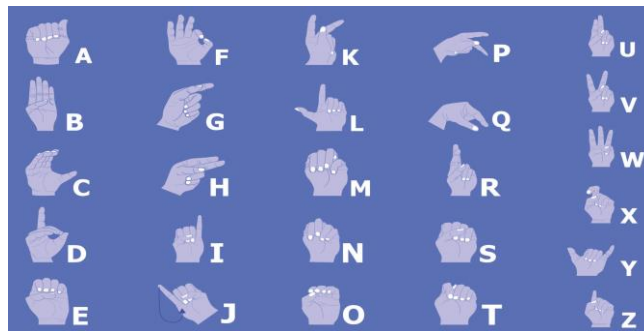


Rapport de mini-projet

Détection de la langue des signes en temps réel



Réalisé par :

- YOUBI Ahlam.
- TADLAOUI Aya.
- KHOLTE Salma.
- MIKH Ayman.

Encadré par :

Pr. Jamal KHARROUBI.

Sommaire

I. Partie 1	3
A. Introduction :	3
B. Objectif :	3
C. Démarche :	4
II. Partie 2	6
A. Structure de l'application :	6
B. Résultat :	7
C. Discussion :	8
III. Conclusion	8

I. Partie 1

A. Introduction :

La langue des signes (**LS**) est une méthode importante de communication pour les gens sourds et muets. Elle est bien structurée avec des gestes codes où chaque geste a une signification qui lui est attribuée. Il existe dans le monde différentes langues des signes chacune avec son propre vocabulaire et gestes. Quelques exemples sont la langue des signes française (LSF), la langue des signes américaine (LSA), la langue des signes britannique (LSB), etc.

Cette langue est couramment utilisée dans les communautés de sourds-muets, y compris les interprètes, les amis et les familles. Cependant, ces langues ne sont pas généralement connues en dehors de ces communautés, et donc les barrières de communication existent entre les sourds et les entendants. Les gens sourds-muets comptent sur des interprètes en langue des signes pour la communication. Cependant, trouver quotidiennement des interprètes qualifiés et expérimentés est une tâche très difficile et aussi inabordable. D'où l'intérêt et le besoin croissant des outils d'analyse et d'annotation automatiques de vidéo de langue des signes. Un tel système permettrait de réduire les obstacles à la communication entre les sourds-muets et les autres personnes.

Dans les dernières années, il y a eu un intérêt accru parmi les chercheurs pour reconnaître l'alphabet et interpréter les mots de la langue des signes et les phrases automatiquement. Récemment, plusieurs approches de reconnaissance de signes ont été proposées; toutefois, un système de reconnaissance automatique de la langue des signes efficace reste un problème ouvert. La communication en langue des signes est multimodale, il s'agit non seulement de gestes de la main, mais également des signes non manuels. C'est une langue où on transmet l'information visuellement par des gestes de mains, par la position de la tête et des parties du corps ou par les expressions faciales. Néanmoins, dans ce projet seulement le geste manuel est concerné et en particulier le geste statique de la main.

B. Objectif :

Notre intérêt se concentrera à la reconnaissance du geste manuel en utilisant le contour de la main seulement.

Dans ce projet, nous allons nous concentrer uniquement sur les alphabets. Nous allons utiliser le modèle de machine learning RandomForest qui sera capable de reconnaître les gestes des mains associés aux lettres de l'alphabet en temps réel. Le modèle sera entraîné à l'aide d'un ensemble de données d'exemples de gestes de la langue des signes associés à chaque lettre. Une fois le modèle entraîné, nous allons l'utiliser pour détecter les gestes de la langue des signes en temps réel. Pour cela nous allons utiliser une caméra pour capturer les gestes de la main de l'utilisateur et les transmettre au modèle pour qu'il les analyse. Le modèle sera capable de reconnaître la lettre associée au geste de la main et de l'afficher à l'utilisateur.

C. Démarche :

➤ Processus :

- ✚ **Collection d'images de mains** : Nous avons collecté un ensemble d'images de mains pour les différentes lettres de l'alphabet de la langue des signes.
- ✚ **Collection des points d'intérêt de la main** : Nous avons utilisé la bibliothèque « **Mediapipe.hands** » pour extraire les points d'intérêt de la main à partir des images de mains. Cela nous permet d'obtenir une représentation numérique des gestes de la main.
- ✚ **Entraînement** : Nous avons entraîné notre jeu de données par le modèle de machine learning « **RandomForest** », en utilisant les points d'intérêt de la main pour chaque lettre de l'alphabet. Le modèle est entraîné à reconnaître les gestes de la main associés à chaque lettre.
- ✚ **Détection des gestes** : Nous avons utilisé le modèle entraîné pour détecter les gestes de la main en temps réel. Pour cela, nous avons utilisé la webcam pour capturer les gestes de la main de l'utilisateur et les transmettre au modèle pour qu'il les analyse.

➤ Algorithmes utilisés :

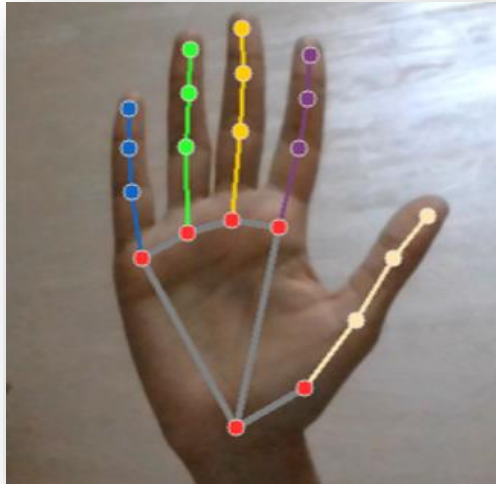
Algorithme 1 : Collection d'images de mains

Cet algorithme parcourt les dossiers contenant les images de chaque classe de la base de données ASL. Il crée un nouveau dossier nommé "data" pour stocker les images redimensionnées. Ensuite, il lit chaque image dans chaque dossier de classe, redimensionne l'image en une dimension de 224x224 pixels en utilisant la fonction `cv2.resize()`, puis enregistre l'image redimensionnée dans un sous-dossier approprié de "data". L'algorithme s'arrête d'enregistrer les images après avoir atteint la taille du jeu de données spécifiée.

Algorithme 2 : Collection des points d'intérêt de la main par **Mediapipe.hands**

MediaPipe est une bibliothèque qui propose des outils pour la reconnaissance faciale, la segmentation d'image, la détection de mouvement, le suivi d'objets et d'autres fonctionnalités de traitement multimédia en temps réel. En résumé, MediaPipe est une bibliothèque de développement logiciel qui fournit des outils pour le traitement multimédia en temps réel, y compris la détection de points clés de la main avec le module **Hands**.

Mediapipe.hands: modèle suit la main détectée par le modèle de détection de la main et fournit des informations supplémentaires telles que les points de repère de la main (bout des doigts, base des doigts, centre de la paume), la direction de la main et l'angle des doigts.



L'algorithme utilise la bibliothèque **OpenCV** pour lire chaque image à partir du jeu de données téléchargé dans le dossier "**data**".

Chaque image est ensuite passée à travers l'algorithme de détection des points clés de la main de la bibliothèque MediaPipe. Lorsqu'une main est détectée, les coordonnées des points clés de la main sont collectées dans une liste appelée "**data_array**". Cette liste est ensuite ajoutée à une autre liste appelée "**data**", qui stocke toutes les données de tous les points clés de chaque image. Les étiquettes correspondantes (classe de l'image) sont stockées dans une autre liste appelée "**labels**". En fin de compte, ces données et étiquettes sont enregistrées dans un fichier binaire appelé "**data.pickle**" pour une utilisation ultérieure dans la formation et l'évaluation de modèles d'apprentissage automatique.

Algorithme 3 : Entraînement par le modèle **RandomForest**

RandomForest est un algorithme de machine learning populaire pour la classification et la régression. Il est souvent utilisé pour des tâches de classification dans lesquelles les données contiennent de nombreuses variables et des relations complexes entre ces variables.

Dans le cas de la détection de la langue des signes, RandomForest est un choix approprié car le modèle doit être capable de reconnaître les gestes de la main associés à chaque lettre de l'alphabet, ce qui peut être considéré comme une tâche de classification complexe.

Algorithme 4 : Interface pour la Détection des gestes de la main avec **tkinter**

Tkinter est une bibliothèque Python pour la création d'interfaces graphiques utilisateur (**GUI**). Elle fournit un ensemble d'outils pour la création de fenêtres, de boutons, de champs de texte, de listes, de menus et de nombreuses autres interfaces utilisateur interactives.

L'interface graphique de notre projet est créée à l'aide de cette bibliothèque. elle comprend un bouton de capture d'image, une zone d'affichage d'image et une zone d'affichage de résultat.

Lorsque l'utilisateur appuie sur le bouton de capture d'image, la webcam est ouverte et l'image est capturée. La bibliothèque MediaPipe est utilisée pour détecter les mains dans l'image capturée. Les points clés de la main sont extraits de l'image capturée et utilisés pour prédire la lettre de l'alphabet associée au geste de la main à l'aide d'un modèle de machine learning de type RandomForest entraîné précédemment. La lettre prédite est ensuite affichée dans la zone d'affichage de résultat et l'image capturée est affichée dans la zone d'affichage d'image.

II. Partie 2

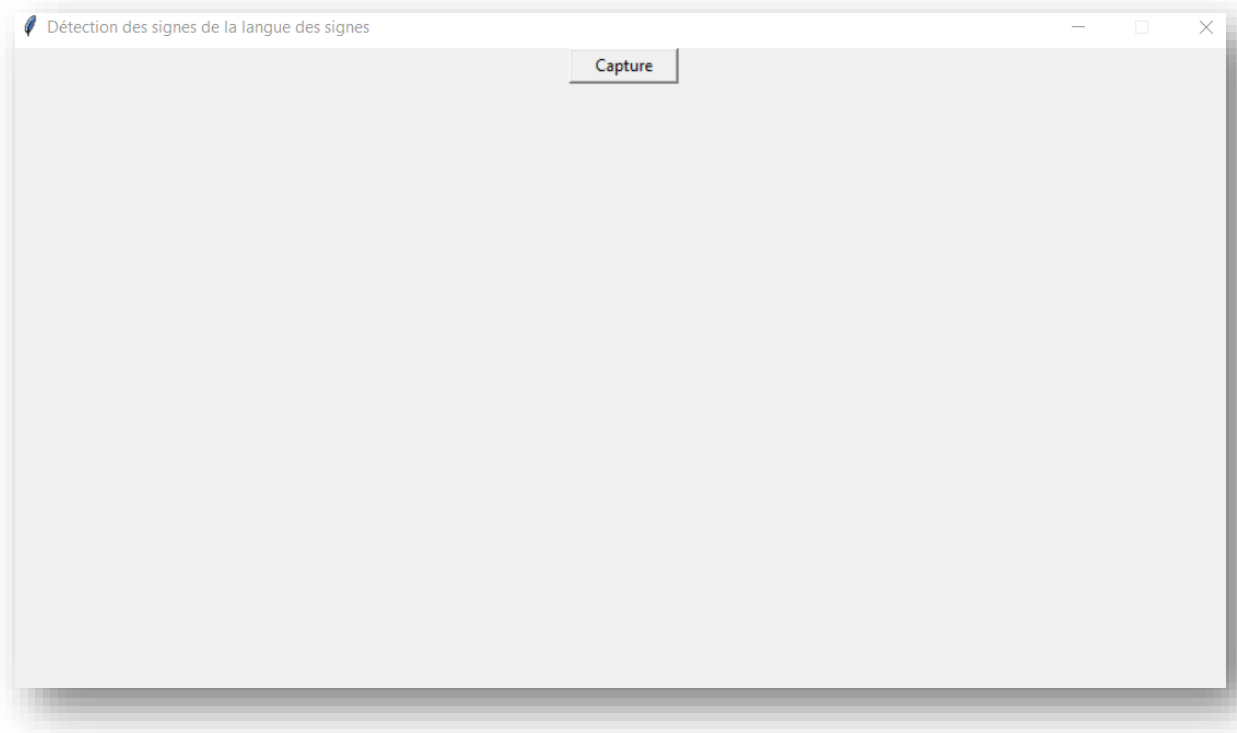
A. Structure de l'application :

L'interface créée par ce code comporte trois éléments principaux :

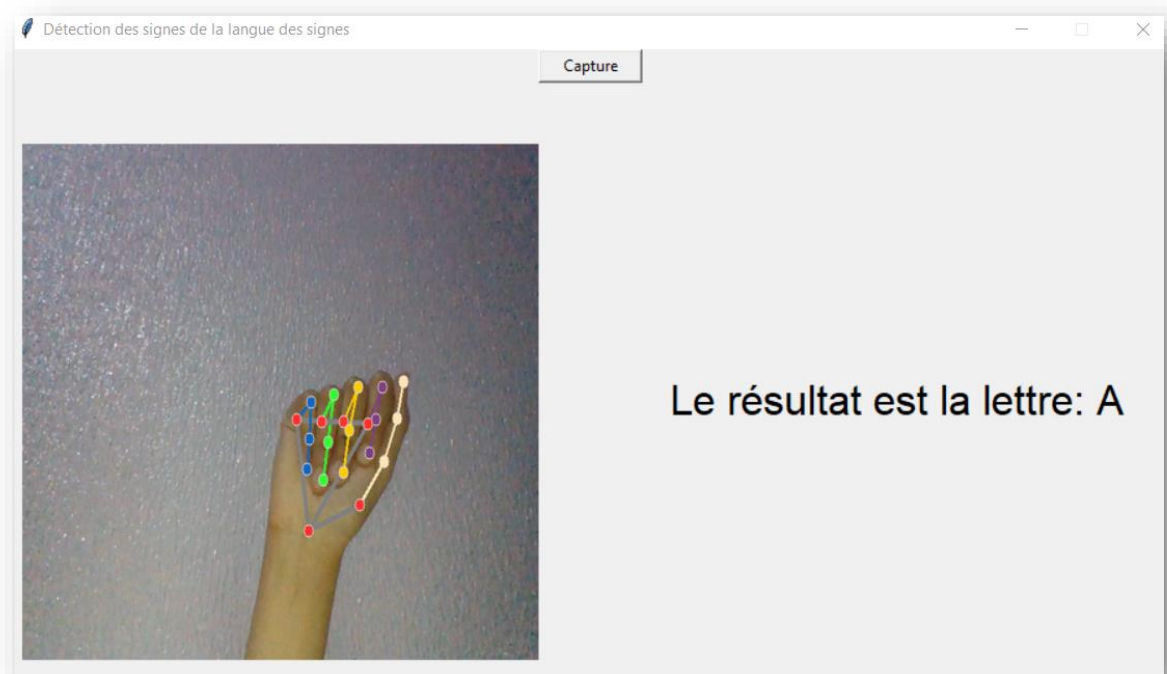
Bouton de capture : Un bouton "**Capture**" est ajouté à l'interface graphique qui permet à l'utilisateur de capturer une image de la webcam pour la reconnaissance de la langue des signes.

Zone d'affichage d'image : Une zone d'affichage d'image est ajoutée à l'interface pour afficher l'image capturée à partir de la webcam. Cette zone d'affichage est mise à jour à chaque fois qu'une nouvelle image est capturée.

Zone d'affichage de résultats : Une zone d'affichage de résultats est ajoutée à l'interface pour afficher le résultat de la reconnaissance de la langue des signes. Cette zone d'affichage est mise à jour à chaque fois qu'une nouvelle image est capturée et que la lettre de l'alphabet associée au geste de la main est prédite par le modèle de machine learning.



B. Résultat :



C. Discussion :

La reconnaissance de la langue des signes est une tâche importante pour faciliter la communication entre les personnes sourdes et muettes et les entendants. Dans ce projet, nous avons utilisé le modèle de machine learning RandomForest pour reconnaître les gestes des mains associés aux lettres de l'alphabet en temps réel à partir de l'image de la main capturée par une webcam. Nous avons entraîné le modèle à l'aide de points clés de la main extraits à partir d'un ensemble de données d'exemples de gestes de la langue des signes associés à chaque lettre.

Les résultats montrent que le modèle de RandomForest peut reconnaître la grande majorité des gestes de la main associées à chaque lettre de l'alphabet. Cependant, certaines lettres ont des gestes de la main similaires qui peuvent être difficiles à distinguer, ce qui entraîne des erreurs de reconnaissance. Par exemple, les gestes de la main pour les lettres "U" et "V" sont très similaires et peuvent être confondus.

Une autre limitation de ce projet est que nous avons utilisé uniquement le contour de la main pour la reconnaissance des gestes de la langue des signes. Cependant, la langue des signes utilise également des signes non manuels tels que la position de la tête et des parties du corps ou les expressions faciales. Par conséquent, notre modèle ne peut pas reconnaître ces signes non manuels, ce qui peut limiter sa précision dans la reconnaissance de la langue des signes.

Enfin, notre modèle a été entraîné sur un ensemble de données limité. L'utilisation d'un ensemble de données plus grand et plus diversifié pourrait permettre d'améliorer la précision de la reconnaissance des gestes de la main pour la langue des signes.

III. Conclusion

En conclusion, ce projet montre que l'utilisation de la technologie de machine learning peut aider à surmonter les barrières à la communication entre les personnes sourdes et muettes et les entendants. Cependant, il est important de reconnaître les limitations de la technologie actuelle et de poursuivre la recherche et le développement de nouvelles approches pour améliorer la reconnaissance de la langue des signes.