and more predictable inference latency, particularly for smaller model variants deployed on CPU-based edge devices, while maintaining detection accuracy comparable to earlier YOLO versions [15].

4. Real-world applications of YOLO

Across the diverse domains discussed in this section, YOLO-based models consistently demonstrate a favorable balance between detection accuracy and real-time inference speed. Table 1 summarizes representative performance results reported in the reviewed studies, spanning autonomous driving and medical imaging applications.

DOI: 10.54254/2755-2721/2026.LD38757

Table 1. Performance comparison of YOLO-based models1

Model	Domain	Dataset	Metric	Result
YOLO-Vehicle-v1s [16]	Autonomous driving	KITTI	Accuracy / FPS	92.1% / 226 FPS
YOLO-Vehicle-Pro [16]	Autonomous driving (fog)	Foggy Cityscapes	mAP@50 / FPS	82.3% / 43 FPS
LEAD-YOLO [17]	Autonomous driving (small object)	VisDrone2019	mAP@0.5 gain	+7.9%

4.1. Object detection in autonomous driving

Autonomous driving represents one of the most demanding real-world settings for object detection, requiring both high accuracy and strict real-time performance, since perception errors can directly translate into safety risks on the road [1]. A persistent challenge in this domain is the degradation of detection performance under low-visibility conditions such as haze and fog, which reduces the reliability of standard YOLO models trained primarily on clear-weather imagery. To address this issue, Li et al. propose YOLO-Vehicle and YOLO-Vehicle-Pro, which combine multimodal image-text fusion, an image dehazing module, and a cloud-edge collaborative deployment scheme that offloads computation to the cloud under complex conditions [16]. On the KITTI dataset, the YOLO-Vehicle-v1s model achieves 92.1% accuracy while maintaining a detection speed of 226 FPS and an inference time of 12 ms, under simulated adverse weather on the Foggy Cityscapes dataset, YOLO-Vehicle-Pro attains an mAP@50 of 82.3% while sustaining 43 FPS, demonstrating that real-time performance can be preserved even under degraded visual conditions [16].

A second major challenge in autonomous driving is the detection of small and distant objects, including pedestrians, cyclists, and vehicles located far from the sensor, which occupy a limited number of pixels and are therefore prone to being missed. LEAD-YOLO, built upon the YOLOv11n backbone, incorporates a lightweight gated transformer module and a dilated feature fusion structure to improve multi-scale context modeling without substantially increasing model size [17]. On the nuImages dataset, LEAD-YOLO achieves improvements of 3.8% and 5.4% in mAP@0.5 and mAP@[0.5:0.95], respectively, while reducing parameter count by 24.1% relative to its baseline, on the more densely populated VisDrone2019 dataset, these gains increase to 7.9% and 6.4% [17]. Together, these studies illustrate that the architectural foundation established in Section 3 provides a general speed-accuracy baseline, which domain-specific modifications subsequently adapt to address real-world degradations, such as adverse weather and scale variation, that are not explicitly handled by the core YOLO architecture.

4.2. Object detection in medical imaging

YOLO's real-time efficiency has similarly driven its adoption in medical image analysis, where automated detection of tumors, lesions, and anatomical structures across modalities such as computed tomography, magnetic resonance imaging, ultrasound, and endoscopy supports diagnostic workflows and clinical decision-making [18]. Interest in this area has grown rapidly, with related publications increasing from four in 2018 to 175 in 2024, reflecting both technical progress and expanding clinical relevance [18]. A representative case is colonoscopic polyp detection, in which a

DOI: 10.54254/2755-2721/2026.LD38757

YOLOv5-based model achieves a precision of 97.26%, a recall of 96.15%, and an mAP@0.5 of 89.44%, results that indicate sufficient reliability for near-real-time assistance during endoscopic procedures [19]. A broader systematic review covering 148 studies further confirms that YOLO-based methods have been applied across diverse imaging modalities and disease categories, including tumor, cardiac, and thyroid applications [19]. However, existing reviews also caution that the generalization of the newest NMS-free architectures, such as YOLOv10 and YOLO26, to specialized medical tasks remains insufficiently validated, since most reported results to date rely on earlier YOLO versions [18]. Taken together, Sections 4.1 and 4.2 demonstrate that the core architectural advances discussed in Section 3 translate into measurable practical benefits, while also revealing that effective real-world deployment typically requires additional domain-specific adaptation beyond the base YOLO framework.

5. Conclusion and future directions

This paper has reviewed the architectural evolution of YOLO, tracing its progression from the grid-based prediction and anchor box mechanisms introduced in early versions to the NMS-free detection paradigm exemplified by YOLO26, and has examined representative applications in autonomous driving and medical imaging. The analysis indicates that successive YOLO versions have consistently improved the balance between detection speed and accuracy, and that recent architectural changes—particularly the removal of NMS and the distribution focal loss module—have further reduced inference latency and improved deployment predictability on edge devices [15]. The case studies in Section 4 further show that this progress translates into tangible benefits across domains, although real-world deployment still depends on domain-specific adaptations, domain-specific adaptations, such as dehazing modules for adverse weather and lightweight attention mechanisms for small-object detection, that extend beyond the core YOLO architecture [16, 17].

Several challenges remain for future research. Small object detection continues to constrain performance in both aerial and street-level scenes, as shown by the substantial gains still achievable through specialized modules such as those in LEAD-YOLO [17]. Similarly, the generalization of the newest NMS-free architectures to specialized domains such as medical imaging requires further validation before widespread clinical adoption [18]. Looking further ahead, YOLO27, expected to be released in September 2026, is set to extend the framework beyond two-dimensional detection toward three-dimensional perception, introducing YOLO-Depth for monocular depth estimation and YOLO-StereoDepth for binocular depth estimation, positioned as a camera-native alternative to lidar for robotics applications [20]. This trajectory suggests that the future development of YOLO is likely to move beyond bounding-box localization toward broader spatial understanding, further expanding the applicability of real-time detection frameworks to robotics and autonomous navigation.

References

[1] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision, 128*(2), 261-318.

[2] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 886-893).

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097-1105).

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

DOI: 10.54254/2755-2721/2026.LD38757

[5] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access, 7*, 128837-128868.
[6] Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection. *Neurocomputing, 396*, 39-64.
[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788).
[8] Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision* (pp. 1440-1448).
[9] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91-99).
[10] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6517-6525).
[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).
[12] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767.*
[13] Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS — improving object detection with one line of code. In *2017 IEEE International Conference on Computer Vision* (pp. 5561-5569).
[14] 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.*
[15] Sapkota, R., Cheppally, R. H., Sharda, A., & Karkee, M. (2025). YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection. *arXiv preprint arXiv:2509.25164.*
[16] Li, X., et al. (2024). YOLO-Vehicle-Pro: A cloud-edge collaborative framework for object detection in autonomous driving under adverse weather conditions. *arXiv preprint arXiv:2410.17734.*
[17] Yang, Y., Yang, S., & Chan, Q. (2025). LEAD-YOLO: A lightweight and accurate network for small object detection in autonomous driving. *Sensors, 25*(15), 4800.
[18] Cai, Z., Zhou, K., & Liao, Z. (2025). A systematic review of YOLO-based object detection in medical imaging: Advances, challenges, and future directions. *Computers, Materials & Continua, 85*(2), 2255-2303.
[19] Wahid, F., Ma, Y., Khan, D., Aamir, M., & Bukhari, S. U. K. (2024). Biomedical image segmentation: A systematic literature review of deep learning based object detection methods. *arXiv preprint arXiv:2408.03393*.
[20] Ultralytics. (2026). Ultralytics YOLO27 real-time object detection. *Ultralytics Docs*. https://docs.ultralytics.com/models/yolo27
[21] Shao, Y., Zhang, D., Chu, H., Zhang, X., & Rao, Y. (2022). A review of YOLO object detection based on deep learning. *Journal of Electronics & Information Technology, 44*(10), 3697-3709.