**Projet reconnaissance\_faciale\_app :**

### 1. app.py (Application principale)

Ce fichier contient l’application Streamlit principale qui :

* Capture une image via webcam
* Permet d’ajouter un visage à la base
* Lance une reconnaissance faciale sur une image capturée

**Fonctionnalités principales :**

st.sidebar.button("...") # Crée des boutons pour interagir  
capture\_image() # Capture via webcam  
mtcnn(img) # Détection du visage avec MTCNN  
model(face.unsqueeze(0)) # Génère l'embedding du visage  
save\_to\_db(nom, embedding) # Sauvegarde dans la base  
log\_recognition(nom) # Journalise une reconnaissance

### 2. models/facenet\_model.py

Ce fichier contient la fonction load\_model() qui :

* Charge le modèle pré-entraîné InceptionResnetV1 de facenet\_pytorch
* Permet de générer un vecteur (embedding) unique pour un visage

from facenet\_pytorch import InceptionResnetV1  
model = InceptionResnetV1(pretrained='vggface2').eval()

### 3. utils/face\_db.py

Contient deux fonctions :

* load\_db() : charge les embeddings à partir d’un fichier .pt
* save\_to\_db(name, embedding) : ajoute un nouvel embedding au fichier

# Chargement :  
db = torch.load("face\_db.pt") if os.path.exists(...) else {}  
  
# Ajout d'un nouvel embedding :  
db[name] = embedding  
torch.save(db, "face\_db.pt")

### 4. utils/journal.py

Permet de journaliser chaque reconnaissance :

import pandas as pd  
with open("data/journal.csv", "a") as f:  
 writer = csv.writer(f)  
 writer.writerow([nom, horodatage])

Cela permet de créer un historique (CSV) des reconnaissances.

### 5. webcam/webcam\_capture.py

Utilise OpenCV pour capturer une image depuis la webcam locale :

* Capture immédiate en appuyant sur une touche
* Retourne l’image capturée

cap = cv2.VideoCapture(0)  
if cv2.waitKey(1) == ord(" "): # Touche Espace pour capturer  
img = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

### 6. requirements.txt

Contient toutes les bibliothèques Python nécessaires :

streamlit  
opencv-python  
facenet-pytorch  
torch  
torchvision  
pandas

Cela permet de recréer l’environnement avec :

pip install -r requirements.txt