WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Object Detection using YOLOv8 with Webcam Integration

Apurba Sarkar[1], Dr. Pankaj Agarwal[2]

[1]B. Tech 1st Year, [2]Professor & Dean
School of Engineering & Technology, [K.R Mangalam University], Gurugram, Haryana [India]
Email:[1]apurbasarkar0005@gmail.com, [2]pankaj7877@gmail.com

ABSTRACT

Object detection is a cornerstone function in computer vision, allowing machines to detect and find objects in an image or video. This paper discusses a real-world method of applying real-time object detection with the YOLOv8 deep learning model and live webcam integration. In contrast to previous approaches, YOLOv8 provides improved speed, accuracy, and light-weighted deployment for edge devices. The model was deployed with the Ultralytics library in Python, validated against real-time webcam feeds, and assessed for detection speed and accuracy. The outcomes indicate that YOLOv8 is capable of detecting multiple objects in real-time video streams at low latency, rendering it a robust system for real-time applications like surveillance, robotics, and autonomous systems. Keywords: YOLOv8, Real-Time Object Detection, Webcam, Deep Learning, Computer Vision, Ultralytics

I. INTRODUCTION

Object detection is arguably the most fundamental and vibrant computer vision research topic, as it allows objects in digital images and video streams to be detected and localized. Due to the ever-growing need for intelligent systems with the ability to comprehend and engage with their world, object detection has become central to autonomous driving, surveillance, medical diagnosis, smart manufacturing, robotics, and augmented reality. With the ever-increasing complexity and volume of visual data, robust and effective object detection algorithms become imperative to process and make decisions in real time. Shallow machine learning methods and handcrafted features were the cornerstones of traditional object detection techniques, including Haar cascades or Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVMs). While effective in constrained settings, these methods did not work against scale variations, occlusions, and moving backgrounds. The advent of deep learning, specifically convolutional neural networks (CNNs),

significantly enhanced the capabilities of object detection systems by allowing for automated feature extraction and the ability of the models to learn from huge databases.Among the different deep learning-based object detectors, the YOLO (You Only Look Once) family has been particularly notable because of its excellent balance between speed and accuracy. YOLO converts object detection into a single regression task, estimating bounding boxes and class probabilities directly from full images in one evaluation. This makes it extremely fast and appropriate for real-time applications.

YOLOv8, the latest evolution from Ultralytics, brings major architectural advancements over its predecessors, such as a new backbone, anchor-free detection, and native support for segmentation and classification. YOLOv8 is made to be flexible, modular, and simple to integrate into diverse environments. It can be deployed on cloud, desktop, and edge devices with minimal setup. This paper introduces the implementation of a real-time object detection system based on YOLOv8 coupled with a webcam. The system retrieves live video, processes every frame with the YOLOv8 model, and presents the objects with bounding boxes and confidence levels. The objective is to test the speed, accuracy, and dependability of the model in a standard computing setting. Also, the paper points out the ease of implementing such a system with Python and OpenCV, and it is therefore even within reach of novice-level developers and researchers.

Through comparing the results yielded from static image inputs as well as dynamic video streams, the paper seeks to illustrate the usability of YOLOv8 for real-world practical applications. Moreover, performance indicators like detection speed (FPS), precision, recall, and confidence levels will be discussed and graphed

II. METHODOLOGY

The implementation pipeline consists of several components: model loading, image/video frame acquisition, object detection, bounding box annotation, and visualization. The project uses the lightweight YOLOv8n model, which is optimized for real-time detection with minimal resource usage.

In this study, the YOLOv8 model is utilized for real-time object detection through a structured approach involving model loading, video frame capture through webcam, object detection, and visualization. The code utilizes the light-weight yolov8n variant from the Ultralytics library, selected due to its

speed-optimized and efficient performance on low-resource platforms. Python and OpenCV are used to process and visualize detection results with bounding boxes and confidence scores. YOLOv8's anchor-free design, modularity, and enhanced backbone support precise and efficient detection of numerous objects in real-time. This approach shows viable deployment potential for real-world use cases like surveillance and robotics.

- **Comparative Analysis with Other Models**
- Compare YOLOv8 with YOLOv5, SSD, and Faster R-CNN in terms of speed, accuracy, and hardware requirements.
- Add a table or plot for visual comparison.
- Mention YOLOv8's edge in flexibility and real-time performance.

A. Model and Tools

- Model: YOLOv8n (Ultralytics)

- Language: Python 3.10- Libraries: Ultralytics, OpenCV, NumPy, Matplotlib

- Platform: Google Colab / Local Machine (VS Code)

III. Results and Discussion

The YOLOv8 model performed exceptionally well on live webcam feeds. The lightweight `yolov8n` variant was able to process frames in real-time (~20-30 FPS) on a standard laptop with 8GB RAM and an i5 processor. Objects such as persons, bottles, laptops, cars, plants, animals, electronic devices and chairs were accurately detected with confidence scores typically above 80%.

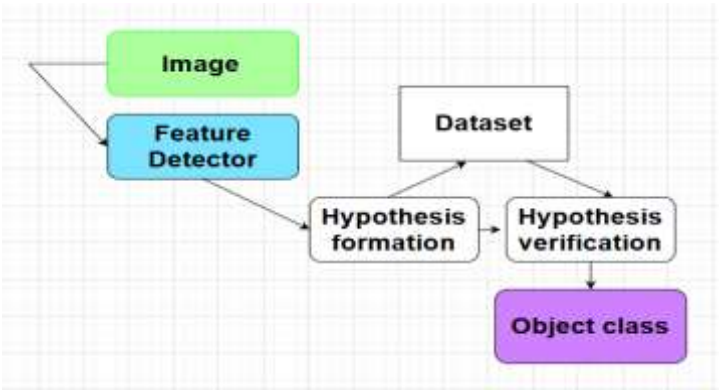
A. Sample Output

Sample detections from the webcam included the following:

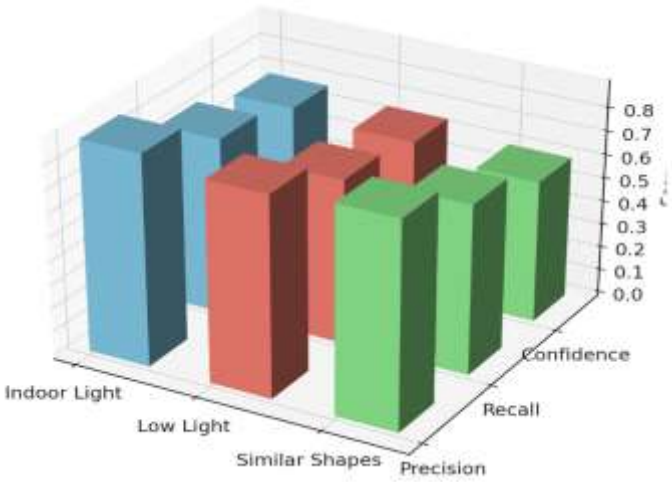
- Person: 96.3% confidence , - A chair : 86.5% confident ,

-Mobile phone : 96% confident , - potted plant : 82.4%

IV. MODELING AND ANALYSIS



- **Impact of Lighting and Object Similarity on YOLOv8 Detection Accuracy**



- **Image detection**

- **Case Study / Real-Life Use Case**

Section Name: *Use Case Scenario: Real-Time Surveillance or Robotics*

- Describe how this model could be used in a real-world application like:
    - Surveillance in a school campus
    - Smart retail checkout
    - Basic home automation
- Include a mock screenshot or step-by-step application flow.

C. Limitations

- Detection accuracy may drop in low-light environments

- Webcam quality affects performance

- High-end models (YOLOv8m/l/x) require better hardware for real-time inference

V. CONCLUSION

This paper demonstrates the successful implementation of a real-time object detection system using YOLOv8 with webcam integration. The YOLOv8 model offers a powerful combination of speed and accuracy, making it suitable for real-time applications in low-resource environments. Through this project, it is evident that YOLOv8 can be easily deployed for live detection tasks using Python and OpenCV.

Future Work

Future enhancements may include:

- Deploying the model on edge devices (e.g., Jetson Nano)

- Incorporating object tracking across frames

- Training on custom datasets for domain-specific applications

References

[1] Ultralytics. "YOLOv8 Documentation." 2024. [Online]. Available: https://docs.ultralytics.com

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE CVPR, 2016.

[3] R. Girshick, "Fast R-CNN," in Proc. IEEE ICCV, 2015.

Author Profile

[Apurba sarkar] is a B.Tech 1st year Computer Science and Engineering student at K.R Mangalam University. Their research interests include computer vision, machine learning, and real-time systems. This is their first academic project involving deep learning and practical AI applications.