Infilect Project

Problem statement

The objective of this assignment is to create and deploy an AI pipeline based on the requirements provided below AI Pipeline application:

This AI pipeline application is a service that serves a simple API request and gives back the response in JSON. The application accepts the list of images in a defined format and the output should be the response containing inference through various AI models present as part of the AI pipeline. The goal is to have an application's latency as minimum as possible and scale it to as many users as possible.

Technology Used:

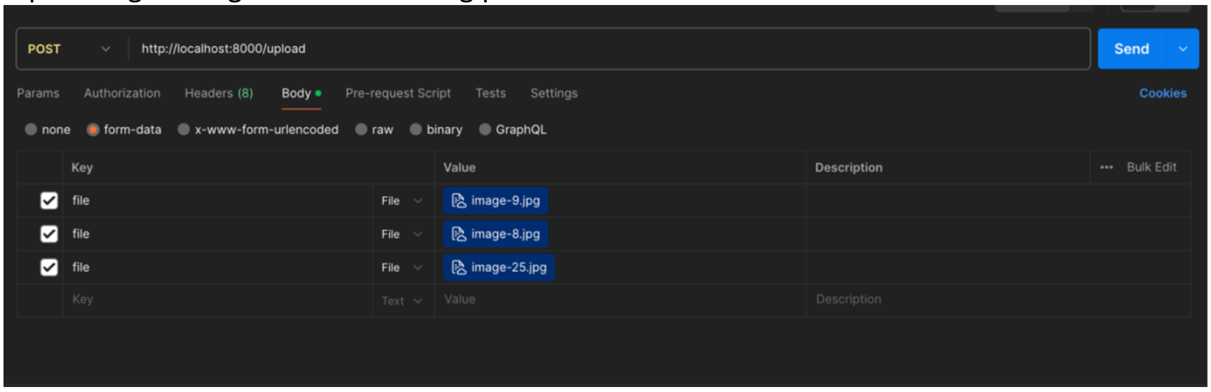
1. Python-Flask as web framework
2. Gunicorn as python web server
3. Yolov5 for detection
4. Keras inception_v3 for crops classification
5. vit-gpt2-image-captioning for captioning entire image
6. Opencv2 for image read/write

Methodology

1. We have 2 processes for detecting image using yolo and another process for captioning
2. Yolo process detects the image and crops image. Then will invoke classifier. Classifier will take all crops from folder and then one by one classify the images, it will return image name with detections of all crops within JSON
3. 2nd process invokes Vision transformer model to caption entire image
4. Results of Transformer and detections are written to image in separate to avoid overlapping of text
5. We have upload html page for uploading images and view html for viewing the results also api returns response as JSON format
6. Images are given to both processes via string separated by , which are then split, and backspaces are removed

Inputs

➔ Input images are given to flask using postman

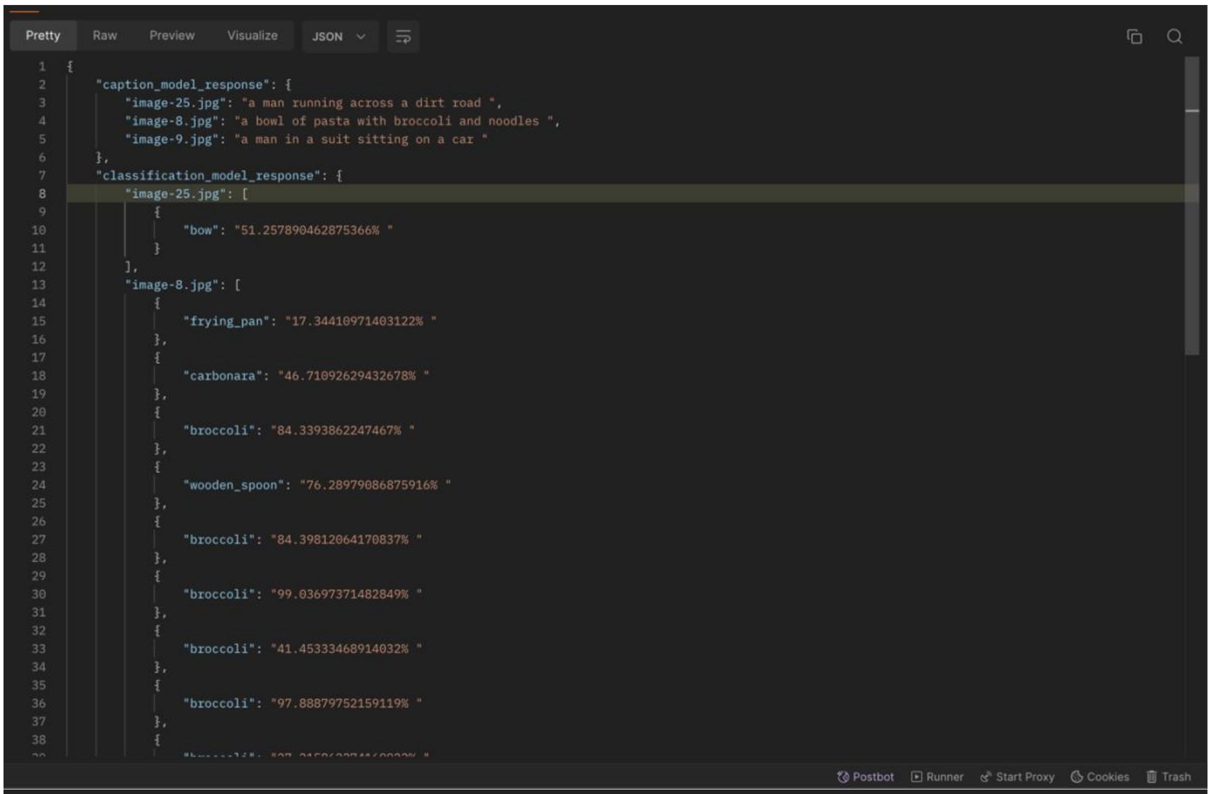


POST http://localhost:8000/upload

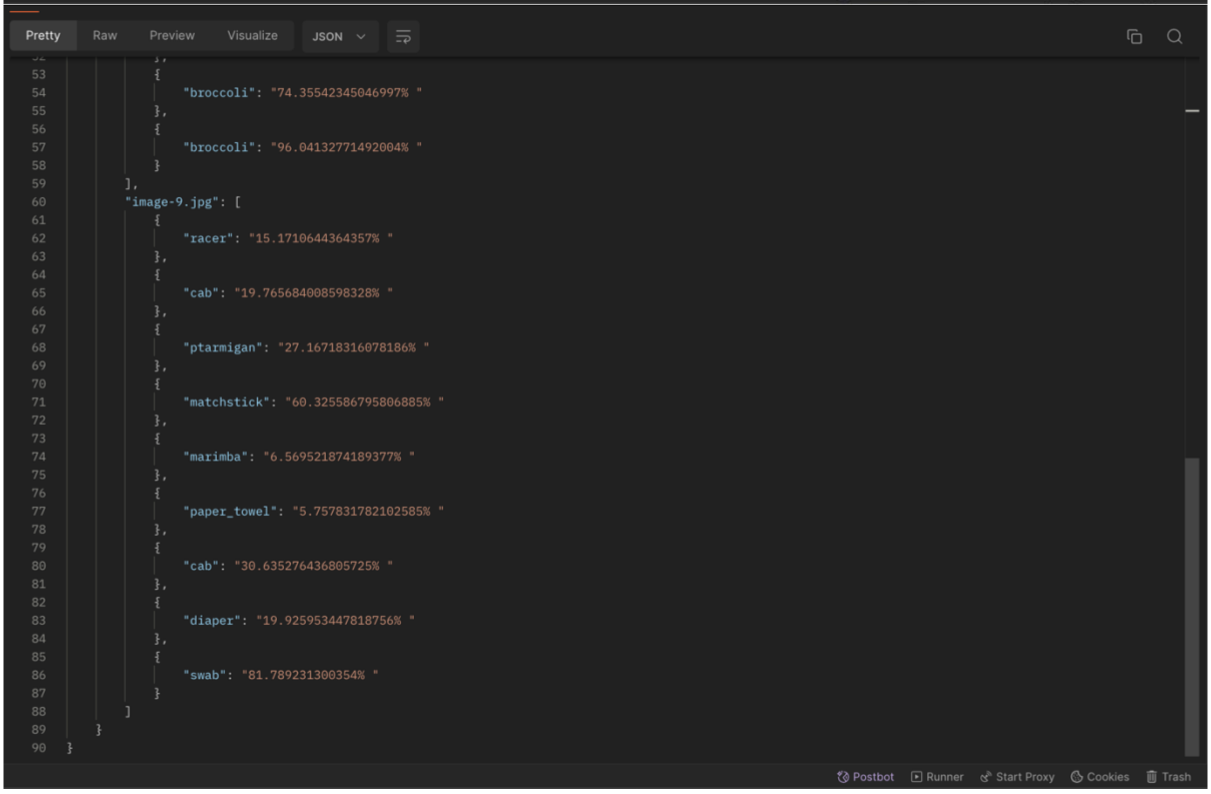
Body: form-data

Key		Value	Description
file	File	image-9.jpg	
file	File	image-8.jpg	
file	File	image-25.jpg	

Output



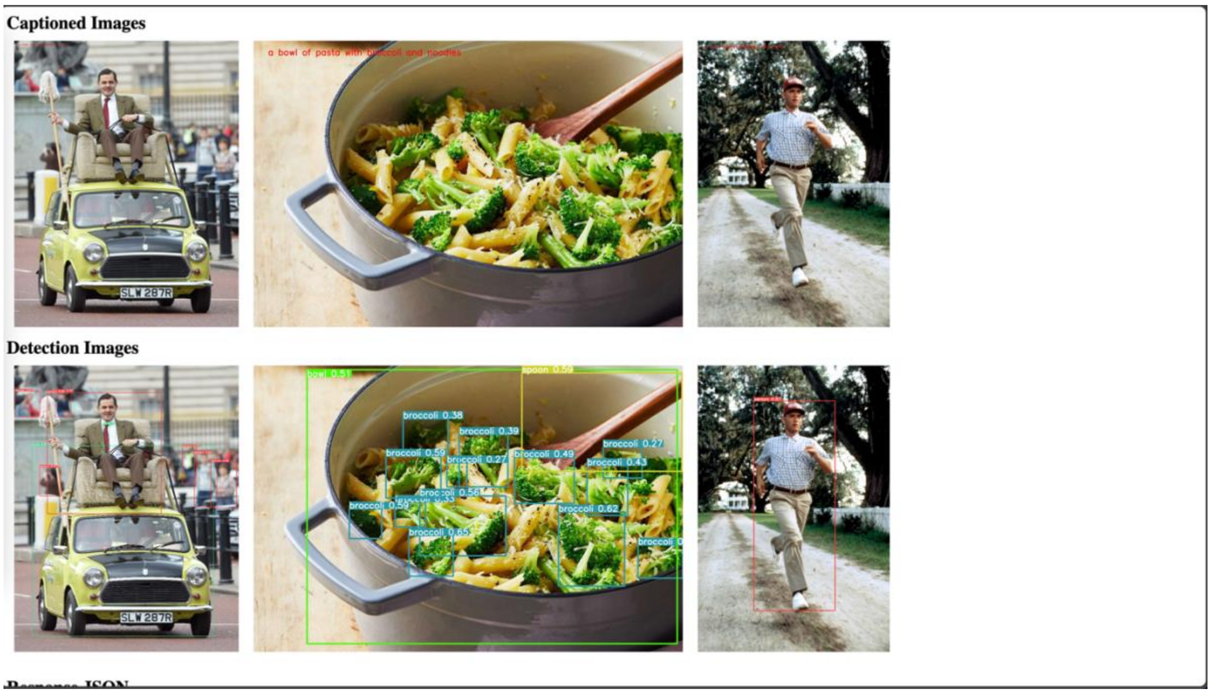
```
{
    "caption_model_response": {
        "image-25.jpg": "a man running across a dirt road ",
        "image-8.jpg": "a bowl of pasta with broccoli and noodles ",
        "image-9.jpg": "a man in a suit sitting on a car "
    },
    "classification_model_response": {
        "image-25.jpg": [
            {
                "bow": "51.257890462875366% "
            }
        ],
        "image-8.jpg": [
            {
                "frying_pan": "17.34410971403122% "
            },
            {
                "carbonara": "46.71092629432678% "
            },
            {
                "broccoli": "84.3393862247467% "
            },
            {
                "wooden_spoon": "76.28979086875916% "
            },
            {
                "broccoli": "84.39812064170837% "
            },
            {
                "broccoli": "99.03697371482849% "
            },
            {
                "broccoli": "41.45333468914032% "
            },
            {
                "broccoli": "97.88879752159119% "
            },
            {
```



```
            {
                "broccoli": "74.35542345046997% "
            },
            {
                "broccoli": "96.04132771492004% "
            }
        ],
        "image-9.jpg": [
            {
                "racer": "15.1710644364357% "
            },
            {
                "cab": "19.765684008598328% "
            },
            {
                "ptarmigan": "27.16718316078186% "
            },
            {
                "matchstick": "60.325586795806885% "
            },
            {
                "marimba": "6.569521874189377% "
            },
            {
                "paper_towel": "5.757831782102585% "
            },
            {
                "cab": "30.635276436805725% "
            },
            {
                "diaper": "19.925953447818756% "
            },
            {
                "swab": "81.789231300354% "
            }
        ]
    }
}
```

Outputs differ from text written in image due to different models used
We also get JSON as response as above, each image has crop from yolo and each crop is appended to list with label and confidence score from inception model

Result from view.html



Setting up the project

// Assuming you are in env

1. Firstly do install all dependencies pip install -r requirements.txt
2. Install all models
    a. yolov5 # https://github.com/ultralytics/yolov5/releases
    b. Vit Image-to-text Transformer model # https://huggingface.co/nlpconnect/vit-gpt2-image-captioning
    c. Inception V3 # https://keras.io/api/applications/

Running the project

1. Open your terminal type gunicorn app:app first app is your file name and 2nd app is your flask application name
2. Now you can got to localhost:8000/ or 127.0.0.1:8000/ Then upload your images and got to localhost:8000/view or 127.0.0.1:8000/view after the API returns successful response
3. You can also upload images using postman