**Documentation : Algorithme de détection de la main et calcul du Digit Ratio (2D:4D)**

**Introduction**

Le Digit Ratio (2D:4D) est le rapport de la longueur de l'index (2D) à celle de l'annulaire (4D). Cet algorithme utilise la bibliothèque **mediapipe** pour détecter les landmarks de la main et calculer le Digit Ratio.

**Dépendances**

* Python
* OpenCV
* Mediapipe

**Structure du code**

Le code est structuré en deux parties principales : une classe **HandDetector** et une fonction principale **main**.

Classe **HandDetector**

Cette classe est responsable de la détection des landmarks de la main.

**Attributs** :

* **mp\_hands** : Module de détection de la main de **mediapipe**.
* **hands** : Instance de la détection de la main.
* **mp\_draw** : Module de dessin de **mediapipe** pour dessiner les landmarks.

**Méthodes** :

* **detect\_hand(image)** : Prend en entrée une image et renvoie l'image avec les landmarks de la main dessinés et le Digit Ratio affiché.

Fonction **main**

Cette fonction est le point d'entrée du script. Elle charge une image, utilise la classe **HandDetector** pour détecter la main et affiche l'image résultante.

**Utilisation**

1. Assurez-vous d'avoir installé toutes les dépendances nécessaires.
2. Modifiez la variable **image\_path** pour pointer vers votre image.
3. Exécutez le script pour voir les résultats.

**Résultats**

L'image affichée montrera la main avec les landmarks détectés par **mediapipe**, et le Digit Ratio (2D:4D) sera affiché directement sur l'image.

**Documentation détaillée : Algorithme de détection de la main et calcul du Digit Ratio (2D:4D)**

**Introduction**

Le Digit Ratio (2D:4D) est le rapport de la longueur de l'index (2D) à celle de l'annulaire (4D). Cet algorithme utilise la bibliothèque **mediapipe** pour détecter les landmarks de la main et calculer le Digit Ratio.

**Fonctionnement détaillé**

1. **Importation des bibliothèques** :
   * **cv2** : Bibliothèque OpenCV pour le traitement d'images.
   * **mediapipe** : Bibliothèque pour la détection des landmarks de la main.
2. **Classe HandDetector** :
   * **Initialisation (\_\_init\_\_)** :
     + **mp\_hands** : Accède au module de détection de la main de **mediapipe**.
     + **hands** : Crée une instance pour la détection de la main.
     + **mp\_draw** : Accède au module de dessin de **mediapipe** pour visualiser les landmarks.
   * **Méthode detect\_hand** :
     + Convertit l'image en couleur RGB car **mediapipe** nécessite une image RGB.
     + Traite l'image avec **mediapipe** pour obtenir les landmarks.
     + Si des landmarks sont détectés, ils sont dessinés sur l'image avec leurs connexions.
3. **Fonction main** :
   * Crée une instance de **HandDetector**.
   * Lit l'image spécifiée par **image\_path**.
   * Appelle la méthode **detect\_hand** pour obtenir une image avec les landmarks dessinés.
   * Redimensionne l'image pour l'affichage.
   * Affiche l'image traitée.
4. **Calcul du Digit Ratio** :
   * Utilise les landmarks 4 (bas de l'index) et 8 (bout de l'index) pour calculer la longueur de l'index.
   * Utilise les landmarks 12 (bas de l'annulaire) et 16 (bout de l'annulaire) pour calculer la longueur de l'annulaire.
   * Calcule le rapport de ces deux longueurs pour obtenir le Digit Ratio.
   * Affiche le Digit Ratio sur l'image.

**Notes techniques**

* **Landmarks de la main** : **mediapipe** fournit 21 landmarks pour la main, représentant des points clés tels que les articulations et les bouts des doigts.
* **Affichage** : L'algorithme utilise OpenCV pour afficher l'image traitée. Les landmarks et le Digit Ratio sont dessinés directement sur cette image.

**Conclusion**

L'algorithme utilise la puissance de **mediapipe** pour détecter les landmarks de la main et calcule ensuite le Digit Ratio en utilisant ces landmarks. L'image résultante, avec les landmarks et le Digit Ratio affichés, donne une visualisation claire de la détection et du calcul effectués.