

# Final Project Report

## Project Title:

Real-Time Object Detection using YOLOv3 and OpenCV

## Objective:

To develop a real-time object detection system that identifies and labels multiple objects from a live webcam feed using the YOLOv3 deep learning model and OpenCV.

## Tools & Technologies Used:

Component	Description
Language	Python
Library	OpenCV (cv2), NumPy
Model	YOLOv3 (You Only Look Once v3)
Pretrained Weights	`yolov3.weights` from official YOLO site
Config File	`yolov3.cfg`
Labels File	`coco.names` – containing 80 object classes
Hardware	Webcam (for real-time detection)
IDE	VS Code (Visual Studio Code)

## Project Description:

This project implements a real-time object detection system using YOLOv3, a state-of-the-art object detection algorithm. The system detects multiple object types (person, dog, car, etc.) from live webcam feed and displays their labels and confidence scores in real-time.

## Dataset & Model:

- COCO Dataset: Common Objects in Context – YOLOv3 is trained on 80 different classes from this dataset.
- YOLOv3 Weights: Pre-trained on COCO dataset, providing fast and accurate detection.

## Key Components:

- • YOLOv3 Model Loading: Load weights and config files using `cv2.dnn.readNet()`.
- • Class Labels: Loaded from `coco.names`.
- • Video Stream: Captured from webcam using `cv2.VideoCapture(0)`.
- • Blob Conversion: Image is preprocessed into a blob using `cv2.dnn.blobFromImage()`.
- • Forward Pass: Model performs prediction on blob to get detections.
- • Bounding Box & Confidence: Boxes with confidence > 0.5 are drawn

- • Display Results: Bounding boxes and labels are shown in real-time using OpenCV window.



### Features:

- • Real-time object detection from webcam
- • Labels with confidence scores
- • Detects all 80 classes from COCO (e.g., person, car, dog, bicycle, tv, etc.)
- • High-speed inference using YOLOv3

### Sample Detected Objects:

- • Person
- • Teddy Bear
- • Cell phone

### Project Folder Structure:

#### object-detection-project/

- └─ object\_detection.py # Main Python script
- └─ yolov3.cfg # YOLOv3 configuration file
- └─ yolov3.weights # YOLOv3 pretrained weights
- └─ coco.names # 80 COCO class labels

### **Future Improvements:**

- • Use YOLOv4 or YOLOv8 for better accuracy.
- • Add logging of detected objects and timestamps.
- • Add voice feedback (text-to-speech).
- • Deploy on mobile or Raspberry Pi for real-world application.
- • Replace webcam input with IP camera or video files.

### **Conclusion:**

This project successfully demonstrates the power of YOLOv3 in detecting multiple objects in real-time using a webcam. By combining OpenCV and deep learning, this project lays the foundation for more complex applications like surveillance, autonomous vehicles, and smart home systems.

***"Vision is the art of seeing what is invisible to others."***

---