

# Object Detection with YOLO Models

## Overview

This project leverages cutting-edge YOLO models (YOLOv3, YOLOv5, YOLOv8) for object detection tasks. Following comparative testing, **YOLOv8** was selected for its superior performance in terms of:

- **Accuracy:** High precision in detecting objects across categories.
- **Speed:** Faster inference times compared to earlier YOLO versions.
- **Small Object Detection:** Enhanced handling of smaller objects.

The **YOLOv8 model** was trained on the **COCO dataset**, ensuring robust detection across a diverse set of object categories.

## Running the Application

Detailed instructions to set up and run the application are provided in the [readme.md](#) file. Once the application is running, follow these steps:

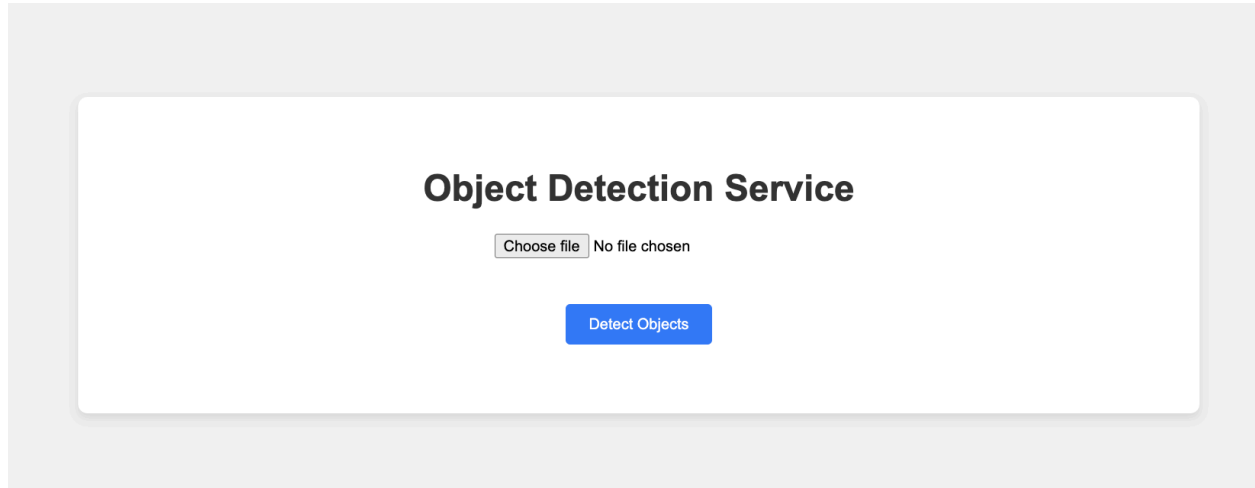
1. **Upload an Image:** Use the provided interface to upload an image.
2. **Detect Objects:** Click on the **"Detect Objects"** button.
3. **View Results:** The application will process the image and return the following:
  - A **JSON object** containing detection details.
  - The **processed image** with detected objects highlighted.

## Output Details

- **Input and Output Images:**
  - Uploaded images are stored in the **temp** folder.
  - Processed images are saved in the **output** folder.
- **JSON Response:** The detection results are returned in a structured JSON format, providing:
  - Object labels
  - Confidence scores
  - Bounding box coordinates

## Interface:

Below are the screenshots of the interface, an example of the output, illustrating the processed image:



## Object Detection Service

Choose file premium\_ph...191935f.jpeg

Detect Objects

### Detection Results

**Class:** person  
**Confidence:** 90.81%  
**Bounding Box:** [747.7742919921875,425.5343933105469,2329.05078125,1979.2108154296875]



### Detailed JSON Output

```
{
  "detections": [
    {
      "bbox": [
        747.7742919921875,
        425.5343933105469,
        2329.05078125,
        1979.2108154296875
      ],
      "class": "person",
      "confidence": 0.9081020355224609
    }
  ],
  "output_image": "output/premium_photo-1664474619075-644dd191935f.jpeg"
}
```

## Resources and References

To implement the solution, I utilized the following resources:

1. [How to Detect Objects in Images Using YOLOv8](#)
2. [Open Images 2019 Object Detection \(Kaggle\)](#)
3. Various **YouTube tutorials**
4. **ChatGPT** for assistance and guidance