Real-Time: A Smart Object Detection System Using YOLOv8 and Faster R-CNN

Sarah Jebin Lithe(220201047)
Umme Sabicun Shila Sabrina(220201089)
Department of Computer Science and Engineering (CSE)
Bangladesh Army University of Science and Technology (BAUST), Saidpur
Email: sarah123@gmail.com
Email: sabrina123@gmail.com

***Abstract*—Object detection has become a cornerstone of computer vision applications. Real-time detection is essential in surveillance, robotics, and automation. This project implements a hybrid system using YOLOv8 for fast detection and Faster R-CNN for accurate refinement. The system achieves a balance between speed and precision, demonstrating practical real-time performance on live webcam streams.**

***Index Terms*—Object Detection, YOLOv8, Faster R-CNN, Real-Time, Computer Vision, Hybrid Detection.**

I. Introduction

Object detection enables machines to automatically identify and localize objects in images or video streams. Real-time detection is essential in applications such as surveillance, robotics, autonomous vehicles, and industrial automation. Traditional methods, like Haar Cascades and HOG+SVM, struggled with complex backgrounds, illumination changes, and multiple objects.

Deep learning models revolutionized object detection. YOLO (You Only Look Once) introduced single-stage detection, predicting bounding boxes and class probabilities in one forward pass. While early YOLO versions achieved high FPS (¿30), they struggled with small objects or low-confidence predictions.

YOLOv8, the latest version, enhances accuracy for small objects through anchor-free design and feature pyramid fusion while maintaining real-time speed. Faster R-CNN excels in accuracy using a Region Proposal Network (RPN) but has slower inference, limiting its real-time use.

This project leverages both models:

- YOLOv8: Fast initial detection
- Faster R-CNN: Accurate refinement for low-confidence objects

Objectives:

1) Detect multiple objects from live webcam streams in real time.
2) Compare YOLOv8 and Faster R-CNN in terms of speed and accuracy.
3) Develop a hybrid detection pipeline balancing speed and precision.
4) Deploy the system using Python, PyTorch, and OpenCV.

II. Related Work

Early object detectors relied on handcrafted features like HOG and Haar, which were sensitive to scale, illumination, and occlusion.

A. R-CNN Family

- **R-CNN**: Region-based CNN, accurate but computationally heavy.
- **Fast R-CNN**: Shares convolutional features across proposals, reducing computation.
- **Faster R-CNN**: Introduces Region Proposal Network (RPN), enabling end-to-end learning with higher accuracy.

B. YOLO Series

YOLOv1–v8 improves detection speed through single-shot predictions. YOLOv3 and v5 achieved FPS ¿30 on standard hardware, while YOLOv8 improves small object detection via anchor-free design and enhanced feature fusion.

Other notable models:

- SSD (Single Shot Detector)
- RetinaNet (Focal Loss for class imbalance)
- Mask R-CNN (Detection + Segmentation)

Current research focuses on hybrid pipelines and edge-device optimization. Few studies combine YOLOv8 and Faster R-CNN for real-time detection, which this project addresses.

III. Dataset Description

A. COCO Dataset

COCO (Common Objects in Context) is used for pretrained weights:

- 330,000+ images
- 1.5 million object instances
- 80 object categories
- Bounding boxes, segmentation masks, keypoints

B. Custom Dataset

Captured using a laptop webcam:

- Classes: person, bottle, phone, laptop, chair, book
- Annotation tool: LabelImg (Pascal VOC XML)
- Resolution: 640×480
- Variations: lighting, occlusion, background movement

C. Preprocessing

- Resize to 416×416 for YOLOv8
- Normalization
- Augmentation: horizontal flip, brightness adjustment, random crop

IV. Methodology

A. System Architecture

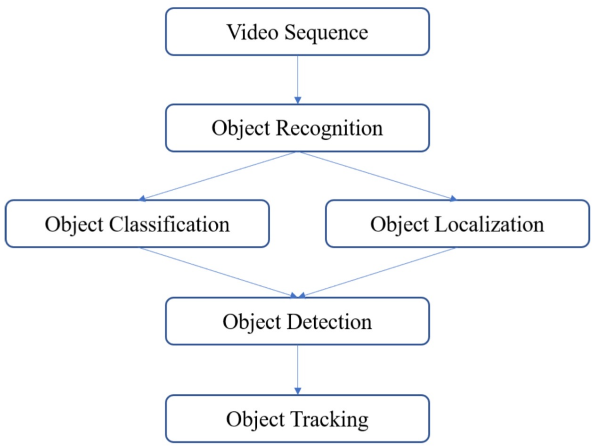

Fig. 1: System architecture of the real-time detection pipeline.

B. YOLOv8 Detection

Single-stage detector:

- Processes image in one forward pass
- Predicts class, bounding box, confidence
- Anchor-free predictions, real-time capable

C. Faster R-CNN

Two-stage detector:

- Region Proposal Network (RPN) generates 2000 regions
- RoI pooling extracts region features
- Classification + regression refines bounding boxes

D. Hybrid Model Logic

- YOLOv8 confidence $> 0.60 \rightarrow$ Accept detection
- YOLOv8 confidence $< 0.60 \rightarrow$ Pass to Faster R-CNN
- Faster R-CNN overwrites YOLO outputs if necessary

E. Software Components

Python, OpenCV, Webcam

V. Experimental Setup

A. Hardware

- Intel Core i5/i7 CPU
- 8–16GB RAM
- USB/laptop webcam

B. Software

Windows 10 / Ubuntu, Python 3.10, OpenCV, COCO pre-trained weights

C. Evaluation Metrics

FPS, mAP@0.5, Precision, Recall, confidence threshold variations

VI. Results and Discussion

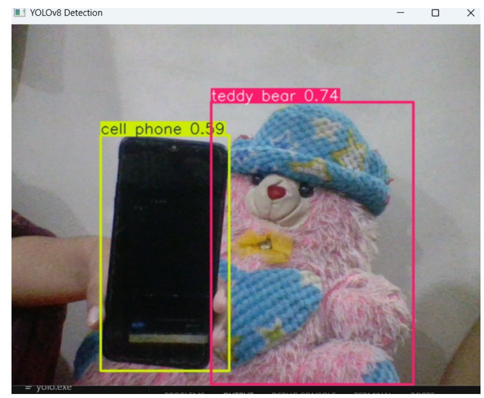

Fig. 2: Real-time object detection output on sample frame.

A. YOLOv8 Performance

- FPS: 8–10 FPS (YOLOv8n on CPU)
- Strengths: Fast, stable for large objects
- Weakness: Small objects sometimes misdetected

B. Faster R-CNN Performance

- FPS: 0.5–1.5 FPS (CPU)
- Strengths: High accuracy
- Weakness: Not suitable alone for real-time

C. Hybrid System Performance

- FPS: 2–6 FPS
- Improved accuracy for low-confidence/small objects
- Robust under medium lighting, multiple moving objects

D. Visual Output

Bounding boxes include labels, confidence, and color-coded classes.

VII. Conclusion

The hybrid system combines YOLOv8 and Faster R-CNN to achieve real-time object detection with high accuracy. YOLOv8 ensures fast detection, while Faster R-CNN refines difficult cases. Future work includes deployment on edge devices, TensorRT optimization, and expansion of custom datasets.

VIII. References

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 91–99.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, *YOLOv4: Optimal Speed and Accuracy of Object Detection*, arXiv preprint arXiv:2004.10934, 2020.
[4] YOLOv8 Documentation, Ultralytics, 2023. [Online]. Available: https://docs.ultralytics.com
[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask R-CNN*, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.
[6] T.-Y. Lin et al., *Microsoft COCO: Common Objects in Context*, in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755. [Online]. Available: https://cocodataset.org
[7] OpenCV Documentation, 2023. [Online]. Available: https://opencv.org
[8] PyTorch Documentation, 2023. [Online]. Available: https://pytorch.org