I. Thư viện cần cài đặt

- Trước tiên là cài đặt môi trường để chạy demo (do mediapipe chỉ chạy trên môi trường python 3.9 và 3.10)

(Cú pháp trong anaconda prompt)

1.

conda create --name mp\_env python=3.9

conda activate mp\_env

pip install mediapipe opencv-python numpy

2.

conda env list

3.

conda activate mp\_env

pip install ipykernel

python -m ipykernel install --user --name=mp\_env --display-name "Python (mp\_env)"

- Các thư viện cần cài đặt:

#Ghi trong cell trong file .ipynb (chọn môi trường chạy là mp\_env)

!pip install opencv-python

!pip install numpy

!pip install matplotlib

!pip install tensorflow

!pip install scikit-learn

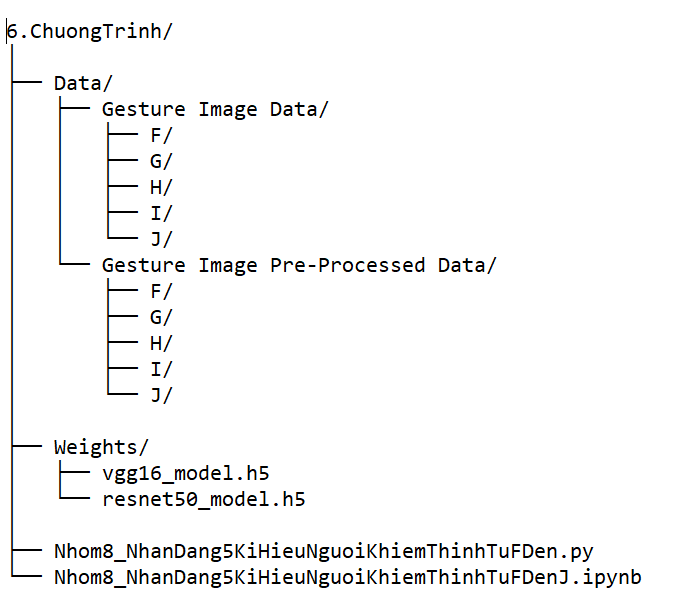
!pip install seaborn

!pip install mediapipe

!pip install --user msvc-runtime

!pip install pyttsx3

II. Cấu trúc thư mục



III. Cách train model

Trong Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.py (hoặc Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.ipynb) , thực hiện các bước sau:

1. Load dữ liệu bằng load\_data(...).
2. Chia train, validation, test bằng train\_test\_split(...).
3. Xây dựng mô hình với VGG16 hoặc ResNet50.
4. Train model:

model.fit(X\_train, y\_train, validation\_data=(X\_valid, y\_valid), epochs=10, batch\_size=32)

1. Lưu mô hình:

model.save("Weights/vgg16\_model.h5")

model.save("Weights/resnet50\_model.h5")

IV. Cách inference model

Trong Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.py (hoặc Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.ipynb):

**1. Load model đã huấn luyện:**

from tensorflow.keras.models import load\_model

model = load\_model("Weights/vgg16\_model.h5")

model = load\_model("Weights/resnet50\_model.h5")

**2. Dự đoán nhãn:**

y\_pred = np.argmax(model.predict(X\_test), axis=1)

**3. Hiển thị kết quả:**

* + Confusion matrix
  + Classification report

V. Hướng dẫn sử dụng giao diện

Trong Trong Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.py (hoặc Nhom8\_NhanDang5KiHieuNguoiKhiemThinhTuFDen.ipynb):

**1. Dự đoán từ webcam:**

test\_video(vgg16\_model, "VGG16")

* Nhấn Q để thoát.
* Kết quả sẽ hiển thị nhãn cử chỉ dự đoán trên màn hình.

test\_video(resnet50\_model, "ResNet50")

* Nhấn Q để thoát.
* Kết quả sẽ hiển thị nhãn cử chỉ dự đoán trên màn hình.

**2. Sử dụng MediaPipe để theo dõi xương tay:**

test\_video\_mediapipe(vgg16\_model, "VGG16")

* Vẽ khung xương tay bằng MediaPipe.
* Hiển thị nhãn dự đoán dựa vào cử chỉ.

test\_video\_mediapipe(resnet50\_model, "ResNet50")

* Vẽ khung xương tay bằng MediaPipe.
* Hiển thị nhãn dự đoán dựa vào cử chỉ.

**3. Tích hợp đọc nhãn bằng giọng nói (Text-to-Speech):**

test\_video\_speech(vgg16\_model, "VGG16")

* Khi phát hiện tay, mô hình sẽ đọc nhãn tương ứng ("F", "G", v.v.).
* Dùng pyttsx3 để phát âm.

test\_video\_speech(resnet50\_model, "ResNet50")

* Khi phát hiện tay, mô hình sẽ đọc nhãn tương ứng ("F", "G", v.v.).
* Dùng pyttsx3 để phát âm.

Lưu ý:

- Ưu tiên nên chạy file .ipynb vì khi chạy mediapipe trên file .py (dù đã ở môi trường python 3.9/3.10 thì chương trình vẫn gặp lỗi về dll (dynamic link library) dù đã install msvc-runtime trước đó)