

Real-Time Object Detection Using Deep Learning

Mrs. S. A. Kulkarni¹, Jayesh Patil², Vedant Patil³, Shantanu Patil⁴, Sajan Koul⁵

¹Guide; Department of Information Technology, PES Modern College of Engineering, Pune, Maharashtra

^{2,3,4,5}Students; Department of Information Technology, PES Modern College of Engineering, Pune, Maharashtra

DOI: <https://dx.doi.org/10.51244/IJRSI.2025.12110094>

Received: 24 November 2025; Accepted: 30 November 2025; Published: 10 December 2025

ABSTRACT

Real-time object detection is a crucial task in computer vision, enabling intelligent systems to identify and classify multiple objects from visual data streams such as images and videos. Traditional detection methods relied heavily on manual feature extraction and suffered from limited scalability in dynamic environments. This paper presents an intelligent system for Real-Time Object Detection Using Deep Learning, utilizing the YOLOv8 (You Only Look Once) architecture integrated with a Flask-based web interface. The proposed system detects and labels multiple objects in live webcam feeds, video inputs, or static images with high accuracy and low latency. It leverages convolutional neural networks (CNNs) for feature extraction and performs training on a custom dataset enhanced through extensive data augmentation. This research demonstrates the potential of integrating deep learning with web-based technologies for real-world applications such as surveillance, industrial monitoring, and autonomous systems.

Keywords: YOLOv8, Object Detection, Deep Learning, Flask Web Application, Computer Vision, Convolutional Neural Networks (CNN).

INTRODUCTION

In the modern era of Artificial Intelligence (AI), computer vision has become a vital branch that enables computers to understand and interpret visual data. Object detection, one of the most fundamental applications of computer vision, involves locating and classifying objects within images or videos. Traditional approaches such as Haar Cascades and Histogram of Oriented Gradients (HOG) depend heavily on handcrafted features and lack adaptability in dynamic real-world environments [1]. With the advent of deep learning and Convolutional Neural Networks (CNNs), object detection has achieved significant improvements in speed, accuracy, and reliability.

The proposed project aims to build a deep learning-based real-time object detection system using the YOLOv8 model. YOLO (You Only Look Once) is a single-stage detector known for its speed and precision, making it ideal for real-time applications. This project integrates YOLOv8 with a Flask-based web application that allows users to interactively upload images, process video feeds, or use live webcams for object detection. Such a system has potential applications in smart surveillance, autonomous driving, and accessibility enhancement for visually impaired individuals [2].

Objectives

The objectives of this project are centered around developing a robust, efficient, and scalable real-time object detection system using deep learning techniques. The detailed objectives include:

- **Accurate Detection:** To design a YOLOv8-based model that can recognize and locate multiple objects in real-time images and videos with high reliability, even in complex scenes.
- **Speed Optimization:** To ensure low-latency processing by optimizing model inference speed using GPU acceleration and efficient model loading techniques for real-time performance.

- **Scalability:** To provide an easily extensible architecture where new object categories can be introduced by retraining on custom datasets without altering the entire model pipeline.
- **Environmental Robustness:** To make the detection system adaptable to changing environmental factors like lighting, motion blur, and occlusion, ensuring stable performance in practical applications.
- **User-Friendly Interface:** To design a responsive, browser-based web interface using Flask that allows users to perform object detection seamlessly through webcam, video, or image uploads.
- **Secure Deployment:** To implement authentication and access control, ensuring that system usage remains secure and consistent with ethical AI principles [3].

LITERATURE SURVEY

Research in real-time object detection has evolved significantly with the advancement of deep learning frameworks. The study by R. K. et al. [1] presented a comparative analysis of YOLO-based architectures, emphasizing how YOLOv8 improves upon earlier versions by adopting a decoupled head, enhanced backbone, and anchor-free detection approach. This enables higher speed and accuracy for small and large objects alike.

Mohammed Kawser Jahan et al. [2] proposed an enhanced YOLOv8 model for improving online platform safety by detecting objectionable content in live streams. Their work introduced fine-tuning techniques and hyperparameter optimization that significantly reduced false positives. The integration of YOLOv8 with edge AI technology demonstrated the model's ability to operate efficiently under limited hardware conditions.

U. Dwivedi et al. [3] explored moving object detection using YOLOv5 and introduced tracking mechanisms using Kalman filters and DeepSORT. This allowed continuous object tracking in videos, a concept that forms the foundation for integrating detection and tracking in future iterations of our system.

S. Borkar et al. [4] proposed a reinforcement learning approach combined with YOLOv4 to dynamically improve detection accuracy over time. Their hybrid framework demonstrated adaptability to environmental changes by rewarding models for successful detections.

Afdhal et al. [5] applied YOLOv8 for self-driving cars, emphasizing performance in mixed traffic environments. Their research validated YOLOv8's scalability and real-time efficiency, proving it to be an ideal choice for critical applications like automated driving.

Problem Statement

Traditional object detection methods are computationally expensive and inefficient in real-time scenarios. They rely heavily on manual feature extraction and often fail in dynamic environments with changing light, occlusion, or motion. Therefore, there is a pressing need for an automated deep learning-based system that can detect and classify multiple objects in real-time from various input sources such as webcams, videos, and images, while maintaining low latency and offering user accessibility through a web platform [1] [2].

Existing System

Existing systems like SSD, Faster R-CNN, and early YOLO versions have contributed immensely to object detection. However, most of these rely on anchor-based detection and complex region proposal mechanisms that increase inference time [3]. Systems such as Faster R-CNN offer high precision but are not suitable for real-time performance due to their two-stage processing nature. SSD and YOLOv3 improved on speed but still struggle with smaller objects or cluttered scenes.

Moreover, current detection solutions lack easy accessibility for non-technical users. They often require installing specialized software or deep learning libraries locally. The absence of an integrated, browser-based interface further limits their usability in practical applications such as surveillance or live analytics [4].

Disadvantages of the Existing System

- **Limited Real-Time Performance:** Conventional models struggle to maintain high frame rates for real-time scenarios [2].

- **Complex Setup:** Most implementations require extensive setup, including GPU configuration and dependency management.
- **Low Adaptability:** Systems often fail in non-ideal environments with poor lighting or multiple overlapping objects [5].
- **No Web Integration:** The lack of a web interface restricts accessibility for broader audiences, especially non-developers.
- **High Computational Requirements:** Training and inference demand high-end hardware, making deployment costly for smaller institutions.

Proposed System

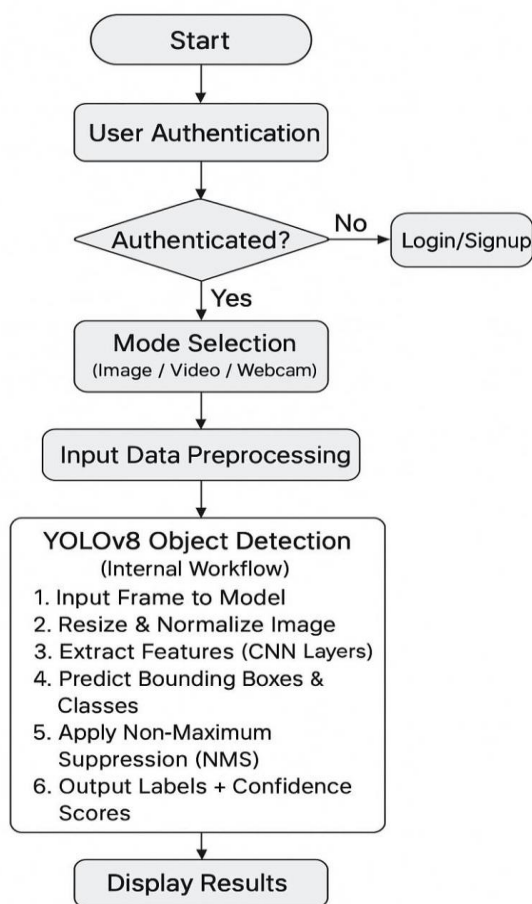
The proposed system integrates YOLOv8 with a Flask-based web platform to build a robust, efficient, and user-friendly real-time object detection solution. YOLOv8 uses a CNN backbone for feature extraction, followed by a detection head that predicts bounding

boxes and class probabilities directly. Its anchor-free architecture reduces complexity, while advanced optimization techniques ensure faster and more accurate inference [1][2].

The system is designed to process multiple input sources including static images, pre-recorded videos, and live webcam streams. The Flask web interface acts as a bridge between the user and the detection model. Once the input is received, the backend preprocesses the data, feeds it into the YOLOv8 model, and visualizes the detection results with bounding boxes and labels in real time.

The architecture includes key modules: data acquisition (input from user), preprocessing (resizing, normalization), model inference (YOLOv8 detection), and result visualization (bounding boxes, labels, and scores). The integration of Flask enables a responsive, platform-independent environment that supports multiple users and devices. Security features such as access control ensure safe use in restricted environments [4][5].

Proposed System Diagram



Working Details

Real-time object detection will be done using a YOLOv8 model integrated with a Flask web interface.

It begins with the preparation of the data, including resizing, normalizing, and augmenting images to increase model accuracy. The YOLOv8 model will be trained based on this dataset. Architecture:

Backbone: It extracts important features using CNN layers.

Neck: It combines multiscale features to detect both small and large objects.

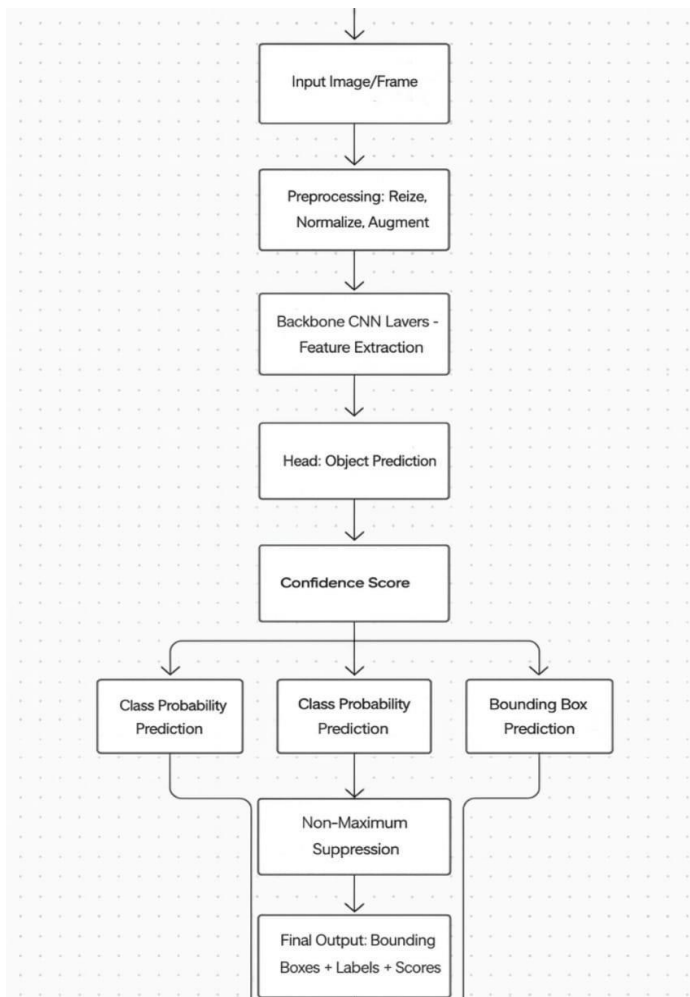
Head: Predicts bounding boxes and class probabilities directly (anchor-free).

This system processes the inputs-image, video, and webcam frame-in the following steps during inference:

1. Preprocess the input frame.
2. Extract features and predict multiple bounding boxes.
3. Apply Non-Maximum Suppression to eliminate overlapping boxes and retain only the best.
4. Display the final detections with bounding boxes, class names, and confidence scores.

This interface can be used through Flask for uploading any files or using a live camera for detection. Results processed will be instantly shown inside the browser fast, accurate, and user- friendly for applications like surveillance and automation.

Workflow of YOLOv8



System Architecture Diagram

The following diagram represents the proposed system architecture for Real-Time Object Detection Using Deep Learning. It illustrates the data flow from user input through the Flask web interface to preprocessing, YOLOv8 model inference, and output.

System Architecture Diagram

Advantages of the Proposed System

- **Real-Time Detection:** Provides immediate detection feedback using live camera feeds or uploaded files.
- **User-Friendly Web Interface:** Offers an intuitive and responsive interface accessible through any browser.
- **Scalability:** New object categories can be added easily by retraining the model with updated datasets.
- **Cross-Platform Support:** Compatible with Windows and Linux operating systems.
- **Secure Access:** Includes authentication mechanisms for controlled user interaction.
- **Data Visualization:** Displays bounding boxes, labels, and confidence scores directly on the video or image stream.

CONCLUSION

This study proposes a comprehensive real-time object detection system integrating YOLOv8 and Flask. The system design prioritizes accessibility, scalability, and real-time interaction. Through a browser-based interface, users can perform object detection on images, videos, or live feeds without technical expertise. This framework lays a strong foundation for future extensions such as object tracking and smart surveillance applications[1][5]

REFERENCES

1. R. K. et al., "A Perspective Study of Real-Time Object Detection Using Deep Learning," IEEE MITADTSOciCon , 2024.
2. Mohammed Kawser Jahan et al., "Enhancing the YOLOv8 Model for Realtime Object Detection to Ensure Online Platform Safety," Scientific Reports 15 (1), 2025.
3. U. Dwivedi et al., "Overview of Moving Object Detection Using YOLO Deep Learning Models," IEEE ICDT, 2024.
4. S. Borkar et al., "Dynamic Approach for Object Detection Using Deep Reinforcement Learning," IEEE SPACE, 2024.
5. Afdhal et al., "Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars," IEEE COSITE, 2023.