PROJECT FOLDER STRUCTURE

PROJECT: ---------- **Root folder**

- **client_app:**
  - **streamlit_app.py**
  - **streamlit_requirements.txt**
- **server_app:**
  - **app_flask:**
    - **Dockerfile** ---------- **1)**
    - **.....**
  - **app_detector:**
    - **Dockerfile** ---------- **2) YOLOv5**
    - **.....**
  - **app_grouper:**
    - **Dockerfile** ---------- **3) CLIP Image encoder("ViT-B/32") DBSCAN clustering, KNN classifier or Cosine similarity**
    - **.....**

HOSTING - SERVER SIDE SETUP(Localhost)

SYSTEM REQUIREMENTS:

1) Install docker :how to install docker
2) GPU host machine installed with nvidia-driver > 550

- *cd server_app*

RUN FLASK APP:(1)

- *cd app_flask*
- *sudo docker build -t flask_app:v1 .*
- *sudo docker run -d --network="host" -p 127.0.0.1:7000:7000 –name flask_app_c1 flask_app:v1*

INSTALL NVIDIA CONTAINER TOOLKIT FOR GPU AVAILABILITY FOR MODELS:

Set up the package repository and GPG key:

- *distribution=$(. /etc/os-release;echo IDVERSION_ID)*
- *curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -*
- *curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list*

Update the package listing and install the toolkit:

- *sudo apt-get update*
- *sudo apt-get install -y nvidia-container-toolkit*

Configure the Docker daemon to recognize the NVIDIA runtime:

- *sudo nvidia-ctk runtime configure –runtime=docker*

Restart the Docker daemon:

- *sudo systemctl restart docker*

RUN DETECTOR APP:(2)

- *cd app_detector*
- sudo docker build -t detector_app:v1 .
- sudo docker run --gpus all -d --*network="host"* -p 127.0.0.1:7001:7001 --name detector_app_c1 detector_app:v1

RUN GROUPER APP:(3)

- *cd app_grouper*
- sudo docker build -t grouper_app:v1 .
- sudo docker run --gpus all -d --*network="host"* -p 127.0.0.1:7002:7002 --name grouper_app_c1 grouper_app:v1

TESTING – CLIENT SIDE SETUP

SETUP:

- *cd client_app*
- *pip install -r streamlit_requirements.txt*

RUN STREAMLIT APP:

- *streamlit run streamlit_app.py*

TO IMPROVE

Deployments:

1) Models hosting in TorchServe
2) Use dockercompose.yml

Model Performance:

1) Used YOLOv5, can use YOLOv11 for better OD.
2) Used DBSCAN clustering algo, that too clustered object only within a single image. But,

“Thought of using **DBSCAN** for initial clustering and save centroids or features.

- For new incoming objects:
- Extract features.
- Use **KNN**, **cosine similarity**, or a distance-based threshold to classify the object into an existing cluster or mark it as new.
- Dynamically update the database with new objects and clusters as needed.”

3) Use OCR to extract the product names and other possible **text** content(keyword search) on the brand lables. **Both image and text vector** index with meta deta, and also trying out Hybrid search especially on the Azure search may give better similarity search.
