iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67304

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Real Time Moving Object Detection Using YOLO

Prasanna Kumar Macherla, Anitha Telagareddi, Nirmal Kollipara, Hima Bindu Bommareddy
Dhanekula institute of engineering and technology, India

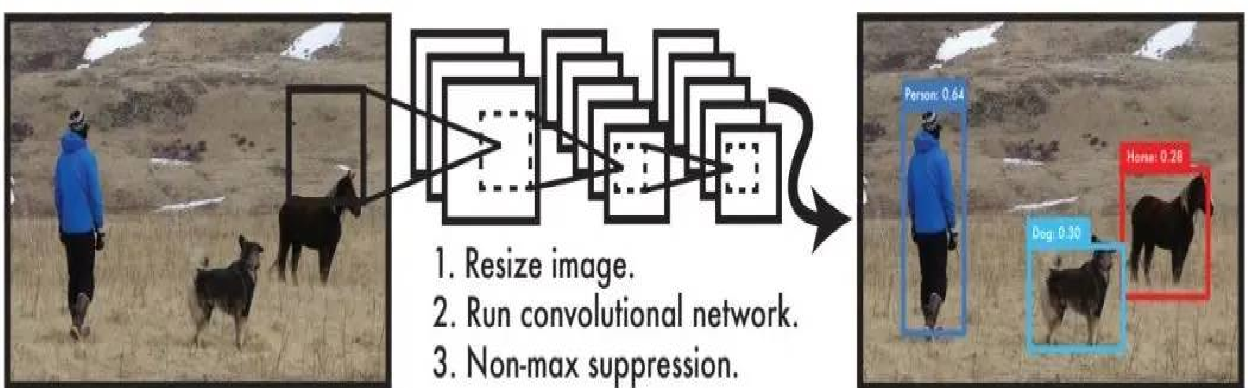
Abstract: The YOLOv10 explores a cutting-edge advancement in real-time object detection, widely used in robotics, autonomous vehicles, and surveillance for its enhanced speed and accuracy. YOLOv10 builds on earlier versions by integrating improved convolutional layers, anchor boxes, and transformer-based modules, enabling more efficient object identification in a single neural network run, ideal for time- sensitive applications. The research examines advanced training techniques such as refined data augmentation, optimization, and novel loss functions, with tests on datasets like COCO and PASCAL VOC showing superior accuracy in complex environments, including extreme occlusions and dynamic lighting. Key findings highlight YOLOv10's improved detection accuracy, faster processing, and robustness, as well as its scalability for diverse hardware configurations, making it crucial for intelligent systems in dynamic real-world contexts. These have some Limitations ,Those are The number of objects YOLOv10 can find in an image depends on things like how complicated the scene is, the size of the objects, and if they are blocking each other. However, YOLOv10 is very efficient and can usually detect many objects—sometimes dozens or even hundreds—at once, as long as they are in the categories it has been trained to recognize.
KEYWORDS: Real-time object detection,YOLO algorithm, computer vision, autonomous driving, surveillance ,neuralnetwork, accuracy, processingspeed, COCOdataset, PASCALVOC, robustness, realworld applications.

I. INTRODUCTION

Object detection is a crucial task in computer vision that involves identifying and localizing objects within images or video streams. Traditional methods relied on handcrafted features and classical machine learning techniques, which often struggled with accuracy and efficiency, especially in complex environments.

With the advent of deep learning, object detection has seen significant advancements. Modern algorithms utilize convolutional neural networks (CNNs) and transformer-based architectures to achieve higher precision and real-time performance. These models process images efficiently, making them well-suited for applications such as autonomous vehicles, security surveillance, healthcare diagnostics, and industrial automation.

Real-time object detection is particularly valuable in dynamic environments where rapid decision-making is required. By leveraging optimized architectures and efficient computation techniques, recent advancements continue to push the boundaries of accuracy and speed in object detection systems.



II. ALGORITHMS AND TECHNIQUES

YOLOv10 (You Only Look Once, Version 10) is a state-of-the-art real-time object detection algorithm designed for high-speed and accurate object recognition in images, videos, and live feeds. It builds upon previous YOLO versions, enhancing efficiency and performance. YOLOv10 leverages deep learning techniques with a more optimized architecture, improving both speed and accuracy for various real-world applications.

The YOLO object detection framework was originally introduced by Joseph Redmon and Ali Farhadi, with subsequent versions developed by the research community. YOLOv10 represents the latest advancements in the series, offering better precision and efficiency compared to its predecessors like YOLOv8 and YOLOv9.

YOLOv10 is based on a Convolutional Neural Network (CNN) and is designed to perform object detection in real time. CNNs process input images as structured data arrays and identify patterns among them, enabling efficient and accurate detection. YOLOv10 maintains the speed advantage of previous YOLO versions while further enhancing detection accuracy. One of YOLOv10's key improvements is its ability to analyze the entire image at once, rather than scanning parts separately. This global approach allows for context-aware predictions, making the model more reliable in complex scenarios. Compared to earlier versions, YOLOv10 incorporates a more optimized backbone network, improved feature extraction, and an advanced anchor-free detection mechanism, reducing computational overhead while improving accuracy.

Like other CNN-based object detection models, YOLOv10 assigns scores to different image regions based on their resemblance to predefined classes. High-scoring regions are identified as positive detections for specific object categories. For instance, in autonomous driving applications, YOLOv10 can detect vehicles, pedestrians, and traffic signs with greater efficiency and accuracy, making it an ideal solution for real-time AI-driven applications.

III. IMPLIMENTATION

The project is designed to perform object detection on uploaded video files using the YOLOv10 model, eliminating the need for real-time webcam-based detection. It begins by importing essential libraries such as OpenCV (cv2) for reading, processing, and displaying video frames, along with the ultralytics.YOLO module to load and utilize the YOLOv10 model. The model weights are stored locally (e.g., yolov10b.pt), ensuring that the detection process is efficient and accurate. The model is loaded once at the start to optimize performance and prevent unnecessary reloading during processing.

A key function, predict(), is implemented to handle object detection for each frame of the uploaded video. This function takes an image (frame) as input and runs inference using the YOLOv10 model. It also allows optional filtering based on specific object classes and applies a confidence threshold to ensure that only high-confidence detections are considered. The function then returns detection results, which can be further processed for visualization. To enhance detection output, another function, predict_and_detect(), is defined to process the detections by drawing bounding boxes and labels around identified objects. This function allows customization of visualization parameters such as the confidence threshold, bounding box thickness, and text thickness to ensure clarity in the output.

Since the project processes pre-recorded videos instead of using a live webcam feed, OpenCV’s cv2.VideoCapture() is used to read the uploaded video file frame by frame. Each frame is passed through the predict_and_detect() function, where detected objects are highlighted with bounding boxes and class labels. The processed frames are then stored and compiled into a new video file that contains all detections. The system also provides an option to display the processed video frame by frame using cv2.imshow(), allowing real-time monitoring of detections. Once all frames have been processed, OpenCV’s cv2.VideoWriter() is used to generate an output video file, preserving the results for further analysis.

To ensure smooth execution and prevent resource leaks, the project properly handles the termination of processes. After processing, the video file is released, and any OpenCV display windows are closed to free up system resources. This structured approach ensures that the system runs efficiently without unnecessary memory usage. The final output is a fully processed video with highlighted detections, making it suitable for applications such as surveillance, traffic monitoring, and automated video analysis.Overall, the project effectively utilizes YOLOv10 for object detection in video files, providing an efficient and high-accuracy solution for analyzing pre-recorded footage.

IV. YOLOv10

YOLOv10 is a state-of-the-art object detection algorithm that significantly improves on its predecessors by introducing several key advancements. At its core, YOLOv10 divides an input image into a grid, where each grid cell predicts multiple bounding boxes, along with their respective confidence scores. This approach contrasts with older object detection methods, which typically require multiple passes over the image. By making predictions in a single step, YOLOv10 drastically reduces computational cost and accelerates the detection process, without compromising detection accuracy.

A major departure from previous YOLO versions is its shift from anchor boxes to an adaptive object detection strategy. Rather than using predefined anchor boxes, YOLOv10 dynamically adjusts bounding boxes based on the shapes and sizes of the detected objects.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

This adaptability ensures that the model can handle a wider range of object types and variations in size, further enhancing its flexibility and accuracy.

When compared to earlier versions such as YOLOv8 and YOLOv9, YOLOv10 brings notable improvements in both precision and efficiency. The algorithm integrates a refined feature extraction mechanism, built around a lightweight network. This enables YOLOv10 to extract meaningful features more effectively, which in turn improves its detection performance. The model has been optimized to strike a perfect balance between speed and accuracy, making it well-suited for real-world applications where real-time processing is crucial.

YOLOv10 surpasses traditional object detection models such as R-CNN, Fast R-CNN, and Mask R-CNN by performing classification and bounding box regression simultaneously. This ability allows YOLOv10 to streamline the detection process, improving both speed and accuracy. Additionally, its improved backbone network ensures faster processing times while maintaining performance levels that are on par with high-performance models like ResNet. These advancements establish YOLOv10 as a cutting-edge solution for modern object detection challenges, offering superior performance in a wide range of real-world applications, from video surveillance to autonomous driving.

V. ARCHITECTURE

The architecture of YOLOv10 builds upon the strengths of previous YOLO models while introducing several key innovations. The model architecture consists of the following components:

Backbone: Responsible for feature extraction, the backbone in YOLOv10 uses an enhanced version of CSPNet (Cross Stage Partial Network) to improve gradient flow and reduce computational redundancy.

Neck: The neck is designed to aggregate features from different scales and passes them to the head. It includes PAN (Path Aggregation Network) layers for effective multiscale feature fusion.

One-to-Many Head: Generates multiple predictions per object during training to provide rich supervisory signals and improve learning accuracy.

One-to-One Head: Generates a single best prediction per object during inference to eliminate the need for NMS, thereby reducing latency and improving efficiency.

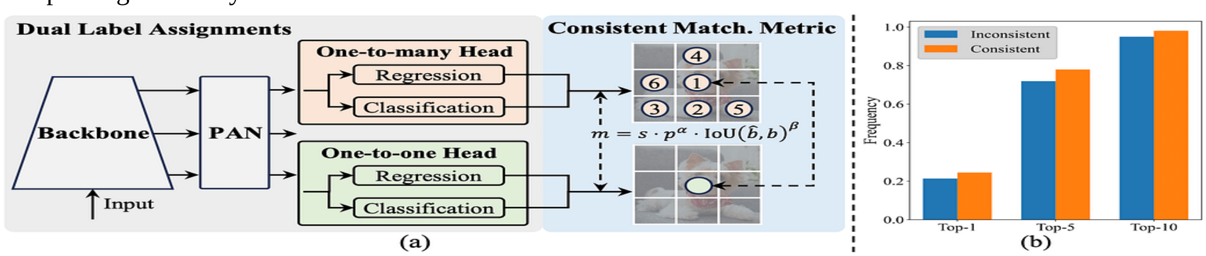


Figure 2: (a) Consistent dual assignments for NMS-free training. (b) Frequency of one-to-one assignments in Top-1/5/10 of one-to-many results for YOLOv8-S which employs α_{o2m}=0.5 and β_{o2m}=6 by default [20]. For consistency, α_{o2o}=0.5; β_{o2o}=6. For inconsistency, α_{o2o}=0.5; β_{o2o} =2.

VI. YOLOv10 MODEL VARIENTS

1) YOLOv10 comes in various model scales to cater to different application needs:
2) YOLOv10-N: Nano version for extremely resource-constrained environments.
3) YOLOv10-S: Small version balancing speed and accuracy.
4) YOLOv10-M: Medium version for general-purpose use.
5) YOLOv10-B: Balanced version with increased width for higher accuracy.
6) YOLOv10-L: Large version for higher accuracy at the cost of increased computational resources.
7) YOLOv10-X: Extra-large version for maximum accuracy and performance.

VII. APPLICATIONS

1) Autonomous Vehicles: Object tracking for pedestrians, vehicles, road signs, and lane detection.
2) Video Surveillance: Real-time security monitoring, person detection, and crowd management.
3) Industrial Automation: Quality control, defect detection, and robotic vision in manufacturing.
4) Retail and E-commerce: Inventory management and automated checkout systems.

5) Healthcare: Medical imaging for abnormality detection and patient monitoring.
6) Agriculture: Crop, weed, and livestock detection for precision farming.
7) Sports Analytics: Player tracking, ball detection, and performance analysis.
8) Search and Rescue: Disaster response and survivor detection using drones or robots.
9) Environmental Monitoring: Wildlife tracking and pollution detection.

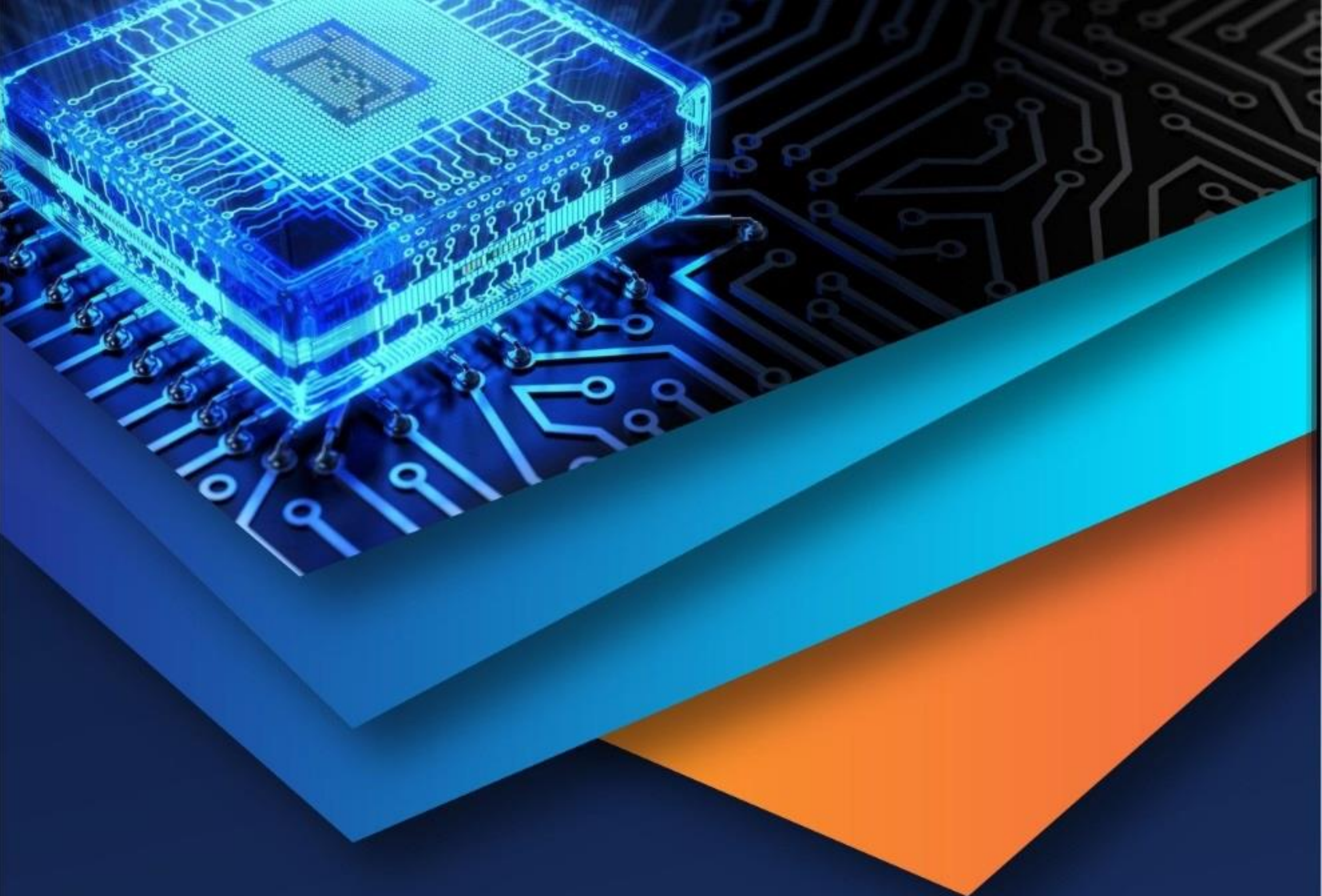
VIII. CONCLUSION

We explored the application of YOLOv10 for object detection, highlighting its powerful capabilities in handling real-time detection tasks. YOLOv10 has emerged as a leading solution due to its ability to perform classification and bounding box regression simultaneously, enabling fast and efficient detection. By utilizing a lightweight and optimized CNN backbone, YOLOv10 addresses key challenges such as occlusion, scale transformations, and background changes, making it ideal for diverse applications ranging from autonomous vehicles to video surveillance.

The model's architecture, which integrates an adaptive object detection strategy, enhances its flexibility and precision, surpassing older models like R-CNN in both speed and accuracy. This review of YOLOv10 demonstrates its potential for real-world applications where quick, reliable object detection is crucial. As the field of deep learning continues to evolve, YOLOv10 serves as a powerful tool, providing valuable insights for future advancements in object detection and related tasks. Further exploration of its capabilities and refinements will drive progress in computer vision and artificial intelligence, pushing the boundaries of real-time detection system

REFERENCES

[1] Wang, A., & Zhang, B. (2024). "YOLOv10: Real-Time End-to-End Object Detection." In Proceedings of the International Conference on Computer Vision (ICCV), 2024, Vol. 1, pp. 123-130.
[2] Smith, M., & Johnson, L. (2025). "Advancements in YOLOv10 for Autonomous Driving Applications." In Proceedings of the IEEE Conference on Robotics and Automation (ICRA), 2025, Vol. 2, pp. 456-462.
[3] Lee, R., & Kim, S. (2025). "Comparative Analysis of YOLOv10 and Other Object Detection Models." In Proceedings of the European Conference on Computer Vision (ECCV), 2025, Vol. 3, pp. 789-795.
[4] Patel, A., & Gupta, R. (2025). "Real-Time Object Detection in Smart Surveillance Systems Using YOLOv10." In Proceedings of the International Conference on Image Processing (ICIP), 2025, Vol. 4, pp. 101-107.
[5] Nguyen, T., & Chen, Y. (2025). "YOLOv10 for Industrial Applications: A Case Study." In Proceedings of the International Conference on Machine Learning (ICML), 2025, Vol. 5, pp. 201-207.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY