

TP : Détection et floutage des visages en temps réel

Contexte

L'objectif de ce TP est de vous initier à la détection de visages en temps réel avec Mediapipe et OpenCV. Ce sera l'occasion de manipuler une webcam d'ordinateur pour traiter chaque frame de la vidéo, extraire les régions (bounding boxes) correspondant aux visages, et appliquer un filtre de floutage (blurring) sur ces zones. Dans une seconde partie, nous irons plus loin en reproduisant un effet similaire à ceux utilisés dans Zoom ou Teams, qui consiste à flouter tous les visages de la vidéo, à l'exception du visage central (supposé être celui de l'utilisateur principal en visioconférence).

Partie 1 : Floutage des visages détectés dans la vidéo

Objectifs :

- Mettre en place un pipeline de capture vidéo en temps réel à l'aide de la webcam.
- Utiliser la librairie Mediapipe pour détecter les visages dans chaque frame de la vidéo.
- Pour chaque visage détecté, récupérer les coordonnées de sa bounding box.
- Appliquer un filtre de floutage (par exemple, Gaussian Blur via OpenCV) sur la zone de la bounding box.
- Afficher la vidéo en temps réel en remplaçant les régions de visages par leur version floutée.

Consignes :

1. Créer une application Python qui capture la vidéo depuis la webcam.
 2. Initialiser le module de détection de visages de Mediapipe.
 3. Pour chaque frame :
 - Convertir l'image dans le format requis par Mediapipe (souvent RGB).
 - Détecter les visages et récupérer leurs coordonnées (bounding boxes) en pixels.
 - Pour chaque bounding box, extraire la région d'intérêt (ROI) correspondant, appliquer un filtre de blurring avec OpenCV (par exemple `cv2.GaussianBlur`) et remettre cette zone floutée dans l'image.
 4. Afficher la vidéo en temps réel avec la fenêtre dédiée (utiliser `cv2.imshow`).
-

Partie 2 : Floutage sélectif des visages en visioconférence

Objectifs :

- Reprendre le pipeline de la partie 1 pour la détection des visages et la capture vidéo.
- Déterminer le visage central dans l'image (le plus vraisemblablement celui de l'utilisateur en visioconférence).
- Appliquer un floutage à tous les visages détectés sauf celui qui est central.

- Penser à définir une stratégie simple pour déterminer le visage central, par exemple en calculant le centre de la frame et en choisissant le visage dont le centre de la bounding box se rapproche le plus de ce point.

Consignes :

1. Pour chaque frame, une fois l'ensemble des visages détectés, calculer le centre de l'image.
 2. Pour chaque visage, calculer la distance entre le centre du visage (milieu de la bounding box) et le centre de la frame.
 3. Identifier le visage dont la distance est minimale et considérer ce visage comme l'utilisateur principal.
 4. Appliquer le floutage à toutes les autres zones contenant des visages, en excluant le visage central.
 5. Afficher le résultat en temps réel en veillant à ce que seul le visage central reste net.
-

Partie 3 : Filtre nez de clown

Dans cette partie nous allons utiliser le FaceMesh de Mediapipe. Le Face Mesh de MediaPipe est un module de détection et de suivi des points de repère faciaux qui identifie jusqu'à 468 landmarks sur le visage en temps réel, permettant ainsi une analyse détaillée de la structure faciale. Cette technologie est utilisée pour diverses applications, telles que les filtres en réalité augmentée, l'animation faciale et des analyses biométriques, grâce à son traitement rapide et précis des visages dans des vidéos ou des images.

consigne :

- Reprendre le pipeline de la partie 1 pour la détection des visages et la capture vidéo.
- Dessiner un cercle rouge au niveau d'un des landmark du nez. Vous trouverez une carte des indices de points ici:
<https://raw.githubusercontent.com/lshmelzeisen/understanding-mediapipe-facemesh-output/main/output/landmarks-numbered.png>