

Documentation du projet :

Détection et suivi des mains

avec MediaPipe et OpenCV

sous macOS

Présentation du projet

Ce projet est une application Python de détection et de suivi des mains en temps réel, utilisant les bibliothèques **MediaPipe** et **OpenCV**. Il est destiné à être exécuté sur macOS et peut servir de base à des applications plus complexes comme le contrôle gestuel du volume, des médias ou de l'interface utilisateur.

Aperçu du projet

Ce projet utilise **OpenCV** et **MediaPipe** pour détecter et suivre les mouvements des mains et du visage via une caméra. Il est composé de trois fichiers principaux :

- **HandTrackingModule.py** : Contient la classe handDetector qui gère la détection et le suivi des mains.
- **FaceTrackingModule.py** : Contient la classe faceDetector qui gère la détection et le suivi du visage.
- **main.py** : C'est le fichier principal qui appelle les fonctions des deux modules (HandTrackingModule.py et FaceTrackingModule.py) pour capturer la vidéo, détecter les mains et les visages, et afficher les résultats.

Le projet permet de suivre en temps réel les mains et le visage de l'utilisateur, d'afficher les coordonnées des landmarks des mains et du visage, et d'effectuer des interactions basées sur ces mouvements.

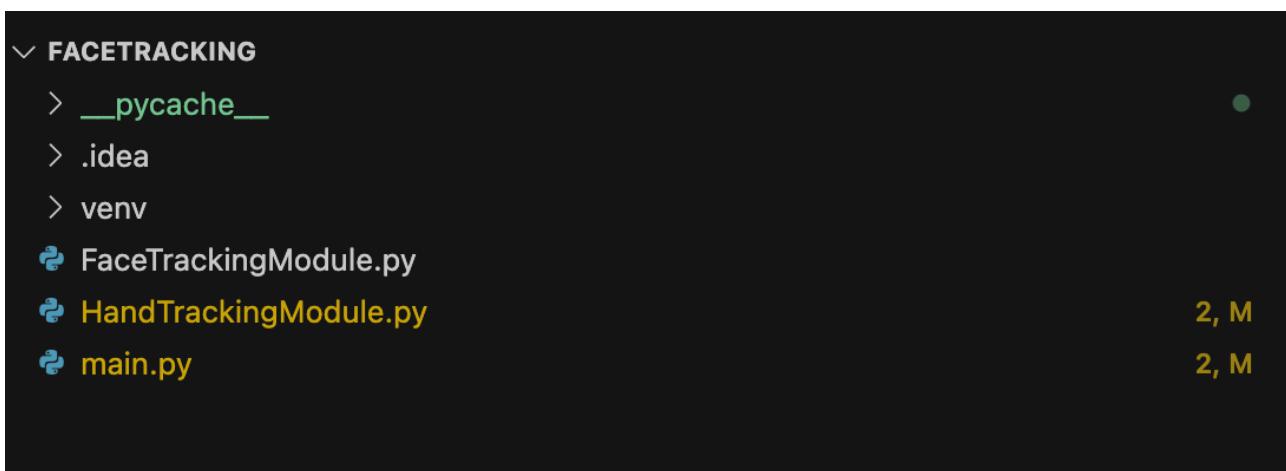
Objectifs

- Déetecter les mains via webcam.
- Suivre les points de repère (landmarks) des mains.
- Afficher en temps réel les coordonnées d'un doigt (ex : le pouce).
- Afficher les FPS pour vérifier la performance.

Technologies utilisées

- **Python 3.10+**
- **OpenCV (cv2)**
- **MediaPipe (solutions.hands)**

Arborescence du projet



Installation et configuration

```
pip install opencv-python mediapipe
```

1. Installation des dépendances

```
python main.py
```

2. Gestion des permissions macOS (caméra)

Accorde l'accès à la caméra dans : **Préférences Système > Sécurité et Confidentialité > Caméra**

3. Exécution

Détail des Fichiers

1. HandTrackingModule.py

Ce fichier contient la classe `handDetector`, qui encapsule les fonctions nécessaires pour détecter et suivre les mains à l'aide de la bibliothèque **MediaPipe**. Il permet d'identifier les mains dans une vidéo en temps réel, d'extraire les coordonnées des repères de la main, et de les afficher ou les manipuler.

Méthodes Principales :

- `init(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5)` :
 - Initialise les paramètres du module de détection des mains.
 - **mode** : Booléen, indique si l'image est statique.
 - **maxHands** : Nombre maximum de mains à détecter.
 - **detectionCon** : Seuil de confiance pour la détection des mains.
 - **trackCon** : Seuil de confiance pour le suivi des mains.
- `findHands(self, img, draw=True)` :
 - **img** : Image d'entrée pour détecter les mains.
 - **draw** : Booléen, si True, dessine les repères sur l'image.
 - **Retour** : Image avec les mains détectées et les repères dessinés.
- `findPosition(self, img, handNo=0, draw=True)` :
 - **img** : Image d'entrée pour récupérer les positions des repères des mains.
 - **handNo** : Index de la main à traiter (par défaut, la première main détectée).
 - **draw** : Booléen, si True, dessine des cercles sur chaque repère détecté.
 - **Retour** : Liste des coordonnées (x, y) des repères des mains dans l'image.

2. FaceTrackingModule.py

Ce fichier gère la détection et le dessin du maillage du visage à l'aide de MediaPipe. Il permet de détecter les visages dans une vidéo en temps réel et de dessiner les connexions entre les différents points du maillage du visage.

Méthodes Principales :

- **FaceMesh** : Utilise la solution Face Mesh de MediaPipe pour détecter les points du visage en temps réel.
- `draw_landmarks()` : Dessine les points de repère et les connexions entre eux sur l'image, permettant de visualiser le maillage du visage.

Exécution du Code :

- Le programme capte l'image de la caméra ou d'une vidéo et détecte les visages.
- Les points du maillage du visage sont dessinés sur l'image avec les connexions entre eux.
- Affichage des FPS dans l'image pour montrer la performance du traitement.

3. main.py

Ce fichier contient le script principal qui intègre les modules précédents pour réaliser un projet complet combinant la détection des mains et la gestion du volume du système à l'aide des gestes des mains.

Composants Principaux :

- **Détection des mains** : Utilise la classe handDetector pour détecter et suivre les mains.
- **Calcul de la distance entre le pouce et l'auriculaire** : Cette distance est utilisée pour ajuster le volume du système macOS.

Fonctions Importantes :

- **set_volume_mac(volume_percent)** :
 - Modifie le volume système de macOS entre 0 et 100% en fonction de l'entrée donnée.
 - Utilise **AppleScript** pour ajuster le volume via la commande osascript.

Déroulement du Programme :

1. **Capture vidéo** : L'image est capturée en temps réel via la webcam.
2. **Détection des mains** : Les coordonnées des repères des mains sont extraites.
3. **Calcul de la distance entre le pouce et l'auriculaire** : Cette distance est utilisée pour déterminer un pourcentage de volume.

4. **Contrôle du volume** : Le pourcentage de volume est appliqué au système macOS en temps réel.
5. **Affichage de la barre de volume** : Une barre de volume et le pourcentage sont affichés sur l'image pour indiquer le niveau actuel.

Fonctionnement :

1. Ouvre la webcam avec `cv2.VideoCapture()`.
2. Lit les frames en boucle.
3. Applique la détection des mains.
4. Affiche les landmarks.
5. Affiche les FPS.
6. Ferme avec 'q'.

Performance:

- Affichage du nombre de FPS en haut à gauche.
- Dépend de la caméra (720p ou 1080p) et de la charge CPU.

Pistes d'amélioration:

- Ajouter la reconnaissance de gestes spécifiques (pouce levé, paume ouverte...).
- Intégrer un contrôle du volume via osascript sur macOS.
- Enregistrer les mouvements en JSON ou CSV pour des analyses.

Captures d'écran:

- Affichage des landmarks de la main
- Console affichant les coordonnées du doigt 4:
- FPS en haut à gauche de la fenêtre

FPS: 6



76 %

