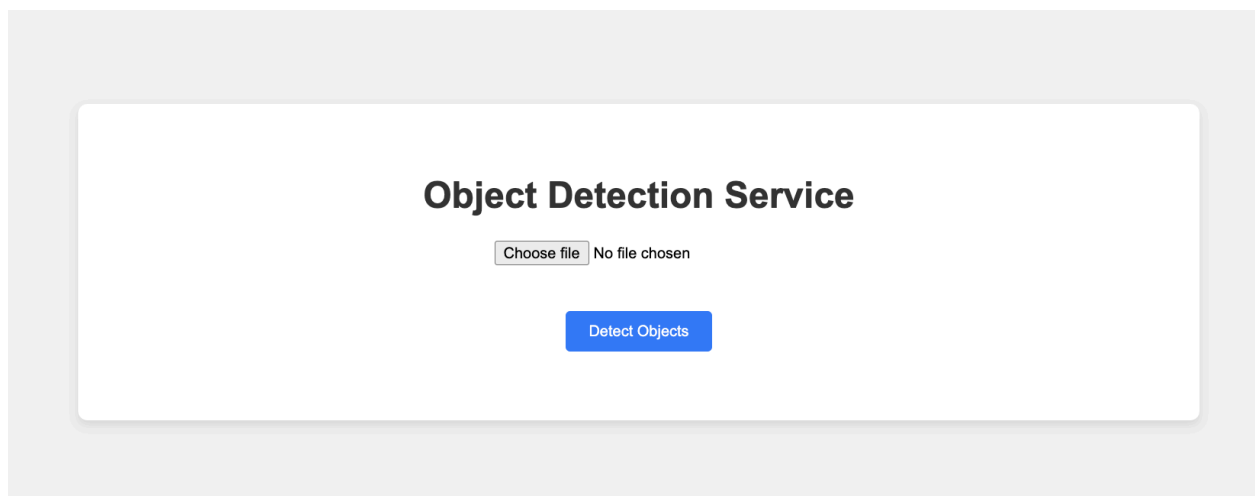


Overview

This project utilizes state-of-the-art YOLO models (YOLOv3, YOLOv5, YOLOv8) for object detection tasks. After comparative testing, **YOLOv8** was chosen due to its superior performance in accuracy, speed, and handling of smaller objects. The YOLOv8 model was trained on the COCO dataset, ensuring robust detection capabilities across diverse object categories.

The steps to run the application are mentioned in readme.md. After running the application the user can upload the image from here:



After uploading the file and clicking the detect objects button the results will look like the below:


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"
}
```

As mentioned in the assessment, the JSON object is returned. I have also returned the processed image. The input and output images are stored in temp and output folders, respectively.

I took help from several articles and resources to reach the solution:

- <https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/>
- <https://www.kaggle.com/code/csmohamedayman/open-images-2019-object-detection>
- Youtube Videos
- ChatGPT