Projet PFA

TRADUCTION DE LA LANGUE DES SIGNES EN TEXTE

Pr. Koulouh Fatima

Realisée par:
Meryem EL HAMMAMI

Filière:
Big Data

Année universitaire: 2024-2025

Table de matières:

Résumé:

Ce projet a pour objectif de développer un système de reconnaissance de la langue des signes capable de traduire en temps réel les signes statiques de l’alphabet en texte écrit. Grâce à l’utilisation de la vision par ordinateur (MediaPipe), de l’apprentissage profond (PyTorch) et d’une interface utilisateur intuitive (OpenCV), le système permet d'afficher la lettre reconnue, de construire des mots et de proposer des suggestions intelligentes.

Introduction:

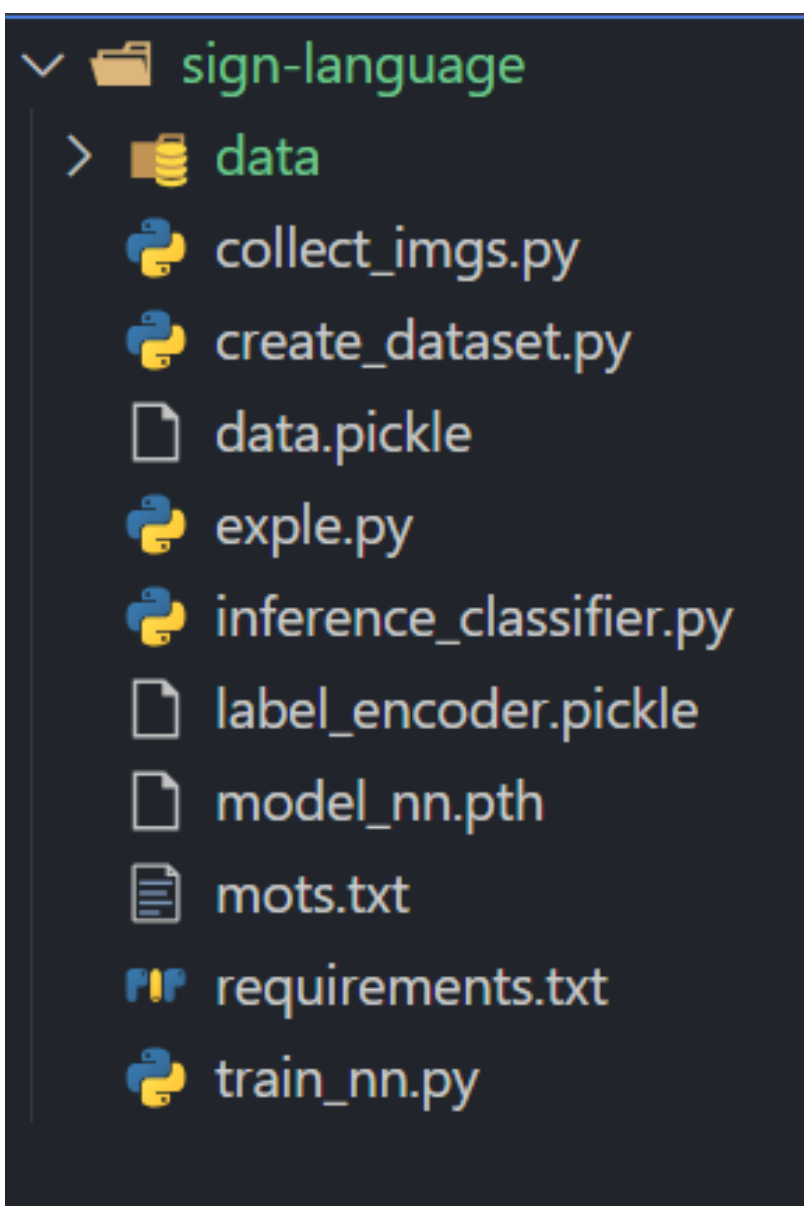
La langue des signes est un outil essentiel pour la communication des personnes sourdes ou malentendantes. Cependant, son intégration dans les systèmes numériques reste limitée. L’objectif de ce projet est de concevoir un prototype de reconnaissance des signes de la main correspondant à l’alphabet, afin de les convertir automatiquement en texte, tout en proposant une interface ergonomique et adaptée.

Problématique:

En explorant des outils d’intelligence artificielle comme ChatGPT, j’ai constaté qu’ils intègrent aujourd’hui des fonctionnalités avancées telles que la reconnaissance vocale, permettant aux utilisateurs de dialoguer à l’oral de manière fluide. Cependant, aucune solution équivalente n’est proposée pour les personnes qui ne peuvent pas parler, notamment celles qui utilisent la langue des signes comme principal moyen de communication. Ce constat met en évidence un manque d’inclusivité dans les interfaces actuelles. C’est dans cette optique que ce projet prend tout son sens : il vise à offrir une alternative accessible en permettant aux personnes sourdes ou muettes de s’exprimer naturellement à travers leurs gestes, tout en bénéficiant d’une traduction automatique en texte dans une interface simple, rapide et adaptable aux technologies existantes.

Architecture du système:

Le système de reconnaissance de la langue des signes repose sur une série d’étapes coordonnées, allant de la capture d’images à la génération de mots. Tout commence par la capture d’images en temps réel via une webcam, où l’utilisateur effectue des signes. Ces images sont ensuite analysées par MediaPipe, qui extrait 21 points clés de la main (landmarks), fournissant une représentation précise de la posture.
Les coordonnées obtenues sont normalisées pour éliminer les variations dues à la taille ou à la position de la main, ce qui permet d’obtenir un vecteur de 42 valeurs numériques prêt à être traité.
Ce vecteur est ensuite transmis à un modèle de deep learning entraîné avec PyTorch, qui prédit la lettre correspondante au geste réalisé. À chaque prédiction, une interface intelligente permet la construction progressive de mots, avec suggestions contextuelles pour faciliter la communication.
L’architecture est reflétée dans la structure du dossier projet, où l’on trouve les scripts de collecte de données, d’entraînement, de prédiction, les fichiers du modèle et des données utiles comme les étiquettes ou les mots proposés.



Intégration du Machine Learning et Deep Learning:

Le système développé repose sur une architecture hybride, combinant des techniques classiques de machine learning et des approches plus avancées de deep learning, réparties sur plusieurs étapes clés.
Dans un premier temps, la partie **Machine Learning** intervient pour le prétraitement et l'extraction des caractéristiques à partir des images capturées. À l'aide de MediaPipe, 21 points clés de la main sont détectés automatiquement, permettant d'obtenir des features 2D représentant avec précision la position des articulations et des doigts. Ces points sont ensuite normalisés : les coordonnées sont ramenées dans un espace standard compris entre 0 et 1, afin d'assurer une invariance aux différences de taille, de distance ou d'orientation de la main. En parallèle, les étiquettes associées à chaque geste sont encodées sous forme numérique à l'aide d'un LabelEncoder, ce qui permet de les exploiter efficacement pour l'apprentissage supervisé.
La seconde phase mobilise les techniques de **deep learning**. Un réseau de neurones multicouche (composé de trois couches entièrement connectées) est implémenté avec PyTorch. Ce modèle est entraîné à partir des vecteurs de caractéristiques extraits pour apprendre à associer chaque vecteur à une lettre de l'alphabet correspondant au signe effectué. Une fois l'entraînement achevé, le modèle est intégré dans le système via le script inference_classifier.py, qui assure la prédiction en temps réel dans l'interface utilisateur.
Ainsi, le modèle appris est capable de généraliser les gestes réalisés en conditions réelles à partir de simples coordonnées de main, et de les transformer avec fiabilité en lettres alphabétiques. L'ensemble de cette architecture permet une interaction fluide, précise et rapide avec l'utilisateur, tout en offrant un socle évolutif pour des améliorations futures comme la reconnaissance de gestes dynamiques ou la génération vocale.

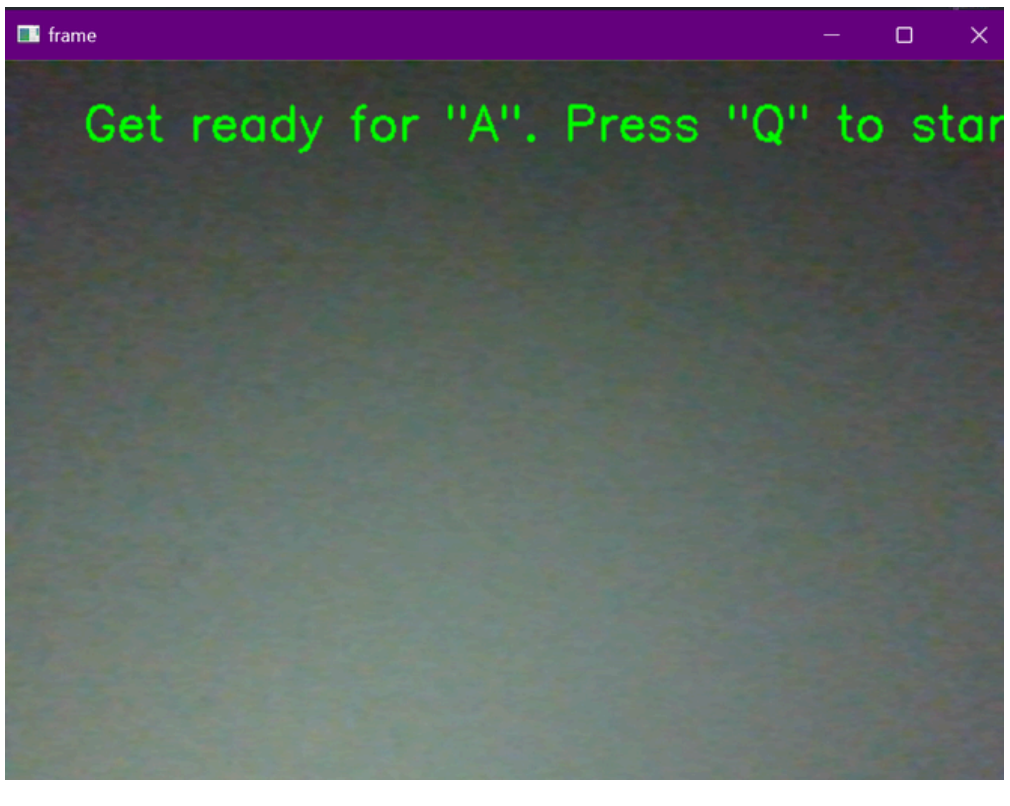
1) Collecte et préparation des données:

La première étape essentielle du projet consiste à constituer manuellement un jeu de données représentatif de l'alphabet gestuel. Pour cela, un script dédié, intitulé collect_imgs.py, a été développé afin de capturer des images en temps réel via la webcam de l'utilisateur. Ce script permet de collecter des données pour chaque lettre de l'alphabet (A à Z), ainsi que pour deux actions supplémentaires : espace (space) et effacement (backspace). À chaque exécution, l'utilisateur effectue un signe avec la main, et le script enregistre une série d'images associées à l'étiquette sélectionnée.

Une fois les images collectées, un second script, create_dataset.py, intervient pour préparer les données d'entraînement. Ce script exploite la bibliothèque MediaPipe pour analyser chaque image et en extraire 21 points clés (landmarks) de la main, chacun représenté par des coordonnées (x, y) dans l'image. Ces 21 points sont ensuite convertis en un vecteur de 42 valeurs numériques, reflétant la position relative de la main.

Afin d'assurer la cohérence du modèle lors de l'entraînement, ces vecteurs sont normalisés, c'est-à-dire que les coordonnées sont transformées pour être comprises entre 0 et 1. Cette normalisation permet d'éliminer les variations liées à la position ou à la taille de la main dans le champ de la caméra.

Les vecteurs normalisés sont ensuite associés à leur étiquette correspondante (la lettre ou la commande représentée) et enregistrés dans un fichier data.pickle. Ce fichier constitue le jeu de données final, prêt à être utilisé pour l'entraînement du modèle de deep learning.



2) Entraînement du modèle:

L’étape suivante du projet consiste à entraîner un modèle de classification supervisée capable d’associer des vecteurs de points de main à la lettre correspondante. Pour cela, un script nommé train_nn.py a été conçu en utilisant le framework PyTorch, largement reconnu pour sa souplesse et sa puissance dans les applications de deep learning.
Le modèle implémenté est un réseau de neurones dense relativement simple, mais efficace. Il est constitué de trois couches entièrement connectées. La couche d’entrée reçoit les 42 valeurs issues de l’extraction des landmarks de la main. Ce vecteur est ensuite traité par deux couches cachées de 64 neurones chacune, activées par la fonction ReLU (Rectified Linear Unit), qui introduit de la non-linéarité et améliore la capacité d’apprentissage du modèle. La couche de sortie produit une prédiction sur n classes, correspondant aux différentes lettres de l’alphabet ainsi qu’aux classes spéciales comme « espace » et « effacement ».
L’optimisation est assurée par l’algorithme Adam, connu pour sa rapidité de convergence et sa stabilité. La fonction de perte utilisée est l’entropie croisée (CrossEntropyLoss), adaptée aux problèmes de classification multiclasse.
L’entraînement du modèle s’est déroulé sur un total de 150 époques, ce qui a permis d’ajuster progressivement les poids du réseau afin de minimiser l’erreur sur l’ensemble des données d’apprentissage. À l’issue de cet entraînement, le modèle a atteint une précision de 96,36 % sur les classes les plus représentées, ce qui montre une capacité significative à généraliser les gestes appris à partir de simples coordonnées de la main.
Ce modèle constitue la base du système de prédiction en temps réel, utilisé dans l’interface utilisateur.

3) Interface utilisateur et prédiction en temps réel:

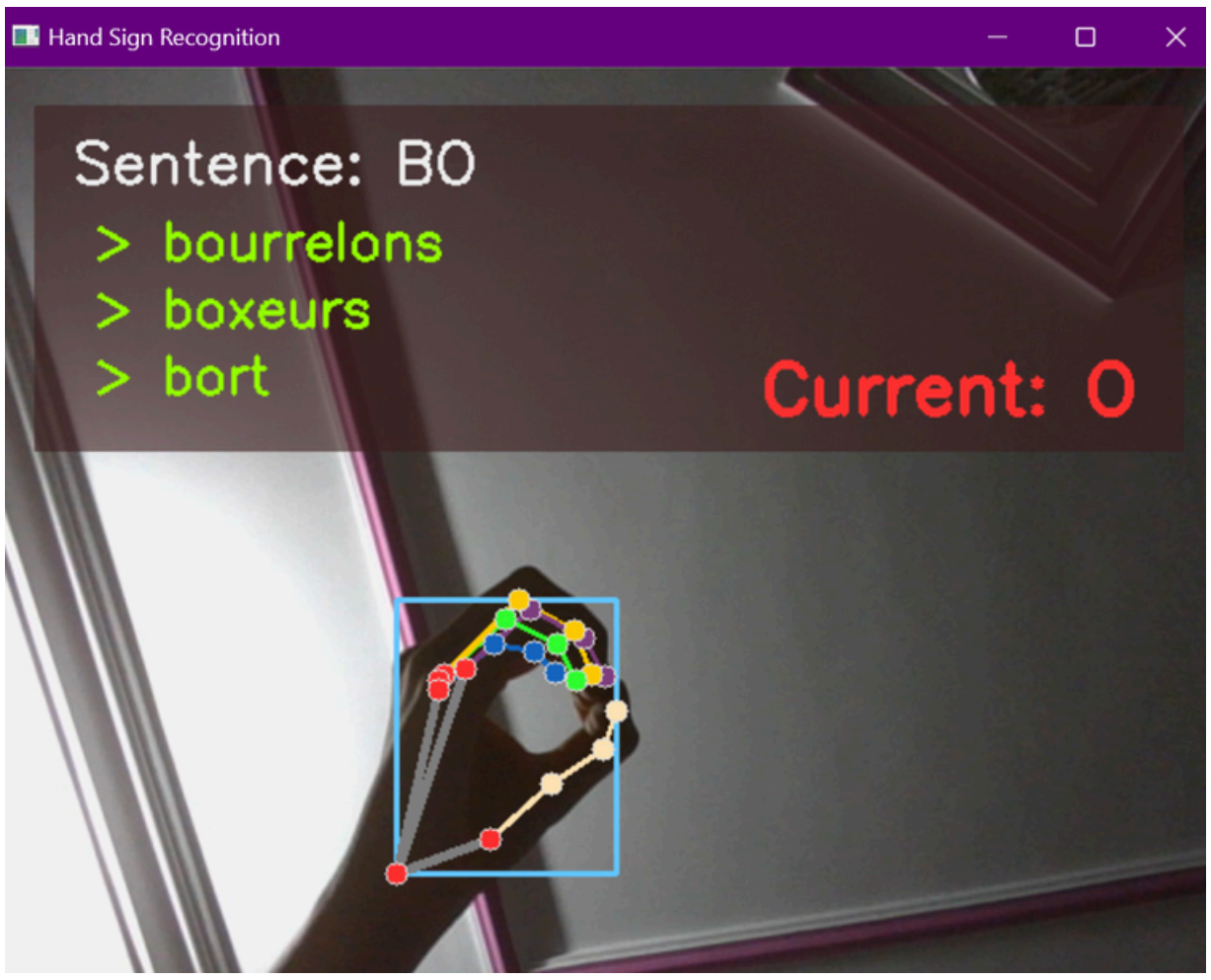
L’interface utilisateur constitue une composante essentielle du système, permettant à l’utilisateur d’interagir de manière fluide et intuitive avec le modèle. Elle est implémentée à l’aide du script inference_classifier.py, qui assure la gestion en temps réel de la détection, de la prédiction et de l’affichage graphique.

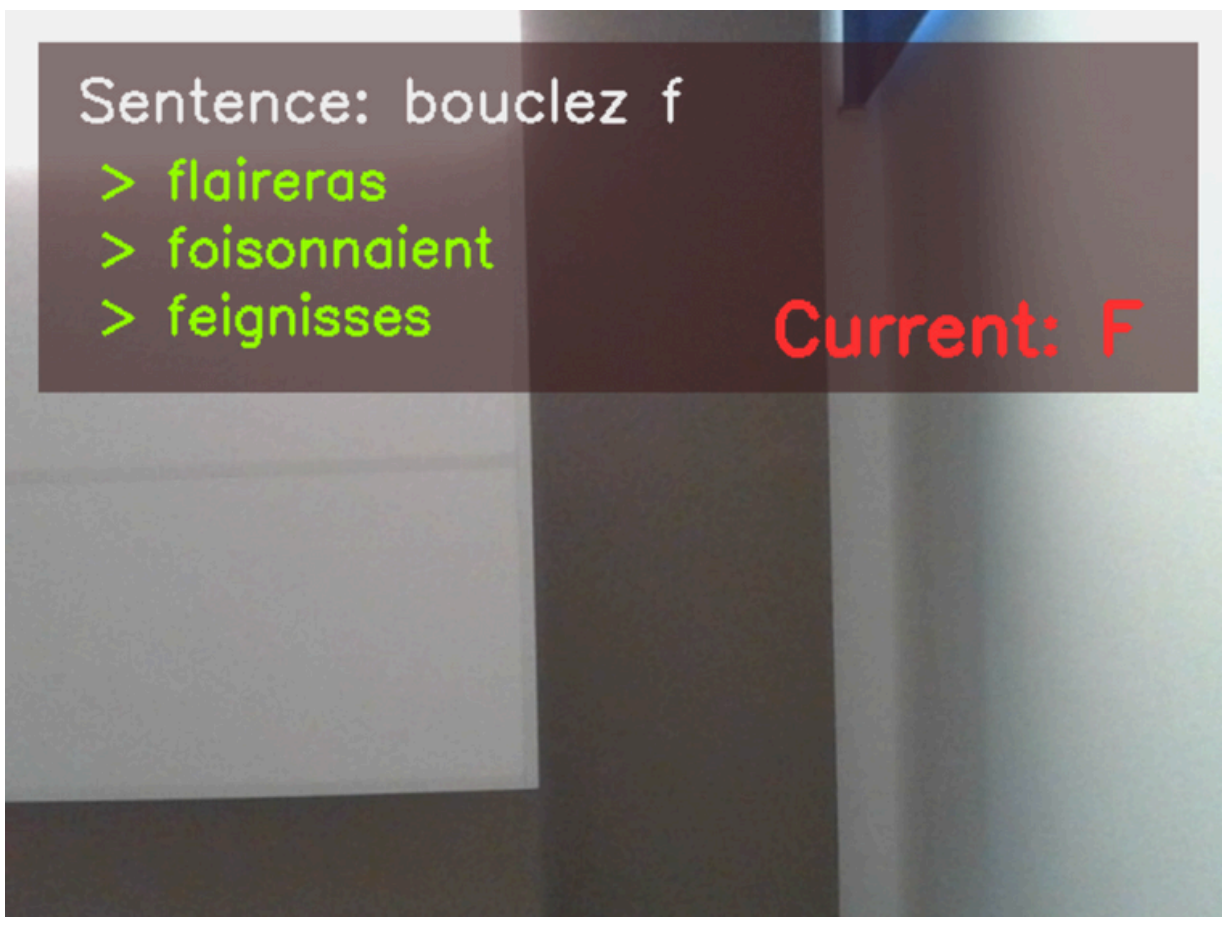
Lors de l’exécution du script, la caméra détecte automatiquement la main de l’utilisateur, en utilisant les capacités de MediaPipe. Une boîte rectangulaire est tracée autour de la main pour mieux délimiter la zone d’intérêt, et les landmarks (points clés) sont affichés en surimpression afin de donner un retour visuel immédiat sur la qualité de la détection.

Simultanément, le modèle de deep learning préalablement entraîné est utilisé pour prédire la lettre correspondant au geste détecté. Afin d’éviter les erreurs dues à des mouvements involontaires ou à une détection instable, l’affichage d’une lettre n’est validé que si le même signe est détecté de manière stable pendant une seconde. Cette logique garantit une meilleure précision dans la saisie des lettres.

Au fur et à mesure que les lettres sont reconnues, l’interface permet de construire dynamiquement un mot, affiché en haut de l’écran. Pour faciliter cette construction, un système de suggestions intelligentes est intégré : à partir des lettres déjà saisies, le programme propose automatiquement des mots issus d’un corpus externe (mots.txt), permettant à l’utilisateur de choisir rapidement un mot sans devoir l’épeler entièrement.

Enfin, une attention particulière a été portée à l’aspect visuel de l’interface. Celle-ci est conçue pour être à la fois esthétique et fonctionnelle, avec des couleurs harmonieuses, des zones semi-transparentes pour une meilleure lisibilité, et une police claire facilitant la lecture. L’ensemble crée une expérience utilisateur agréable et accessible, même pour les personnes peu familiarisées avec les outils technologiques.


Hand Sign Recognition
Sentence: BO
> bourrelons
> boxeurs
> bort
Current: O


Sentence: bouclez f
> flaireras
> foisonnaient
> feignisses
Current: F

Résultats et performances:

- Le système développé pour la reconnaissance de la langue des signes présente des résultats très encourageants. Le modèle d'apprentissage profond utilisé atteint une précision de 96,36 % sur le jeu de test, ce qui témoigne d'une performance fiable et robuste dans la classification des signes statiques.
- Par ailleurs, le système fonctionne en temps réel, avec une latence très faible estimée à 0,4 seconde, complétée par une stabilité de l'affichage de 1 seconde, garantissant ainsi une expérience utilisateur fluide et réactive. De plus, des suggestions de lettres sont proposées à l'utilisateur de manière aléatoire mais pertinente, directement affichées en haut de l'écran pour faciliter l'interaction.

Limites et perspectives:

- Malgré les résultats satisfaisants, certaines limites importantes demeurent. Actuellement, le système est uniquement capable de reconnaître des signes statiques, principalement les lettres de l'alphabet. Il ne prend donc pas en charge la reconnaissance de gestes dynamiques tels que les mouvements ou les séquences gestuelles correspondant à des mots entiers ou des phrases complètes.
- Ces limitations réduisent le champ d'application du système dans un contexte de communication fluide et naturelle en langue des signes, qui repose généralement sur une combinaison de mouvements complexes.

Améliorations possibles :

Afin d’enrichir et de perfectionner le système, plusieurs pistes d’amélioration peuvent être envisagées :

- Ajout d’un module de reconnaissance vocale en sortie, permettant une conversion automatique des signes reconnus en parole synthétisée. Cette fonctionnalité permettrait une interaction plus directe et accessible entre les personnes sourdes et les entendants.
- Extension de la reconnaissance aux gestes dynamiques, en intégrant des modèles temporels (tels que les réseaux neuronaux récurrents ou les transformers), afin de pouvoir reconnaître des séquences de gestes représentant des mots ou des phrases.

Conclusion:

Ce projet a permis de prototyper un traducteur de langue des signes en texte basé sur la vision par ordinateur et le deep learning.
Il met en œuvre une chaîne complète allant de la capture des données à l'affichage interactif, en passant par l'entraînement d'un modèle personnalisé. L'approche peut être généralisée et améliorée pour couvrir des scénarios plus complexes de communication inclusive.