Intermediate Report: A YOLOv8-Based System for Real-Time Object Detection and Localization

Ramazan YILDIZ
Department of
Computer Engineering,
Faculty of Engineering,
Ataturk University,
Erzurum, Turkey
Email: ry.yildizramazan@gmail.com
Telephone: +90 544 448 3169

Beyza GÜLER
Department of
Computer Engineering,
Faculty of Engineering,
Ataturk University
Erzurum, Turkey
Email: beyzafeyza.2001@gmail.com
Telephone: +90 552 647 5822

Abdelrahman MOHAMED
Department of
Computer Engineering,
Faculty of Engineering,
Ataturk University,
Erzurum, Turkey
Email: abdoessammo@gmail.com
Telephone: +90 552 750 8202

mdy
November 27, 2025

***Abstract*—This intermediate report presents the current progress of our real-time object detection and localization system based on the YOLOv8 architecture. At this stage, the system operates using a pre-trained YOLOv8 model without custom fine-tuning. A subset of 12 fruit classes was selected from the Fruits-262 dataset, and 100 image samples per class were extracted for future annotation and training purposes. The current system performs real-time inference through a Streamlit-based web interface with an observed frame rate of approximately 30 FPS. Since the annotation and model training processes have not yet been completed, this system is considered an intermediate prototype. Future stages will include dataset labeling, model fine-tuning, and full quantitative evaluation.**

***Index Terms*—Object Detection, YOLOv8, Real-Time Detection, Computer Vision, Streamlit, Prototype System.**

I. Introduction

Object detection is one of the most fundamental problems in computer vision, aiming to identify and localize objects within an image or a video stream. It is widely used in applications such as autonomous driving, surveillance, medical imaging, smart agriculture, and industrial automation. Recent advancements in deep learning have significantly improved detection accuracy and inference speed.

Among modern object detection models, the YOLO (You Only Look Once) family stands out for its ability to perform real-time object detection with high accuracy. In this project, we employ YOLOv8 for real-time object detection and visualization through a web-based interface. At the current stage, the system utilizes a pre-trained YOLOv8 model for inference only, and functions as an intermediate prototype. Custom dataset labeling and model training will be performed in the next phase.

II. Related Work

Redmon et al. introduced the YOLO architecture as a single-stage detector that performs detection in real time. Subsequent versions significantly improved both speed and accuracy. YOLOv7 further optimized real-time performance through architectural improvements and better training strategies.

Ultralytics introduced YOLOv8 as a fully redesigned architecture that offers state-of-the-art detection performance with improved efficiency and modularity. YOLOv8 outperforms previous versions in both speed and accuracy while simplifying the training and deployment process.

In fruit detection studies, deep learning-based approaches have demonstrated promising results for agricultural automation. However, many previous works rely on small and domain-specific datasets with limited generalization capability. Moreover, most studies focus on offline detection rather than real-time deployment.

The major limitation of previous studies is the lack of real-time deployment with lightweight web-based interfaces. Our work aims to bridge this gap by integrating YOLOv8 with a real-time Streamlit-based deployment pipeline.

III. Problem Definition and Importance

A. Problem Definition

The primary problem addressed in this project is the real-time detection and localization of multiple object classes using a lightweight and deployable deep learning-based system. Although high-performance object detectors exist, their integration into real-time, user-friendly web platforms remains a practical challenge.

B. Importance of the Problem

Real-time object detection systems are essential for safety-critical and automation-oriented applications such as traffic monitoring, smart surveillance, and agricultural product classification. Developing an easy-to-deploy, browser-based real-time detection system increases accessibility and practical usability for non-expert users.

IV. Proposed Work

The proposed system is based on the YOLOv8 object detection architecture integrated with a real-time video processing and visualization pipeline. In the current phase, only the pre-trained YOLOv8 nano model is used for inference without additional training.

A subset of 12 fruit classes was selected from the Fruits-262 dataset, and 100 image samples were collected per class (1200 images in total). These samples will be manually annotated in the next stage to construct a custom detection dataset.

The system pipeline consists of:

- Real-time video capture using a webcam,
- Object detection using pre-trained YOLOv8,
- Visualization of bounding boxes and class labels,
- Deployment via a Streamlit web interface.

At this stage, the system functions as an intermediate prototype. Final model training and optimization will be performed in future work.

V. Methodology

A. Dataset and Tools

The Fruits-262 dataset was used as the primary data source. From this dataset, 12 fruit classes were selected, and 100 images per class were sampled. The tools used in this study include:

- Python
- Ultralytics YOLOv8
- Streamlit
- OpenCV
- PyTorch

B. Data Preprocessing

A custom Python script was developed to randomly sample 100 images from each selected class. At this stage, the images have not yet been annotated with bounding boxes. No resizing, normalization, or augmentation has been applied yet since model training has not started.

C. Detection Model

YOLOv8 nano (YOLOv8n) is currently used as the detection model. The model is pre-trained on the COCO dataset and is employed directly for inference. No fine-tuning has been performed so far. The model processes each frame and outputs bounding boxes, confidence scores, and predicted class labels.

VI. Evaluation

A. Main Hypothesis

The main hypothesis of this study is that a YOLOv8-based system can perform real-time object detection efficiently within a web-based interface while maintaining acceptable detection accuracy.

B. Evaluation Metrics

The final evaluation will be based on Precision, Recall, mean Average Precision (mAP), and Intersection over Union (IoU). These metrics cannot yet be computed because model training has not been completed.

C. Experimental Results

At the current stage, only qualitative results are available. The system successfully performs real-time detection using the webcam with an observed frame rate of approximately 30 frames per second. Sample detections demonstrate that the model can accurately detect common objects using the pre-trained weights.

D. Methods for Comparison

At this intermediate stage, no comparative analysis has been conducted. In the final stage, the fine-tuned YOLOv8 model will be compared with the pre-trained model to evaluate performance improvements.

VII. Conclusion

This intermediate report presented the current state of a YOLOv8-based real-time object detection system. The system currently operates as a prototype using a pre-trained model without fine-tuning. Dataset sampling has been completed for 12 fruit classes, but labeling and training have not yet begun.

The system demonstrates stable real-time inference performance with approximately 30 FPS using a Streamlit-based web interface. The next phase of the project will focus on image annotation, model training, quantitative evaluation, and performance comparison. At this stage, the system is considered an intermediate prototype rather than a finalized solution.

VIII. References

References

[1] J. Redmon et al., ”You Only Look Once: Unified, Real-Time Object Detection,” CVPR, 2016.

[2] C. Wang et al., ”YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” arXiv, 2023.

[3] Ultralytics, ”YOLOv8 Documentation,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[4] S. Bargoti and J. Underwood, ”Deep Fruit Detection in Orchards,” IEEE Robotics and Automation Letters, 2021.

[5] A. ElChimminut, ”Fruits-262 Dataset,” Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/aelchimminut/fruits262