

Problem Statement : -

Build an Object Detection Model which takes an image as input and returns the predictions of the detections and print the image with Bounding Boxes over the images and also return the detections in JSON format

Solution Proposed: -

Build a Frontend and Back end application to execute the project. For Frontend i have used Flask API and for Back end i have used Python. A HTML file also developed for UI interaction of the Application.

Algorithm used for object detection is the YOLOv8 model, which is the most recent stable model developed on COCO data sets.

Approach to the solution is arrived at by doing little research on different versions of the YOLO models and ease of use from models.

Execution steps to replicate the results: -

Approach -1 : - Using the zip file provided in the mail

Step : 1. Download the zip file and unzip it

Step : 2. Open the folder in VSCode

Step : 3. Open the terminal in VSCode and point to the current folder using cd command

Example: **cd YOLO_AI_MONK**

Step : 3 Create a new virtual environment with python 3.9 or greater

The code worked well with Python 3.9.6

Run the following command in terminal to create a new venv

python3 -m venv yolo_exp

Yolo_exp is the new venv created

Step : 4. Activate the new venv with following command in terminal

source yolo_exp/bin/activate

This will activate the new environment

Step : 5. Install the requirements using requirements.txt file, the file has all the dependencies need for running the current application

pip install -r requirements.txt

Step : 6. Once the requirements.txt is completed then run the following code to test the application locally

python yolo_detection.py

This open the application on your default browser chrome or safari

Step : 7. Open your default browser and run the following url in the browser

http://localhost:8080/

You will be able to see a HTML file where you can upload the image and see the results like image with detection showing in the image and below the detection image you can see the predictions with labels and bounding box coordinates

Sample results are shown in the result section of the documents

Approach 2: cloning the source code from git repository

Create a new folder and cd to your new folder using VS code terminal

Run the following command in your VScode terminal

Run : git clone https://github.com/sheri-tsnlgit/AI_Monk_Assessment.git

Once the code is cloned to your folder then follow the same steps as mentioned in approach 1

Step : 2. Open the folder in VSCode

Step : 3. Open the terminal in VSCode and point to the current folder using cd command

Example: **cd YOLO_AI_MONK**

Step : 3 Create a new virtual environment with python 3.9 or greater

The code worked well with Python 3.9.6

Run the following command in terminal to create a new venv

python3 -m venv yolo_exp

Yolo_exp is the new venv created

Step : 4. Activate the new venv with following command in terminal

source yolo_exp/bin/activate

This will activate the new environment

Step : 5. Install the requirements using requirements.txt file, the file has all the dependencies need for running the current application

pip install -r requirements.txt

Step : 6. Once the requirements.txt is completed then run the following code to test the application locally

python yolo_detection.py

This open the application on your default browser chrome or safari

Step : 7. Open your default browser and run the following url in the browser

http://localhost:8080/

You will be able to see a HTML file where you can upload the image and see the results like image with detection showing in the image and below the detection image you can see the predictions with labels and bounding box coordinates

Sample results are shown in the result section of the documents

Approach 3: - If docker is installed in your laptop

Once the source code is downloaded or unzipped in your pc

cd YOLO_AI_MONK

Build a docker image using the docker file provided in the code

Build Docker image with following code by running it in your VScode terminal

```
docker build -t <docker-username>/yolo_exp:v01 .
```

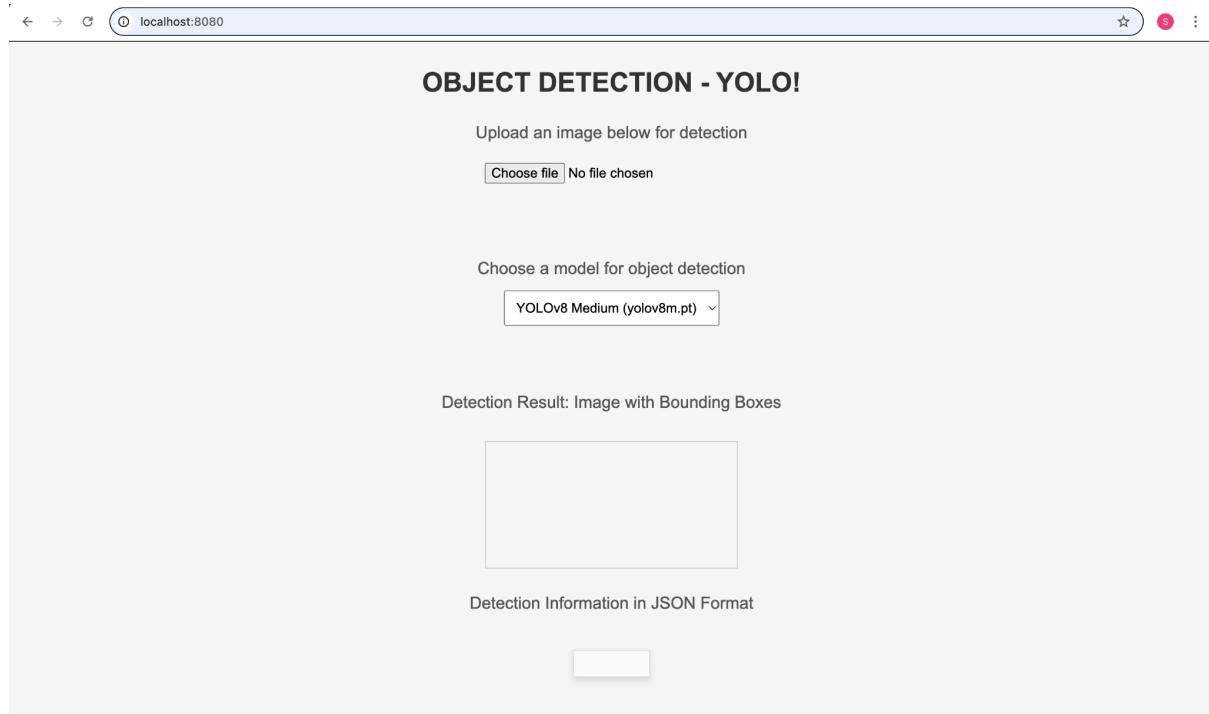
This takes time to build the docker image. Once built, run the docker container to see the results. Run the following code to see the flask application running on your default browser on <http://localhost:8080>/

```
docker run -p 8080:8080 <docker-username>/yolo_exp:v01
```

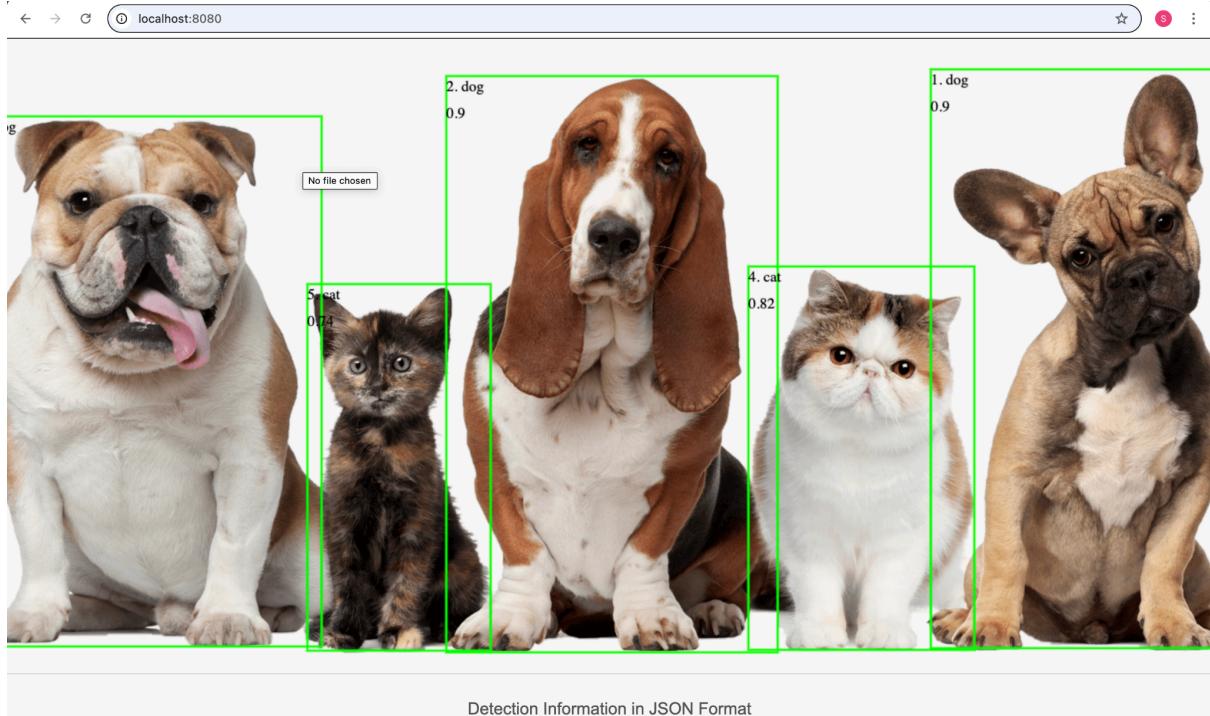
Yolo_exp is my image name, you can have your own docker image name of your choice

Sample results can be seen in the results section of the folder

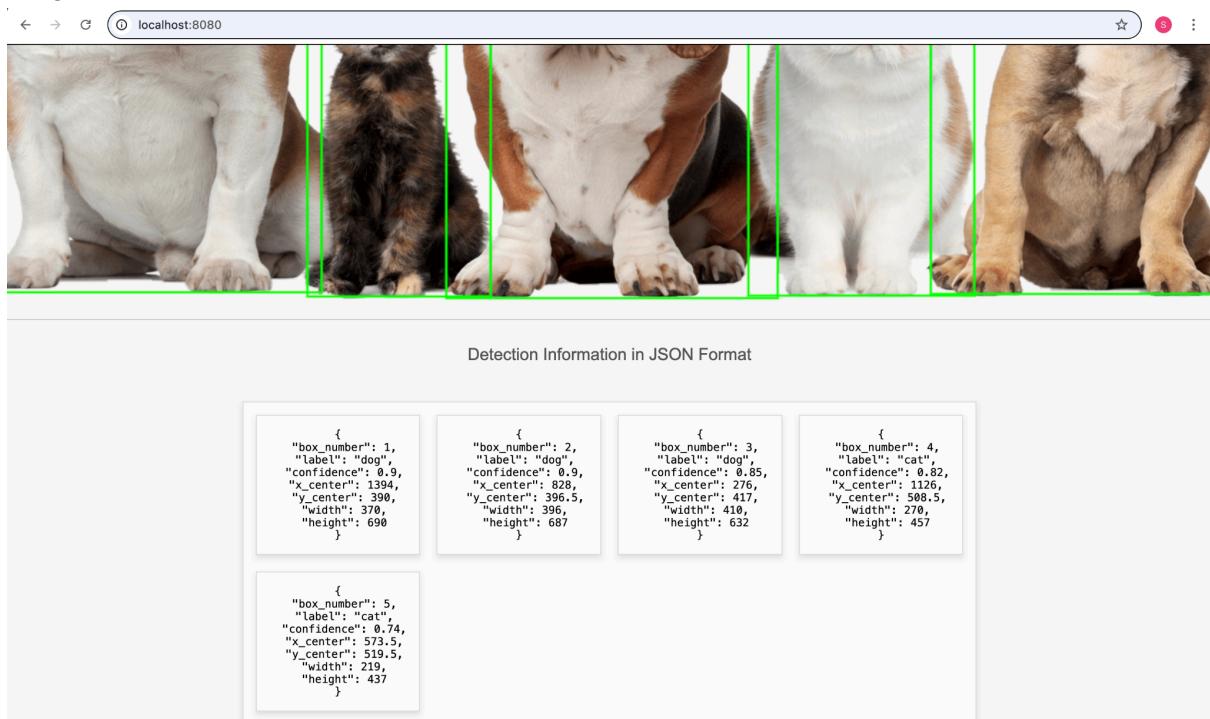
Results 🎉



The plain HTML page where one can upload your image for object detection



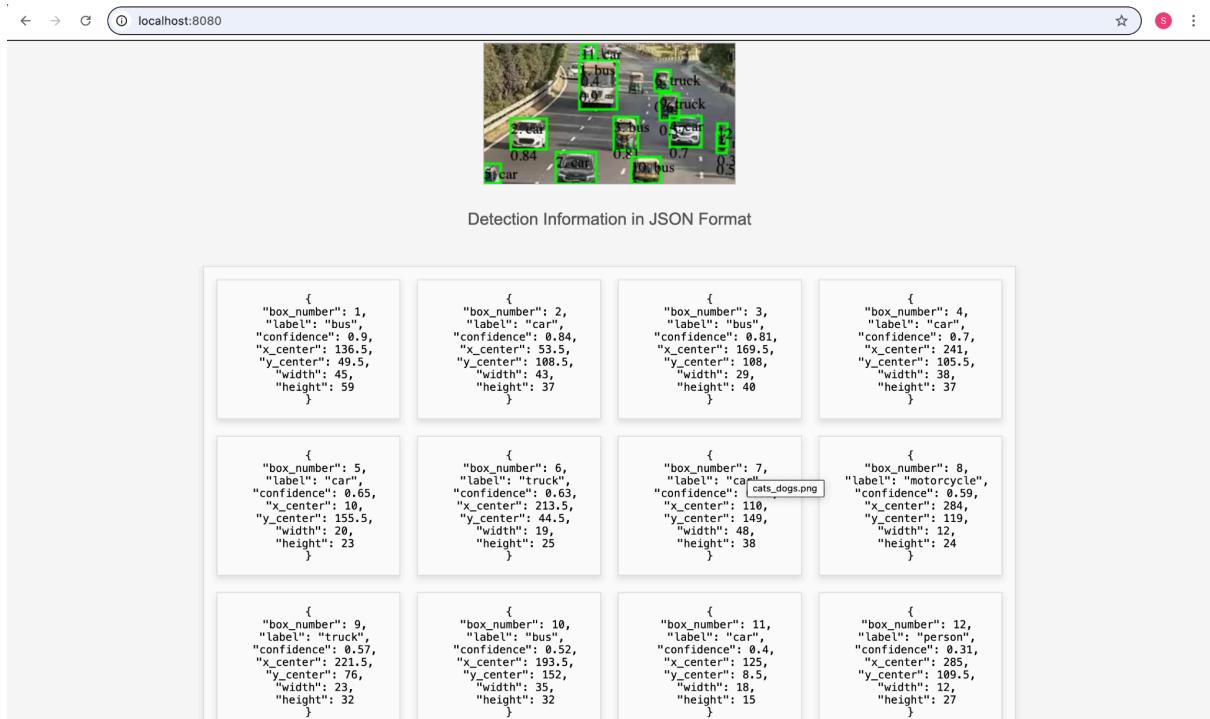
Sample output which shows the image with Bounding boxes and label name in the above image



The Above image shows JSON format of the detection results along with the label. Confidence and Bounding box center coordinates and width and height of the box.

You can find the BOX number also which helps in easy identification if the which box corresponds to which number as you can see the box number in image also

Second Example Image



References: -

1. <https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/>
2. <https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb>
3. <https://www.analyticsvidhya.com/blog/2024/03/live-object-detection-and-image-segmentation-with-yolov8/>