

Enhancing Real-Time Object Detection With Yolo Algorithm

Madhuri Nanasaheb Borse*, Vijaykumar M. P.

Dept. Of Computer science & Engineering, Shreeyash college of Engineering, Chh. Sambhajinagar, India

ABSTRACT

Object detection is an important area in computer vision and is widely used in applications such as security systems, autonomous vehicles, robotics, traffic monitoring, and healthcare. In recent years, there has been a growing need for fast and accurate object detection methods that can work in real time. To achieve this, deep learning-based algorithms have become very popular, especially the YOLO (You Only Look Once) algorithm. YOLO is one of the fastest and most efficient object detection algorithms because it detects and classifies objects in a single step using Convolutional Neural Networks (CNN). Unlike traditional methods that require multiple stages for detection, YOLO processes the entire image at once, which improves speed and makes it suitable for real-time applications. This review paper presents the working principle, architecture, and development of different YOLO versions such as YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv7, and YOLOv8. The study also explains how each version improves detection accuracy, processing speed, and overall performance. In addition, various real-world applications of YOLO in surveillance systems, smart transportation, industrial automation, and medical image analysis are discussed. The paper also highlights the advantages of YOLO, including high speed, simple architecture, and real-time performance. At the same time, some limitations such as difficulty in detecting very small objects and crowded scenes are also mentioned. Overall, this paper provides a detailed review of the YOLO algorithm and its role in enhancing real-time object detection systems.

Keywords: computer vision, image processing, object detection, CNN, Accuracy.

INTRODUCTION

Object detection is one of the most important applications of computer vision and artificial intelligence. It is used to identify and locate objects within images and videos. In recent years, object detection technology has gained significant attention because of its wide use in areas such as video surveillance, autonomous vehicles, robotics, healthcare, industrial automation, and smart traffic systems. The main objective of object detection is not only to recognize objects but also to determine their exact position in an image using bounding boxes.

Traditional object detection methods were mainly based on manual feature extraction techniques and machine learning algorithms. These methods required separate processes for feature extraction, classification, and localization, making them slower and less effective for real-time applications. With the development of deep learning and Convolutional

Neural Networks (CNNs), object detection systems have become faster, more accurate, and more reliable.

Among the various deep learning-based object detection algorithms, YOLO (You Only Look Once) has become one of the most popular approaches for real-time object detection. YOLO treats object detection as a single regression problem and performs classification and localization in one step. Unlike traditional region-based detection methods, YOLO processes the entire image at once through a neural network, which significantly improves detection speed. Due to this capability, YOLO is highly suitable for real-time applications where quick decision-making is required.

The YOLO algorithm has evolved through several versions such as YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv7, and YOLOv8. Each version introduced improvements in accuracy, speed, and detection performance. These advancements have

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

made YOLO one of the leading object detection frameworks in modern computer vision research.

This paper focuses on enhancing real-time object detection using the YOLO algorithm. It presents an overview of the YOLO architecture, working mechanism, different versions, advantages, limitations, and practical applications. The study also discusses how YOLO contributes to efficient and accurate object detection in real-world scenarios.

Traditional object detection methods are often slow and computationally expensive, making them unsuitable for real-time applications such as surveillance, autonomous vehicles, and robotics. Although the YOLO algorithm provides fast and accurate object detection, challenges such as small object detection, crowded scenes, and deployment on low-resource devices still exist. Therefore, there is a need to analyze and improve YOLO-based techniques to achieve better accuracy and real-time performance in practical applications.

Problem Statement

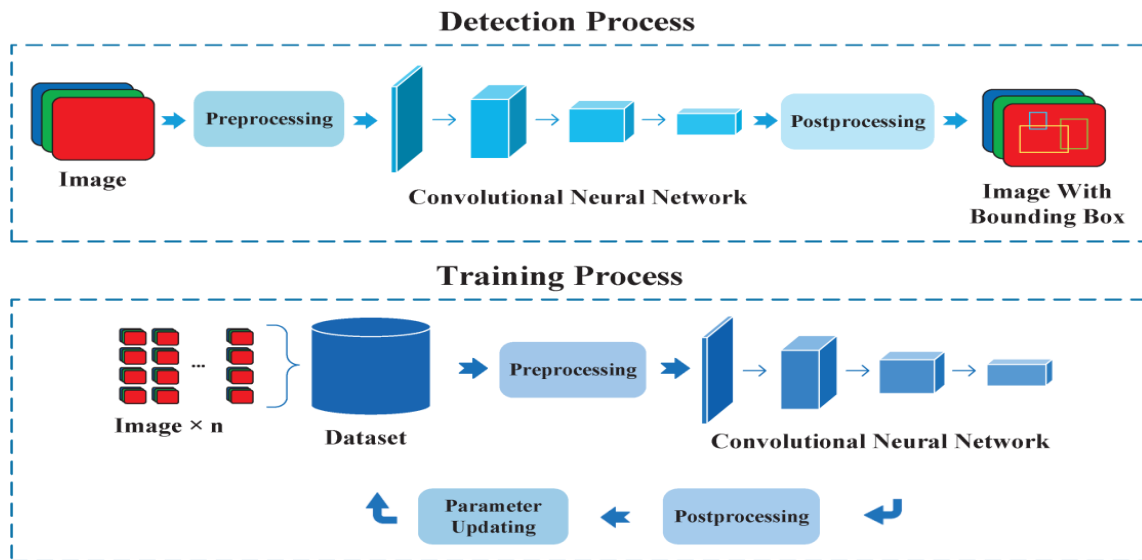


Fig.1: Block diagram

2. LITERATURE SURVEY

YOLO Version	Year	Main Feature	Advantages	Limitations
YOLOv1	2016	Single-stage object detection	Very fast detection speed	Lower accuracy for small objects
YOLOv2	2017	Batch normalization and anchor boxes	Improved accuracy and speed	Difficulty in dense object detection
YOLOv3	2018	Multi-scale prediction	Better small object detection	Larger model size
YOLOv4	2020	CSPDarknet53 backbone	High speed and accuracy	Requires powerful GPU

YOLOv5	2020	Lightweight and easy implementation	Faster training and deployment	Not officially released as research paper initially
YOLOv6	2022	Industrial optimization	Efficient for industrial applications	Less academic documentation
YOLOv7	2022	Enhanced trainable features	High real-time accuracy	Complex architecture
YOLOv8	2023	Anchor-free detection	Better accuracy and flexibility	Higher computational requirement

Table 1: Comparison of Different YOLO Versions

3. RESEARCH GAP ANALYSIS

Although YOLO algorithms have shown significant improvements in real-time object detection, several research gaps still exist. Early versions such as YOLOv1 and YOLOv2 mainly focused on increasing detection speed, but their accuracy for detecting small and overlapping objects was limited. Later versions improved accuracy and feature extraction, but challenges still remain in complex environments.

One major research gap is the detection of very small objects in crowded scenes. In applications such as traffic monitoring, drone surveillance, and medical imaging, small objects are difficult to detect accurately. Another challenge is maintaining high accuracy under poor lighting conditions, fog, rain, or blurred images.

Most advanced YOLO models also require high computational power and GPU resources, which limits their use in low-cost embedded systems and mobile devices. Researchers are still working on lightweight YOLO models that can provide both high accuracy and low processing time.

Another research gap is real-time detection in edge computing and IoT devices. Many existing models perform well in high-performance systems but face difficulties when deployed on microcontrollers, Raspberry Pi, or low-memory devices. In addition, there is a need for improving object tracking, reducing false detection rates, and enhancing performance for

multi-object detection in dynamic environments. Future research can focus on combining YOLO with artificial intelligence, edge computing, and optimization techniques to improve overall detection performance.

4. LIMITATIONS OF YOLO ALGORITHM

Although the YOLO (You Only Look Once) algorithm is widely used for real-time object detection because of its high speed and efficiency, it still has several limitations that affect its performance in certain situations. **Difficulty in Detecting Small Objects:** YOLO sometimes struggles to detect very small objects, especially when multiple small objects are present in the same image. Since the image is divided into grids, smaller objects may not be represented properly within a grid cell.

- **Lower Accuracy in Crowded Scenes:** In crowded environments where many objects overlap with each other, YOLO may fail to identify all objects correctly. This can reduce detection accuracy in applications such as traffic monitoring or public surveillance.
- **Localization Errors:** Although YOLO is fast, it may produce less precise bounding boxes compared to region-based detection methods like Faster R-CNN. This can affect accurate object localization.

- **High Computational Requirement:** Advanced versions such as YOLOv7 and YOLOv8 require powerful GPUs and high computational resources for training and real-time detection. This makes deployment difficult on low-cost devices.
- **Performance in Low-Light Conditions:** YOLO performance may decrease in poor lighting conditions, fog, rain, blurred images, or low-resolution video streams. Environmental factors can affect detection accuracy.
- **Requires Large Training Dataset:** To achieve high accuracy, YOLO requires a large amount of labeled training data. Preparing and annotating datasets is time-consuming and expensive.
- **Difficulty in Detecting Objects with Unusual Shapes:** YOLO may face challenges when detecting irregularly shaped objects because the algorithm mainly depends on rectangular bounding boxes.
- **Trade-off Between Speed and Accuracy:** Although YOLO is optimized for real-time speed, increasing speed can sometimes reduce accuracy. Lightweight models are faster but may miss certain objects.
- **Limited Performance on Edge Devices:** Deployment of complex YOLO models on embedded systems, IoT devices, or mobile platforms can be difficult due to memory and processing limitations.
- **False Positives and False Negatives:** YOLO may sometimes detect objects incorrectly (false positives) or fail to detect existing objects (false negatives), especially in complex backgrounds.

CONCLUSION

The YOLO algorithm has become one of the most powerful and widely used object detection techniques in computer vision. Its ability to perform object localization and classification in a single step makes it highly suitable for real-time applications such as surveillance systems, autonomous vehicles, robotics, healthcare, and industrial automation.

Over the years, different YOLO versions have improved significantly in terms of speed, accuracy, and detection performance. From YOLOv1 to YOLOv8, each version introduced new techniques to overcome the limitations of previous models. Among these versions, YOLOv8 provides the best balance of accuracy, flexibility, and real-time performance.

Despite these advancements, challenges such as small object detection, high computational requirements, and deployment on low-cost devices still exist. Future research can focus on lightweight architectures, improved feature extraction, and AI-based optimization methods to further enhance object detection systems.

Overall, YOLO continues to play an important role in the development of intelligent real-time vision systems and remains a leading solution in modern object detection research.

REFERENCES

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
2. Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271.
3. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.
4. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.
5. Jocher, G. (2020). YOLOv5 by Ultralytics. *GitHub Repository*. Available: <https://github.com/ultralytics/yolov5>
6. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv preprint arXiv:2207.02696*.
7. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLOv8 Documentation and Implementation. Ultralytics.

8. Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1440–1448.
9. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision (ECCV)*, 21–37.
11. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
12. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117–2125.
13. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303–338.
14. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, 740–755.
15. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NIPS)*, 1097–1105.

HOW TO CITE: Madhuri Nanasaheb Borse*, Vijaykumar M. P., Enhancing Real-Time Object Detection With Yolo Algorithm, *Int. J. Sci. R. Tech.*, 2026, 3 (6), 992-996. <https://doi.org/10.5281/zenodo.20717957>