

Rapport de projet

Contrôle des gérances de PC utilisant la reconnaissance des mouvements de la main

Membres :

El Moussaoui El Mustapha
Soukaina zaka
Nour el houda zeroual
Ismail Iaabidi
Abdelmajid oumalki
Hicham Redoine
Bouchra Elkheir

Encadrants :

-Madame Zrikem
-Mr hatim
-Mr EL BEID

Soutenu le 10 / 06 / 2023

Plan :

- Problématique
- Cahier de charge
- Analyse des besoins
- Outils & bibliothèques utilisées
- Réalisation du projet
- Les difficultés rencontrées et les solutions proposées.
- Remerciement.

●Problématique

Problématique : Dans notre quotidien, l'utilisation d'ordinateurs est devenue incontournable pour diverses raisons. Cependant, certaines actions courantes comme ajuster le volume d'une vidéo, prendre une capture d'écran ou modifier la taille une image peuvent se révéler fastidieuses et prendre un temps précieux. Il est donc nécessaire de trouver des solutions pratiques et efficaces pour simplifier ces tâches.

Notre proposition consiste à développer des scripts en Python qui permettront de résoudre ces problématiques de manière optimale.

Grâce à ces scripts, les utilisateurs pourront effectuer ces actions en quelques clics,

sans avoir à chercher les options appropriées dans différentes interfaces ou applications. L'objectif est de rendre ces fonctionnalités facilement accessibles, offrant ainsi un gain de temps considérable et une expérience utilisateur améliorée.

- Cahier de charge

● Cahier de charge

Le projet vise à développer un système de contrôle des gestions de PC utilisant la reconnaissance des mouvements de la main à l'aide de la bibliothèque OpenCV et du framework MediaPipe. L'objectif est de permettre aux utilisateurs d'interagir avec leur ordinateur en effectuant des gestes prédéfinis avec leur main devant une caméra

Fonctionnalités principales :

Voici quelques exemples de gestes que nous pouvons implémenter dans notre projet de contrôle des gestions de PC avec la reconnaissance des mouvements de la main

1. Contrôle du volume :

Utiliser un geste de rotation de la main dans le sens des aiguilles d'une montre ou dans le sens inverse pour augmenter ou diminuer le volume du son de l'ordinateur.

2. Zoom in/zoom out :

Éloigner ou rapprocher la main de la caméra pour simuler le zoom avant ou arrière sur une page ou une image affichée à l'écran

3. Capture d'écran :

Utiliser un geste de la main pour déclencher la capture d'écran, ce qui peut être utile pour prendre rapidement une capture d'écran pendant une présentation ou une conférence

Contraintes techniques :

Le projet doit être développé en utilisant Python 3.8.

Le projet doit être développé en utilisant des bibliothèques open source telles que OpenCV, PyAudio, et autres.

Le projet doit être conçu pour fonctionner sur des systèmes d'exploitation Windows, MacOS et Linux

- Analyse des besoins

- ANALYSE DE BESION :

Notre projet consiste à développer une application capable de détecter les doigts de la main en utilisant la puissante bibliothèque MediaPipe. L'objectif est d'afficher le nombre de doigts détectés à l'écran et de permettre à l'utilisateur d'accéder à différentes options en fonction de ce nombre. Nous avons choisi MediaPipe en raison de sa facilité d'apprentissage et de sa capacité à réaliser des suivis en temps réel avec une grande précision.

Pour commencer, nous utiliserons la fonctionnalité de suivi de main de MediaPipe pour détecter et suivre les mouvements de la main dans une séquence vidéo ou en temps réel via la caméra. Grâce aux points de repère fournis par MediaPipe, nous pourrons déterminer la position et l'état de chaque doigt, qu'il soit plié ou tendu.

En comptant le nombre de doigts tendus, nous obtiendrons le chiffre correspondant.

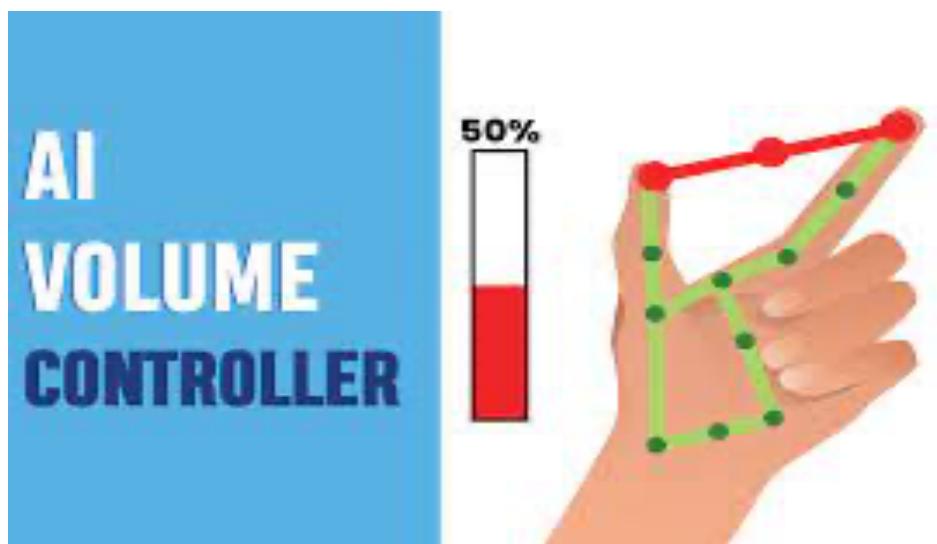
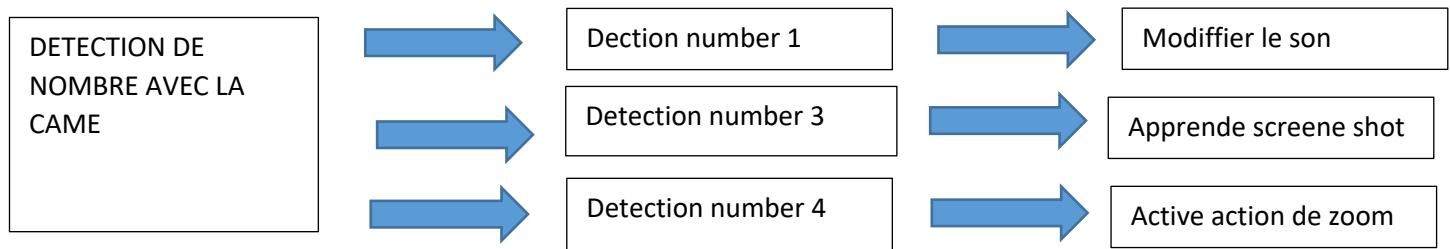
Une fois le nombre détecté, nous l'afficherons clairement à l'écran à l'aide d'une interface utilisateur conviviale. Nous utiliserons des graphiques ou des éléments visuels appropriés pour permettre à l'utilisateur de visualiser facilement le nombre détecté.

Enfin, pour accéder aux différentes options, nous mettrons en place une logique de contrôle basée sur le nombre détecté. À l'aide d'instructions conditionnelles, nous vérifierons quelles options doivent être activées en fonction du chiffre détecté. Par exemple, si le chiffre est 1, nous activerons l'option sonore. Si le chiffre est 3, nous donnerons accès à l'option de capture d'écran.

Si le chiffre est 4, nous effectuerons un zoom sur une image préalablement importée dans notre programme.

Dans le cadre de ce projet, nous utiliserons également d'autres bibliothèques spécialisées pour certaines fonctionnalités spécifiques. Par exemple, nous intégrerons la bibliothèque PyAudio pour la gestion du son, une bibliothèque dédiée à la capture d'écran (comme pyautogui) et OpenCV pour appliquer des transformations de zoom sur les images.

Grâce à cette approche, nous créerons une application interactive et intuitive qui permettra à l'utilisateur d'interagir avec les options en utilisant simplement les mouvements de sa main.



• Outils &
bibliothèques utilisées

:

Outils & bibliothèques utilisées :

Python est un langage de programmation couramment utilisé sur différents systèmes d'exploitation tels que Linux, Mac OS X et Windows. Il est facile à installer sur ces plateformes. Cependant, dans ce livre, nous abordons l'utilisation de modules spécifiques qui sont très utiles dans le domaine de la science des données, et le traitement d'image tels que ,NumPy, Mediapipe, Opencv, Math, Ctypes, Comtypes, Pycaw et Cvzone Nous faisons également référence à l'utilisation des notebooks Pycharm.

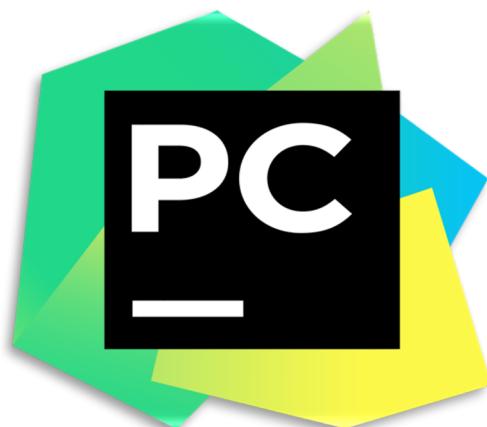
Pour faciliter l'installation de ces modules supplémentaires, nous recommandons l'utilisation d'un gestionnaire de paquets. Cela permettra de simplifier le processus d'installation et de gestion des dépendances. Il est important de noter que ce gestionnaire de paquets est également disponible pour Windows, afin de garantir une expérience cohérente pour tous les utilisateurs, quel que soit leur système d'exploitation.

Il y a deux grandes alternatives :

-PyCharm : PyCharm est un environnement de développement intégré (IDE) spécialement conçu pour le langage de programmation Python. Il offre un éditeur de code avancé, des fonctionnalités de débogage, une gestion des versions, une intégration avec des outils de gestion de packages, ainsi qu'un support étendu pour le développement web avec Python.

C'est un outil puissant et convivial pour les développeurs Python. PyCharm propose un éditeur de code avancé avec des fonctionnalités telles que la coloration syntaxique, l'autocomplétion intelligente, la mise en forme automatique, la navigation dans le code, la recherche et le remplacement, ainsi que des suggestions contextuelles. Il permet également de déboguer efficacement des programmes Python, d'inspecter les variables et d'exécuter le code pas à pas.

En plus de l'éditeur de code. Il prend également en charge la gestion des packages Python via des gestionnaires de paquets tels que pip et conda. On plus de ça il est plus simple pour le téléchargement des bibliothèques.



-Pip : PIP est un gestionnaire de packages pour Python. Il permet d'installer, mettre à jour et gérer les bibliothèques et les modules tiers développés par la communauté Python. PIP simplifie le processus d'installation des packages en automatisant la recherche, le téléchargement et l'installation des dépendances requises. C'est un outil essentiel pour accéder à un vaste écosystème de packages Python et faciliter le développement de projets en utilisant des fonctionnalités supplémentaires disponibles dans ces packages.

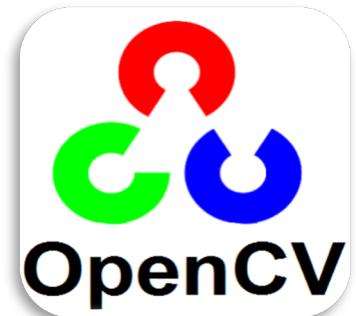
On va parler maintenant des modules utilisés :

NumPy : NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.



Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes. NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique.

Opencv : OpenCV est une bibliothèque de vision par ordinateur open source utilisée pour le traitement d'images et de vidéos, la détection d'objets et la reconnaissance faciale. Elle offre de nombreuses fonctionnalités et est largement utilisée dans des domaines variés tels que la robotique, la vision industrielle et la réalité augmentée.



Mediapipe : MediaPipe est une bibliothèque open source développée par Google pour le traitement en temps réel des flux multimédias. Elle fournit des outils et des modèles pré-entraînés pour des tâches telles que la vision par ordinateur, la détection d'objets et la reconnaissance faciale.



MediaPipe permet aux développeurs de créer des pipelines de traitement personnalisés et est utilisé pour le développement d'applications de réalité augmentée, de suivi de pose et d'autres applications de vision par ordinateur en temps réel.

PyCaw : PyCAW est une bibliothèque open source en Python qui permet de contrôler et manipuler les périphériques audios sur les systèmes Windows en utilisant l'API Core Audio. Elle offre des fonctionnalités pour ajuster le volume, sélectionner les périphériques audios, obtenir des informations sur les périphériques et effectuer d'autres opérations liées à l'audio.

Ctypes : ctypes est une bibliothèque Python qui permet d'interagir avec du code natif écrit en langages tels que C, C++ et autres. Elle fournit des fonctionnalités pour charger des bibliothèques partagées, appeler des fonctions et manipuler des données provenant de ces bibliothèques. C'est un outil essentiel pour intégrer du code natif dans des projets Python et accéder à des fonctionnalités avancées présentes dans ces bibliothèques. ctypes est inclus dans la bibliothèque standard de Python, ce qui en facilite l'utilisation et la rend compatible avec différentes plateformes.

CVZone : CVZone est une bibliothèque open source en Python dédiée au traitement d'images et de vidéos dans le domaine de la vision par ordinateur. Elle offre des modules prêts à l'emploi pour des tâches courantes telles que la détection faciale, la détection d'objets et bien d'autres. CVZone simplifie le développement d'applications de vision par ordinateur en fournissant des solutions efficaces et précises, tout en étant adaptée aux débutants et aux développeurs expérimentés.



- Réalisation du
projet:

la réalisation de notre projet se décompose en trois parties intéressantes:

1^{er} étape : préparation de l'environnement de travaille

Installer les bibliothèques OpenCV et MediaPipe sur l'ordinateur, puis on va initialiser la caméra pour capturer la vidéo en utilisant OpenCV et on va préparer le flux vidéo pour le traitement ultérieur.

2éme étape : Détection de la main et suivi des mouvements

- Utilisez les fonctionnalités de détection d'objets d'OpenCV pour détecter la main dans chaque image du flux vidéo. Appliquez des techniques de suivi des mouvements à la détection de la main en utilisant MediaPipe.

- Utilisation de MediaPipe pour détecter les points clés de la main :

MediaPipe fournit un modèle pré-entraîné pour la détection des points clés de la main. Vous pouvez utiliser **mp.solutions.hands.Hands()** pour initialiser le détecteur de main.

- Obtention des coordonnées des points clés de la main :

Les résultats de détection des mains rentrés par MediaPipe contiennent les coordonnées normalisées des points clés de la main (par exemple, les coordonnées x, y et z).

Après avoir les résultats de détection des points de la main on choisit deux points de référence Par exemple, vous pourriez choisir les points de pouce et d'index .pour mesurer l'angle et changer le volume

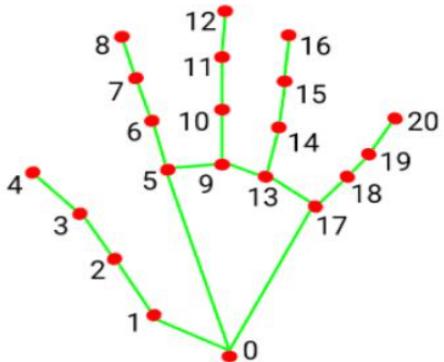
Pour la capture on va prendre une Screenshot si les deux points de pouce et l'index sont plus proches (distance entre eux < 30)

Pour le zoom deux mains sont détectées (**len(hands) == 2**), le code vérifie si les doigts index et majeur des deux mains sont levés. Si c'est le cas, cela signifie que l'utilisateur effectue le geste de zoom.

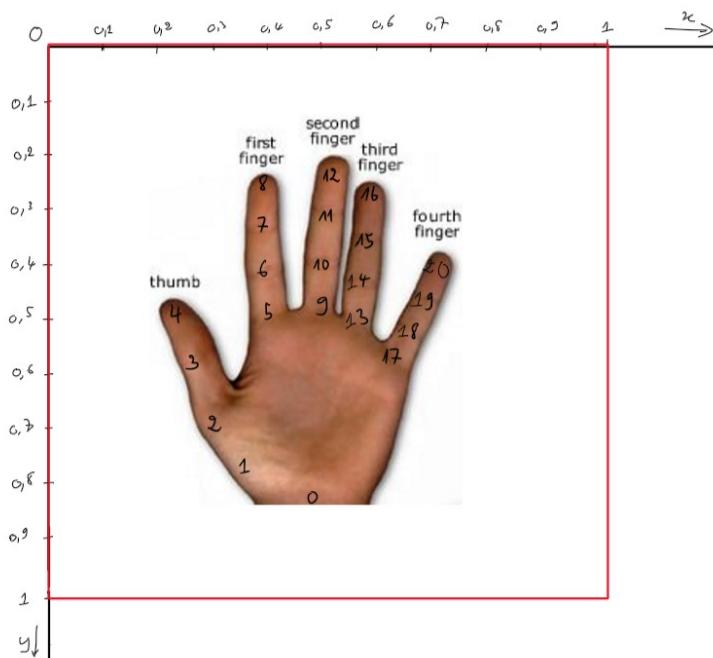
La distance actuelle entre les mains est comparée à la distance initiale pour déterminer le facteur d'échelle (**scale**). Les variables **cx** et **cy** stockent les coordonnées du centre de la région zoomée.

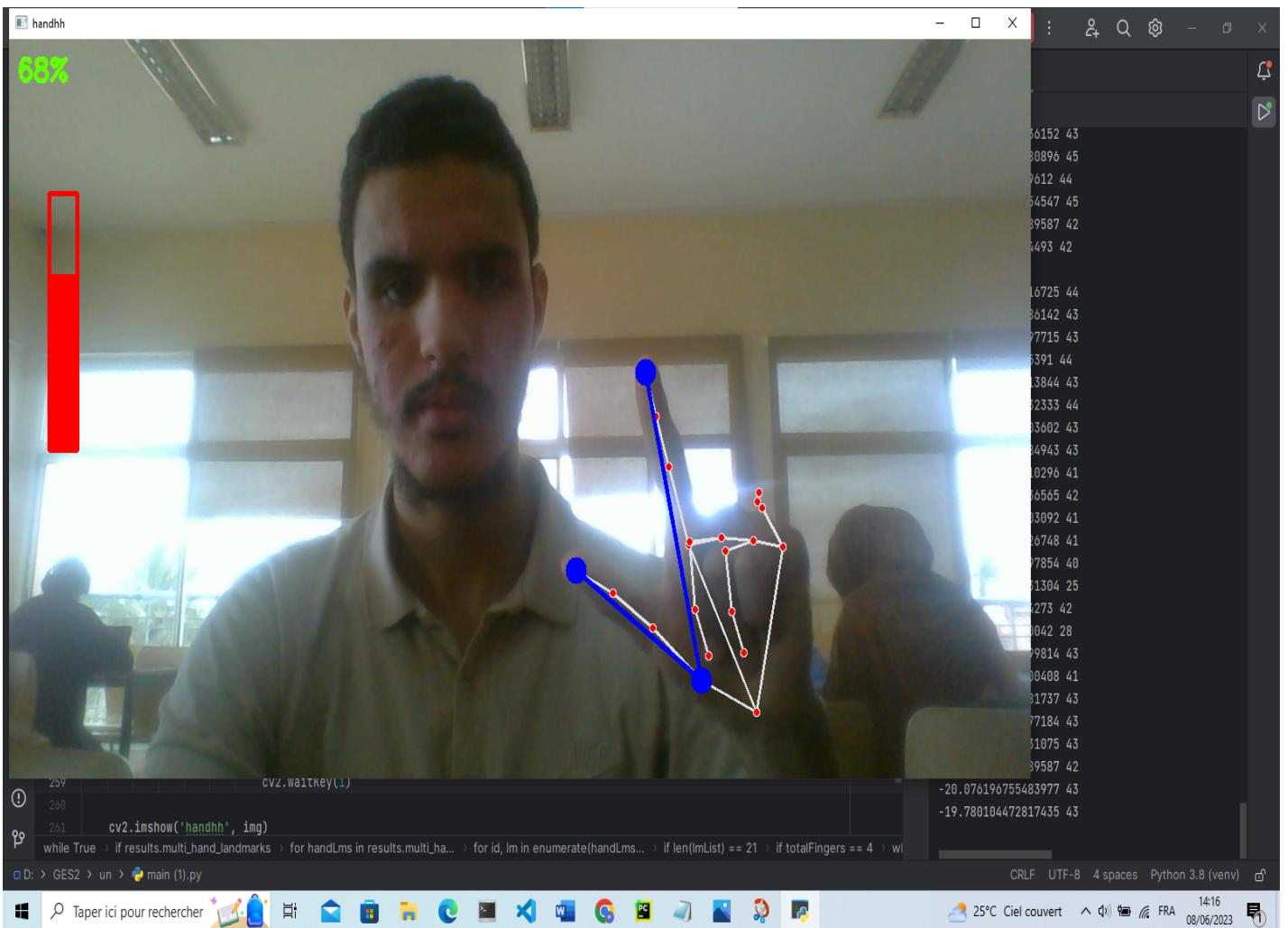
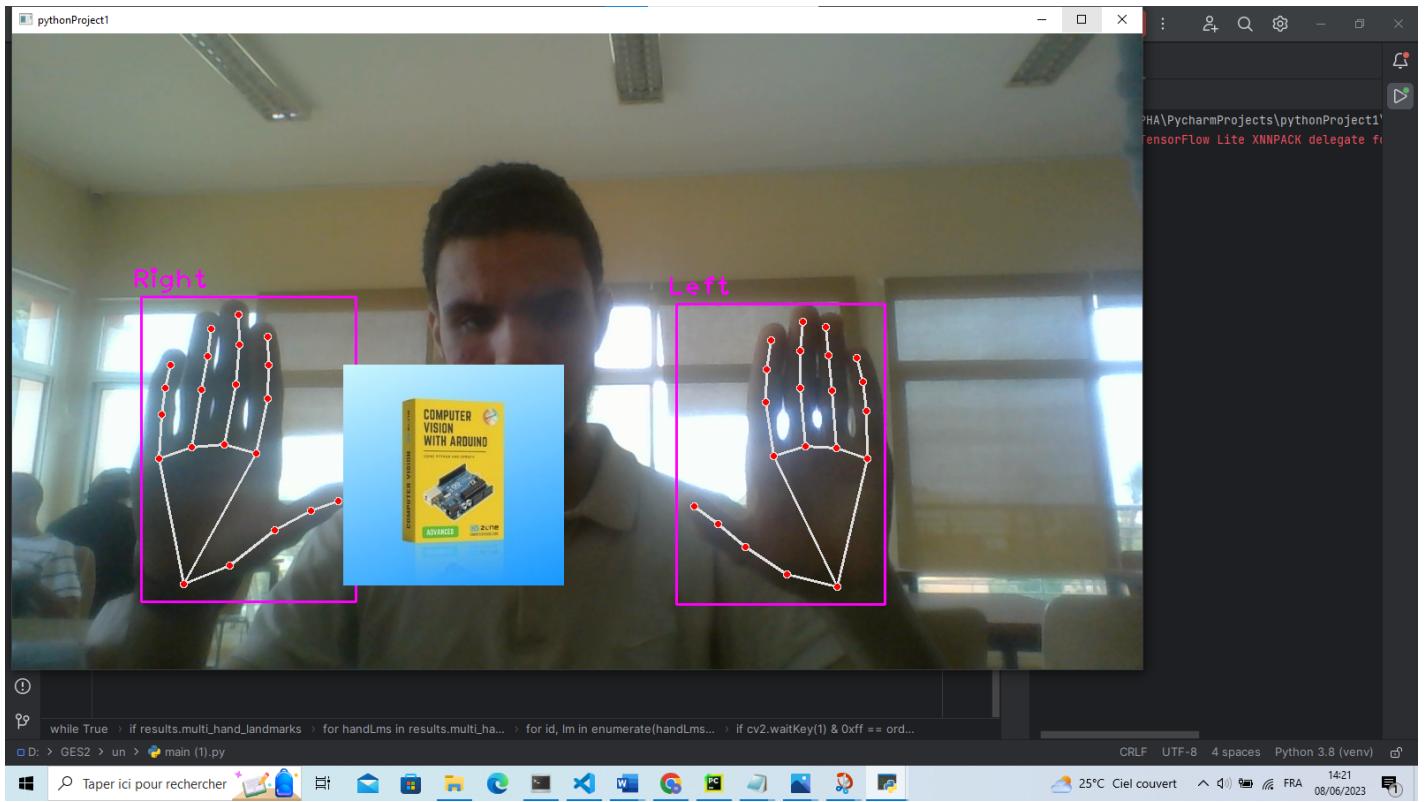
Ensuite placé sur l'image **img** aux coordonnées (**cy - newH//2, cx - newW//2**) pour créer l'effet de zoom

Hand Land Marks



- 0. WRIST
- 1. THUMB_CMC
- 2. THUMB_MCP
- 3. THUMB_IP
- 4. THUMB_TIP
- 5. INDEX_FINGER_MCP
- 6. INDEX_FINGER_PIP
- 7. INDEX_FINGER_DIP
- 8. INDEX_FINGER_TIP
- 9. MIDDLE_FINGER_MCP
- 10. MIDDLE_FINGER_PIP
- 11. MIDDLE_FINGER_DIP
- 12. MIDDLE_FINGER_TIP
- 13. RING_FINGER_MCP
- 14. RING_FINGER_PIP
- 15. RING_FINGER_DIP
- 16. RING_FINGER_TIP
- 17. PINKY_MCP
- 18. PINKY_PIP
- 19. PINKY_DIP
- 20. PINKY_TIP





3ème étape : détecter les doigts de la main en utilisant la puissante bibliothèque MediaPipe.

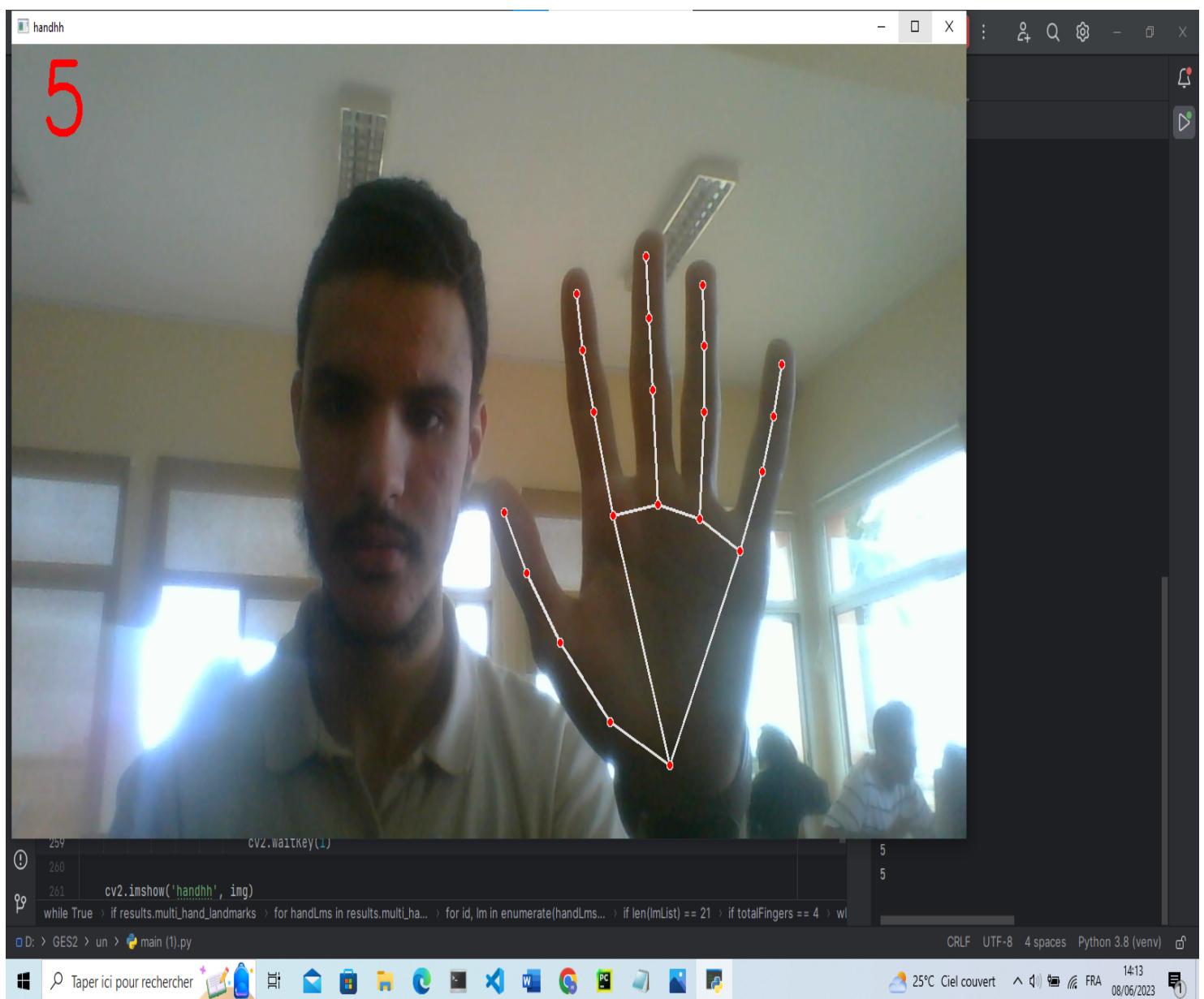
L'objectif est d'afficher le nombre de doigts détectés à l'écran et de permettre à l'utilisateur d'accéder à différentes options, une fois que vous avez obtenu les points clés de la main à l'aide de MediaPipe, vous pouvez extraire les coordonnées des points clés pertinents pour détecter l'état des doigts. Les points clés qui sont généralement utilisés pour cela sont les points de l'extrémité des doigts (les points 8 à 12 dans la numérotation des points clés de MediaPipe).

Pour chaque doigt, vous pouvez comparer les coordonnées du point clé de l'extrémité du doigt avec les coordonnées des points clés des phalanges correspondantes. Par exemple, pour détecter si le pouce est levé, vous pouvez vérifier si la coordonnée y du point clé de l'extrémité du pouce est supérieure à la coordonnée y du point clé de la phalange proximale du pouce.

En utilisant cette logique, vous pouvez détecter l'état de chaque doigt (levé ou baissé). Vous pouvez stocker ces informations dans une liste ou un tableau pour représenter l'état des cinq doigts de la main.

Une fois que vous avez détecté l'état des doigts, vous pouvez utiliser cette information pour afficher le nombre de doigts levés à l'écran. Par exemple, vous pouvez utiliser la fonction **cv2.putText** de OpenCV pour afficher le nombre de doigts détectés dans le coin de l'image.

En fonction de l'état des doigts, vous pouvez permettre à l'utilisateur d'accéder à différentes options ou actions. Par exemple, si tous les doigts sont baissés, cela peut correspondre à une option de "sélection". Si le pouce et l'index sont levés, cela peut correspondre à une option de "zoom". Vous pouvez définir les règles et les gestes associés à chaque option en fonction de l'état des doigts détectés.



● Les difficultés rencontrées et les solutions proposées.

Les difficultés rencontrées et les solutions proposées :

Durant la réalisation de notre projet on a rencontré plusieurs problèmes et difficultés, ce qui a arrêté le déroulement du travail dans certaines périodes. Parmi ces difficultés nous citons :

- L'installation et la mise à jour continue de certaines bibliothèques, ce qui nécessite à chaque fois la recherche des dernières versions de celle-ci puis modifier notre code.
- Dans la partie du contrôle du volume, nous avions initialement utilisé la distance entre les deux doigts comme variable pour cette fonction. Cependant, lors de la mise en œuvre, nous avons constaté que la distance variait en fonction de la position de la main par rapport à la caméra de l'ordinateur, ce qui compromettait la stabilité du contrôle du volume. Afin de résoudre ce problème, nous avons opté pour une approche alternative : l'utilisation de l'angle formé entre le pouce et l'index.

En utilisant l'angle plutôt que la distance, nous avons pu maintenir une stabilité du volume indépendamment de la position de la main par rapport à la caméra.

L'angle entre le pouce et l'index fournit une mesure plus fiable et constante, ce qui permet de contrôler le volume de manière cohérente, quelles que soient les variations de distance entre la main et la caméra. Cette solution a permis d'obtenir un contrôle précis et stable du volume, améliorant ainsi l'expérience utilisateur.

- Au début du projet, nous travaillons sur chaque tâche de manière indépendante. Nous réglons les codes de chaque tâche individuellement. Cependant, nous rencontrons des problèmes lorsque nous essayons de combiner les codes entre eux.

Le problème que nous avons rencontré était exactement comment accéder au code de chaque tâche et comment passer d'une tâche à l'autre de manière fluide. Comme solution à ce problème, nous avons implémenté un module de code indépendant qui détecte précisément le nombre de doigts levés.

Lorsqu'un seul doigt est détecté, il déclenche la fonctionnalité de contrôle du volume correspondante. De même, lorsqu'il détecte trois doigts, il lance la fonctionnalité de zoom. Enfin, lorsqu'il détecte quatre doigts levés, il déclenche la fonctionnalité de capture d'écran. Cette approche garantit une interaction fluide et une exécution précise des tâches souhaitées en fonction du nombre de doigts levés.

Pour sortir de chaque tâche, nous avions initialement envisagé d'utiliser des délais, mais nous avons rencontré un problème où l'utilisation de la fonction de délai bloquait le programme et empêchait le fonctionnement de la caméra. Pour résoudre ce problème, nous avons adopté une approche différente.

Nous avons introduit une variable qui s'incrémenter dans la boucle du chaque sous-programme. Lorsque cette variable atteint une valeur spécifique, le programme sort de cette boucle et passe au programme initial qui lire le nombre de doigts levés. Cette approche nous permet de contrôler la durée d'exécution de chaque tâche sans bloquer le programme ni affecter le fonctionnement de la caméra.

En utilisant cette variable comme condition de sortie de la boucle, nous pouvons garantir que chaque tâche est exécutée pendant la durée souhaitée, puis passer à la tâche suivante sans provoquer de blocage indésirable. Cela nous permet d'assurer un fonctionnement fluide du programme

- Remerciement.

Remerciement :

Nous tenons à remercier sincèrement nos professeurs encadrants, madame **ZRIKEM**, monsieur **EL BEID**, et monsieur **HATIM** pour leurs efforts, et leur disponibilité, ainsi que leurs conseils, durant toutes les périodes de travail. Grace à eux on a pu résoudre plusieurs problèmes afin de bien finaliser notre projet.